

Agenda item: 6.1.4
Source: TD Tech
Title: ACK/NACK Multiplexing in PUSCH in TDD
Document for: Discussion and decision

1 Introduction

One issue related to ACK/NACK transmission in the PUSCH is how to avoid data being interpreted as ACK/NACK at eNB when a DL grant is missed by a UE.

The discussing solution is to have 1 bit in the UL grant to indicate to the UE that eNB expects it to transmit ACK/NACK, or always reserving space in the PUSCH for ACK/NAK transmission. As ACK/NACK in uplink control sub-frame is strictly corresponding to one downlink sub-frame with time alignment in FDD with HARQ process, the 1 bit indication in UL grant can guarantee UE be able to include the corresponding ACK/NACK in PUSCH and reserve the right and enough place for ACK/NACK, and eNodeB would not mistake the data as ACK/NACK. However, in TDD, the issue is a bit more complicated as there exist configurations with more sub-frames configured as DL than UL, and it requires several ACK/NACK be multiplexed in one PUSCH. As there is no one-to-one association between DL and UL sub-frames as in FDD, the eNodeB would mix up the associations between the ACK/NACK and the DL grants when one or more DL grants are missed by UE and there is no explicitly downlink indication of the number and allocation of the ACK/NACK.

For ACK/NACK multiplexing in TDD, [3] suggests “AND” operation on all the ACK/NACKs and only one bit of the AND result is allocated in the UL PUSCH to solve the coverage problem, as the high order modulation for multiplexing the multiple ACK/NACK would greatly reduce the UL coverage. With this “AND” approach, the UL coverage problem is solved, but when there is NACK for one DL sub-frame is detected by UE, the “AND” operation makes UE feedback only the NACK to eNB, no matter the ACK or NACK feedback corresponding to other DL sub-frames, and in this case, eNB would not know which transmission goes wrong and it would simply retransmit all the DL blocks, which would greatly reduces the DL link throughput.

2 Why ACK/NACK Multiplexing in TDD

In [3] it agrees to consider both bundling ACK and multiple ACK strategies in asymmetrical TDD systems. To our understanding, the discussing multiple TTI in both UL and DL requires either the bundling ACK or multiple ACK. With the bundling strategy, we notice two issues not solved at moment:

1. **The bundling solves the UL coverage, but not the DL capacity reduction.**
2. **In case DL grant is missed by UE, how to detect the DL grant missing or DTX-ACK without mistaking the ACK/NACK as data.**

3 ACK/NACK Multiplexing in PUSCH in TDD

In this contribution, we propose use multiple ACK/NACK to avoid the difficulties by bundling ACK. The following problems are addressed by the ACK/NACK multiplexing method.

1. *How to multiplex the several ACK/NACK in PUSCH without reducing the UL coverage?*
2. *How the ACK/NACK multiplexing works, when there are DL grants missed by UE, eNB can still detect the missing without mixing up the ACK/NACK, i.e. not mis-associating one ACK/NACK to other DL grant.*

In solving the above problems, we extend the idea of transmitting 1 bit in UL grant in [1] by inserting more bits in UL grant to indicate the UE how many ACK/NACK spaces or bits need to be reserved for UL ACK/NACK in PUSCH. In DL grant the same number of bits is included for indicating the exact start position (How to multiplexing the multiple ACK/NACK is FFS, for example, the multiple ACK/NACK bits are jointly coded and spread with ZC sequence. Anyway, multiple ACK/NACK shall be in a kind of ordering, no matter it is a exact position or CS shift of a ZC sequence, we can just call it “position”) of the corresponding DL ACK/NACK in the PUSCH. Denote the bits associating to the number of reserved ACK/NACK spaces in UL grant as “ACK-Num-bits” and the bits associating to the ACK/NACK start position index in DL grant as “Index-bits” respectively. “ACK-Num-bits” in the UL grant is to tell the UE the spaces are required to be reserved for ACK/NACK and eNB would not mistake the ACK/NACK reservation as the data in PUSCH. The “Index-bits” in the DL grant is used to tell the NodeB, where the UL ACK/NACK for the corresponding DL transmission insertion in the PUSCH starts from.

The idea behind including the same number of bits in DL as in UL grant is that in case the DL grant is missed by UE, UE would still reserve the ACK/NACK rooms for the missed DL grant. When PUSCH is received by eNB, eNB can be sure there are the reservations for ACK/NACK in PUSH. The exact region of the reservation is known both to UE and eNB, such that eNB would not interpret the ACK/NACK in PUSCH as data. With this method, each DL transmission maps to one bit UL ACK/NACK feedback, which incurs no UL coverage loss and performs more like one-to-one mapping if one naked bit to one ACK/NACK.

For telling the eNB whether there is DL grant missed by UE, the method from [2] can be applied, which employs different CRC masking to distinguish the DL grant missed or non-missing cases: when it has a DL grant missed, the total number of ACK/NACK reservation counted in DL grant will not be equal to the number of ACK/NACK indicated in the UL grant. After simple operation, UE knows there are DL grant missed, it would transmit a masking CRC which is “AND” or “XOR” by the UE ID. UE detects the DL grant missing by checking the start points and the number of streams indicated by all the DL grants.

A simple example with non-MIMO case is used to illustrate the method: Suppose, a cell is with DL:UL = 3:1 configuration, 4 DL (3DL frames + DwPTS) for PDSCH and 1 UL for PUSCH. Say DL grant 2 is missed by UE due to the interference and noise. The “Index-bits” in the received DL grants detected by UE are 0, 1 and 3. Further assume UL grant is correctly decoded by UE and the “ACK-Num-bits” =4 indicating to UE that there 4 spaces or bits needs to be reserved for ACK/NACK. After comparing the total number “3” ACK/NACK spaces or bits required in DL grant and the reservation “4” indicated in DL grant, UE knows there is a DL grant is missing, further, it knows it is “2” DL grant missing and it would inform eNB there is DL grant missing by the CRC masking method and set the corresponding “2” reservation to NACK. At eNB, after the data decoding and the masking CRC examination, eNB knows a DL grant is missed (Assuming no error is in UL data). Then, eNB would exam the NACKs it receives and retransmits the corresponding DL grants and corresponding DL data. Here, when there is only one NACK in the PUSCH, eNodeB know which DL grant is missed by UE or eNode would not exactly know which DL grant is missed by UE in more NACKs feedback case and it would retransmit the DL grants and data corresponding to the NACKs feedback ignoring the corresponding DL grants are correctly received or not.

When no DL grant is missed, no masking operation is applied to CRC in PUSCH. When eNB checks the CRC is a non-masking CRC, it just retransmits the data corresponding to the NACKs following the normal HARQ process. When the non-masking-CRC and masking-CRC are all wrong, there are errors in received data, eNB will retransmit the DL grant and DL data corresponding to the NACK. Table 1 lists the possible CRC cases.

Table 1.

Masking CRC wrong, Non Masking CRC correct	No DL grant missing, eNB retransmits the DL data
--	--

Masking CRC correct, Non Masking CRC correct	corresponding to the NACK.
Masking CRC correct, Non Masking CRC wrong	DL grant missing, eNB retransmits the DL grant and DL data corresponding to the NACK.
Masking CRC wrong, Non Masking CRC wrong	

As the reservation region in PUSCH is known in advance, the eNB only needs to calculate the CRC out of the data region and check the CRC is masked or unmasked with user ID by either a “AND” or “XOR” operation. With ACK/NACK insertion or not, eNB only needs to calculate CRC once and can reduce additional decoding computational complexity required by the blind decoding.

Considering saving the “Index-Bits” in DL grant, the HARQ process ID can be employed for indicated the corresponding ACK/NACK position in PUSCH, if the HARQ processes IDs are in continuous order.

4 Conclusion

In this contribution, we propose a multiple ACK/NACK multiplexing transmission in PUSCH method, which can solve both the UL coverage and DL grant failure problems. By extending the 1bit in [1] to more bits in UL grant and inserting the same number of bits in DL grant as in UL grant (the exact number of bits depending on the maximum number of ACK/NAK for multiplexing), the proposed scheme also overcomes the downlink throughput reduction problem of the ACK/NACK bundling in [1].

References

- [1] Nokia Siemens Networks, Nokia, R1-074331, “Avoiding PUSCH error situation caused by DL allocation grant signaling failure”, 3GPP RAN1 meeting #50bis.
- [2] Samsung., R1-080676, “ACK/NAK DTX detection in the PUSCH”, 3GPP RAN1 meeting #52
- [3] Ericsson, et al, R1-081110, “Multiple ACK/NAK for TDD”, 3GPP RAN1 meeting #52