3GPP TSG-RAN WG2 Meeting #116bis-e R2-22XXXX

Electronic Meeting, 17 – 25 January 2022

**Agenda item: 8.8.1**

**Source: CMCC**

**Title: Report for [AT116bis-e][240][Slicing] Remaining details for slice groups (CMCC)**

**WID/SID: FS\_NR\_slice**

**Document for: Discussion and Decision**

# Introduction

This document aims at address the remaining details for slice groups

* [AT116bis-e][240][Slicing] Remaining details for slice groups (CMCC)

Scope: Discuss the slice group aspects: 1) discuss what should be the definition of slice group (based on latest RAN2 and SA2 agreements)? 2) how to resolve the TA boundary aspects? 3) does UE select different slice group if no cell supporting that slice group is available?

Intended outcome: Discussion summary in R2-2201708.

Deadline: Deadline 3

Comment deadline: Thursday W1, 1600 UTC (for collecting views)

Rapporteur proposals: Friday W1, 0900 UTC (proposed resolution of issues)

Document deadline: Monday W2, 1200 UTC (report or agreed CRs)

**Contact List**

|  |  |  |
| --- | --- | --- |
| Company | Name | Email |
| Qualcomm | Peng Cheng | chengp@qti.qualcomm.com |
| Lenovo | Prateek Basu Mallick | [pmallick@lenovo.com](mailto:pmallick@lenovo.com) |
| CMCC | Jiayao Tan | tanjiayao@chinamobile.com |
| Intel Corporation | Seau Sian Lim | seau.s.lim@intel.com |
| Apple | Yuqin Chen | yuqin\_chen@apple.com |
| LGE | HyunJung Choe | [stella.choe@lge.com](mailto:stella.choe@lge.com) |
| Huawei, HiSilicon | Jun Chen | jun.chen@huawei.com |
| Nokia | Gyuri Wolfner | gyorgy.wolfner@nokai.com |
| Radisys | Manasi Padhy | manasi.padhy@radisys.com |
| NEC | Yuhua Chen | Yuhua.chen@emea.nec.com |
| OPPO | Zhe Fu | fuzhe@OPPO.com |
| Sharp | Art Ishii | [ishiia@sharplabs.com](mailto:ishiia@sharplabs.com) |
| Spreadtrum | Xiaoyu Chen | xiaoyu.chen@unisoc.com |
| Xiaomi | Xiaofei Liu | [liuxiaofei@xiaomi.com](mailto:liuxiaofei@xiaomi.com) |
| KDDI | Hiroki TAKEDA | [ho-takeda@kddi.com](mailto:ho-takeda@kddi.com) |
| Kyocera | Mitsutaka Hata | mitsutaka.hata.gt@kyocera.jp |
| CATT | Haocheng Wang | wanghaocheng@catt.cn |

# Discussion

***Open issue 1: Definition of slice grouping***

Previous agreements in RAN2#116-e:

* A network slice can be associated to none or only one slice group.

Since SA2 hasn’t reached agreement on the definition of slice grouping, RAN2 can try to define one based our understanding. A candidate definition is shown below:

**Slice group: A group which is associated with one or multiple slices. And a slice is associated to none or only one slice group.**

**Q1.1: Do you agree with the above definition for slice group? Other suggestions are also welcome.**

|  |  |
| --- | --- |
| **Company** | **Comments** |
| Qualcomm | Agree. |
| Lenovo | No. There’s no need to restrict that one slice can belong to only one slice group. This would create deployment configuration issues e.g., if a slice can’t belong to more than one group, network might be forced to keep only one slice in a group, taking away all the advantages associated to a slice group. The definition can be modified as:  **A network slice can be associated to none or more slice groups.** |
| CMCC | Agree.  RAN2 has agreed that a slice can be associated to none or only one slice group in last meeting. If a slice is allowed to be associated to more slice groups, there will be ambiguous for slice based RACH configuration, i.e. the UE can be confused to choose RACH partition. |
| Intel | Agree that from a UE perspective (that is, a slice is only associated with a max of one slice group amongst the slice groups that are in the UE’s neighbouring cells). |
| Apple | Agree. This has been already agreed, no need to re-visit. |
| LGE | Agree |
| Huawei, HiSilicon | Agree. This has been agreed in previous RAN2 meeting. If we are to change it, more issues have to be considered. |
| Nokia | No.  As we discussed in our paper R2-2200947 the limitation that maximum a slice can belong to maximum one slice group creates a dependency between slice-based cell reselection and slice specific RACH enhancements. It also makes difficult to create appropriate slice groups for cell reselection. This may make impossible to optimize the slice grouping and may make very difficult to deploy both slice-based reselection and in a network and slice specific RACH enhancements together. |
| Radisys | No.  It is better not to have any restriction of the slice association strictly to one slice. The slice grouping helps the network to allow the UE to use the similar services at times hence one slice can belong to multiple slice groups. |
| NEC | No,  In general, it is better to wait for SA2 to provide the definition.  if RAN2 define one, we would like to delete “**a slice is associated to none or only one slice group** “. Because we see a bit issue with this limitation:  For slice specific RACH, a small slice group e.g., including only one slice (slice X ) might be needed  For slice specific cell reselection, we may need a bigger slice group for signalling overhead reduction, e.g., a group including all slice (slice X and other more slice) with similar latency/capacity requirement.  In above example, with the limitation, network has to create two slice groups, slice group#1 including only X, and slice group#2 including all other slices, but both slice groups will have same slice specific cell reselection configuration: lead to duplicated signalling  Without the limitation, network can create two slice groups, slice group#1 including slice X for slice specific RACH configuration, and slice group#2 including slice X and other slice for slice specific cell reselection configuration. this has lower signalling overhead. |
| OPPO | Agree |
| Sharp | Agree |
| Spreadtrum | Agree. |
| Xiaomi | Agree. |
| KDDI | Agree |
| Kyocera | Agree. Currently, we don’t have any clear reason to allow to associate a slice with multiple slice groups. |
| CATT | Agree  We think this has been agreed in previous meeting. If we revert to the agreements, great impacts may be brought to the progress. |

Regarding to the maximum number of slice groups, the contributions [19, 30, 47] suggested the maximum number of slice groups as at most 16 slice groups, or as less as possible (e.g. 3 slice groups representing for high/medium/low degree of importance). While, [5] suggested to postpone the decision on maximum number of slice grouping configured by Network after it is clear what dedicate configurations can be configured for one slice group.

**Q1.2: What is the maximum number of slice group?**

**Option 1: 16**

**Option 2: other numbers.**

|  |  |  |
| --- | --- | --- |
| Company | Which option do you prefer | Comments |
| Qualcomm | See comments | We still see more than 1 companies are proposing to modify previous RAN2 WA on per-TA slice grouping to per-PLMN with some technique justifications. If it is per-PLMN, it is obviously up to 16 groups is not sufficient. That is why we think it seems still not a good time to determine the maximum number of slice group. We think RAN2 can agree max slice grouping number only after:   * The definition of slice grouping is finalized (including its definition and granularity) * What configuration (and its payload size) can be configured to one slice group |
| Lenovo | 2 | Assuming this question is from network’s perspective.  **Max Slice Groups:** Very unclear now. Logically, number of slice groups must be less than possible slices itself (S-NSSAIs) but it is difficult to put a number now. Number of slices itself could be in thousands:  S-NSSAI ::= CHOICE{  sst BIT STRING (SIZE (8)),  sst-SD BIT STRING (SIZE (32))  }  Calculating just with sst, we can have upto 256 slices but it is clear that even sst-SD can be used for forming slice identifications. Assuming about 1000 slice signaling possibility to be supported by Broadcast signalling and assuming that around 8 slices can be clubbed together in a slice group, we might end up supporting signalling possibility for up to 128 slice groups (7bits).  Since it is difficult to see the necessity of so many slice groups in Rel. 17/ 18 timeframe, we can go for a lower number now like 16/ 32/ 64 slice groups. |
| CMCC | Open with the number like 16/32/64 | We think the number of slice group should not be too large considering the signalling payload based on the current working assumption, i.e. the granularity is per TA.  We are open with the number like 16/32/64 slice groups. |
| Intel | 16 | From a UE perspective, 16 is sufficient as UE can be updated when there is a TA change. |
| Apple | Open with the number | We can also ask CT1 if they have any preference among the potential numbers proposed in RAN2. |
| LGE | See comments | We are fine to make Option1 as working assumption. Less number is preferred but it seems too early to decide the exact maximum number. |
| Huawei, HiSilicon | Option 1 | For slice groups, the following information need to be decided (they would impact ASN.1 part):   * For slice information in SIB/RRCRelease, what the maximum number of slice group is. We think Q1.2 is related to it * The size of slice group id   It is our understanding that not all slices need to be put to slice groups, and operators may only want to prioritize some slices for fast access, so the max number of slice group and the size of slice group id can be small values. In our paper R2-2200974, we propose the following:  **Proposal 4: As the base of further discussion, it is proposed that size of slice group ID could be 4 bits, 8 bits, or 16 bits.**  **Proposal 5: It is proposed to have at most 16 slice groups for slice information.** |
| Nokia | See comments | The maximum number of slice groups may also depend on the design of slice groups (e.g., see question above) and SA2 decisions. This is not an important issue and it can be decided later. It is a constant that can be specified during stage 3 work (ASN.1 review) |
| Radisys | Open with the number like 16/32/64 | The possible number of slice group can be large number as per range of SST and SD, however rconsidering the applicability and use cases, we are open with the number like 16/32/64 slice groups. |
| NEC | Option1 | In [40], we propose to have maximum 8 or 16 slice groups.  Whether slice group ID should be consistent in TA or in PLAN, this will impact how many bits we need to indicate a group ID. It is another issue that maximum number of slice groups will be included in sliceinfor. We think there will not be too many, otherwise the cell reselection procedure would be too heavy. |
| OPPO | See comments | At this stage, we have no strong view on the max number of slice groups, and less number is preferred. Also, we also think it is a bit early to decide the exact number. |
| Sharp | Open with the number |  |
| Spreadtrum | See comments | We are open with the number, but hope the number should be determined with the consideration of following aspects and take a good balance:   1. Slice deployment: the number of slices supported in TA in Rel-17. 2. The granularity of slice group, whether we confirm per TA configuration or need to re-consider. 3. Limitation of SIB payload size. |
| Xiaomi | See comments | We are fine to decide later, but less number is preferred as it may have much impacts on the RACH partitions number. |
| KDDI | Option1 | No strong opinion, but we don’t see much benefit on having a too finer granularity. |
| Kyocera | Option 2  16/32/64 | Considering one-to-one mapping between slice and slice group is allowed, it may be better to have more slice groups. However, a larger number of slice groups would impact signalling efficiency. Therefore, either 16, 32 or 64 is preferable. |
| CATT | Comments | We have no strong view on the max number of the slice groups. But considering the design for preamble partition in RACH common, the number of slice group should be as less as possible. |

***Open Issue 2: TA boundary***

Previous agreements in RAN2#116-e

* 3: Working assumption: The granularities of the slice groups for cell reselection are per TA. FFS on the details (e.g. how to resolve TA boundaries).

A number of contributions [4, 12,16, 19, 25, 30, 49] see issues for TA boundary and suggested to resolve the issues, while the contributions [22, 48] thought that there is no issue and no spec impacts on TA boundary.

The first open issue is that, when UE is checking whether the highest ranked cell support the highest priority slice, how the UE is aware of the slice supported by the neighbouring cell belonging to other TA?

According to the contributions, there are several potential solutions for resolving the TA boundary issue.

**Q2.1: How UE can know the supported slice for neighbouring cell at TA boundary?**

**Option A: The gNBs exchange the supported slices (S-NSSAI/NSSAI) through Xn interface, then serving gNB can map the slices supported by neighbour cells to slice groups based on the slice group mapping relationship in current TA and broadcast it to the UEs. [19]**

**Option B: As assistance information, an optional PCI list is introduced to indicate the cells supporting one slice group in a new SIB. And if NW don’t provide such info on the best ranked cell, the UE may skip the checking on slice support in best ranked cell. [4]**

**Option C: Add the association of slice group IDs and their valid TACs in the slice-specific cell reselection info for inter-TA cell reselection. The UE can determine whether to camp on the highest ranked cell based on the TAC received in its SIB1.[30]**

|  |  |  |
| --- | --- | --- |
| Company | Which option do you prefer | comments |
| Qualcomm | Option B in RAN2 | We are proponent of Option B. It is also one solution to reduce signalling overhead of slice info in SIB. In this way, the UE behaviour is clearly specified, and no RAN3 and SA2 impacts are foreseen (i.e., it is up to source cell implementation how to provide the optional PCI list)  For Option A and C, it obviously has at least RAN3 impacts. And we don’t see they are mutual exclusive to Option B. Because they are not RAN2 expertise, we suggest proponents to propose these solution in RAN3 and SA2 first. |
| Lenovo | A | Our understanding is that the slice group should have the same meaning (i.e., the associated slices) for a UE in its registration area. Therefore, the TA-list received as part of the registration area should have consistent use of Slice group. On top of this, we think that the agreement from the last meeting applies:  ***A serving cell can provide slice support of neighbour cells.***  In the (TA-list) boundary cases, the UE needs to perform RAU when reselecting to a cell of another RA (based on slice based or legacy reselection). The RAU procedure will reveal the applicable association of slice->slice group and therefore, optimizing for boundary cases is not necessary to eliminate cases when the meaning of slice group changes from one RA to another. |
| CMCC | Option A | RAN2#116-e has agreed that the serving cell can provide slice support of neighbour cells, and option A is better to achieve it.  We would like to explain a bit about option A, gNBs would exchange the supported S-NSSAI during the NG/Xn interface setup procedure, and serving gNB can map the slices supported by neighbour cells to current slice groups and broadcast it to the UEs. Then the UE will be aware of the supported slices by neighbour cells and perform slice-based cell reselection based on the broadcast slice group and the mapping between slice and slice group.    Option B can provide which neighbour cell supports current slice group of serving cell. As shown in TR 38.832, different slices can be supported on different frequencies. For example, in Location 3 showed in the figure, slice 1 is deployed in F1 and slice 2 is deployed in F2. With option B, cell 6 (as serving cell) will only provide which neighbour cell supports slice 1, and the UE cannot be aware that slice 2 can be available in F2. This will cause that the UE camps on F1 all the time and never selects F2 to camp on.  For option C, the UE needs to read SIB1 from neighbour cells to determine whether the neighbour cell supports the selected slice. In our understanding, the agreement “*A serving cell can provide slice support of neighbour cells*” means that the UE can determine whether the neighbour cell supports the selected slice based on the slice info provided by serving cell, and doesn’t need to read SIB1 in this stage. If the neighbour cell supports the selected slice and is suitable, then the UE will read SIB1 for the final check as specified in TS 38.304. But if the neighbour cell doesn’t support the selected slice, the UE will not read SIB1 and can check the next frequency if any. Option C may increase unnecessary latency and power consumption for reading SIB1 in advance. |
| Intel | Option A with comments | Serving gNB broadcast the supported slices (as slice groups) in the current cell and neighbouring cells if it is different to the current cell (as it might be at TA border). The serving gNB obtains this information about slices supported by the neighbouring gNBs – this could be in form of the slice groups or slices and the mapping could be done by the serving or neighbouring gNB. Whether the supported slices are exchange between the gNB or via RAN OAM or whether it is slices or slice groups is not in the scope of RAN2. |
| Apple | See comment | Regarding the original question, we think among multiple TA in the same RA, the configuration on slice grouping should be homogeneous. Otherwise it would have problem since UE does not perform TAU when moving across TA boundary but within RA.  However, from further check with proponent on the issue, we understand it is actually asking “**How UE can know the supported slice for neighbouring cell at ~~TA~~ RA boundary”**.  To support this, we think it requires that the mapping should be homogeneous per PLMN. But last meeting we responded to CT1/SA2 that the configuration is per TA. Do we really want to change our preference?  Assuming this is feasible, I think the global mapping between slice to slice grouping in the PLMN should be available to both UE (by NAS signaling) and gNB (by OAM). And the neighbour cell’s slicing info can be known at gNB by either OAM or Xn interface exchange, which should not be discussed here. |
| LGE | Option A |  |
| Huawei, HiSilicon | Option B, Option C | Firstly, we do not think all options are mutual exclusive, and they focus on different aspects of slice group mechanism.  Secondly, we proposed option C in our paper, and option C is based on the assumption that the serving cell can not provide slice support of neighbour cells via PCI list, so the UE may need to read neighbour cell’s SIB1 to check the slice support. In option B, an optional PCI list can be used to indicate the slice support of neighbour cells. For CMCC’s comments on option B, we think cell 6 can provide neighbour cell supportings slice 2 via slice info.  Thirdly, we are open about option A, and how slice information are exchanged between network nodes may be checked (and thus no RAN3 impacts). |
| Nokia | Option C and Option B optionally, but comments | Option A: We do not see how this can work, as TA may have many neighbouring TAs. We think that using the RA concept is not feasible as RA is UE specific, while slice grouping is valid for all UEs. On the other hand, using Xn to exchange slice grouping information may be useful.  Option B: This solution does not mandate the use of new SIB. Having PCI lists per slice group may be feasible, but overhead could be a problem. Therefore, we think that it may be applied optionally in some deployments.  Option C: We think that adding TACs for slice groups could solve the issue. It is enough to add them for slice groups that do not belong to the TA of the current cell. Option C can be used together with option B. |
| Radisys | Option A | Simpler approach to inform the slice group of the neighbour cells and enable UE to do the cell reselection based on cell rank and slice group in the TA. Assuming different cells may support different slice groups in the TA. |
| NEC | See comment | We share the same understanding as Lenovo that slice to slice grouping mapping relationship should be consistent within RA, or in another angle, it should be consistent within a serving cell’s neighbouring hood. Otherwise it will cause UE misunderstanding on the meaning of slice group.  This consistence should be guaranteed by OAM configuration in our understanding, no need further solution. Solution like option A should be discussed in other WG |
| OPPO | See comments | We already agreed that the association between the slice and the slice group is per TA. Logically, the core network should further keep the unified association configuration among the UE’s registration area. Thus, we understand the case we focused on here is that the UE is at the RA boundary where different slices are supported on the different cells belonging to the same frequency or different slices are supported on the different frequencies.  In the previous RAN2 meeting, it is agreed that the serving cell can provide slice support of neighbour cells. Thus, the UE can know the supported slice group info for the neighbouring cell belonging to another TA once the serving cell provides such information. When the UE is going to move to the cell that belonged to a different TA, the UE should perform slice-specific cell reselection by using the obtained slice group support information of serving/neighbour cells and the association that the UE receives in the current TA. The association is considered valid until TAU.  For the purpose of providing slice support of neighbour cells, the serving gNB needs to know the slice group support info of the neighbour gNBs, no matter it is the case within or across the RA boundary. How the serving gNB obtains such information should be decided by SA2 or RAN3. In addition, the slice association which is received at the current TA is different from the one of another TA. Thus, it is possible that the UE considers slice 1 is supported at another TA based on the slice association but it is actually not. Based on this, improper cell camping may be induced. We understand this drawback, but we think it is a corner case and can be tolerable. However, if companies consider this is a problem, we understand Option A can be a potential solution but it also should be discussed in RAN3 or SA2. |
| Sharp | Option A with comments | We prefer this option, but there is something to consider. Suppose a slice group of the currently serving cell consists of slices S1, S2 and S3, whereas the neighbouring cell of concern supports only S1 and S2. In such a case, the slice group cannot be used for specifying the supported slices in the neighbouring cell. Instead, these slices must be specified individually.  As pointed out by other companies, Option C requires SIB1 acquisition before determining camping, which is not in favour of UE performance. |
| Spreadtrum | See comments | For option A, it relies on gNB to remapping slices into slice group supported in current TA. One issue is that some slices cannot be remapped into existing slice groups. Another issue is that though some slice of neighbour TA can be remapped into slice group A which is supported in current TA, it doesn’t mean that neighbour TA support all the slices in the slice group A. The description of option A is still unclear that how UE can figure out the differences. And the solution should be further discussed in RAN3.  For option B, just wonder that how UE know the mapping of slice and slice group in neighbour TA. And the achieved agreement is “A serving cell can provide slice support of neighbour cells.” If slice info is broadcast per slice group, UE still cannot understand whether neighbour cell support slice group A represents that slice1 (the selected slice) is supported. So, it seems that pure option B cannot solve the issue.  For option C, the slice group is coupled with TAC. One issue is the same as option B that how UE know the mapping of slice and slice group in neighbour TA. Another issue is that UE has to read SIB1 which cause extra delay.  Just from our side, one assumption should be confirmed firstly, i.e., the mapping of slice and slice group of neighbour TA should be known to UE in advance. (e.g., The mapping of slice and slice group of neighbour TA could be indicated by NAS in advance and coupled with TAC.)  Then, a solution combined option B and C could be considered. Serving cell in TA boundaries could broadcast supported slice groups of neighbour cells and also with TAC. (e.g., provide in SIB3/4, which already includes PCI). To make UE aware of two points:   1. The neighbouring cell belongs to which TA, to avoid reading SIB1; 2. Help UE confirm the valid slice group in the TA, and check whether the selected slice is included in the supported slice group in the cell. |
| Xiaomi | See comments | From RAN2 respective, we think there is no spec impacts as UE only needs to apply the slice group identity provided by NAS and slice group specific cell reselection provided by serving cell. As for how to guarantee the consistent configuration on slice group in RA boundary can be left to NW implementation and option A can be considered as candidate solution but it is not RAN2 scope. |
| KDDI | Option A |  |
| Kyocera | See comments | The UE performs TAU whenever it goes across the TA boundary, and we assume the new slice group mapping is provided during TAU procedure. Therefore, we think that there is no issue and no specification impact on the TA boundary. |
| CATT | Option A or option D | The main issue at TA boundary is that how gNB provides the slice support information of other TAs so that UE can identify correctly. There are two alternatives:  Alt1: gNB **directly** provide slice support information in other TAs in SIB, including slice group ID in other TA+TAC+PCI list.  Alt2: gNB provide the slice support information in other TAs **according to the slice group definition in current TA in SIB**, including slice group ID in current TA + PCI list.  The option B and C are related to Alt1, but both don’t solve the issue completely. As **UE still can’t correctly identify the slice group ID in other TA**. UE only get the mapping between slice and slice group in current TA though NAS signalling. Another solution D is provided in [12] which can work with Option B and Option C to solve the issue in alt1.  **Option D：The UE obtains the slice group definition of the adjacent TAs via RA registration procedure.**  In Option D, UE can obtain the slice group definition (mapping) of all the TAs within RA via RA registration procedure. In this way, UE can correctly identify the slice group ID in other TAs. Within RA, if Option D is adopted, the issue in Q2.2 will not exist.  The Option A is also proposed in [12], which is the implement of Alt2. gNB firstly exchange the slice support information of neighbour TAs though Xn interface. Then the gNB can translate the slice support information in neighbour TAs into the slice support information according to the definition of current TA. Then the UE can correctly identify the slice support information. The option A has less impact on UE, but option A has the issued described in Q2.2. If neighbour cell supports a slice which cannot be mapped into any current slice group, gNB cannot provide the corresponding slice support information.  In summary, we are fine with Option A and Option D. |

The contribution [19, 12] pointed out the case that the neighbour cell supports a slice which cannot be mapped into any current slice group and suggested to handle this issue. There are several potential solutions for this issue:

**Q2.2: How to handle the case if the gNB doesn’t support the slice group mapping for the slice of the neighbouring cell? Do we need to send LS to RAN3/SA2?**

**Option 1: The gNB can request CN to update the mapping to involve the new slice.[19]**

**Option 2: The gNB can request RAN OAM to update the mapping to involve the new slice.**

**Option 3: Restrict that one slice is mapped to only one slice group. [12]**

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Which option do you prefer | Send LS to RAN3/SA2? | comments |
| Qualcomm | See comments | No | As we replied in Q2.1, Option 1 and Option 2 are RAN3 expertise. We suggest proponents to propose these solutions in RAN3 first. We don’t think any issue to resolve in RAN2, and also we don’t think RAN2 should trigger such discussion to RAN3. |
| Lenovo | See comments | Not sure | Not supporting should not mean a gNB can’t advertise slice group mapping for the slice of the neighbouring cell. |
| CMCC | Option1 or option 2 | Yes | As we replied in Q2.1, we understand this will happen during the the NG/Xn interface setup procedure, and the UE can obtain more assistance information which is beneficial to slice based cell reselection. |
| Intel | See comments | No | We think that this can be resolved through network deployment and there is no need to discuss this in RAN2. |
| Apple | See comments | No | Not for RAN2 to discuss. |
| LGE | See comments | No | Not for RAN2 to discuss. |
| Huawei, HiSilicon | See comments | Not sure | We think that Q2.2 is related to Q2.1. For slice group information transmission in Uu interface, there are two aspects: (1) how the slice group information is defined in RRC spec; (2) how the network generates such information. Q2.2 seems to be more about (2), and we wonder whether this issue can be handled by RAN-OAM interactions or not. |
| Nokia | See comments | No | We also think that this is mainly in the scope of RAN3. We think that RAN3 can discuss it without an LS from RAN2. This is an issue for RAN3. Companies shall contribute to RAN3 on this. |
| Radisys | Option 2 | Yes | This shall be based on network deployment and configuration. |
| NEC | See comment | No | Not for RAN2 to discuss |
| OPPO | See comments | No | Not for RAN2 to discuss. |
| Sharp | See comments | No | Not for RAN2 to discuss. |
| Spreadtrum | See comments | No | It should not be discussed by RAN2. |
| Xiaomi | See comments | No | Not for RAN2 to discuss. |
| KDDI | See comments | No | Not for RAN2 to discuss. |
| Kyocera | See  Comments | No | Same comment with Q2.1, we think there is no issue needs to be solved. |
| CATT | Option 3 | Not sure | If option 3 is adopted, this can avoid the case that the slices within a slice group in other TA map to the different slice groups in current TA. |

***Open issue 3: Consider low priority slice or not***

The contribution [8, 19] have proposed that low priority slice is considered with iteration but needs to some enhancements, for example, reuse the immediate past measurements, or set the maximum number of iterations, or set a timer for iteration, etc. Another contribution [30] proposed that RAN can indicate the UE whether to perform Step 7 or the limit times of iterations.

On the other hand, the contribution [35, 37] proposed new algorithm on frequency priority handling, in which the low priority slice is considered without iteration.

Some contributions [4, 16, 11, 33, 41] suggested to only consider the highest priority slice.

**Q3: Does UE select different slice if no cell supporting that slice is available, which option do the companies prefer?**

**Option A1: Low priority slice is considered with iteration. [8, 19, 44, 45]**

**Option A2: Low priority slice is considered, but without iteration. [35, 37]**

**Option B: Only highest priority slice considered. [4, 16, 11, 33, 41]**

|  |  |  |
| --- | --- | --- |
| Company | Which option do you prefer | Comments |
| Qualcomm | Option B with comments | We support a clearer Option B:  **Option B-1: Only highest priority slice considered, then legacy priorities considered. [4, 16, 11, 33, 41]**  For Option A1 and A2:   * We do not support them as we commented before that it will increase UE’s latency and power consumptions during cell reselection, especially if number of slice group is large. It is conflicted with the intention to introduce slice specific cell reselection. * At such late stage of Rel-17, we have to emphasize that RAN2 need to focus on how to close open issues in the remaining two meetings. In our understanding, there is even not a converged baseline solution on how to consider the low priority slice (e.g., how to converge Option A1 and Option A2?). And more and more enhancements are being proposed as Rapporteur mentioned (e.g., set the maximum number of iterations, or set a timer for iteration), which we are not sure whether they are on top of Option A1 or Option A2? We don’t think how RAN2 can make progress for these on-fly proposals in the remaining 2 meetings.   In all, although we understand Option B-1 has some performance restriction, we think it is a reasonable way forward to finalize Rel-17 RAN slicing enhancement, given the current situation. |
| Lenovo | A1 | **In case of Option A2** [35, 37]: UE after an unsuccessfully try to reselect a cell for the highest priority slice, will:  *…the UE shall use the CellReselectionPriority as reselection priority for this frequency until the highest ranked cell changes on the frequency, or new slice priorities are received from NAS.*  This approach has the following issues:  Then this will/ can lead to a situation where the UE must start with measurement of other frequencies afresh.  [Apple feedback]: This is not true. UE performs RRM measurement as legacy way. It’s not a afresh measurement.  It is possible that the highest ranked cell supports the next highest prioritized slice, but since the UE is not going to consider this frequency again until the highest ranked cell changes, the second highest ranked slice can’t be attained.  [Apple feedback]: I may not get the question. If the highest ranked cell does not support the highest prioritized slice, and assuming no other frequency to look into, and UE figures out the second highest prioritized slice is supported, UE will camp this cell.  The condition “until the highest ranked cell changes” can lead to UE continuously monitoring the highest RC – leading to battery loss.  **In case of B**, the importance of this work item is reduced to a bare minimum and is therefore un-acceptable; if e.g., there’s no frequency supporting UE’s highest priority slice, the UE falls back immediately to legacy cell reselection procedure. |
| CMCC | Option A1 or A2 | With option A1 or A2, more slices can be considered in slice-based cell reselection, and it’s beneficial when the highest priority slice is not available but the second or lower slices can be available. Option A1 and A2 can better satisfy the intention of slice-based cell reselection.  For the concern on the UE complexity and power consumption, we think the enhancements (e.g. set the maximum number of iterations, or add a timer for iteration or RAN indicate whether to perform iteration) can be considered. |
| Intel | A2 | As explained in our contribution R2-2200510, if we strictly follow step 2 in ‘Annex A’ as stated in the email disc, then in our understanding, if the highest priority slice in the UE slice list provided by NAS (i.e., URLCC) is not available in any of the frequencies in the coverage region of the UE and Step 7 in ‘Annex A’ is removed, the UE will fall back to legacy reselection mechanism.  That is, a consequence of not supporting step 7 in the current flow chart seems to be that UE will fall back to legacy frequency prioritisation if the highest priority slice is not found.  For example, if a UE has URLLC and eMBB in its slice list from NAS, and URLLC is not available in any of the inter-frequency cells in that geographical region, UE will fall back to legacy reselection and will not follow the slice based frequency priority for eMBB.  That is, UE will follow legacy cell reselection in all of the region that URLLC is not available.  We think this is not acceptable.    With Option A2, this is resolved |
| Apple | A2 (without formula) | Our contribution R2-2201110 is revised to R2-2201686 to add BT plc as a co-source.  From our perspective, A2 (without formula) is a simple way to maintain the merit of Solution 4. The only change is to let UE create the frequency pool at once, but not upon NAS selecting a slice. |
| LGE | Option A1 | We don’t see the measurement issue such that the measurement results in the previous step can be reused in the next iteration.  The UE could control the slice aware cell reselection procedure e.g., the UE can perform iteration for a few high priority slices. We should not restrict the procedure just for the highest priority slice. |
| Huawei, HiSilicon | A1 or B | As a trade-off proposed in our paper R2-2200974, RAN can indicate the UE whether to perform Step 7 or the limit times of iterations.  For A2, as proposed by some companies, it still needs more time to discuss the solution with/without formula and how formula works (more details). |
| Nokia | See comments | If RAN2 selects the approach of Annex B of [R2-2200043](https://www.3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_116bis-e/Docs/R2-2200043.zip), then option A2  If RAN2 selects the approach of Annex A of [R2-2200043](https://www.3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_116bis-e/Docs/R2-2200043.zip), then option B. Having multiple iterations (A1) has technical issues and requires RAN4 involvement.  Note also that we assume that the question is about slice groups, as slice groups are used for cell reselection. |
| Radisys | A1 | Option A1, as it is possible to achieve slice based cell reselection in case of the highest ranked cell does not support the highest priority slice. Further improvements can be done to reduce signalling overhead, latency and power consumption. |
| NEC | Option B | Reasons:   1. Clarification on option B: “**highest priority slice”**  is highest priority slice group among the ones indicated in Sliceinfo and also supported by UE. if network indicates the slice group in the SliceInfor, it means this slice group is supported in general in this area, hence it is likely that UE will find a best ranked cell on a frequency support the highest priority slice and consequently need to fall back to secondary priority slice or default priority. in summary, option B should work well in most cases 2. Even though it is ideal to take all slice groups into account, we have had lengthy discussion on option A1, we cannot conclude it at all but with more options like A2 and Ax coming up. we do not know how to converge the discussion without compromise. 3. Consider the limit time we have for Rel17, compromise to a simple but yet work solution is necessary.   NEC provided TP for option B in [41], the specification change is straightforward, moreover, it should have very little impact on UE implementation and also easy for network to tune the load among frequencies. |
| OPPO | B | We support Option B.  We understand that Option A increases the UE complexity or effort on further checking more slice support. Even if we choose Option B, i.e. the UE directly falls back to the legacy cell reselection, there is some probability that the UE can reselect to a cell which supports the second priority slice. |
| Sharp | See comments | We have no strong opinion, but it may depend on how UE derives/obtains slice priorities. May need to wait for SA2 inputs. |
| Spreadtrum | Option B | To make it simple to converge, the option B is enough in Rel-17. The UE does best effort to find a cell but excessive optimization should also be avoided.  At least for the highest priority slices, UE measures all the frequencies supporting the selected slices.  As for low priority slice, no matter with reduced iteration or no iteration, the previous measurements may cannot be totally reused. UE needs to measure new frequency. It may still hard to reach a convincing consensus like the length of timer or the max number of iterations, and to explain why those settings are beneficial for slice cell reselection.  Moreover, we still can’t guarantee that a suitable cell supporting low priority slice can be found. |
| Xiaomi | A2(without formula) | We agree with Lenovo’s comments on option B that only considering highest priority slice for cell reselection has minimum benefits.  Considering some companies have concerns on the latency caused by slice iteration in option A1, thus we’d like to adopt A2 to have the all frequency priorities determined with taking all slice into consideration but without slice iteration to avoid **measurement** latency, but to reduce the complexity, any form for the priorities calculation formula is not allowed.  We are fine with apple’s approach to have the priorities decided by text instead of formula, but it needs clarification that the priorities considering all slice is only used for measurement procedure, while for the reselection procedure, the cell check should be performed based on the frequency priorities per slice. |
| KDDI | Option A1 or A2 | With Option A1/A2, in our understanding additional mechanism are required so that the cells having second (or third or less) priority are measured in advance. It is because based on the measurement rule, those cells may have lower priorities before the iteration therefore those cells may not be included in the cell ranking list. To address this, Lenovo proposes to set the current priority the lowest (A1: relative value approach), on the other hand Ericsson/Apple proposes to set the all slice related cells above the current priorities. (A2 : absolute value approach)  We think both approaches are functionally equivalent, from our perspective, it seems to be a matter of preference. |
| Kyocera | Option A1z | When the UE-AS received multiple intended slices form the UE-NAS, we think all intended slices, i.e., 1st priority as well as other priorities, are important, since it would be considered that the UE-NAS intends to use these slices. Therefore, we prefer Option A1.  In case the UE wants to improve power consumption and cell reselection efficiency, the UE-NAS may provide only one intended slice, which behaves as same with Option B. |
| CATT | B or A1 with enhancement | We slightly prefer B. this solution is the simplest and can work well in most of cases in this stage.  If the majority companies prefer A1, considering the potential latency and power consumptions, we can accept A1 with enhancement. |

# Summary

# References

1. R2-2200043 [Post116-e][242][Slicing] Slice-based cell re-selection algorithm Ericsson
2. R2-2200044 Running 38.304 CR for RAN slicing Ericsson
3. R2-2200055 List of open issues for RAN slicing WI CMCC
4. R2-2200179 Remaining issues on slice specific cell reselection Qualcomm Incorporated
5. R2-2200180 Remaining issues on slice specific RACH Qualcomm Incorporated
6. R2-2200181 Further discussion on UE capability related to RAN slicing enhancement Qualcomm Incorporated
7. R2-2200406 Optimizations for signalling Slice Information Lenovo, Motorola Mobility
8. R2-2200407 RAN Slicing CR to 38.304 Lenovo, Motorola Mobility
9. R2-2200408 Triggers for initiating RAN slicing based cell reselections Lenovo, Motorola Mobility
10. R2-2200409 Principles of Slice based reselection Lenovo, Motorola Mobility
11. R2-2200416 Discussion on Slice based Cell Reselection CATT
12. R2-2200417 Analysis on issues of slice groups at TA boundaries CATT
13. R2-2200418 Analysis on UE capability for RAN slicing enhancement CATT
14. R2-2200510 Further considerations of slice based cell reselection Intel Corporation
15. R2-2200511 UE capability for Slicing enhancement Intel Corporation
16. R2-2200636 Consideration on slice based cell reselection Spreadtrum Communications
17. R2-2200697 Considerations on UE capability for RAN slicing Beijing Xiaomi Software Tech
18. R2-2200844 Open issues list for RAN Slicing CMCC
19. R2-2200845 Discussion on open issues for slice based cell reselection CMCC
20. R2-2200846 Discussion on open issues for slice based RACH configuration CMCC
21. R2-2200847 Discussion on UE capability for RAN slicing enhancement CMCC
22. R2-2200929 Consideration on slice-specific cell reselection OPPO
23. R2-2200930 Consideration on slice-specific RACH OPPO
24. R2-2200931 Consideration on UE capability for Slicing OPPO
25. R2-2200947 Considerations on slice groups Nokia, Nokia Shanghai Bell
26. R2-2200948 Text Proposals for the draft 38.304 PCR Nokia, Nokia Shanghai Bell
27. R2-2200949 Cell reselection delay for option B and option C Samsung R&D Institute India
28. R2-2200972 Report of [Post116-e][243][Slicing] Running NR RRC CR for RAN slicing (Huawei) Huawei
29. R2-2200973 Running NR RRC CR for RAN slicing Huawei, HiSilicon
30. R2-2200974 Discussion on slice based cell reselection under network control Huawei, HiSilicon
31. R2-2200975 Discussion on slice based RACH configuration Huawei, HiSilicon
32. R2-2200976 Discussion on UE capabilities for RAN slicing Huawei, HiSilicon
33. R2-2201005 Leftover issues in slice based cell reselection ZTE corporation, Sanechips
34. R2-2201050 Detailed RRC signalling for RACH prioritization configuration Nokia, Nokia Shanghai Bell
35. R2-2201110 Text proposal for slice based cell reselection under NW control Apple
36. R2-2201111 Slice based RACH configuration Apple
37. R2-2201169 On slice-based cell re-selection TP for 38.304 Ericsson
38. R2-2201170 RACH for RAN slicing enhancement Ericsson
39. R2-2201171 UE Capabilities for Slice- based Cell re-selection Ericsson
40. R2-2201190 Slice-Info provision NEC Telecom MODUS Ltd.
41. R2-2201192 Slice-based cell re-selection TP for solution 4C NEC Telecom MODUS Ltd.
42. R2-2201200 Slice information provided by RRCRelease Sharp
43. R2-2201208 Discussion on signalling slice information LG Electronics UK
44. R2-2201209 Discussion on slice based cell reselection LG Electronics UK
45. R2-2201389 A couple of FFS for Cell Reselection Kyocera
46. R2-2201406 Discussion on Slice Aware UL BSR RadiSys, Reliance JIO
47. R2-2201409 Considerations on remaining issues for slice based RACH Beijing Xiaomi Software Tech
48. R2-2201410 Resolving the common issues in slice based cell reselection Beijing Xiaomi Software Tech
49. R2-2201417 Further consideration on slice specific RACH ZTE corporation, Sanechips
50. R2-2201418 TP for system information and slice based reselection priority handling ZTE corporation, Sanechips
51. R2-2201422 On selection of Solution 4 Option A, B and C Samsung R&D Institute UK
52. R2-2201443 Remaining Issues on Slice Information Samsung R&D Institute UK
53. R2-2201475 Remaining issues on slice based RACH prioritization LG Electronics Inc.
54. R2-2201536 38.321 running CR for RAN Slicing OPPO