**TSG-RAN Meeting #22**                                            *RP-030710*
**Maui, USA, 9-12 December 2003**

**Title:**          **Proposed update of agreed CRs 2150-2152 to TS 25.331 on Minimum UE capability class**

**Source:**          **Ericsson**

**Agenda item:**          **7.3.3**

| Spec | CR | Rev | Phase | Subject | Cat | Version-Current | Version-New | Doc-2nd-Level | Workitem |
|------|------|-----|-------|---------|-----|-----------------|-------------|---------------|----------|
| 25.331 | 2150 | 2 | R99 | Minimum UE capability class | F | 3.16.0 | 3.17.0 | R2-032713 | TEI |
| 25.331 | 2151 | 2 | Rel-4 | Minimum UE capability class | A | 4.11.0 | 4.12.0 | R2-032714 | TEI |
| 25.331 | 2152 | 2 | Rel-5 | Minimum UE capability class | F | 5.6.0 | 5.7.0 | R2-032715 | TEI5 |

CR-Form-v7

# CHANGE REQUEST

| ⌘ | **25.331** CR **2150** | ⌘ rev | **2** | ⌘ | Current version: | **3.16.0** | ⌘ |

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

**Proposed change affects:**  UICC apps⌘ ☐   ME **X** Radio Access Network ☐  Core Network ☐

| **Title:** | ⌘ | Minimum UE capability class |
|---|---|---|

| **Source:** | ⌘ | Ericsson |
|---|---|---|

| **Work item code:** | ⌘ | TEI | | **Date:** ⌘ | Nov 2003 |
|---|---|---|---|---|---|

| **Category:** | ⌘ | **F** | | **Release:** ⌘ | R99 |
|---|---|---|---|---|---|

Use <u>one</u> of the following categories:
**F** *(correction)*
**A** *(corresponds to a correction in an earlier release)*
**B** *(addition of feature),*
**C** *(functional modification of feature)*
**D** *(editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP <u>TR 21.900</u>.

Use <u>one</u> of the following releases:
2 *(GSM Phase 2)*
R96 *(Release 1996)*
R97 *(Release 1997)*
R98 *(Release 1998)*
R99 *(Release 1999)*
Rel-4 *(Release 4)*
Rel-5 *(Release 5)*
Rel-6 *(Release 6)*

| **Reason for change:** | ⌘ | A minimum radio access capability parameter combination is not defined. |
|---|---|---|

| **Summary of change:** | ⌘ | The radio access parameter values for a new reference combination called 12kbps class UE are added for FDD. It is defined that this 12kbps class UE is the minimum UE.

1. The capability values that exist in signalling but is below the capabilty of the 12 kbps class UE will never be used and are removed from the signalling.

2. A sentence is added to make sure that a UE with the lowest possible UE capability still should perform in a predictable manner in NWs utilising a high bitrate common channel configuration.
For 12 kbps class Ues it is a shall in R'99 and onwards.
For other Ues there is only a shall in Rel-5.

Impact analysis:
UEs with capabilities lower than for 32kbps UE class are affected.
UEs that do not implement the CR may be unable to support RAB combinations e.g. on common channels, like SCCPCH combinations. Specifically they can fail to read the FACH. For these Ues there are examples in 34.108 that will not be supported.
UEs with capabilities equal to or higher than the 32kbps class are also affected in Rel-5. If these Ues do not implement the CR they may be unable to support RAB combinations other than existing 34.108 common channels configurations, if these require higher capabilities than the 32kbps class.

The CR has no impact on UTRAN. |
|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| **Consequences if not approved:** | ⌘ | There is no definition of the minimum UE capabilities that can be expected by the network. Operators may use RAB combinations that can not be supported by UEs with insufficient UE capabilities. In consequence UEs may be unable to access the system. | | | |

| | | | | | |
|---|---|---|---|---|---|
| **Clauses affected:** | ⌘ | 8.1.1.6.5, 8.1.1.6.6, 8.5.19, 10.3.3.25, 10.3.3.34, 10.3.3.40, 11.3 | | | |

| | | **Y** | **N** | | | |
|---|---|---|---|---|---|---|
| **Other specs affected:** | ⌘ | X | | Other core specifications | ⌘ | 25.306 |
| | | | X | Test specifications | | |
| | | | X | O&M Specifications | | |

| | | | |
|---|---|---|---|
| **Other comments:** | ⌘ | In revision 2 of this CR the last paragraph changed in section 8.1.1.6.5 and 8.1.1.6.6 are rephrased to be normative text instead of a NOTE. | |

**How to create CRs using this form:**

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks"  feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text.  Delete those parts of the specification which are not relevant to the change request.

### 8.1.1.6.5    System Information Block type 5

The UE should store all relevant IEs included in this system information block. The UE shall:

1> if in connected mode, and System Information Block type 6 is indicated as used in the cell:

2> read and act on information sent in System Information Block type 6.

1> replace the TFS of the RACH with the one stored in the UE if any;

1> let the physical channel(s) of type PRACH given by the IE(s) "PRACH info" be the default in uplink for the PRACH if UE is in CELL_FACH state;

1> use the first instance of the list of transport formats as in the IE "RACH TFS" for the used RACH received in the IE "PRACH system information list" when using the CCCH;

1> start to receive the physical channel of type AICH using the parameters given by the IE "AICH info" (FDD only) when given allocated PRACH is used;

1> replace the TFS of the FACH/PCH with the one stored in the UE if any;

1> select a Secondary CCPCH as specified in [4] and in subclause 8.5.19, and start to receive the physical channel of type PICH associated with the PCH carried by the selected Secondary CCPCH using the parameters given by the IE "PICH info" if UE is in Idle mode or in CELL_PCH or URA_PCH state;

1> start to monitor its paging occasions on the selected PICH if UE is in Idle mode or in CELL_PCH or URA_PCH state;

1> start to receive the selected physical channel of type Secondary CCPCH using the parameters given by the IE(s) "Secondary CCPCH info" if UE is in CELL_FACH state;

1> in TDD:

2> use the IE "TDD open loop power control" as defined in subclause 8.5.7 when allocated PRACH is used;

2> if the IE "PDSCH system information" and/or the IE "PUSCH system information" is included:

3> store each of the configurations given there with the associated identity given in the IE "PDSCH Identity" and/or "PUSCH Identity" respectively. For every configuration, for which the IE "SFN Time info" is included, the information shall be stored for the duration given there.

If a UE is a 12 kbps class UE according to [35] and the UE has a lower capability than required to support all transport channel configurations mapped on a specific Secondary CCPCH, the UE shall at a certain time instant still be able to decode those transport channels mapped on this Secondary CCPCH that do match the capability supported by the UE. The UE shall use the TFCI bits for that Secondary CCPCH, to distinguish a transport channel configuration that is supported by the UE from a transport channel configuration that is not supported by the UE.

In particular if the UE is a 12 kbps class UE according to [35] and it does not support the processing requirement at a given point in time for a Secondary CCPCH, it shall still be able to decode the same Secondary CCPCH when the processing requirement is consistent with the UE capability. Or if the UE does not support the number of TFs or the coding of a certain transport channel on a Secondary CCPCH, it shall still be able to decode other transport channels mapped on the same Secondary CCPCH that is consistent with what is supported by the UE.

### 8.1.1.6.6    System Information Block type 6

If in connected mode, the UE should store all relevant IEs included in this system information block. The UE shall:

1> replace the TFS of the RACH with the one stored in the UE if any;

1> let the physical channel(s) of type PRACH given by the IE(s) "PRACH info" be the default in uplink if UE is in CELL_FACH state. If the IE "PRACH info" is not included, the UE shall read the corresponding IE(s) in System Information Block type 5 and use that information to configure the PRACH;

1> start to receive the physical channel of type AICH using the parameters given by the IE "AICH info" when associated PRACH is used. If the IE "AICH info" is not included, the UE shall read the corresponding IE in System Information Block type 5 and use that information (FDD only);

1> replace the TFS of the FACH/PCH with the one stored in the UE if any;

1> select a Secondary CCPCH as specified in [4] and in subclause 8.5.19, and start to receive the physical channel of type PICH associated with the PCH carried by the selected Secondary CCPCH using the parameters given by the IE "PICH info" if the UE is in CELL_PCH or URA_PCH state. If the IE "PICH info" is not included, the UE shall read the corresponding IE in System Information Block type 5 and use that information;

1> start to monitor its paging occasions on the selected PICH if the UE is in CELL_PCH or URA_PCH state;

1> start to receive the selected physical channel of type Secondary CCPCH using the parameters given by the IE(s) "Secondary CCPCH info" if the UE is in CELL_FACH state. If the IE "Secondary CCPCH info" is not included, the UE shall read the corresponding IE(s) in System Information Block type 5 and use that information;

1> in TDD: use the IE "TDD open loop power control" as defined in subclause 8.5.7;

1> in TDD: if the IE "PDSCH system information" and/or the IE "PUSCH system information" is included, store each of the configurations given there with the associated identity given in the IE "PDSCH Identity" and/or "PUSCH Identity" respectively. For every configuration, for which the IE "SFN Time info" is included, the information shall be stored for the duration given there.

If in idle mode, the UE shall not use the values of the IEs in this system information block.

If a UE is a 12 kbps class UE according to [35] and the UE has a lower capability than required to support all transport channel configurations mapped on a specific Secondary CCPCH, the UE shall at a certain time instant still be able to decode those transport channels mapped on this Secondary CCPCH that do match the capability supported by the UE. The UE shall use the TFCI bits for that Secondary CCPCH, to distinguish a transport channel configuration that is supported by the UE from a transport channel configuration that is not supported by the UE.

In particular if the UE is a 12 kbps class UE according to [35] and it does not support the processing requirement at a given point in time for a Secondary CCPCH, it shall still be able to decode the same Secondary CCPCH when the processing requirement is consistent with the UE capability. Or if the UE does not support the number of TFs or the coding of a certain transport channel on a Secondary CCPCH, it shall still be able to decode other transport channels mapped on the same Secondary CCPCH that is consistent with what is supported by the UE.

## 8.5.19   Secondary CCPCH selection

In UTRAN Connected mode, the UE shall select the Secondary CCPCH according to the following rules:

1> in Cell_DCH state:

2> select Secondary CCPCH according to subclause 8.6.6.4.

1> in Cell_FACH state:

2> if System Information Block type 6 is defined and includes one or more SCCPCH that carry a FACH, compile a list of candidate SCCPCH that consists of these SCCPCH, in the order of appearance in System Information Block type 6.

NOTE 1:   An SCCPCH carries a FACH if the size of the "FACH/PCH information" list within the IE "Secondary CCPCH system information" exceeds 1 or if the size of this list equals 1 while IE "Secondary CCPCH system information" does not contain an IE "PICH info".

2> otherwise:

3> compile a list of candidate SCCPCH that consists of the SCCPCH(s) included in System Information Block type 5 that carry a FACH, in the order of appearance in System Information Block type 5.

2> select an SCCPCH from the list of candidate SCCPCHs based on U-RNTI as follows:

"Index of selected SCCPCH" = U-RNTI mod K,

where K is equal to the number of candidate SCCPCHs.

1> in Cell_PCH and URA_PCH states:

2> if System Information Block type 6 is defined and includes one or more SCCPCH that carry a PCH, compile a list of candidate SCCPCH that consists of these SCCPCH, in the order of appearance in System Information Block type 6

NOTE 2: An SCCPCH carries a PCH if the IE "Secondary CCPCH system information" contains IE "PICH info"

2> otherwise compile a list of candidate SCCPCH that consists of the SCCPCH(s) included in System Information Block type 5 that that carry a PCH , in the order of appearance in System Information Block type 5

2> select an SCCPCH from the list of candidate SCCPCHs based on U-RNTI as follows:

"Index of selected SCCPCH" = U-RNTI mod K,

where K is equal to the number of candidate SCCPCHs.

The UE shall set the CFN in relation to the SFN of the current cell according to subclause 8.5.15.

The UE shall ~~support reception~~ decode ~~of~~ all transport formats on all FACHs multiplexed on the selected S-CCPCH according to its UE capability, as defined in section 8.1.1.6.5 and 8.1.1.6.6, to find blocks adressed to the UE.

## 10.3.3.25    Physical channel capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| Downlink physical channel capability information elements | | | | |
| FDD downlink physical channel capability | CH-*fdd_req_sup* | | | |
| >Max no DPCH/PDSCH codes | MP | | Integer (1..8) | Maximum number of DPCH/PDSCH codes to be simultaneously received |
| >Max no physical channel bits received | MP | | Integer (~~600,~~1200, 2400, 3600, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 48000, 57600, 67200, 76800) | Maximum number of physical channel bits received in any 10 ms interval (DPCH, PDSCH, S-CCPCH) |
| >Support for SF 512 | MP | | Boolean | TRUE means supported |
| >Support of PDSCH | MP | | Boolean | TRUE means supported |
| >Simultaneous reception of SCCPCH and DPCH | MP | | Boolean | TRUE means supported |
| >Simultaneous reception of SCCPCH, DPCH and PDSCH | CV-*if_sim_rec_pdsch_sup* | | Boolean | TRUE means supported |
| >Max no of S-CCPCH RL | CV-*if_sim_rec* | | Integer(1) | Maximum number of simultaneous S-CCPCH radio links |
| >Support of dedicated pilots for channel estimation | MD | | Enumerated (true) | Presence of this element means supported and absence not supported. Note 1. |
| TDD downlink physical channel capability | CH-*tdd_req_su* | | | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| *p* | | | | |
| >Maximum number of timeslots per frame | MP | | Integer (1..14) | |
| >Maximum number of physical channels per frame | MP | | Integer (~~1~~5..224) | |
| >Minimum SF | MP | | Integer (1, 16) | |
| >Support of PDSCH | MP | | Boolean | TRUE means supported |
| >Maximum number of physical channels per timeslot | MP | | Integer (~~1~~5..16) | |
| **Uplink physical channel capability information elements** | | | | |
| FDD uplink physical channel capability | CH-*fdd_req_sup* | | | |
| >Maximum number of DPDCH bits transmitted per 10 ms | MP | | Integer (600, 1200, 2400, 4800. 9600, 19200. 28800, 38400, 48000, 57600) | |
| >Support of PCPCH | MP | | Boolean | TRUE means supported |
| TDD uplink physical channel capability | CH-*tdd_req_sup* | | | |
| >Maximum Number of timeslots per frame | MP | | Integer (1..14) | |
| >Maximum number of physical channels per timeslot | MP | | Integer (1, 2) | |
| >Minimum SF | MP | | Integer (1, 2, 4, 8, ~~16~~) | |
| >Support of PUSCH | MP | | Boolean | TRUE means supported |

| Condition | Explanation |
|---|---|
| *if_sim_rec_pdsch_sup* | The IE is mandatory present if the IE "Simultaneous reception of SCCPCH and DPCH" = True and IE Support of PDSCH = True. Otherwise this field is not needed in the message. |
| *if_sim_rec* | The IE is mandatory present if the IE "capability Simultaneous reception of SCCPCH and DPCH" = True. Otherwise this field is not needed in the message. |
| *tdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "TDD" or "FDD/TDD" and a TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |
| *fdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "FDD" or "FDD/TDD" and a FDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |

NOTE 1: These performance requirements are defined in Release 5.

## 10.3.3.34 RLC capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| Total RLC AM buffer size | MP | | Integer (~~2,~~10,50,100,150,500,1000) | Total receiving and transmitting RLC AM buffer capability in kBytes. One spare value is needed. |
| Maximum RLC AM Window Size | MP | | Integer(2047,4095) | Maximum supported RLC TX and RX window in UE |
| Maximum number of AM entities | MP | | Integer (~~3,~~4,5,6,8,16,30) | |

## 10.3.3.40 Transport channel capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| **Downlink transport channel capability information elements** | | | | |
| Max no of bits received | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all transport blocks received at an arbitrary time instant |
| Max convolutionally coded bits received | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all convolutionally coded transport blocks received at an arbitrary time instant |
| Max turbo coded bits received | CV-*turbo_dec_sup* | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all turbo coded transport blocks received at an arbitrary time instant |
| Maximum number of simultaneous transport channels | MP | | Integer(4, 8, 16, 32) | |
| Maximum number of simultaneous CCTrCH | MP | | Integer (1..8) | |
| Max no of received transport blocks | MP | | Integer(4, 8, 16, 32, 48, 64, 96, 128, 256, 512) | Maximum total number of transport blocks received within TTIs that end at within the same 10ms interval |
| Maximum number of TFC | MP | | Integer(16, 32, 48, 64, 96, 128, 256, 512, 1024) | |
| Maximum number of TF | MP | | Integer(32, 64, 128, 256, 512, 1024) | |
| Support for turbo decoding | MP | | Boolean | TRUE means supported |
| **Uplink transport channel capability information** | | | | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| **elements** | | | | |
| Max no of bits transmitted | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all transport blocks transmitted at an arbitrary time instant |
| Max convolutionally coded bits transmitted | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all convolutionally coded transport blocks transmitted at an arbitrary time instant |
| Max turbo coded bits transmitted | CV-*turbo_enc_sup* | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all turbo coded transport blocks transmitted at an arbitrary time instant |
| Maximum number of simultaneous transport channels | MP | | Integer(~~2,~~ 4, 8, 16, 32) | |
| Maximum number of simultaneous CCTrCH of DCH type | CH-*tdd_req_sup* | | Integer (1..8) | |
| Max no of transmitted transport blocks | MP | | Integer(~~2,~~ 4, 8, 16, 32, 48, 64, 96, 128, 256, 512) | Maximum total number of transport blocks transmitted within TTIs that start at the same time |
| Maximum number of TFC | MP | | Integer(~~4, 8,~~ 16, 32, 48, 64, 96, 128, 256, 512, 1024) | |
| Maximum number of TF | MP | | Integer(32, 64, 128, 256, 512, 1024) | |
| Support for turbo encoding | MP | | Boolean | TRUE means supported |

| Condition | Explanation |
|---|---|
| *turbo_dec_sup* | The IE is mandatory present if the IE "Support of turbo decoding" = True. Otherwise this field is not needed in the message. |
| *turbo_enc_sup* | The IE is mandatory present if the IE "Support of turbo encoding" = True. Otherwise this field is not needed in the message. |
| *tdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "TDD" or "FDD/TDD" and a TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |

## 11.3	Information element definitions

```
[…]

RLC-Capability ::=                      SEQUENCE {
    totalRLC-AM-BufferSize                  TotalRLC-AM-BufferSize,
    maximumRLC-WindowSize                   MaximumRLC-WindowSize,
    maximumAM-EntityNumber                  MaximumAM-EntityNumberRLC-Cap
}
[…]

TotalRLC-AM-BufferSize ::=               ENUMERATED {
                                            kb2dummy, kb10, kb50, kb100,
                                            kb150, kb500, kb1000, spare }
[…]
MaximumAM-EntityNumberRLC-Cap ::=        ENUMERATED {
                                            am3dummy, am4, am5, am6,
                                            am8, am16, am30 }
[…]
DL-TransChCapability ::=                 SEQUENCE {
    maxNoBitsReceived                       MaxNoBits,
    maxConvCodeBitsReceived                 MaxNoBits,
    turboDecodingSupport                    TurboSupport,
    maxSimultaneousTransChs                 MaxSimultaneousTransChsDL,
    maxSimultaneousCCTrCH-Count             MaxSimultaneousCCTrCH-Count,
    maxReceivedTransportBlocks              MaxTransportBlocksDL,
    maxNumberOfTFC                          MaxNumberOfTFC-DL,
    maxNumberOfTF                           MaxNumberOfTF
}
[…]
MaxNoBits ::=                            ENUMERATED {
                                            b640, b1280, b2560, b3840, b5120,
                                            b6400, b7680, b8960, b10240,
                                            b20480, b40960, b81920, b163840 }
[…]
MaxSimultaneousTransChsDL ::=            ENUMERATED {
                                            e4, e8, e16, e32 }
[…]
MaxTransportBlocksDL ::=                 ENUMERATED {
                                            tb4, tb8, tb16, tb32, tb48,
                                            tb64, tb96, tb128, tb256, tb512 }
[…]
MaxNumberOfTFC-DL ::=                    ENUMERATED {
                                            tfc16, tfc32, tfc48, tfc64, tfc96,
                                            tfc128, tfc256, tfc512, tfc1024 }
[…]
TurboSupport ::=                         CHOICE {
    notSupported                            NULL,
    supported                               MaxNoBits
[…]
UL-TransChCapability ::=                 SEQUENCE {
    maxNoBitsTransmitted                    MaxNoBits,
    maxConvCodeBitsTransmitted              MaxNoBits,
    turboEncodingSupport                    TurboSupport,
    maxSimultaneousTransChs                 MaxSimultaneousTransChsUL,
    modeSpecificInfo                        CHOICE {
        fdd                                     NULL,
        tdd                                     SEQUENCE {
            maxSimultaneousCCTrCH-Count             MaxSimultaneousCCTrCH-Count
        }
    },
    maxTransmittedBlocks                    MaxTransportBlocksUL,
    maxNumberOfTFC                          MaxNumberOfTFC-UL,
    maxNumberOfTF                           MaxNumberOfTF
}
[…]
MaxSimultaneousTransChsUL ::=            ENUMERATED {
                                            e2dummy, e4, e8, e16, e32 }
[…]
MaxTransportBlocksUL ::=                 ENUMERATED {
                                            tb2dummy, tb4, tb8, tb16, tb32, tb48,
                                            tb64, tb96, tb128, tb256, tb512 }
[…]
MaxNumberOfTFC-UL ::=                    ENUMERATED {
                                            tfc4dummy1, tfc8dummy2, tfc16, tfc32, tfc48, tfc64,
                                            tfc96, tfc128, tfc256, tfc512, tfc1024 }
```

```
[…]
DL-PhysChCapabilityFDD ::=              SEQUENCE {
    maxNoDPCH-PDSCH-Codes                   INTEGER (1..8),
    maxNoPhysChBitsReceived                 MaxNoPhysChBitsReceived,
    supportForSF-512                        BOOLEAN,
    supportOfPDSCH                          BOOLEAN,
    simultaneousSCCPCH-DPCH-Reception    SimultaneousSCCPCH-DPCH-Reception
}
[…]
DL-PhysChCapabilityFDD-v380ext ::=        SEQUENCE {
    supportOfDedicatedPilotsForChEstimation    SupportOfDedicatedPilotsForChEstimation    OPTIONAL
}

SupportOfDedicatedPilotsForChEstimation ::=         ENUMERATED { true }
[…]
MaxNoPhysChBitsReceived ::=          ENUMERATED {
                                         b600dummy, b1200, b2400, b3600,
                                         b4800, b7200, b9600, b14400,
                                         b19200, b28800, b38400, b48000,
                                         b57600, b67200, b76800 }
[…]
DL-PhysChCapabilityTDD ::=           SEQUENCE {
    maxTS-PerFrame                       MaxTS-PerFrame,
    maxPhysChPerFrame                    MaxPhysChPerFrame,
    minimumSF                            MinimumSF-DL,
    supportOfPDSCH                       BOOLEAN,
    maxPhysChPerTS                       MaxPhysChPerTS
}
[…]
-- the values 1 …4 for MaxPhysChPerFrame are not used in this version of the protocol
MaxPhysChPerFrame ::=                INTEGER (1..224)
[…]
-- the values 1 …4 for MaxPhysChPerTS are not used in this version of the protocol
MaxPhysChPerTS ::=                   INTEGER (1..16)
[…]
UL-PhysChCapabilityFDD ::=           SEQUENCE {
    maxNoDPDCH-BitsTransmitted           MaxNoDPDCH-BitsTransmitted,
    supportOfPCPCH                       BOOLEAN
}
[…]
MaxNoDPDCH-BitsTransmitted ::=       ENUMERATED {
                                         b600, b1200, b2400, b4800,
                                         b9600, b19200, b28800, b38400,
                                         b48000, b57600 }
[…]
UL-PhysChCapabilityTDD ::=           SEQUENCE {
    maxTS-PerFrame                       MaxTS-PerFrame,
    maxPhysChPerTimeslot                 MaxPhysChPerTimeslot,
    minimumSF                            MinimumSF-UL,
    supportOfPUSCH                       BOOLEAN
}
[…]
MinimumSF-UL ::=                     ENUMERATED {
                                         sf1, sf2, sf4, sf8, sf16 dummy }
```

⌘ CR-Form-v7

# CHANGE REQUEST

| ⌘ | **25.331 CR 2151** | ⌘ **rev** | **2** | ⌘ | Current version: | **4.11.0** | ⌘ |

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

**Proposed change affects:** UICC apps⌘ ☐    ME **X** Radio Access Network ☐   Core Network ☐

| | | | | |
|---|---|---|---|---|
| **Title:** | ⌘ | Minimum UE capability class | | |
| **Source:** | ⌘ | Ericsson | | |
| **Work item code:** ⌘ | TEI | | **Date:** ⌘ | Nov 2003 |
| **Category:** | ⌘ **A** | | **Release:** ⌘ | Rel-4 |

Use <u>one</u> of the following categories:
   **F** *(correction)*
   **A** *(corresponds to a correction in an earlier release)*
   **B** *(addition of feature),*
   **C** *(functional modification of feature)*
   **D** *(editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP <u>TR 21.900</u>.

Use <u>one</u> of the following releases:
  *2*     *(GSM Phase 2)*
  *R96*  *(Release 1996)*
  *R97*  *(Release 1997)*
  *R98*  *(Release 1998)*
  *R99*  *(Release 1999)*
  *Rel-4*  *(Release 4)*
  *Rel-5*  *(Release 5)*
  *Rel-6*  *(Release 6)*

| | |
|---|---|
| **Reason for change:** ⌘ | A minimum radio access capability parameter combination is not defined. |

**Summary of change:** ⌘ The radio access parameter values for a new reference combination called 12kbps class UE are added for FDD. It is defined that this 12kbps class UE is the minimum UE.

1. The capability values that exist in signalling but is below the capabilty of the 12 kbps class UE will never be used and are removed from the signalling.

2. A sentence is added to make sure that a UE with the lowest possible UE capability still should perform in a predictable manner in NWs utilising a high bitrate common channel configuration.
For 12 kbps class Ues it is a shall in R'99 and onwards.
For other Ues there is only a shall in Rel-5.

Impact analysis:
UEs with capabilities lower than for 32kbps UE class are affected.
UEs that do not implement the CR may be unable to support RAB combinations e.g. on common channels, like SCCPCH combinations. Specifically they can fail to read the FACH. For these Ues there are examples in 34.108 that will not be supported.
UEs with capabilities equal to or higher than the 32kbps class are also affected in Rel-5. If these Ues do not implement the CR they may be unable to support RAB combinations other than existing 34.108 common channels configurations, if these require higher capabilities than the 32kbps class.

The CR has no impact on UTRAN.

| | | |
|---|---|---|
| ***Consequences if not approved:*** | ⌘ | There is no definition of the minimum UE capabilities that can be expected by the network. Operators may use RAB combinations that can not be supported by UEs with insufficient UE capabilities. In consequence UEs may be unable to access the system. |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 8.1.1.6.5, 8.1.1.6.6, 8.5.19, 10.3.3.25, 10.3.3.34, 10.3.3.40, 11.3 |

| | | | | |
|---|---|---|---|---|
| | | **Y** | **N** | |
| ***Other specs affected:*** | ⌘ | **X** | | Other core specifications ⌘ 25.306 |
| | | | **X** | Test specifications |
| | | | **X** | O&M Specifications |

| | | |
|---|---|---|
| ***Other comments:*** | ⌘ | In revision 2 of this CR the last paragraph changed in section 8.1.1.6.5 and 8.1.1.6.6 are rephrased to be normative text instead of a NOTE. |

**How to create CRs using this form:**

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

### 8.1.1.6.5 System Information Block type 5

The UE should store all relevant IEs included in this system information block. The UE shall:

1> if in connected mode, and System Information Block type 6 is indicated as used in the cell:

2> read and act on information sent in System Information Block type 6.

1> replace the TFS of the RACH with the one stored in the UE if any;

1> let the physical channel(s) of type PRACH given by the IE(s) "PRACH info" be the default in uplink for the PRACH if UE is in CELL_FACH state;

1> start to receive the physical channel of type AICH using the parameters given by the IE "AICH info" (FDD only) when given allocated PRACH is used;

1> use the first instance of the list of transport formats as in the IE "RACH TFS" for the used RACH received in the IE "PRACH system information list" when using the CCCH;

1> replace the TFS of the FACH/PCH with the one stored in the UE if any;

1> select a Secondary CCPCH as specified in [4] and in subclause 8.5.19, and start to receive the physical channel of type PICH associated with the PCH carried by the selected Secondary CCPCH using the parameters given by the IE "PICH info" if UE is in Idle mode or in CELL_PCH or URA_PCH state;

1> start to monitor its paging occasions on the selected PICH if UE is in Idle mode or in CELL_PCH or URA_PCH state;

1> start to receive the selected physical channel of type Secondary CCPCH using the parameters given by the IE(s) "Secondary CCPCH info" if UE is in CELL_FACH state;

1> in 3.84 Mcps TDD:

2> use the IE "TDD open loop power control" as defined in subclause 8.5.7 when allocated PRACH is used.

1> in TDD:

2> if the IE "PDSCH system information" and/or the IE "PUSCH system information" is included:

3> store each of the configurations given there with the associated identity given in the IE "PDSCH Identity" and/or "PUSCH Identity" respectively. For every configuration, for which the IE "SFN Time info" is included, the information shall be stored for the duration given there.

If a UE is a 12 kbps class UE according to [35] and the UE has a lower capability than required to support all transport channel configurations mapped on a specific Secondary CCPCH, the UE shall at a certain time instant still be able to decode those transport channels mapped on this Secondary CCPCH that do match the capability supported by the UE. The UE shall use the TFCI bits for that Secondary CCPCH, to distinguish a transport channel configuration that is supported by the UE from a transport channel configuration that is not supported by the UE.

In particular if the UE is a 12 kbps class UE according to [35] and it does not support the processing requirement at a given point in time for a Secondary CCPCH, it shall still be able to decode the same Secondary CCPCH when the processing requirement is consistent with the UE capability. Or if the UE does not support the number of TFs or the coding of a certain transport channel on a Secondary CCPCH, it shall still be able to decode other transport channels mapped on the same Secondary CCPCH that is consistent with what is supported by the UE.

### 8.1.1.6.6 System Information Block type 6

If in connected mode, the UE should store all relevant IEs included in this system information block. The UE shall:

1> replace the TFS of the RACH with the one stored in the UE if any;

1> let the physical channel(s) of type PRACH given by the IE(s) "PRACH info" be the default in uplink if UE is in CELL_FACH state. If the IE "PRACH info" is not included, the UE shall read the corresponding IE(s) in System Information Block type 5 and use that information to configure the PRACH;

1> start to receive the physical channel of type AICH using the parameters given by the IE "AICH info" when associated PRACH is used. If the IE "AICH info" is not included, the UE shall read the corresponding IE in System Information Block type 5 and use that information (FDD only);

1> replace the TFS of the FACH/PCH with the one stored in the UE if any;

1> select a Secondary CCPCH as specified in [4] and in subclause 8.5.19, and start to receive the physical channel of type PICH associated with the PCH carried by the selected Secondary CCPCH using the parameters given by the IE "PICH info" if the UE is in CELL_PCH or URA_PCH state. If the IE "PICH info" is not included, the UE shall read the corresponding IE in System Information Block type 5 and use that information;

1> start to monitor its paging occasions on the selected PICH if the UE is in CELL_PCH or URA_PCH state;

1> start to receive the selected physical channel of type Secondary CCPCH using the parameters given by the IE(s) "Secondary CCPCH info" if the UE is in CELL_FACH state. If the IE "Secondary CCPCH info" is not included, the UE shall read the corresponding IE(s) in System Information Block type 5 and use that information;

1> in 3.84 Mcps TDD: use the IE "TDD open loop power control" as defined in subclause 8.5.7;

1> in TDD: if the IE "PDSCH system information" and/or the IE "PUSCH system information" is included, store each of the configurations given there with the associated identity given in the IE "PDSCH Identity" and/or "PUSCH Identity" respectively. For every configuration, for which the IE "SFN Time info" is included, the information shall be stored for the duration given there.

If in idle mode, the UE shall not use the values of the IEs in this system information block.

If a UE is a 12 kbps class UE according to [35] and the UE has a lower capability than required to support all transport channel configurations mapped on a specific Secondary CCPCH, the UE shall at a certain time instant still be able to decode those transport channels mapped on this Secondary CCPCH that do match the capability supported by the UE. The UE shall use the TFCI bits for that Secondary CCPCH, to distinguish a transport channel configuration that is supported by the UE from a transport channel configuration that is not supported by the UE.

In particular if the UE is a 12 kbps class UE according to [35] and it does not support the processing requirement at a given point in time for a Secondary CCPCH, it shall still be able to decode the same Secondary CCPCH when the processing requirement is consistent with the UE capability. Or if the UE does not support the number of TFs or the coding of a certain transport channel on a Secondary CCPCH, it shall still be able to decode other transport channels mapped on the same Secondary CCPCH that is consistent with what is supported by the UE.

## 8.5.19 Secondary CCPCH selection

In UTRAN Connected mode, the UE shall select the Secondary CCPCH according to the following rules:

1> in Cell_DCH state:

2> select Secondary CCPCH according to subclause 8.6.6.4.

1> in Cell_FACH state:

2> if System Information Block type 6 is defined and includes one or more SCCPCH that carry a FACH, compile a list of candidate SCCPCH that consists of these SCCPCH, in the order of appearance in System Information Block type 6.

NOTE 1: An SCCPCH carries a FACH if the size of the "FACH/PCH information" list within the IE "Secondary CCPCH system information" exceeds 1 or if the size of this list equals 1 while IE "Secondary CCPCH system information" does not contain an IE "PICH info".

2> otherwise:

3> compile a list of candidate SCCPCH that consists of the SCCPCH(s) included in System Information Block type 5 that carry a FACH, in the order of appearance in System Information Block type 5.

2> select an SCCPCH from the list of candidate SCCPCHs based on U-RNTI as follows:

"Index of selected SCCPCH" = U-RNTI mod K,

where K is equal to the number of candidate SCCPCHs.

1> in Cell_PCH and URA_PCH states:

2> if System Information Block type 6 is defined and includes one or more SCCPCH that carry a PCH, compile a list of candidate SCCPCH that consists of these SCCPCH, in the order of appearance in System Information Block type 6

NOTE 2: An SCCPCH carries a PCH if the IE "Secondary CCPCH system information" contains IE "PICH info"

2> otherwise compile a list of candidate SCCPCH that consists of the SCCPCH(s) included in System Information Block type 5 that that carry a PCH , in the order of appearance in System Information Block type 5

2> select an SCCPCH from the list of candidate SCCPCHs based on U-RNTI as follows:

"Index of selected SCCPCH" = U-RNTI mod K,

where K is equal to the number of candidate SCCPCHs.

The UE shall set the CFN in relation to the SFN of the current cell according to subclause 8.5.15.

The UE shall ~~support reception of~~ decode all transport formats on all FACHs multiplexed on the selected S-CCPCH according to its UE capability, as defined in section 8.1.1.6.5 and 8.1.1.6.6, to find blocks adressed to the UE.

## 10.3.3.25	Physical channel capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description | Version |
|---|---|---|---|---|---|
| Downlink physical channel capability information elements | | | | | |
| FDD downlink physical channel capability | CH-*fdd_req_su p* | | | | |
| >Max no DPCH/PDSCH codes | MP | | Integer (1..8) | Maximum number of DPCH/PDSCH codes to be simultaneously received | |
| >Max no physical channel bits received | MP | | Integer (~~600,~~ 1200, 2400, 3600, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 48000, 57600, 67200, 76800) | Maximum number of physical channel bits received in any 10 ms interval (DPCH, PDSCH, S-CCPCH) | |
| >Support for SF 512 | MP | | Boolean | TRUE means supported | |
| >Support of PDSCH | MP | | Boolean | TRUE means supported | |
| >Simultaneous reception of SCCPCH and DPCH | MP | | Boolean | TRUE means supported | |
| >Simultaneous reception of SCCPCH, DPCH and PDSCH | CV-*if_sim_rec _pdsch _sup* | | Boolean | TRUE means supported | |
| >Max no of S-CCPCH RL | CV-*if_sim_rec* | | Integer(1) | Maximum number of simultaneous S-CCPCH radio links | |
| >Support of dedicated pilots for channel estimation | MD | | Enumerated (true) | Presence of this element means | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description | Version |
|---|---|---|---|---|---|
| | | | | supported and absence not supported. If the UE notifies support of this functionality, it should comply with the corresponding performance requirements. Note 1. | |
| 3.84 Mcps TDD downlink physical channel capability | CH-*3.84_Mcps_tdd_req_sup* | | | | Name changed in REL-4 |
| >Maximum number of timeslots per frame | MP | | Integer (1..14) | | |
| >Maximum number of physical channels per frame | MP | | Integer (5~~1~~..224) | | |
| >Minimum SF | MP | | Integer (1, 16) | | |
| >Support of PDSCH | MP | | Boolean | TRUE means supported | |
| >Maximum number of physical channels per timeslot | MP | | Integer (5~~1~~..16) | | |
| 1.28 Mcps TDD downlink physical channel capability | CH-*1.28_Mcps_tdd_req_sup* | | | | REL-4 |
| >Maximum number of timeslots per subframe | MP | | Integer (1..6) | | REL-4 |
| >Maximum number of physical channels per subframe | MP | | Integer (1..96) | | REL-4 |
| >Minimum SF | MP | | Integer (1, 16) | | REL-4 |
| >Support of PDSCH | MP | | Boolean | TRUE means supported | REL-4 |
| >Maximum number of physical channels per timeslot | MP | | Integer (1..16) | | REL-4 |
| >Support of 8PSK | MP | | Boolean | TRUE means supported | REL-4 |
| **Uplink physical channel capability information elements** | | | | | |
| FDD uplink physical channel capability | CH-*fdd_req_sup* | | | | |
| >Maximum number of DPDCH bits transmitted per 10 ms | MP | | Integer (600, 1200, 2400, 4800. 9600, 19200. 28800, 38400, 48000, 57600) | | |
| >Support of PCPCH | MP | | Boolean | TRUE means supported | |
| 3.84 Mcps TDD uplink physical channel capability | CH-*3.84_Mcps_tdd_req_sup* | | | | Name changed in REL-4 |
| >Maximum Number of timeslots per frame | MP | | Integer (1..14) | | |
| >Maximum number of physical | MP | | Integer | | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description | Version |
|---|---|---|---|---|---|
| channels per timeslot | | | (1, 2) | | |
| >Minimum SF | MP | | Integer (1, 2, 4, 8, ~~16~~) | | |
| >Support of PUSCH | MP | | Boolean | TRUE means supported | |
| 1.28 Mcps TDD uplink physical channel capability | CH-*1.28_Mcps_tdd_req_sup* | | | | REL-4 |
| >Maximum Number of timeslots per subframe | MP | | Integer (1..6) | | REL-4 |
| >Maximum number of physical channels per timeslot | MP | | Integer (1, 2) | | REL-4 |
| >Minimum SF | MP | | Integer (1, 2, 4, 8, 16) | | REL-4 |
| >Support of PUSCH | MP | | Boolean | TRUE means supported | REL-4 |
| >Support of 8PSK | MP | | Boolean | TRUE means supported | REL-4 |

| Condition | Explanation |
|---|---|
| *if_sim_rec_pdsch_sup* | The IE is mandatory present if the IE "Simultaneous reception of SCCPCH and DPCH" = True and IE Support of PDSCH = True. Otherwise this field is not needed in the message. |
| *if_sim_rec* | The IE is mandatory present if the IE "capability Simultaneous reception of SCCPCH and DPCH" = True. Otherwise this field is not needed in the message. |
| *3.84_Mcps_tdd_req_sup* | The IE is mandatory present if the IE "TDD RF capability" is present with the IE "Chip rate capability" set to "3.84 Mcps" and a 3.84 Mcps TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |
| *1.28_Mcps_tdd_req_sup* | The IE is mandatory present if the IE "TDD RF capability" is present with the IE "Chip rate capability" set to "1.28 Mcps" and a 1.28 Mcps TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |
| *fdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "FDD" or "FDD/TDD" and a FDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |

NOTE 1:  These performance requirements are defined in Release 5.

## 10.3.3.34    RLC capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| Total RLC AM buffer size | MP | | Integer (~~2,~~10,50,100,150,500,1000) | Total receiving and transmitting RLC AM buffer capability in kBytes. One spare value is needed. |
| Maximum RLC AM Window Size | MP | | Integer(2047,4095) | Maximum supported RLC TX and RX window in UE |
| Maximum number of AM entities | MP | | Integer (~~3,~~4,5,6,8,16,30) | |

## 10.3.3.40 Transport channel capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| **Downlink transport channel capability information elements** | | | | |
| Max no of bits received | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all transport blocks received at an arbitrary time instant |
| Max convolutionally coded bits received | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all convolutionally coded transport blocks received at an arbitrary time instant |
| Max turbo coded bits received | CV-*turbo_dec_sup* | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all turbo coded transport blocks received at an arbitrary time instant |
| Maximum number of simultaneous transport channels | MP | | Integer(4, 8, 16, 32) | |
| Maximum number of simultaneous CCTrCH | MP | | Integer (1..8) | |
| Max no of received transport blocks | MP | | Integer(4, 8, 16, 32, 48, 64, 96, 128, 256, 512) | Maximum total number of transport blocks received within TTIs that end at within the same 10ms interval |
| Maximum number of TFC | MP | | Integer(16, 32, 48, 64, 96, 128, 256, 512, 1024) | |
| Maximum number of TF | MP | | Integer(32, 64, 128, 256, 512, 1024) | |
| Support for turbo decoding | MP | | Boolean | TRUE means supported |
| **Uplink transport channel capability information elements** | | | | |
| Max no of bits transmitted | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all transport blocks transmitted at an arbitrary time instant |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| Max convolutionally coded bits transmitted | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all convolutionally coded transport blocks transmitted at an arbitrary time instant |
| Max turbo coded bits transmitted | CV-*turbo_enc_sup* | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all turbo coded transport blocks transmitted at an arbitrary time instant |
| Maximum number of simultaneous transport channels | MP | | Integer(~~2,~~ 4, 8, 16, 32) | |
| Maximum number of simultaneous CCTrCH of DCH type | CH-*tdd_req_sup* | | Integer (1..8) | |
| Max no of transmitted transport blocks | MP | | Integer(~~2,~~ 4, 8, 16, 32, 48, 64, 96, 128, 256, 512) | Maximum total number of transport blocks transmitted within TTIs that start at the same time |
| Maximum number of TFC | MP | | Integer(~~4, 8,~~ 16, 32, 48, 64, 96, 128, 256, 512, 1024) | |
| Maximum number of TF | MP | | Integer(32, 64, 128, 256, 512, 1024) | |
| Support for turbo encoding | MP | | Boolean | TRUE means supported |

| Condition | Explanation |
|---|---|
| *turbo_dec_sup* | The IE is mandatory present if the IE "Support of turbo decoding" = True. Otherwise this field is not needed in the message. |
| *turbo_enc_sup* | The IE is mandatory present if the IE "Support of turbo encoding" = True. Otherwise this field is not needed in the message. |
| *tdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "TDD" or "FDD/TDD" and a TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |

## 11.3    Information element definitions

```
[…]
TotalRLC-AM-BufferSize ::=          ENUMERATED {
                    kb2dummy, kb10, kb50, kb100,
                    kb150, kb500, kb1000, spare }
[…]
MaximumAM-EntityNumberRLC-Cap ::=   ENUMERATED {
                    am3dummy, am4, am5, am6,
                    am8, am16, am30 }
[…]
MaxSimultaneousTransChsUL ::=       ENUMERATED {
                    e2dummy, e4, e8, e16, e32 }
[…]
```

```
MaxTransportBlocksUL ::=               ENUMERATED {
                                          tb2dummy, tb4, tb8, tb16, tb32, tb48,
                                          tb64, tb96, tb128, tb256, tb512 }
[…]
MaxNumberOfTFC-UL ::=                  ENUMERATED {
                                          tfc4dummy1, tfc8dummy2, tfc16, tfc32, tfc48, tfc64,
                                          tfc96, tfc128, tfc256, tfc512, tfc1024 }

[…]
MaxNoPhysChBitsReceived ::=            ENUMERATED {
                                          b600dummy, b1200, b2400, b3600,
                                          b4800, b7200, b9600, b14400,
                                          b19200, b28800, b38400, b48000,
                                          b57600, b67200, b76800 }
[…]
-- the values 1 …4 for MaxPhysChPerFrame are not used in this version of the protocol
MaxPhysChPerFrame ::=                  INTEGER (1..224)
[…]
-- the values 1 …4 for MaxPhysChPerTS are not used in this version of the protocol
MaxPhysChPerTS ::=                     INTEGER (1..16)
[…]
MinimumSF-UL ::=                       ENUMERATED {
                                          sf1, sf2, sf4, sf8, sf16 dummy}
```

CR-Form-v7

# CHANGE REQUEST

| ⌘ | **25.331 CR 2152** | ⌘ **rev** | **2** | ⌘ | Current version: | **5.6.0** | ⌘ |

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

**Proposed change affects:**    UICC apps⌘ ☐    ME **X** Radio Access Network ☐    Core Network ☐

| **Title:** | ⌘ | Minimum UE capability class |

| **Source:** | ⌘ | Ericsson |

| **Work item code:** ⌘ | TEI5 | | **Date:** ⌘ | Nov 2003 |

| **Category:** | ⌘ | **F** | | **Release:** ⌘ | Rel-5 |

*Use one of the following categories:*
　**F** *(correction)*
　**A** *(corresponds to a correction in an earlier release)*
　**B** *(addition of feature),*
　**C** *(functional modification of feature)*
　**D** *(editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP <u>TR 21.900</u>.

*Use one of the following releases:*
　2　　　*(GSM Phase 2)*
　R96　*(Release 1996)*
　R97　*(Release 1997)*
　R98　*(Release 1998)*
　R99　*(Release 1999)*
　Rel-4　*(Release 4)*
　Rel-5　*(Release 5)*
　Rel-6　*(Release 6)*

| **Reason for change:** | ⌘ | A minimum radio access capability parameter combination is not defined. |

| **Summary of change:** ⌘ | The radio access parameter values for a new reference combination called 12kbps class UE are added for FDD. It is defined that this 12kbps class UE is the minimum UE.

1. The capability values that exist in signalling but is below the capabilty of the 12 kbps class UE will never be used and are removed from the signalling.

2. A sentence is added to make sure that a UE with the lowest possible UE capability still should perform in a predictable manner in NWs utilising a high bitrate common channel configuration.
For 12 kbps class Ues it is a shall in R'99 and onwards.
For other Ues there is only a shall in Rel-5.

Impact analysis:
UEs with capabilities lower than for 32kbps UE class are affected.
UEs that do not implement the CR may be unable to support RAB combinations e.g. on common channels, like SCCPCH combinations. Specifically they can fail to read the FACH. For these Ues there are examples in 34.108 that will not be supported.
UEs with capabilities equal to or higher than the 32kbps class are also affected in Rel-5. If these Ues do not implement the CR they may be unable to support RAB combinations other than existing 34.108 common channels configurations, if these require higher capabilities than the 32kbps class.

The CR has no impact on UTRAN. |

| | | | | |
|---|---|---|---|---|
| **Consequences if not approved:** | ⌘ | There is no definition of the minimum UE capabilities that can be expected by the network. Operators may use RAB combinations that can not be supported by UEs with insufficient UE capabilities. In consequence UEs may be unable to access the system. | | |

| | | | | |
|---|---|---|---|---|
| **Clauses affected:** | ⌘ | 8.1.1.6.5, 8.1.1.6.6, 8.5.19, 10.3.3.25, 10.3.3.34, 10.3.3.40, 11.3 | | |

| | | Y | N | | | |
|---|---|---|---|---|---|---|
| **Other specs affected:** | ⌘ | X | | Other core specifications | ⌘ | 25.306 |
| | | | X | Test specifications | | |
| | | | X | O&M Specifications | | |

| | | | | |
|---|---|---|---|---|
| **Other comments:** | ⌘ | In revision 2 of this CR the last paragraph changed in section 8.1.1.6.5 and 8.1.1.6.6 are rephrased to be normative text instead of a NOTE. | | |

**How to create CRs using this form:**

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

### 8.1.1.6.5 System Information Block type 5

The UE should store all relevant IEs included in this system information block. The UE shall:

1> if in connected mode, and System Information Block type 6 is indicated as used in the cell:

2> read and act on information sent in System Information Block type 6.

1> replace the TFS of the RACH with the one stored in the UE if any;

1> let the physical channel(s) of type PRACH given by the IE(s) "PRACH info" be the default in uplink for the PRACH if UE is in CELL_FACH state;

1> start to receive the physical channel of type AICH using the parameters given by the IE "AICH info" (FDD only) when given allocated PRACH is used;

1> use the first instance of the list of transport formats as in the IE "RACH TFS" for the used RACH received in the IE "PRACH system information list" when using the CCCH;

1> replace the TFS of the FACH/PCH with the one stored in the UE if any;

1> select a Secondary CCPCH as specified in [4] and in subclause 8.5.19, and start to receive the physical channel of type PICH associated with the PCH carried by the selected Secondary CCPCH using the parameters given by the IE "PICH info" if UE is in Idle mode or in CELL_PCH or URA_PCH state;

1> start to monitor its paging occasions on the selected PICH if UE is in Idle mode or in CELL_PCH or URA_PCH state;

1> start to receive the selected physical channel of type Secondary CCPCH using the parameters given by the IE(s) "Secondary CCPCH info" if UE is in CELL_FACH state;

1> in 3.84 Mcps TDD:

2> use the IE "TDD open loop power control" as defined in subclause 8.5.7 when allocated PRACH is used.

1> in TDD:

2> if the IE "PDSCH system information" and/or the IE "PUSCH system information" is included:

3> store each of the configurations given there with the associated identity given in the IE "PDSCH Identity" and/or "PUSCH Identity" respectively. For every configuration, for which the IE "SFN Time info" is included, the information shall be stored for the duration given there.

If a UE has a lower capability than required to support all transport channel configurations mapped on a specific Secondary CCPCH, the UE shall at a certain time instant still be able to decode those transport channels mapped on this Secondary CCPCH that do match the capability supported by the UE. The UE shall use the TFCI bits for that Secondary CCPCH, to distinguish a transport channel configuration that is supported by the UE from a transport channel configuration that is not supported by the UE.

In particular if the UE does not support the processing requirement at a given point in time for a Secondary CCPCH, it shall still be able to decode the same Secondary CCPCH when the processing requirement is consistent with the UE capability. Or if the UE does not support the number of TFs or the coding of a certain transport channel on a Secondary CCPCH, it shall still be able to decode other transport channels mapped on the same Secondary CCPCH that is consistent with what is supported by the UE.

### 8.1.1.6.6 System Information Block type 6

If in connected mode, the UE should store all relevant IEs included in this system information block. The UE shall:

1> replace the TFS of the RACH with the one stored in the UE if any;

1> let the physical channel(s) of type PRACH given by the IE(s) "PRACH info" be the default in uplink if UE is in CELL_FACH state. If the IE "PRACH info" is not included, the UE shall read the corresponding IE(s) in System Information Block type 5 and use that information to configure the PRACH;

1> start to receive the physical channel of type AICH using the parameters given by the IE "AICH info" when associated PRACH is used. If the IE "AICH info" is not included, the UE shall read the corresponding IE in System Information Block type 5 and use that information (FDD only);

1> replace the TFS of the FACH/PCH with the one stored in the UE if any;

1> select a Secondary CCPCH as specified in [4] and in subclause 8.5.19, and start to receive the physical channel of type PICH associated with the PCH carried by the selected Secondary CCPCH using the parameters given by the IE "PICH info" if the UE is in CELL_PCH or URA_PCH state. If the IE "PICH info" is not included, the UE shall read the corresponding IE in System Information Block type 5 and use that information;

1> start to monitor its paging occasions on the selected PICH if the UE is in CELL_PCH or URA_PCH state;

1> start to receive the selected physical channel of type Secondary CCPCH using the parameters given by the IE(s) "Secondary CCPCH info" if the UE is in CELL_FACH state. If the IE "Secondary CCPCH info" is not included, the UE shall read the corresponding IE(s) in System Information Block type 5 and use that information;

1> in 3.84 Mcps TDD: use the IE "TDD open loop power control" as defined in subclause 8.5.7;

1> in TDD: if the IE "PDSCH system information" and/or the IE "PUSCH system information" is included, store each of the configurations given there with the associated identity given in the IE "PDSCH Identity" and/or "PUSCH Identity" respectively. For every configuration, for which the IE "SFN Time info" is included, the information shall be stored for the duration given there.

If in idle mode, the UE shall not use the values of the IEs in this system information block.

If a UE has a lower capability than required to support all transport channel configurations mapped on a specific Secondary CCPCH, the UE shall at a certain time instant still be able to decode those transport channels mapped on this Secondary CCPCH that do match the capability supported by the UE. The UE shall use the TFCI bits for that Secondary CCPCH, to distinguish a transport channel configuration that is supported by the UE from a transport channel configuration that is not supported by the UE.

In particular if the UE does not support the processing requirement at a given point in time for a Secondary CCPCH, it shall still be able to decode the same Secondary CCPCH when the processing requirement is consistent with the UE capability. Or if the UE does not support the number of TFs or the coding of a certain transport channel on a Secondary CCPCH, it shall still be able to decode other transport channels mapped on the same Secondary CCPCH that is consistent with what is supported by the UE.

## 8.5.19    Secondary CCPCH selection

In UTRAN Connected mode, the UE shall select the Secondary CCPCH according to the following rules:

1> in Cell_DCH state:

2> select Secondary CCPCH according to subclause 8.6.6.4.

1> in Cell_FACH state:

2> if System Information Block type 6 is defined and includes one or more SCCPCH that carry a FACH, compile a list of candidate SCCPCH that consists of these SCCPCH, in the order of appearance in System Information Block type 6.

NOTE 1: An SCCPCH carries a FACH if the size of the "FACH/PCH information" list within the IE "Secondary CCPCH system information" exceeds 1 or if the size of this list equals 1 while IE "Secondary CCPCH system information" does not contain an IE "PICH info".

2> otherwise:

3> compile a list of candidate SCCPCH that consists of the SCCPCH(s) included in System Information Block type 5 that carry a FACH, in the order of appearance in System Information Block type 5.

2> select an SCCPCH from the list of candidate SCCPCHs based on U-RNTI as follows:

"Index of selected SCCPCH" = U-RNTI mod K,

where K is equal to the number of candidate SCCPCHs.

1> in Cell_PCH and URA_PCH states:

2> if System Information Block type 6 is defined and includes one or more SCCPCH that carry a PCH, compile a list of candidate SCCPCH that consists of these SCCPCH, in the order of appearance in System Information Block type 6

NOTE 2: An SCCPCH carries a PCH if the IE "Secondary CCPCH system information" contains IE "PICH info"

2> otherwise compile a list of candidate SCCPCH that consists of the SCCPCH(s) included in System Information Block type 5 that that carry a PCH , in the order of appearance in System Information Block type 5

2> select an SCCPCH from the list of candidate SCCPCHs based on U-RNTI as follows:

"Index of selected SCCPCH" = U-RNTI mod K,

where K is equal to the number of candidate SCCPCHs.

The UE shall set the CFN in relation to the SFN of the current cell according to subclause 8.5.15.

The UE shall ~~support reception of~~ decode all transport formats on all FACHs multiplexed on the selected S-CCPCH according to its UE capability, as defined in section 8.1.1.6.5 and 8.1.1.6.6, to find blocks adressed to the UE.

## 10.3.3.25 Physical channel capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description | Version |
|---|---|---|---|---|---|
| Downlink physical channel capability information elements | | | | | |
| FDD downlink physical channel capability | CH-*fdd_req_sup* | | | | |
| >Max no DPCH/PDSCH codes | MP | | Integer (1..8) | Maximum number of DPCH/PDSCH codes to be simultaneously received | |
| >Max no physical channel bits received | MP | | Integer (~~600,~~ 1200, 2400, 3600, 4800, 7200, 9600, 14400, 19200, 28800, 38400, 48000, 57600, 67200, 76800) | Maximum number of physical channel bits received in any 10 ms interval (DPCH, PDSCH, S-CCPCH) | |
| >Support for SF 512 | MP | | Boolean | TRUE means supported | |
| >Support of PDSCH | MP | | Boolean | TRUE means supported | |
| >CHOICE *Support of HS-PDSCH* | MP | | | | REL-5 |
| >>Supported | | | | | REL-5 |
| >>>HS-DSCH physical layer category | MP | | Integer (1..64) | | REL-5 |
| >>>Support of dedicated pilots for channel estimation of HS-DSCH | MP | | Boolean | TRUE means supported | REL-5 |
| >>Unsupported | | | | (no data) | REL-5 |
| >Simultaneous reception of SCCPCH and DPCH | MP | | Boolean | TRUE means supported | |
| >Simultaneous reception of | CV- | | Boolean | TRUE means | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description | Version |
|---|---|---|---|---|---|
| SCCPCH, DPCH and PDSCH | *if_sim_rec _pdsch _sup* | | | supported | |
| >Max no of S-CCPCH RL | CV-*if_sim_rec* | | Integer(1) | Maximum number of simultaneous S-CCPCH radio links | |
| >Support of dedicated pilots for channel estimation | MD | | Enumerated (true) | Presence of this element means supported and absence not supported. This IE shall be set to TRUE in this version of the protocol. | |
| 3.84 Mcps TDD downlink physical channel capability | CH-*3.84_Mcps _tdd_req_s up* | | | | Name changed in REL-4 |
| >Maximum number of timeslots per frame | MP | | Integer (1..14) | | |
| >Maximum number of physical channels per frame | MP | | Integer (~~1~~5..224) | | |
| >Minimum SF | MP | | Integer (1, 16) | | |
| >Support of PDSCH | MP | | Boolean | TRUE means supported | |
| >CHOICE *Support of HS-PDSCH* | MP | | | | REL-5 |
| >>Supported | | | | | REL-5 |
| >>>HS-DSCH physical layer category | MP | | Integer (1..64) | | REL-5 |
| >>Unsupported | | | | (no data) | REL-5 |
| >Maximum number of physical channels per timeslot | MP | | Integer (~~1~~5..16) | | |
| 1.28 Mcps TDD downlink physical channel capability | CH-*1.28_Mcps _tdd_req_s up* | | | | REL-4 |
| >Maximum number of timeslots per subframe | MP | | Integer (1..6) | | REL-4 |
| >Maximum number of physical channels per subframe | MP | | Integer (1..96) | | REL-4 |
| >Minimum SF | MP | | Integer (1, 16) | | REL-4 |
| >Support of PDSCH | MP | | Boolean | TRUE means supported | REL-4 |
| >CHOICE *Support of HS-PDSCH* | MP | | | | REL-5 |
| >>Supported | | | | | REL-5 |
| >>>HS-DSCH physical layer category | MP | | Integer (1..64) | | REL-5 |
| >>Unsupported | | | | (no data) | REL-5 |
| >Maximum number of physical channels per timeslot | MP | | Integer (1..16) | | REL-4 |
| >Support of 8PSK | MP | | Boolean | TRUE means supported | REL-4 |
| **Uplink physical channel capability information elements** | | | | | |
| FDD uplink physical channel capability | CH-*fdd_req_su p* | | | | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description | Version |
|---|---|---|---|---|---|
| >Maximum number of DPDCH bits transmitted per 10 ms | MP | | Integer (600, 1200, 2400, 4800. 9600, 19200. 28800, 38400, 48000, 57600) | | |
| >Support of PCPCH | MP | | Boolean | TRUE means supported | |
| 3.84 Mcps TDD uplink physical channel capability | CH-*3.84_Mcps_tdd_req_sup* | | | | Name changed in REL-4 |
| >Maximum Number of timeslots per frame | MP | | Integer (1..14) | | |
| >Maximum number of physical channels per timeslot | MP | | Integer (1, 2) | | |
| >Minimum SF | MP | | Integer (1, 2, 4, 8, ~~16~~) | | |
| >Support of PUSCH | MP | | Boolean | TRUE means supported | |
| 1.28 Mcps TDD uplink physical channel capability | CH-*1.28_Mcps_tdd_req_sup* | | | | REL-4 |
| >Maximum Number of timeslots per subframe | MP | | Integer (1..6) | | REL-4 |
| >Maximum number of physical channels per timeslot | MP | | Integer (1, 2) | | REL-4 |
| >Minimum SF | MP | | Integer (1, 2, 4, 8, 16) | | REL-4 |
| >Support of PUSCH | MP | | Boolean | TRUE means supported | REL-4 |
| >Support of 8PSK | MP | | Boolean | TRUE means supported | REL-4 |

| Condition | Explanation |
|---|---|
| *if_sim_rec_pdsch_sup* | The IE is mandatory present if the IE "Simultaneous reception of SCCPCH and DPCH" = True and IE Support of PDSCH = True. Otherwise this field is not needed in the message. |
| *if_sim_rec* | The IE is mandatory present if the IE "capability Simultaneous reception of SCCPCH and DPCH" = True. Otherwise this field is not needed in the message. |
| *3.84_Mcps_tdd_req_sup* | The IE is mandatory present if the IE "TDD RF capability" is present with the IE "Chip rate capability" set to "3.84 Mcps" and a 3.84 Mcps TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |
| *1.28_Mcps_tdd_req_sup* | The IE is mandatory present if the IE "TDD RF capability" is present with the IE "Chip rate capability" set to "1.28 Mcps" and a 1.28 Mcps TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |
| *fdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "FDD" or "FDD/TDD" and a FDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |

## 10.3.3.34	RLC capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| Total RLC AM buffer size | MP | | Integer (2,10,50,100 ,150,200,300 ,400,500,750 ,1000) | Total receiving and transmitting RLC AM buffer and MAC-hs reordering buffer capability in kBytes. |
| Maximum RLC AM Window Size | MP | | Integer(2047 ,4095) | Maximum supported RLC TX and RX window in UE |
| Maximum number of AM entities | MP | | Integer (3,4,5,6,8,16 ,30) | |

## 10.3.3.40	Transport channel capability

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| **Downlink transport channel capability information elements** | | | | |
| Max no of bits received | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all transport blocks received at an arbitrary time instant |
| Max convolutionally coded bits received | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all convolutionally coded transport blocks received at an arbitrary time instant |
| Max turbo coded bits received | CV-*turbo_dec_ sup* | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all turbo coded transport blocks received at an arbitrary time instant |
| Maximum number of simultaneous transport channels | MP | | Integer(4, 8, 16, 32) | |
| Maximum number of simultaneous CCTrCH | MP | | Integer (1..8) | |
| Max no of received transport blocks | MP | | Integer(4, 8, 16, 32, 48, 64, 96, 128, 256, 512) | Maximum total number of transport blocks received within TTIs that end at within the same 10ms interval |
| Maximum number of TFC | MP | | Integer(16, 32, 48, 64, 96, 128, 256, 512, 1024) | |
| Maximum number of TF | MP | | Integer(32, 64, 128, 256, 512, 1024) | |
| Support for turbo decoding | MP | | Boolean | TRUE means supported |
| **Uplink transport channel** | | | | |

| Information Element/Group name | Need | Multi | Type and Reference | Semantics description |
|---|---|---|---|---|
| **capability information elements** | | | | |
| Max no of bits transmitted | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all transport blocks transmitted at an arbitrary time instant |
| Max convolutionally coded bits transmitted | MP | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all convolutionally coded transport blocks transmitted at an arbitrary time instant |
| Max turbo coded bits transmitted | CV-*turbo_enc_sup* | | Integer(640, 1280, 2560, 3840, 5120, 6400, 7680, 8960, 10240, 20480, 40960, 81920, 163840) | Maximum sum of number of bits of all turbo coded transport blocks transmitted at an arbitrary time instant |
| Maximum number of simultaneous transport channels | MP | | Integer(2, 4, 8, 16, 32) | |
| Maximum number of simultaneous CCTrCH of DCH type | CH-*tdd_req_sup* | | Integer (1..8) | |
| Max no of transmitted transport blocks | MP | | Integer(2, 4, 8, 16, 32, 48, 64, 96, 128, 256, 512) | Maximum total number of transport blocks transmitted within TTIs that start at the same time |
| Maximum number of TFC | MP | | Integer(4, 8, 16, 32, 48, 64, 96, 128, 256, 512, 1024) | |
| Maximum number of TF | MP | | Integer(32, 64, 128, 256, 512, 1024) | |
| Support for turbo encoding | MP | | Boolean | TRUE means supported |

| Condition | Explanation |
|---|---|
| *turbo_dec_sup* | The IE is mandatory present if the IE "Support of turbo decoding" = True. Otherwise this field is not needed in the message. |
| *turbo_enc_sup* | The IE is mandatory present if the IE "Support of turbo encoding" = True. Otherwise this field is not needed in the message. |
| *tdd_req_sup* | The IE is mandatory present if the IE "Multi-mode capability" has the value "TDD" or "FDD/TDD" and a TDD capability update has been requested in a previous message. Otherwise this field is not needed in the message. |

## 11.3 Information element definitions

```
[…]
TotalRLC-AM-BufferSize ::=        ENUMERATED {
                                      kb2dummy, kb10, kb50, kb100,
                                      kb150, kb500, kb1000, spare }
[…]
MaximumAM-EntityNumberRLC-Cap ::=  ENUMERATED {
                                      am3dummy, am4, am5, am6,
                                      am8, am16, am30 }

[…]
MaxSimultaneousTransChsUL ::=     ENUMERATED {
                                      e2dummy, e4, e8, e16, e32 }
[…]
MaxTransportBlocksUL ::=          ENUMERATED {
                                      tb2dummy, tb4, tb8, tb16, tb32, tb48,
                                      tb64, tb96, tb128, tb256, tb512 }

[…]
MaxNumberOfTFC-UL ::=             ENUMERATED {
                                      tfc4dummy1, tfc8dummy2, tfc16, tfc32, tfc48, tfc64,
                                      tfc96, tfc128, tfc256, tfc512, tfc1024 }
[…]
MaxNoPhysChBitsReceived ::=       ENUMERATED {
                                      b600dummy, b1200, b2400, b3600,
                                      b4800, b7200, b9600, b14400,
                                      b19200, b28800, b38400, b48000,
                                      b57600, b67200, b76800 }
[…]
-- the values 1 …4 for MaxPhysChPerFrame are not used in this version of the protocol
MaxPhysChPerFrame ::=             INTEGER (1..224)
[…]
-- the values 1 …4 for MaxPhysChPerTS are not used in this version of the protocol
MaxPhysChPerTS ::=                INTEGER (1..16)
[…]
MinimumSF-UL ::=                  ENUMERATED {
                                      sf1, sf2, sf4, sf8, sf16 dummy }
```