3GPP TSG-RAN WG1 Meeting #100bis-e R1-20xxxxx

e-Meeting, April 20th – 30th, 2020

Agenda Item: 6.2.1.3

Source: Moderator (Ericsson)

Title: Feature lead summary #2 for Multi-TB scheduling for LTE-MTC

Document for: Discussion, Decision

# Introduction

In the Rel-16 work item on “Additional MTC enhancements for LTE” [1], one of the objectives is to specify support for scheduling of multiple DL/UL transport blocks.

|  |
| --- |
| The objective is to specify the following set of improvements for machine-type communications for BL/CE UEs.  [...]  **Scheduling enhancement:**   * Specify scheduling multiple DL/UL transport blocks with single DCI for SC-PTM and unicast [RAN1, RAN2] |

RAN1 agreements made until RAN1#99 are summarized in [2] and RAN1 agreements made in RAN1#100e are listed below. RAN2 agreements are summarized in [3]. The endorsed L1 configuration parameter list can be found in [4], the initial RAN1 UE feature list in [5], and the endorsed RAN1 CRs in [6] – [16].

|  |
| --- |
| [**R1-2001056**](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001056.zip) Feature lead summary for Multi-TB scheduling for LTE-MTC Ericsson  [**R1-2001185**](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001185.zip) Feature lead summary#2 for Multi-TB scheduling for LTE-MTC Ericsson  [**R1-2001220**](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001220.zip) Feature lead summary#3 for Multi-TB scheduling for LTE-MTC Ericsson  [100e-LTE-eMTC5-Multi-TB-01] – Johan (Ericsson)  Email discussion/approval onHARQ/NDI/RV/FH encoding for both FDD and TDDby 2/27; if there is a spec impact, followed by endorsing the corresponding TP by 3/2  **Conclusion**  For FDD case:   * For 36.212, use Futurewei’s TP in [R1-2001086](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001086.zip) as a basis, possibly with the clarification “From MSB to LSB” in each section. * For 36.211 and 36.213, take the provided comments and proposals into account in contributions to the next meeting.   For TDD case:   * There is no consensus in RAN1#100e for optimization (or elimination) of the TDD HARQ process grouping. The 36.212 seems adequate and potential corresponding 36.213 text can be added in the next meeting.   As per email decision posted on Mar. 4th, two companies prefer not to add “From MSB to LSB”, so:  Agreement: The text proposal in [R1-2001086](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001086.zip) is endorsed for inclusion into TS36.212 editor’s CR.  [100e-LTE-eMTC5-Multi-TB-02] – Johan (Ericsson)  Email discussion/approval onHARQ-ACK bundling for both FDD and TDDby 2/27; if there is a spec impact, followed by endorsing the corresponding TP by 3/2  As per email decision posted on Mar. 5th,:  Agreement: The TP provided in [R1-2001214](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001214.zip) for TS36.213 section 10.2 is endorsed. To be included as part of the editor’s CR for TS36.213.  [100e-LTE-eMTC5-Multi-TB-03] – Johan (Ericsson)  Email discussion/approval onscheduling gaps for both unicast and multicastby 2/27; if there is a spec impact, followed by endorsing the corresponding TP by 3/2  **Conclusion**  For the unicast case   * There is no consensus in RAN1#100e for the proposal to specify explicit unicast scheduling gaps. * Since unicast scheduling gaps are included in the draft RAN1 UE feature list, there may be a need to update the feature list, and this is something that can be brought up in the email discussion for the feature list.   For the multicast case   * There is no consensus in RAN1#100e for the proposal to insert the scheduling gaps before each TB instead of after each TB. |

This document provides a prioritized list of issues and proposals based on the contributions in [17] – [23].

# Issue #3: HARQ-ACK bundling size

RAN1#100e identified a need to define the mapping between DCI field ‘Multi-TB HARQ-ACK bundling size’ in 36.212 and parameter ‘M’ in 36.213. The 36.212 editor’s interpretation of the earlier RAN1 agreements is presented in Futurewei’s contribution [23].

Huawei’s contribution [17] and ZTE’s contribution [18] propose to map 0-3 in 36.212 to 1-4 in 36.213, whereas Qualcomm’s contribution [20] proposes to use 1 instead of 2 bits in the DCI and derive the bundle size from a table in the specification (see Section 2.3 in Huawei’s contribution, Section 2.2.4 in ZTE’s contribution, Issue #1 in Qualcomm’s contribution and Issue #2 in Ericsson’s contribution for further discussion).

Proposal 3-1: Discuss and decide on a mapping between DCI field ‘Multi-TB HARQ-ACK bundling size’ in 36.212 and parameter ‘M’ in 36.213.

|  |  |
| --- | --- |
| **Company** | **Comments on Proposal 3-1** |
| Qualcomm | We propose a **„1-bit field in the DCI“** that determines the size of the bundles for the purposes of HARQ-ACK bundling. We propose to **endorse TP1 in Section 1 of** [**R1-2002174**](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002174.zip)**.**  Please note the following **„technical“** reasons for this—in addition to the point that there is **no „agreement“ on this field**, as has been outlined by us in the past.   1. For less than 4 TBs scheduled (which is a practical use case), **we don‘t even have 4 states** to be represented by 2 bits. Please recall the often painful negotiations during this work item on **„saving every bit“ in the DCI design**—we must respect that principle here. Indeed, each bit added to the DCI reduces MPDCCH coverage. 2. In several cases, **many of the states (represented by 4 codepoints) are worse than others, making the worse states redundant**. See the examples below, where **for Fig. 1, [4,4] is worse than [2,3,3]** and in **Fig. 2, [2,2] is worse than [2,1,1]** when number of PUCCH & PDSCH repetitions are 1.     Figure 1: Throughput comparison for 8 HARQ processes.    Figure 2: Throughput comparison for 4 HARQ processes.  Moreover, as we highlight in the examples above, the „timeline limitations“ become an important issue when PUCCH has 1 repetition, and PDSCH has one repetition or PDSCH interleaving (with granularity 1) is enabled. As a result, this case should be treated differently from the case where this „timeline limitation“ is no longer a bottleneck.  Keeping these in mind, we propose to **endorse TP1 in Section 1 of** [**R1-2002174**](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002174.zip), that essentially implements the following proposal:  **Proposal 3-1-QC: The field “Multi-TB HARQ-ACK bundling size” is 1 bit that enables and disables HARQ-ACK bundling.**  **- The bundle sizes are fixed in the specification depending on (#repetitions for PDSCH, #TBs, #reps for PUCCH, interleaving ON/OFF) as in the table below, where:**  **- Case 1 is used if (“number of PDSCH repetitions = 1” or “interleaving is enabled”) and “number of PUCCH repetitions = 1”**  **- Otherwise, case 2 is used**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **1TB** | **2TB** | **4TB** | **6TB** | **8TB** | | **Case 1** | **M=[1]** | **M=[2]** | **M=[2,1,1]** | **M=[2,2,2]** | **M=[2,3,3]** | | **Case 2** | **M=[1]** | **M=[2]** | **M=[2,2]** | **M=[3,3]** | **M=[4,4]** |   ***FURTHER NOTES: To us, it is extremely disappointing how some companies are trying to imply that a certain wording allows for a certain way of DCI signalling, while expressly precluding other forms of DCI signalling. This cannot be farther from the truth.***  ***The agreement was to signal the „actual bundle size“ in the DCI. „How“ this was to be signalled, was NOT discussed. To us, „all“ the proposals on the table are an equally „allowed“ means to signal the bundle size. That’s exactly what our proposal does too: a DCI field that—in conjunction with the RRC parameter—tells the UE the size of the bundles.***  ***The interpretation that the bundle size determination will be done only according to the current equations in the specification, and any other forms of determination will be expressly prohibited, is extremely unfortunate—and if we may say so, plain wrong.***  ***We would also like to point out that even the current equation doesn‘t lead to „one size for all bundles“ in many instances. It cannot be „implied understanding“ that the „only allowed way“ to have different sizes for different bundles in a multi-TB PDSCH when a „remainder operation after a division“ dictates it. To us, such lines of reasoning are a disappointing means to block the discussion of alternative proposals that may have technical merit.***  ***We would kindly request the companies to try to judge each solution—all of which (including ours, and including the current placeholder text) are „not precluded“—on the technical merits of each. We hope we can do this. As a group, we deserve better for ourselves than to try to disallow legitimate solutions from discussion and consideration, based on arbitrarily-constructed and flimsy technicalities.*** |
| FUTUREWEI | As stated by the moderator, the views of the editor on the earlier agreements and how the specifications work are in [R1-2002654](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002654.zip), “HARQ-ACK bundling for Multi-TB scheduling”. We encourage companies to read that document, as we only summarize a few points here:   1. The editors made their best effort to produce specifications from the available agreements, and their efforts should not be disparaged. 2. The current specifications work. 3. The proposal from Qualcomm, in our view, does not follow the agreements. We understand from the statements above that Qualcomm feels it does. 4. Our paper clearly states that we are not trying to stop *any* technical discussion, not only for aspects noted during the endorsement process, but any other proposed implementation.   The moderator had previously suggested that we need to sort out the understanding of the agreements. Our view is in the paper. However, in our limited time it may be more productive to focus on the technical discussion. Technically, it appears to be flexibility/simplicity versus lower overhead and the ability to select certain preferered bundling options. FUTUREWEI has a slight preference for flexibility/simplicity, but may be open to update our view after hearing other views. |
| ZTE,Sanechips | For comparison, we should look at the issue from all aspects, peak transmission rate, uplink usage, total transmission time etc. Therefore we think 2bit method is the best. It is also aligned with the original intention of the agreement. We have given our detailes analysis in our tdoc , there's no need to paste the same content here, but we just want to recap the key observation/conclusion.  1. The design should enable better performance for large #TB, i.e, 6 or 8 TB scheduling, when this happens flexible bundling pattern is needed to adapt to different status (initial scheduling vs retransmission) of each TB. Single or very limited fixed bundling pattern would cause the performance loss especially for the large TB number or colossal repetitions.  2. Uplink resource usage should be considered. For example, when compare the bundling method [4,4] and [2,3,3]  Case a: A total of 8 TB, includes first 4 new TBs and last 4 retransmitted TBs. Obviously, the former method is better and has less retransmission.  Case b: All 8 TBs are new or retransmitted. It seems that latter method would be better, at the cost of more uplink usage. However, in this case, [2,2,2,2] seems to better than [2,3,3], and non-bundling may be the best.  So, from the perspective of resource saving, [2,3,3] does not show any benefits compared with [4,4]. if we want to reduce the PDSCH retransmission, then non-bundling is preferred. If we want to reduce uplink subframes, then [4,4] is preferred. It is just a tradeoff, pattern [2,3,3] is not the best choice.  3.Gap caused of timeline limitation is that important issue and the peak throughput performance is the same for [2,2] and [2,1,1]. Additionally, in many cases, when number of PUCCH & PDSCH repetitions are larger than 1, the throughput performance for [2,1,1] and [2,3,3] would be worse than [2,2] and [4,4] |
| Sierra Wireless | Firstly, I highly commend all editors as they do an amazing job with the agreements that we provide them. Thanks Brian for [R1-2002654](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100b_e/Docs/R1-2002654.zip) - it explains how we got here. I truly believe everyone wants two things – a clear/complete specification and good efficient design.  WRT a clear/complete specification: If you assume the “Multi-TB HARQ-ACK bundling size” is linear from 1-4, we still need to define the bundle size to a bundle pattern. Good examples are shown in [R1-2002654](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100b_e/Docs/R1-2002654.zip) e.g. if NTB=8 and M=3 the bundle pattern could be {3,3,2} or {2,3,3} or {3,2,3} or {invalid} - this still needs to be defined so the specification in not complete yet.  The only agreement we have is this general agreement to “Strive to reuse”:   * Strive to reuse Rel-14 HARQ-ACK bundling feature as baseline at least for the non-interleaving case   We have no agreements that the bundling size DCI field is 2 bits - thus Sierra feels this is an open issue similar to the bundling pattern being an open issue and both should be taken together.  Sierra’s preference is to have a 1:1 mapping of NTB scheduled and bundle patterns so that no DCI bits are needed. We express this view last meeting via email and in [R1-2000507](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2000507.zip).   |  |  | | --- | --- | | # of TBs scheduled | HARQ-ACK Bundle Pattern | | 1 | 1 | | 2 | {1,1} | | 4 | {2,1,1} | | 6 | {2,2,2) | | 8 | {3,3,2} |   As with Rel14 Ack bundling design, bundling can be disable when PDSCH repeats are indicated in DCI.  We can also unenthusiastically accept a 1-bit field and are open to bundle mappings. For example:   |  |  | | --- | --- | | # of TBs scheduled | HARQ-ACK Bundle Patterns (1-bit DCI) | | 1 | 1, ?? | | 2 | {1,1}, ?? | | 4 | {2,1,1}, [{1,1,1,1}] | | 6 | {2,2,2), [{1,1,2,2}] | | 8 | {3,3,2}, [{2,2,2,2}] |   We are not open to a 2-bit field, as we do not feel this a good use of valuable DCI bits and two options per NTB is sufficiently flexibility. |
| Lenovo&Moto | We prefer 2 bit indication in DCI solution, which gives full flexibility to eNB scheduling, and should make the relationship between bit indication={00,01,10,11} and M={1,2,3,4}in TS36.213. |
| Ericsson | With a 2-bit DCI field, is the common understanding that each bundle will have the same size, meaning e.g. that when 8 TBs are scheduled, all bundles have either size 1, 2 or 4, and only 3 of the 4 values that can be indicated by the 2-bit DCI field have a meaning? |
| LG | We are fine with either 1-bit or 2 bit DCI field for budle size indication. In any cases, we prefer that all bundle have the same size.  For example, if 1-bit indication is adopted, we would like to seggest to use only case 2 from proposal 3-1-QC. In our view, introducing additional bundling pattern only for repetition=1 is not a essential issue and benefits from it is marginal.  Regarding Ericsson’s comment, we prefer that all bundles have the same size if 2-bit indication is adopted. |
| Nokia, NSB | Our preference is to use 2-bit DCI indication and clarify the mapping in the specifications as 0-3 in 36.212 to 1-4 in 36.21.  The 1-bit solution provides some optimization but we feel the flexibility trade-off is not good enough and therefore prefer to have the full flexibility. |
| Huawei, HiSilicon | As we discussed in our contribution, we support 2 bits considering aspect considering the resource utilization, delay, flexiblity.  2 bits field can provide more choices for determining the actual bundling size to fit the variable channel condition and the traffic.  As shown in following figure, the [4,4] has the same time delay and resource utilization with [3,3,2] and [2,3,3], while [4,4] can save one PUCCH resource.  In addition, as it has been agreed that the actual bundling size is indicated by DCI, at the maintenance phase, we don’t think we should easily revert an agreement if there’s no errors in it. |
| FUTUREWEI  (2nd comment) | In response the Ericsson’s question, in my reading in [R1-2002654](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002654.zip) of the current 36.213 is the following for the 8 TB case:  = 8  *M* = 1,2,3,4 bundle sizes  = 8,4,3,2 bundles  Bundles are (1,1,1,1,1,1,1,1) or (2,2,2,2) or (3,3,2) or (4,4)  So if you signal M=3, you would have three bundles of 3, 3, and 2 TBs. Lenovo or others can confirm. |
| ZTE,Sanechips | Regarding the exact meaning of 2-bit indication, we have same understanding as Futurewei, i.e , the total number of TB is evenly divided by the bundle size indicated, each bundle has same integer number of Tb , except for the last bundle which could equal to the remainder of the integer division. |
| Qualcomm | We will try to reply to some of the comments made:  Futurewei: From your contribution:  **Observation 1: All bundles contain the same number of TBs, except possibly the last bundle.**  This is completely arbitrary. Even if you argue that the *bundle size* is constant, we need to define the way to distribute the TBs in case there is some remainder. Having said this, there is a single entry in the QC proposal that does not meet your criteria. The remaining cases can be written as a bundle size.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **1TB** | **2TB** | **4TB** | **6TB** | **8TB** | | **Case 1** | **M=[1]** | **M=[2]** | **M=[2,1,1]** | **M=[2,2,2]** | **M=[2,3,3]** | | **Case 2** | **M=[1]** | **M=[2]** | **M=[2,2]** | **M=[3,3]** | **M=[4,4]** |   In the Sierra proposal, same thing.  ZTE: Gap caused of timeline limitation is that important issue and the peak throughput performance is the same for [2,2] and [2,1,1].  > This argument doesn’t make much sense. If you argue this, then you could say that the peak throughput performance is the best if we just report single HARQ-ACK bit. For a given peak throughput, we should transmit as many HARQ-ACKs as possible.  Additionally, in many cases, when number of PUCCH & PDSCH repetitions are larger than 1, the throughput performance for [2,1,1] and [2,3,3] would be worse than [2,2] and [4,4]  > Correct, that is why we have the two rows in our contribution.  The design should enable better performance for large #TB, i.e, 6 or 8 TB scheduling, when this happens flexible bundling pattern is needed to adapt to different status (initial scheduling vs retransmission) of each TB.  > This statement is not very specific. If the initial transmission uses bundling, you cannot know which TBs failed (only which bundles failed). If the initial transmission did not use bundling, how do you plan to adjust? Could you give some examples? From the figure in your paper:  6  How does the eNB even know which TBs failed? For case 1 it seems you use bundle size of 2 to know that (bundle size of 4 would not allow for the granularity you have). In retx you would use bundle size of 2. For Case 2, you may have used bundle size of 4, you would use the same one. For case 3, in any case you cannot group them together, and if you have that pattern I guess that you didnt use bundling in the previous transmission, since otherwise you cannot get that granularity. Just to be clear, regarding your observations for case 3 in your contribution, our solution allows to dynamically enable/disable HARQ bundling.  Single or very limited fixed bundling pattern would cause the performance loss especially for the large TB number or colossal repetitions.  > Again, this is a very general statement.  Lenovo: We prefer 2 bit indication in DCI solution, which gives full flexibility to eNB scheduling, and should make the relationship between bit indication={00,01,10,11} and M={1,2,3,4}in TS36.213.  Nokia: The 1-bit solution provides some optimization but we feel the flexibility trade-off is not good enough and therefore prefer to have the full flexibility.  > The term flexibility is widely used in 3GPP, but most of the time I am not sure whether it is good or bad. Is it good for the spec to waste some overhead to give the possibility to the eNB to do something wrong? Could you elaborate how you would use this flexibility?  Huawei:  > If we follow this reasoning, we should do a bundling of 8 TBs, since it has the same time-delay and has the lowest resource utilization. The argument of saving of PUCCH resources doesnt seem to have a lot of thought behind it, honestly.  ZTE: Regarding the exact meaning of 2-bit indication, we have same understanding as Futurewei, i.e , the total number of TB is evenly divided by the bundle size indicated, each bundle has same integer number of Tb , except for the last bundle which could equal to the remainder of the integer division.  > Do you mean the reading of the current spec? Or the agreements?  Having said all this, let me write down the table by following FW’s interpretation of the agreement (with which we disagree):   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **1TB** | **2TB** | **4TB** | **6TB** | **8TB** | | **Case 1** | **M=1** | **M=2** | **M=2** | **M=2** | **M=3** | | **Case 2** | **M=1** | **M=2** | **M=2** | **M=3** | **M=4** |   With the 2-bit approach, we can signal M={1,2,3,4}, but most of these combinations are useless – let’s take the single repetition case – highlighted ones are signaled by QC proposal:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **1TB** | **2TB** | **4TB** | **6TB** | **8TB** | | **M=1** |  |  |  |  |  | | **M=2** | **Useless** | **Useless** |  |  |  | | **M=3** | **Useless** | **Useless** | **Useless** | **Useless** |  | | **M=4** | **Useless** | **Useless** | **Useless** | **Useless** | **Useless** |   So, for the single repetition case – which is the, the additional “flexibility” allows to signal an additional 11 useless states and 1 *useful* state (M=2 for 8TBs). Actually, the QC original proposal would signal the same number of “useful” states, since it would signal [2,1,1] which is better than [2,2]. |
| Sierra Wireless | To answer Ericsson’s Question: We strongly support different sized bundles in the bundling pattern. To optimize speed and minimize bundle sizes, the optimal number of bundles is three but since 8 doesn’t divide into 3 evenly, the bundles sizes must be different e.g. {3,3,2}.  Observation – The 2-bit option has been presented by some as having no or little restrictions which is simply not true. Since there are WAY more than 4 bundling patterns (especially when NTB=8), even when 2-bits are used to indicate the bundling pattern - there are scheduling restrictions so all options presented have some degree of scheduling restrictions.  For those companies that support a 2-bit field to indicate the bundling pattern, what bundling patterns do you propose? You will have to make hard choices as well! E.g. from R1-2002654, I see these choices made:   |  |  | | --- | --- | | # of TBs scheduled | HARQ-ACK Bundle Patterns (2-bit DCI) | | 1 | (1) | | 2 | (1,1) or (2) | | 4 | (1,1,1,1) or (2,2) or (3,1) or (4) | | 6 | (1,1,1,1,1,1) or (2,2,2) or (3,3) or (4,2) | | 8 | (1,1,1,1,1,1,1,1) or (2,2,2,2) or (3,3,2) or (4,4) | |
| ZTE,Sanechips 3 | To answer Qualcomm's questions  Qualcomm > This argument doesn’t make much sense. If you argue this, then you could say that the peak throughput performance is the best if we just report single HARQ-ACK bit. For a given peak throughput, we should transmit as many HARQ-ACKs as possible.  ZTE:We are saying that with good channel condition large bundling size is an advantage. We don't agree the statement “For a given peak throughput, we should transmit as many HARQ-ACKs as possible.” The uplink resource usage is more important, otherwise you can just disable bundling to get maximum number of HARQ-ACK since you want " as many HARQ-ACKs as possible."  ZTE: Additionally, in many cases, when number of PUCCH & PDSCH repetitions are larger than 1, the throughput performance for [2,1,1] and [2,3,3] would be worse than [2,2] and [4,4]  Qualcomm > Correct, that is why we have the two rows in our contribution.  ZTE: That's not still not flexible as we expect. When repetition is one and the channel condition is good, what is the meaning of using more ACK/NACK feedback? In this case, [2,1,1] and [2,3,3]are still worse than [2,2] and [4,4]  Qualcomm > This statement is not very specific. If the initial transmission uses bundling, you cannot know which TBs failed (only which bundles failed). If the initial transmission did not use bundling, how do you plan to adjust? Could you give some examples? From the figure in your paper:  ZTE: You cannot know which TBs failed, but you know which bundles failed, and the TB in the failed bundle either need retransmission or already succeeded. Then when you assign bundle pattern next time, you should choose a pattern that can bundle new TB together, at the same time those TBs from the previous failed bundle together,as we indicated in our figure. This is not possible with single pattern design.  Lastly, the so called usage table is misleading.  Qualcomm : With the 2-bit approach, we can signal M={1,2,3,4}, but most of these combinations are useless –  ZTE: for 4TB, M=3 and m=4 are not useless; similarly for 6TB, m=3 and 4 are also useful. For 8TB, M=4 isuseful. Again, we should look at uplink usage ,which is much more important. Otherwise why do we need to enable bundling? |
| Qualcomm | *Again, we should look at uplink usage ,which is much more important. Otherwise why do we need to enable bundling?*  The objective of bundling is, for HD-FDD UEs, to maximize throughput, since the UE can spend less time transmitting PUCCH. This should be clear from the Rel-14 WI:    Also, your reasoning seems a bit fuzzy here:  ZTE:We are saying that with good channel condition large bundling size is an advantage. We don't agree the statement “For a given peak throughput, we should transmit as many HARQ-ACKs as possible.” The uplink resource usage is more important, otherwise you can just disable bundling to get maximum number of HARQ-ACK since you want " as many HARQ-ACKs as possible."  > Our statement was “**For a given peak throughput**, transmit as many HARQ-ACKs as possible” (as you copied right before your conclusion). If we disable bundling completely, we get less peak throughput. |
| SONY | We are basically OK with the specification as it is. We are OK if there is a clarification of a mapping between 0->3 in 36.212 and 1->4 in 36.213.  The guidance for the meeting is that we should be focussing on “essential corrections” and a lot of the “fancier” proposals here do not seem to be essential, when there is already a working solution in the current version of the specs.  The discussion above between ZTE and Qualcomm brings up the issue (maybe rhetorically) about wherever large bundle sizes are better and / or whether no bundling at all is better. We are OK with having larger bundling sizes: fewer PUCCH means less power consumption. This is consistent with our central proposal on HARQ A/N feedback throughout this work item, which was that ACK / NACKs are bundled to a single A/N bit and if that single A/N bit indicates “NACK”, then the UE further indicates the A/N status of each TB of the MTBG transmission.  And a big thank you to the spec editors for all their diligent work in their efforts in implementing the RAN1 agreements. |

Based on the comments provided above, a majority seems to prefer to keep the currently specified 2-bit DCI field. Companies are invited to comment on their preference for how the 2-bit value should be interpreted in terms of number(s) of TBs.

|  |  |
| --- | --- |
| **Company** | **Comments on interpretation of 2-bit value** |
| Example | E.g. copy this example (picked from one of the comments above) into one of the table rows below and modify according to your preferences.   |  |  | | --- | --- | | # of TBs scheduled | HARQ-ACK Bundle Patterns (2-bit DCI) | | 1 | (1) | | 2 | (1,1) or (2) | | 4 | (1,1,1,1) or (2,2) or (3,1) or (4) | | 6 | (1,1,1,1,1,1) or (2,2,2) or (3,3) or (4,2) | | 8 | (1,1,1,1,1,1,1,1) or (2,2,2,2) or (3,3,2) or (4,4) | |
| SONY | Interpretation in the current specs is OK. Hence we are OK with the table in the “example” row above. |
| Sierra Wireless | |  |  | | --- | --- | | # of TBs scheduled | HARQ-ACK Bundle Patterns (2-bit DCI) | | 1 | (1) | | 2 | (1,1) or (2) | | 4 | (1,1,1,1) or (2,1,1) or (2,2) **or [(4)]** | | 6 | (1,1,1,1,1,1) or (2,2,2) or (3,3) **or [(4,2)]** | | 8 | (1,1,1,1,1,1,1,1) or (3,3,2) or (4,4) **or [(2,2,2,2)]** |   The patterns in [red], I don’t have a strong view on.  The logic behind this set is:   * 1 pattern for disabled bundling {1,…} * 1 pattern to optimize throughput and minimize bits/bundle for 1 repeat case – three bundles * 1 pattern to optimize throughput for > 1 repeat – two bundles * 1 pattern to fill up the list – [red]   Note: the only difference between the example bundling patterns from R1-2002654 and this one is for NTB=4 where (3,1) is replaced with (2,1,1). |
| Lenovo&MotoM | Interpretation in the current specs is OK, no need to generate any table. Only thing need to be done is: make the relationship between bit indication={00,01,10,11} in TS36.212 and M={1,2,3,4} in TS36.213.  The current spec is: the total number of TB is evenly divided by the bundle size indicated, each bundle has same integer number of Tb , except for the last bundle which could equal to the remainder of the integer division. |
| LG | |  |  | | --- | --- | | # of TBs scheduled | HARQ-ACK Bundle Patterns (2-bit DCI) | | 1 | (1) | | 2 | (1,1) or (2) | | 4 | (1,1,1,1) or (2,2) or [(2,1,1) or] (4) | | 6 | (1,1,1,1,1,1) or (2,2,2) or (3,3) ~~or (4,2)~~ | | 8 | (1,1,1,1,1,1,1,1) or (2,2,2,2) or [(3,3,2)] or (4,4) |   4 TB: Compare to (3,3) bundling, we can not find benefit from (4,2) bundling. Our best preference is to remove (3,1) bundling from the example above, and define 3 bundling pattern only. If further optimization is needed, we can consider replacing (3,2) with (2,1,1) as suggested by Sierra.  6 TB: If 2 HARQ-ACK channel can be used, we think it is better to use (3,3) bundling rather than (4,2). Thus we prefer to erase (4,2) from above example.  8 TB: Our best preference is using only evenly distributed bundle pattern. If further optimization is needed, we are ok with using (3,3,2) bundling. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# References

1. [RP-192875](https://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_86/Docs/RP-192875.zip), “Revised WID: Additional MTC enhancements for LTE”

1. [R1-1913594](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_99/Docs/R1-1913594.zip), “RAN1 agreements for Rel-16 Additional MTC Enhancements for LTE”

1. [R2-2001886](https://www.3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_109_e/Docs/R2-2001886.zip), “RAN2 agreements for Rel-16 additional enhancements for NB-IoT and MTC”

1. [R1-2001477](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001477.zip), “Updated consolidated parameter list for Rel-16 LTE”

1. [R1-2001485](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001485.zip), “RAN1 UE features list for Rel-16 LTE after RAN1#100-E”

1. [R1-1913610](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913610.zip), Addition of feature for 36.211
2. [R1-1913611](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913611.zip), Addition of feature for 36.212
3. [R1-1913612](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913612.zip), Addition of feature for 36.213 (s00-s05)
4. [R1-1913613](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913613.zip), Addition of feature for 36.213 (s06-s07)
5. [R1-1913614](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913614.zip), Addition of feature for 36.213 (s08-s09)
6. [R1-1913684](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913684.zip), Addition of feature for 36.213 (s10-s13)
7. [R1-1913615](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913615.zip), Addition of feature for 36.213 (s14-sxx)

1. [R1-1913683](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_99/Docs/R1-1913683.zip), Addition of feature for 36.214
2. [R1-2001427](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001427.zip), Corrections for 36.211
3. [R1-2001431](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001431.zip), Corrections for 36.212

1. [R1-2001433](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100_e/Docs/R1-2001433.zip), Corrections for 36.213

1. [R1-2001568](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2001568.zip), “Corrections on scheduling of multiple transport blocks”, Huawei, HiSilicon

1. [R1-2001852](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2001852.zip), “Remaining issues on scheduling enhancement for MTC”, ZTE

1. [R1-2001928](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2001928.zip), “Remaining issues on multiple transport blocks scheduling in MTC”, LG Electronics

1. [R1-2002174](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002174.zip), “Scheduling of multiple DL/UL transport blocks”, Qualcomm Incorporated

1. [R1-2002504](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002504.zip), “Corrections for Multi-TB scheduling for LTE-MTC”, Ericsson

1. [R1-2002642](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002642.zip), “Remaining issues for scheduling of multiple TBs”, Nokia, Nokia Shanghai Bell

1. [R1-2002654](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100b_e/Docs/R1-2002654.zip), “HARQ-ACK bundling for Multi-TB scheduling”, Futurewei

1. [R1-2000507](http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_100_e/Docs/R1-2000507.zip), “LTE-M Multiple Transport Block Grant Design Considerations”, Sierra Wireless
2. [R1-2002512](https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_100b_e/Docs/R1-2002512.zip), “Feature lead summary for Multi-TB scheduling for LTE-MTC”