Agenda Item:

Source: **NTT DoCoMo**

Title: **RLC Confirm primitive**

Document for: **Decision**

_____

1. Introduction

This paper shows problems of "Confirm" primitive from RLC to L3 and proposes the solution.

2. When "Confirm" primitive is necessary?

"Confirm" primitive is useful when time sensitive L3 (RRC) messages are transmitted that is the case the L3 (RRC) entity wants to have immediate acknowledgement for the transmitted message to perform next action as soon as possible.

Example: When "RRC CONNECTION RELEASE" is transmitted.
         RRC entity will be released as soon as the "Confirm" is received from RLC.

3. Problems of "Confirm" primitive

3.1 Concept of protocol

L3 (RRC) is the layer to generate massages for signaling. And L2 (RLC) is the layer to transfer the signaling messages.
Using RLC acknowledgement in place of RRC acknowledgement message makes each layer's (RLC & RRC) responsibility obscure.
-> It is inconsistent to the stratified protocol concept.

3.2 Actions for unnecessary "Confirm" primitive

Supposing that "Confirm" primitive is used for all of the L3 messages, L3 has to perform based on "Confirm" primitive. Hence L3 must take "Confirm" primitive into the consideration in the state that does not need "Confirm" primitive and include the mechanism to receive "Confirm" primitive in any state and to act against the "Confirm" primitive.
-> It makes L3 state complicated in unnecessarily.

3.3 Relationship between "Confirm" primitive and L3 messages

If "Confirm" primitive is used, the mechanism that detects the relationship between "Confirm" primitive and L3 message should be needed. Following two mechanisms are considerable.
    1)   Getting acknowledgement against a L3 message certainly
    2)   Giving reference number
In case of 1), if there are some L3 messages should be transmitted continuously, the next message can not be transmitted unless "Confirm" is received.
In case of 2), reference number is necessary for each L3 messages and reference number check mechanism is needed for L3.
-> These extra mechanisms make L3 heavier.

4. Proposal

Considering the situation that "Confirm" primitive is used is limited, following two methods are considerable to solve the above problems.
    1)   Using L3 Acknowledgement message instead of RLC Acknowledgement and transmitting it with unacknowledged mode of RLC.

-> RLC and L3 can be separated clearly.

  -> This method may cause higher error probability if the L3 message is quite long.

2) "Confirm" primitive is requested only when it is necessary by adding extra parameter in data request primitive (RLC-AMDATA req.) or providing a new primitive.

  -> This method can reduce the impact against L3.

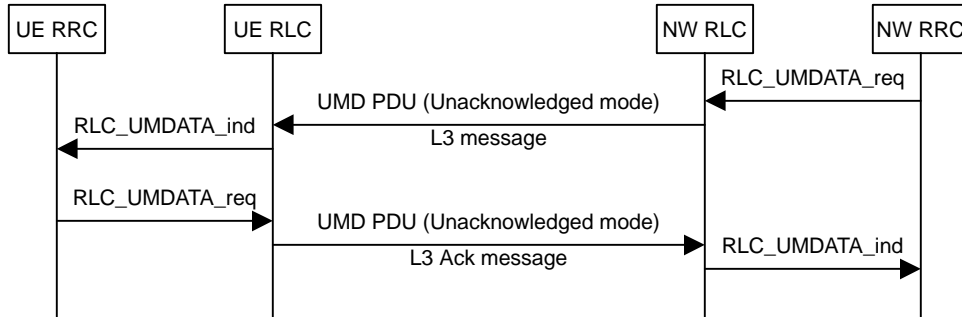It is proposed that "Confirm" primitive should be studied based on these two methods.

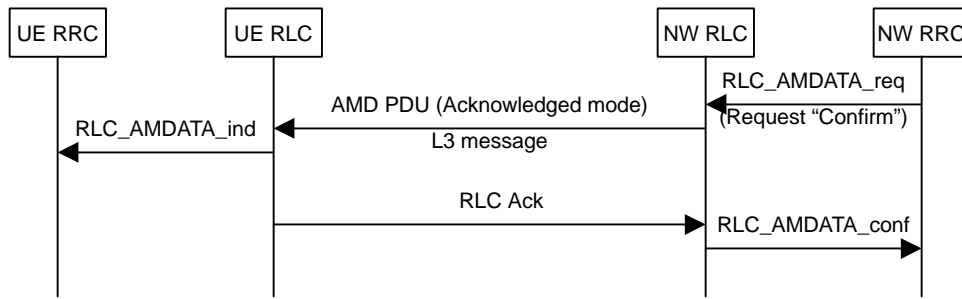| UE RRC | UE RLC | | NW RLC | NW RRC |

RLC_UMDATA_req

UMD PDU (Unacknowledged mode)

RLC_UMDATA_ind

L3 message

RLC_UMDATA_req

UMD PDU (Unacknowledged mode)

L3 Ack message

RLC_UMDATA_ind

Fig. 1 Method 1

| UE RRC | UE RLC | | NW RLC | NW RRC |

RLC_AMDATA_req

AMD PDU (Acknowledged mode)

(Request "Confirm")

RLC_AMDATA_ind

L3 message

RLC Ack

RLC_AMDATA_conf

Fig. 2 Method 2