SIB1
(Coreset 0)

Radha Krishna Ganti

Feb 2020

What is a Coreset?

- Coreset is a set of RE (in time and frequency) that is used to carry PDCH/DCI

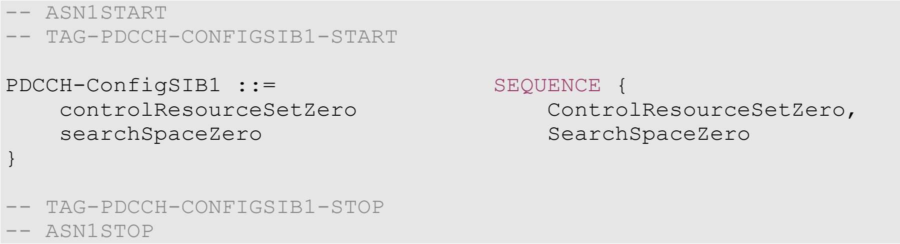
- More about Coresets later
- Currently we focus on Coreset 0
- MIB (SSB) -----> SIB1

Presence of a Coreset-0

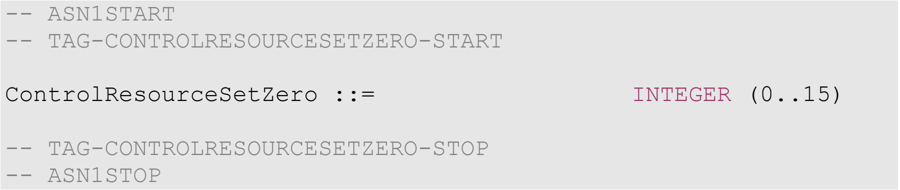
- Determined from MSB
  - If Kssb <=23 (for FR1) or Kssb <=11 (for FR2)
  - Coreset-0 is present
- Otherwise Coreset-0 is not present
  - *PDCCH-ConfigCommon* will be used
- The SIB1 SCS information is also present in MIB

Coreset 0

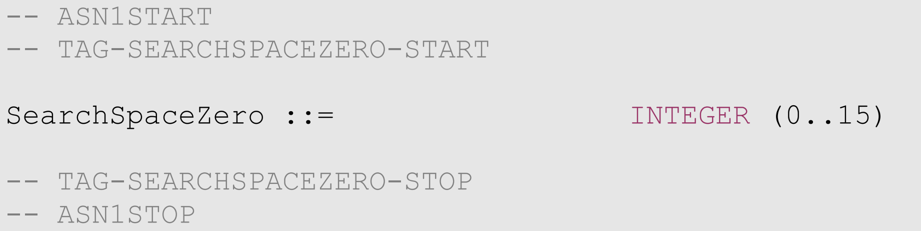
- Coreset 0 is the control information that is decoded after decoding SSB
  - Coreset 0 contains the DCI which contains the location of SIB1.
- The location of Coreset 0 (in a cryptic manner) is provided in MIB
  - pdcch-ConfigSIB1
    - ControlResourceSetZero (4 bits)
    - SearchSpaceZero (4 bits)
- Once the MIB is decoded, based on the above IE, the UE has to
  - find location of coreset0
  - decode coreset0 which has SIB1



```
-- ASN1START
-- TAG-PDCCH-CONFIGSIB1-START

PDCCH-ConfigSIB1 ::=                    SEQUENCE {
    controlResourceSetZero                  ControlResourceSetZero,
    searchSpaceZero                         SearchSpaceZero
}

-- TAG-PDCCH-CONFIGSIB1-STOP
-- ASN1STOP
```



```
-- ASN1START
-- TAG-CONTROLRESOURCESETZERO-START

ControlResourceSetZero ::=                      INTEGER (0..15)

-- TAG-CONTROLRESOURCESETZERO-STOP
-- ASN1STOP
```



```
-- ASN1START
-- TAG-SEARCHSPACEZERO-START

SearchSpaceZero ::=                 INTEGER (0..15)

-- TAG-SEARCHSPACEZERO-STOP
-- ASN1STOP
```

Location of Coreset0 (partially)

38.213 [13]

MIB->pdcch-ConfigSIB1->ControlResourceSetZero (4 bits)

- 4 bits are used as index in tables 13-1 through 13-10 [38.213]
  - See examples of tables in the next slide
- How do you choose the (correct) table?
  - SCS of PBCH (you have just decoded it)
  - SCS of PDCCH (MIB information)
  - The frequency band
    - Minimum BW supported is given per frequency band (corresponding to scs of PDCCH) in 38.101-1 (and 38.101-2)
      - You get the frequency band when the UE locks-on to SSB.
      - See Table 5.3.5-1 in 38.101-1 (and 38.101-2)

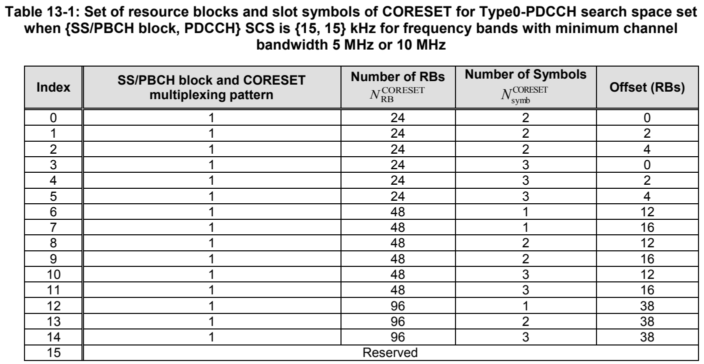


Table 13-1: Set of resource blocks and slot symbols of CORESET for Type0-PDCCH search space set when {SS/PBCH block, PDCCH} SCS is {15, 15} kHz for frequency bands with minimum channel bandwidth 5 MHz or 10 MHz

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{\mathrm{RB}}^{\mathrm{CORESET}}$	Number of Symbols $N_{\mathrm{symb}}^{\mathrm{CORESET}}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	2
2	1	24	2	4
3	1	24	3	0
4	1	24	3	2
5	1	24	3	4
6	1	48	1	12
7	1	48	1	16
8	1	48	2	12
9	1	48	2	16
10	1	48	3	12
11	1	48	3	16
12	1	96	1	38
13	1	96	2	38
14	1	96	3	38
15	Reserved			

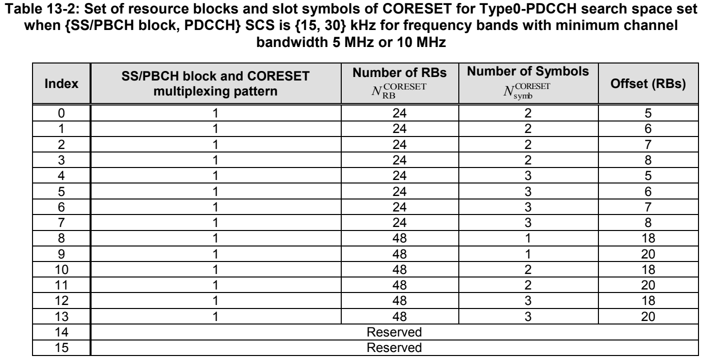


Table 13-2: Set of resource blocks and slot symbols of CORESET for Type0-PDCCH search space set when {SS/PBCH block, PDCCH} SCS is {15, 30} kHz for frequency bands with minimum channel bandwidth 5 MHz or 10 MHz

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{\mathrm{RB}}^{\mathrm{CORESET}}$	Number of Symbols $N_{\mathrm{symb}}^{\mathrm{CORESET}}$	Offset (RBs)
0	1	24	2	5
1	1	24	2	6
2	1	24	2	7
3	1	24	2	8
4	1	24	3	5
5	1	24	3	6
6	1	24	3	7
7	1	24	3	8
8	1	48	1	18
9	1	48	1	20
10	1	48	2	18
11	1	48	2	20
12	1	48	3	18
13	1	48	3	20
14	Reserved			
15	Reserved			

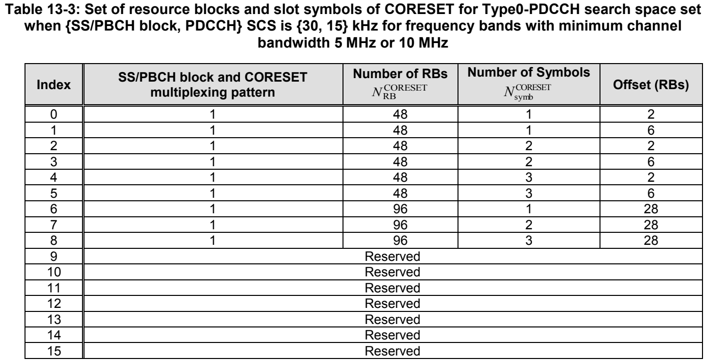


Table 13-3: Set of resource blocks and slot symbols of CORESET for Type0-PDCCH search space set when {SS/PBCH block, PDCCH} SCS is {30, 15} kHz for frequency bands with minimum channel bandwidth 5 MHz or 10 MHz

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{\mathrm{RB}}^{\mathrm{CORESET}}$	Number of Symbols $N_{\mathrm{symb}}^{\mathrm{CORESET}}$	Offset (RBs)
0	1	48	1	2
1	1	48	1	6
2	1	48	2	2
3	1	48	2	6
4	1	48	3	2
5	1	48	3	6
6	1	96	1	28
7	1	96	2	28
8	1	96	3	28
9	Reserved			
10	Reserved			
11	Reserved			
12	Reserved			
13	Reserved			
14	Reserved			
15	Reserved			

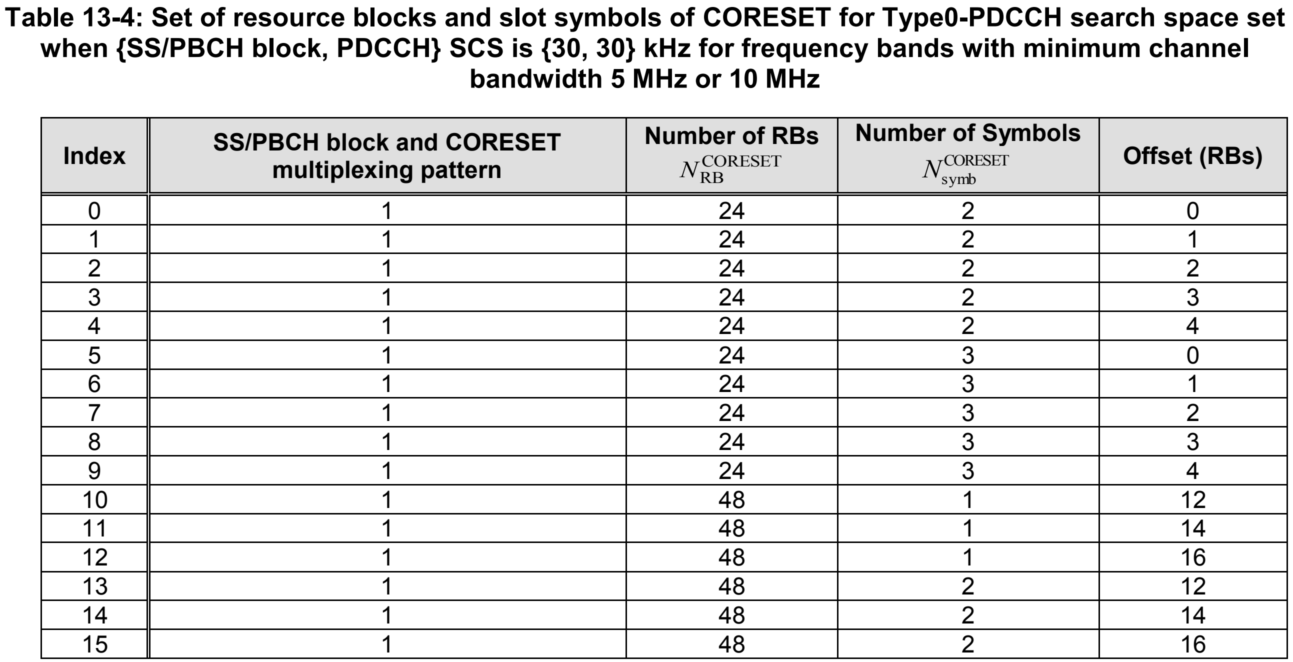


Table 13-4: Set of resource blocks and slot symbols of CORESET for Type0-PDCCH search space set when {SS/PBCH block, PDCCH} SCS is {30, 30} kHz for frequency bands with minimum channel bandwidth 5 MHz or 10 MHz

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{\mathrm{RB}}^{\mathrm{CORESET}}$	Number of Symbols $N_{\mathrm{symb}}^{\mathrm{CORESET}}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	1
2	1	24	2	2
3	1	24	2	3
4	1	24	2	4
5	1	24	3	0
6	1	24	3	1
7	1	24	3	2
8	1	24	3	3
9	1	24	3	4
10	1	48	1	12
11	1	48	1	14
12	1	48	1	16
13	1	48	2	12
14	1	48	2	14
15	1	48	2	16

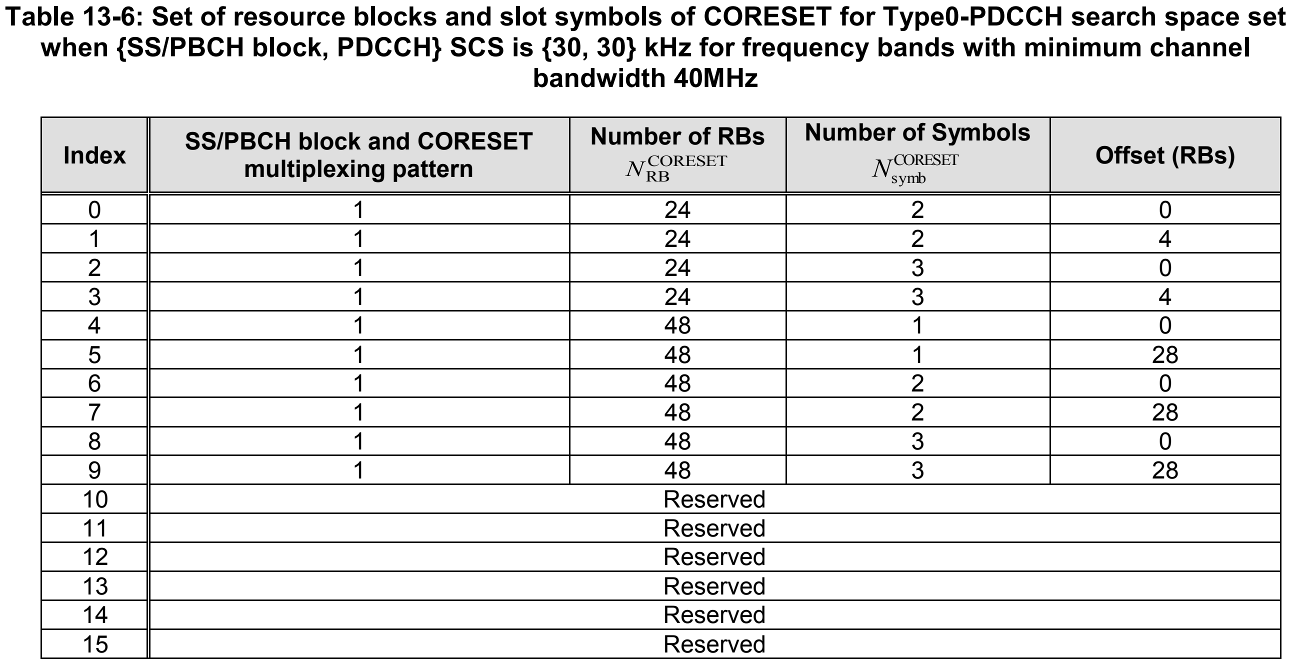


Table 13-6: Set of resource blocks and slot symbols of CORESET for Type0-PDCCH search space set when {SS/PBCH block, PDCCH} SCS is {30, 30} kHz for frequency bands with minimum channel bandwidth 40MHz

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{\mathrm{RB}}^{\mathrm{CORESET}}$	Number of Symbols $N_{\mathrm{symb}}^{\mathrm{CORESET}}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	4
2	1	24	3	0
3	1	24	3	4
4	1	48	1	0
5	1	48	1	28
6	1	48	2	0
7	1	48	2	28
8	1	48	3	0
9	1	48	3	28
10	Reserved			
11	Reserved			
12	Reserved			
13	Reserved			
14	Reserved			
15	Reserved			

Summary of the SCS, Minimum BW and table

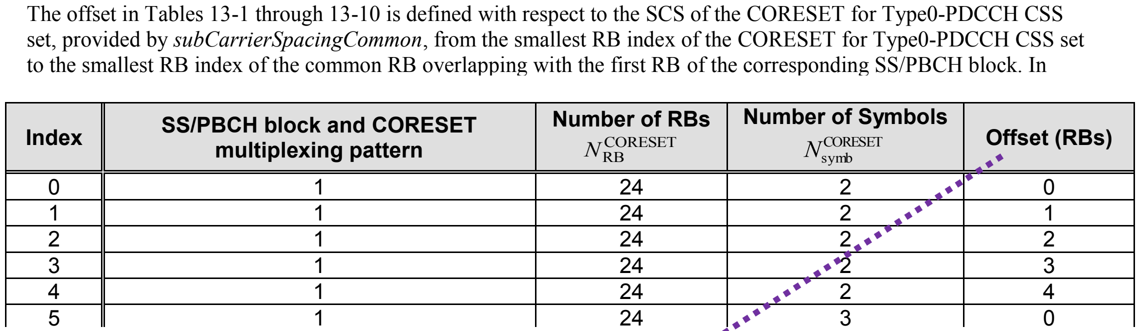
SSB SCS (KHz)	PDCCH SCS (KHz)	Minimum BW (MHz)	Table # in 38.213
15	15	5 or 10	13-1
15	30	5 or 10	13-2
30	15	5 or 10	13-3
30	30	5 or 10	13-4
30	15	40	13-5
30	30	40	13-6
120	60	-----	13-7
120	120	----	13-8
240	60	----	13-9
240	120	----	13-10

For example,
Band 78,
SSB SCS =30
PDCCH =30

Minimum BW from
38.101-1 is 10

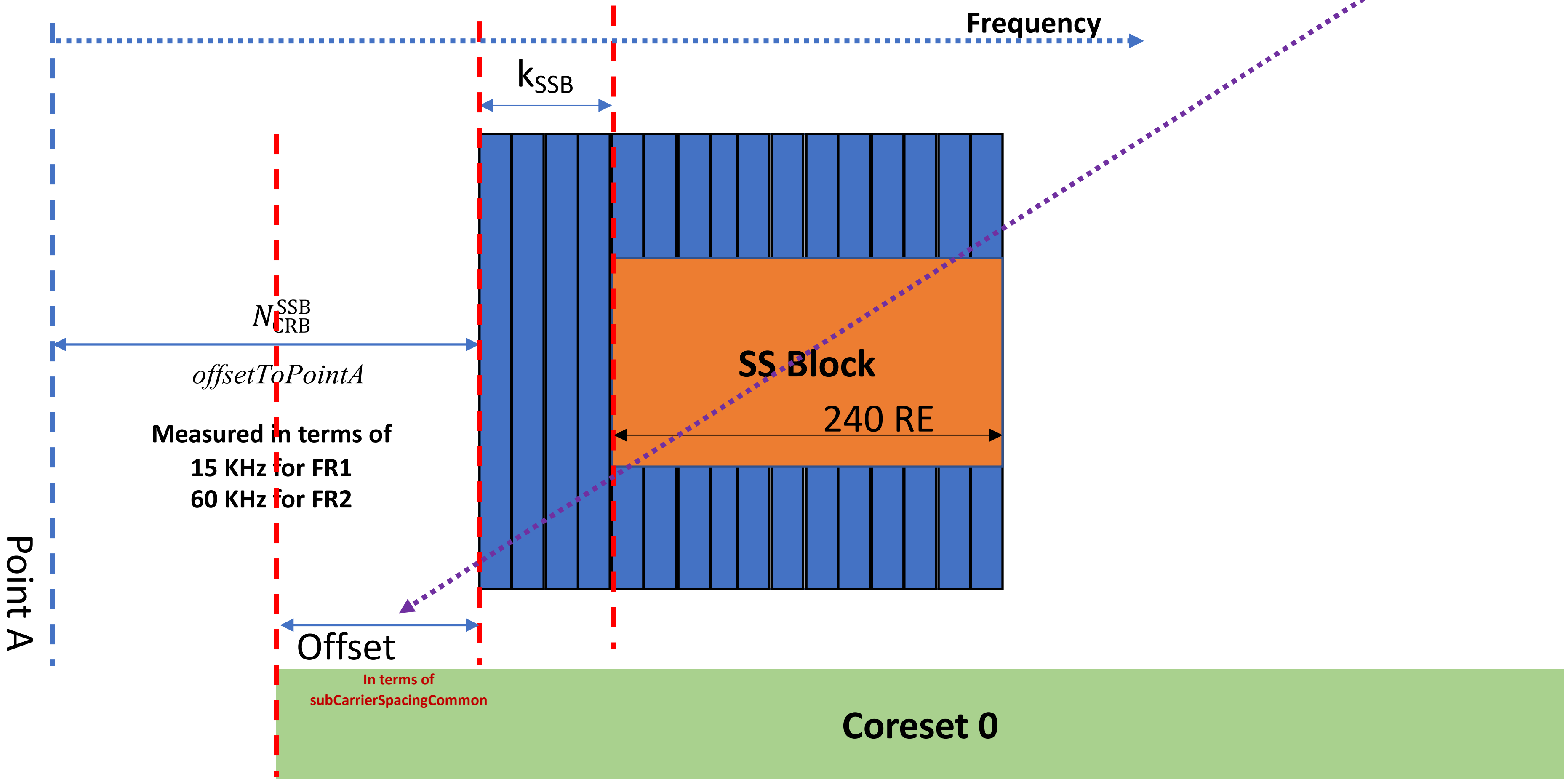
38.213 [13]

- From the previous procedure, the size of the coreset and its relative location can be found out.
- Example:
  - 30 KHz SCS for both SSB and PDCCH, Band n78
  - ControlResourceSetZero = 4 (from MIB)
    - Table 13.4, row index =4
  - So the coreset has 24 RB and spans 2 symbols.
  - It is at an offset of 4 RBs from SSB

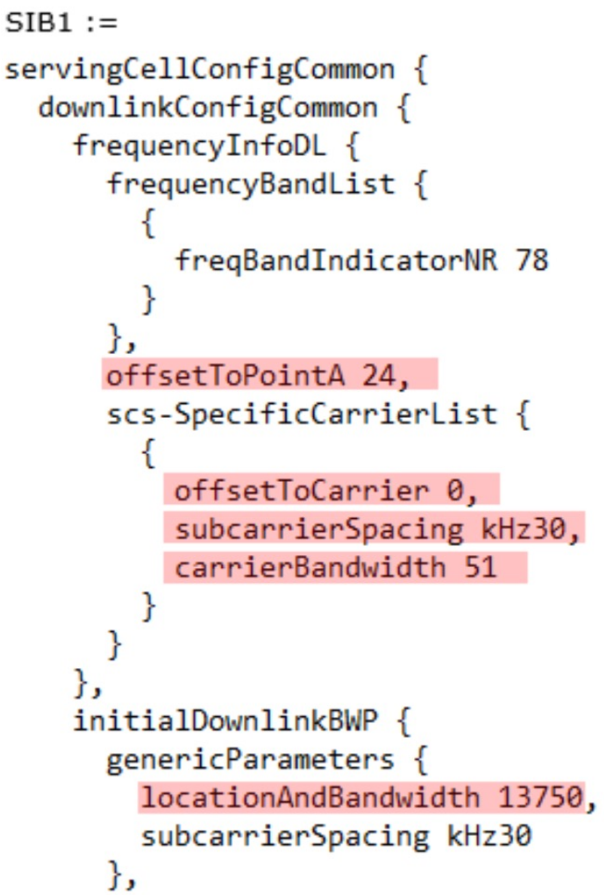


The offset in Tables 13-1 through 13-10 is defined with respect to the SCS of the CORESET for Type0-PDCCH CSS set, provided by *subCarrierSpacingCommon*, from the smallest RB index of the CORESET for Type0-PDCCH CSS set to the smallest RB index of the common RB overlapping with the first RB of the corresponding SS/PBCH block. In

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{RB}^{CORESET}$	Number of Symbols $N_{symb}^{CORESET}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	1
2	1	24	2	2
3	1	24	2	3
4	1	24	2	4
5	1	24	3	0

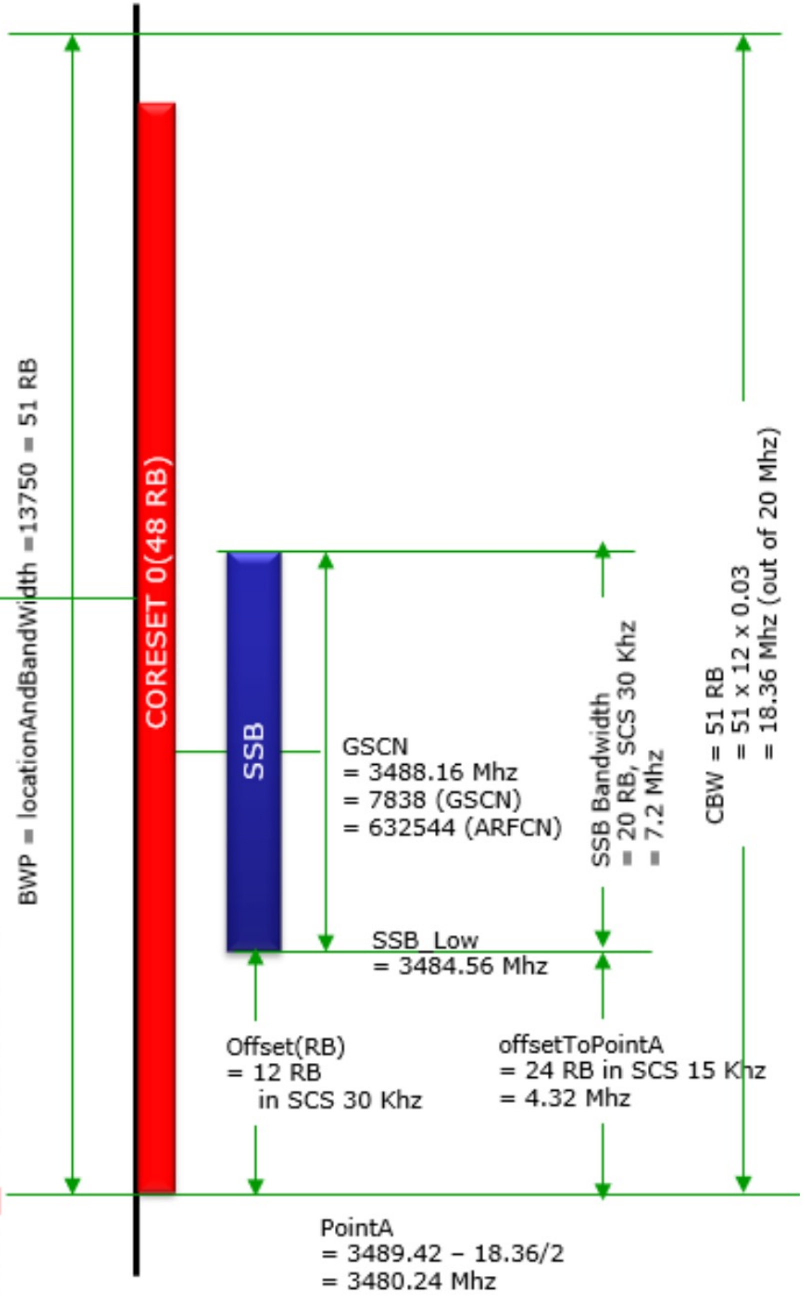


MIB.pdcch-ConfigSIB1.controlResourceSetZero = 10 , MIB.ssb-SubcarrierOffset= 0



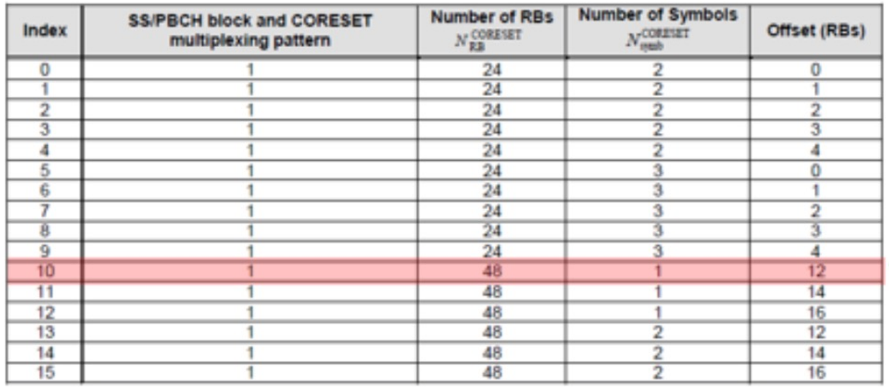
```
SIB1 :=
servingCellConfigCommon {
  downlinkConfigCommon {
    frequencyInfoDL {
      frequencyBandList {
        {
          freqBandIndicatorNR 78
        }
      },
      offsetToPointA 24,
      scs-SpecificCarrierList {
        {
          offsetToCarrier 0,
          subcarrierSpacing kHz30,
          carrierBandwidth 51
        }
      }
    },
    initialDownlinkBWP {
      genericParameters {
        locationAndBandwidth 13750,
        subcarrierSpacing kHz30
      },
```

Center Freq
= 632628 (ARFCN)
= 3489.42 (Mhz)



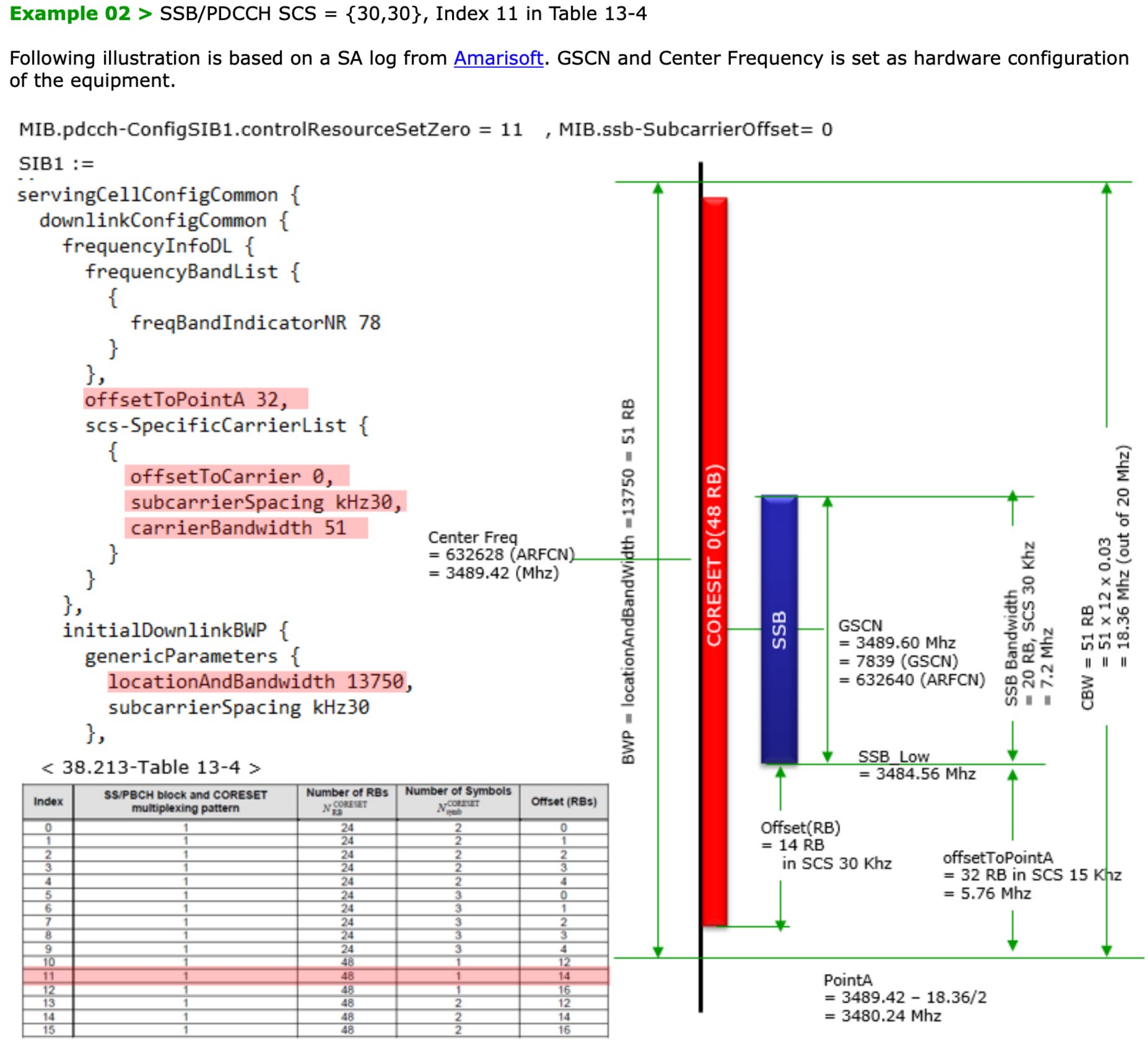
SSB/PDCCH SCS = {30,30}, Index 10 in Table 13-4

< 38.213-Table 13-4 >



Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{RB}^{CORESET}$	Number of Symbols $N_{symb}^{CORESET}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	1
2	1	24	2	2
3	1	24	2	3
4	1	24	2	4
5	1	24	3	0
6	1	24	3	1
7	1	24	3	2
8	1	24	3	3
9	1	24	3	4
10	1	48	1	12
11	1	48	1	14
12	1	48	1	16
13	1	48	2	12
14	1	48	2	14
15	1	48	2	16

From:https://www.sharetechnote.com/html/5G/5G_CommonSearchSpace_Type0_PDCCH.html



Example 02 > SSB/PDCCH SCS = {30,30}, Index 11 in Table 13-4

Following illustration is based on a SA log from Amarisoft. GSCN and Center Frequency is set as hardware configuration of the equipment.

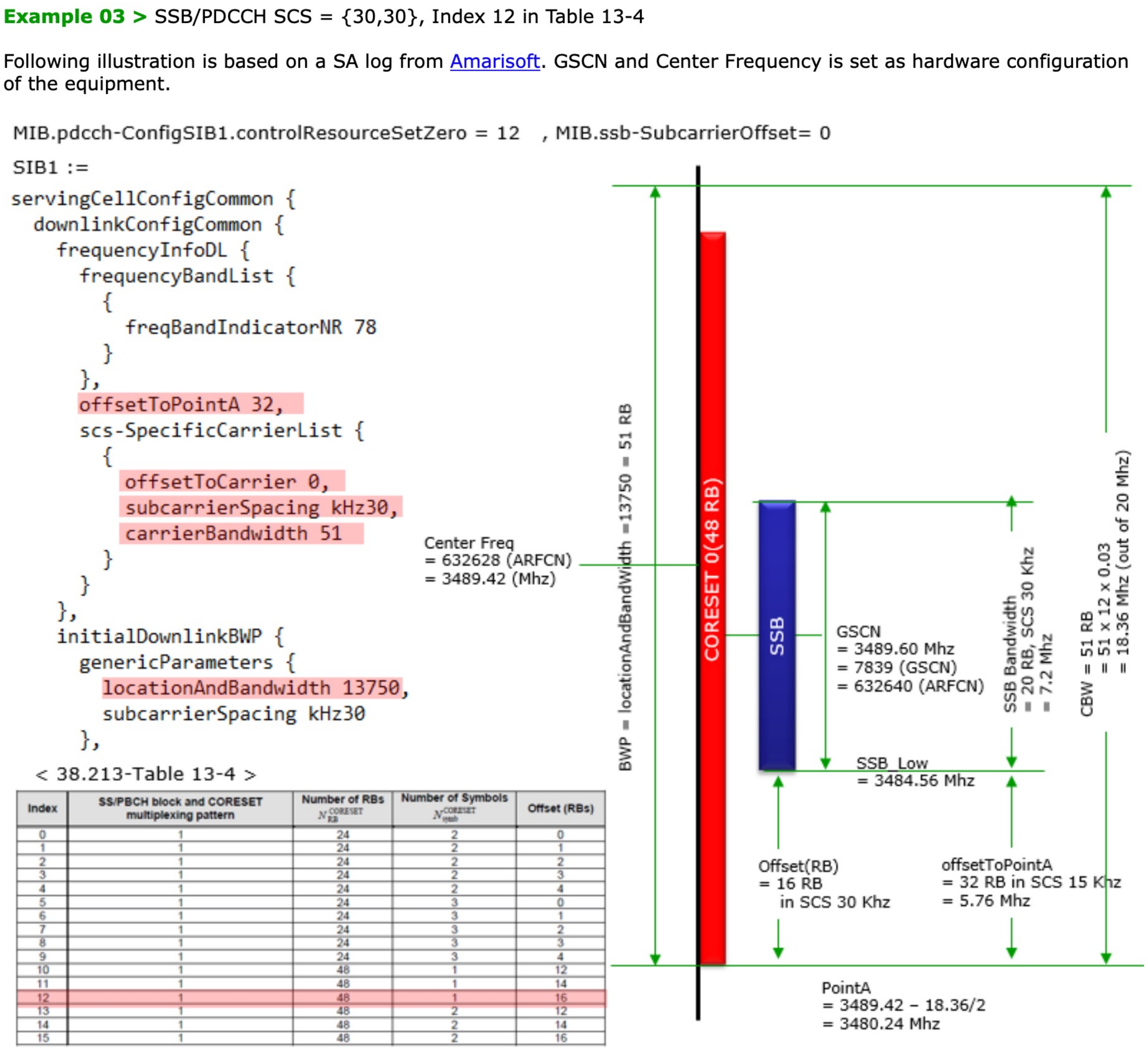
MIB.pdcch-ConfigSIB1.controlResourceSetZero = 11 , MIB.ssb-SubcarrierOffset= 0

```
SIB1 :=
servingCellConfigCommon {
  downlinkConfigCommon {
    frequencyInfoDL {
      frequencyBandList {
        {
          freqBandIndicatorNR 78
        }
      },
      offsetToPointA 32,
      scs-SpecificCarrierList {
        {
          offsetToCarrier 0,
          subcarrierSpacing kHz30,
          carrierBandwidth 51
        }
      }
    },
    initialDownlinkBWP {
      genericParameters {
        locationAndBandwidth 13750,
        subcarrierSpacing kHz30
      },
```

< 38.213-Table 13-4 >

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{RB}^{CORESET}$	Number of Symbols $N_{symb}^{CORESET}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	1
2	1	24	2	2
3	1	24	2	3
4	1	24	2	4
5	1	24	3	0
6	1	24	3	1
7	1	24	3	2
8	1	24	3	3
9	1	24	3	4
10	1	48	1	12
11	1	48	1	14
12	1	48	1	16
13	1	48	2	12
14	1	48	2	14
15	1	48	2	16

From:https://www.sharetechnote.com/html/5G/5G_CommonSearchSpace_Type0_PDCCH.html



Example 03 > SSB/PDCCH SCS = {30,30}, Index 12 in Table 13-4

Following illustration is based on a SA log from Amarisoft. GSCN and Center Frequency is set as hardware configuration of the equipment.

```
MIB.pdcch-ConfigSIB1.controlResourceSetZero = 12   , MIB.ssb-SubcarrierOffset= 0

SIB1 :=
servingCellConfigCommon {
  downlinkConfigCommon {
    frequencyInfoDL {
      frequencyBandList {
        {
          freqBandIndicatorNR 78
        }
      },
      offsetToPointA 32,
      scs-SpecificCarrierList {
        {
          offsetToCarrier 0,
          subcarrierSpacing kHz30,
          carrierBandwidth 51
        }
      }
    },
    initialDownlinkBWP {
      genericParameters {
        locationAndBandwidth 13750,
        subcarrierSpacing kHz30
      },
```

< 38.213-Table 13-4 >

Index	SS/PBCH block and CORESET multiplexing pattern	Number of RBs $N_{RB}^{CORESET}$	Number of Symbols $N_{symb}^{CORESET}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	1
2	1	24	2	2
3	1	24	2	3
4	1	24	2	4
5	1	24	3	0
6	1	24	3	1
7	1	24	3	2
8	1	24	3	3
9	1	24	3	4
10	1	48	1	12
11	1	48	1	14
12	1	48	1	16
13	1	48	2	12
14	1	48	2	14
15	1	48	2	16

From:https://www.sharetechnote.com/html/5G/5G_CommonSearchSpace_Type0_PDCCH.html

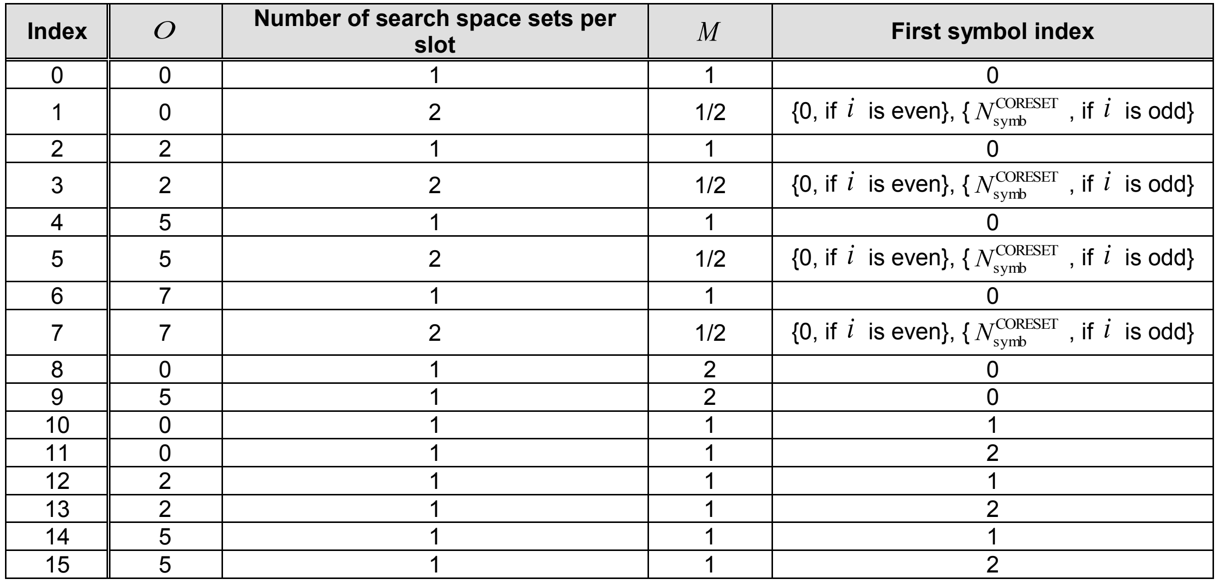
38.213 [13]

Time location

- From the previous table we also find the value of
  - Coresetmultiplexingpattern
- From the value of Coresetmultiplexingpattern, and frequency range (Fr1/Fr2) we choose a table from 13-11 to 13-15
- The index is chosen from the other 4 bits
  - SearchSpaceZero
- From this table, we can derive the slot locations and other parameters (next slide)

Example table

Table 13-11: Parameters for PDCCH monitoring occasions for Type0-PDCCH CSS set - SS/PBCH block and CORESET multiplexing pattern 1 and FR1



Index	O	Number of search space sets per slot	M	First symbol index
0	0	1	1	0
1	0	2	1/2	{0, if i is even}, { $N_{\text{symb}}^{\text{CORESET}}$, if i is odd}
2	2	1	1	0
3	2	2	1/2	{0, if i is even}, { $N_{\text{symb}}^{\text{CORESET}}$, if i is odd}
4	5	1	1	0
5	5	2	1/2	{0, if i is even}, { $N_{\text{symb}}^{\text{CORESET}}$, if i is odd}
6	7	1	1	0
7	7	2	1/2	{0, if i is even}, { $N_{\text{symb}}^{\text{CORESET}}$, if i is odd}
8	0	1	2	0
9	5	1	2	0
10	0	1	1	1
11	0	1	1	2
12	2	1	1	1
13	2	1	1	2
14	5	1	1	1
15	5	1	1	2

i: SSB block index

38.213 [13]

Slot # of Coreset 0 (Multiplexing pattern =1)

- Two consecutive slots starting at slot index n$_0$
  - UE monitors **both of these** slots. BS **need not** send the SIB1 in both.
- For SSBlock with index i

$$n_0 = \left(O \cdot 2^{\mu} + \lfloor i \cdot M \rfloor\right) \bmod N_{\text{slot}}^{\text{frame},\mu}$$

From the table (see previous slide)

- The SFN index is
  - Even frames if

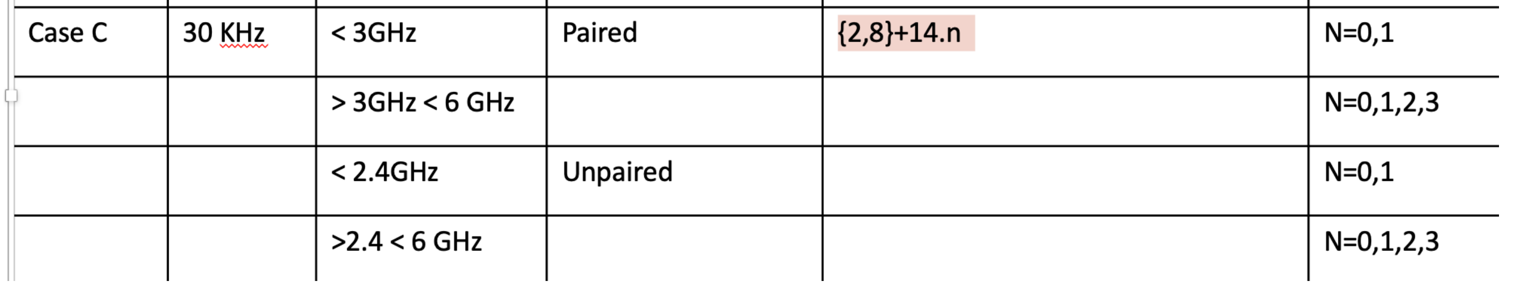
$$\left\lfloor \left(O \cdot 2^{\mu} + \lfloor i \cdot M \rfloor\right) / N_{\text{slot}}^{\text{frame},\mu} \right\rfloor \bmod 2 = 0$$

  - Odd frame if

$$\left\lfloor \left(O \cdot 2^{\mu} + \lfloor i \cdot M \rfloor\right) / N_{\text{slot}}^{\text{frame},\mu} \right\rfloor \bmod 2 = 1 .$$

- The index of the starting symbol in a slot is provided in the table (previous slide)

Example (with SSB) 3.5 GHz

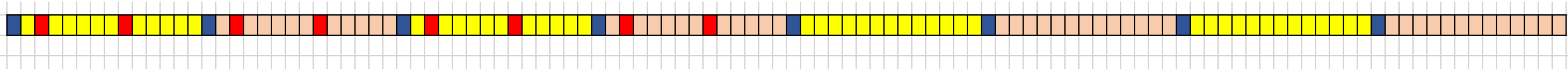


Case C	30 KHz	< 3GHz	Paired	{2,8}+14.n	N=0,1
		> 3GHz < 6 GHz			N=0,1,2,3
		< 2.4GHz	Unpaired		N=0,1
		>2.4 < 6 GHz			N=0,1,2,3

Assumptions
Lmax = 8 (Assume all of them are there)
First half frame
Periodicity of 10ms
SearchSpaceZero =0
Controlreourceset =0

SLOTS in which SSB is present: 0, 1, 2, 3 (Two instances in each slot)
ControlResourceSet = 0 → 24 RB and 2 symbols and multiplexing pattern =1;
SearchSpaceZero = 0 → M=1, O=0 and starting symbol = 0
→ N_0 = (0+i)mod14 → N_0 = i.
→ Floor(i/14)mod 2 =0 → Even SFN.

So SIB1 will be there in Slots 0,1,2,3,4,5,6,7,8 (possible locations) and Even SF.



Red: SSB
Blue: SIB1

Coreset0 has the control information for SIB1

- SIB1 is in PDSCH
- Coreset0 (PDCCH) provides the location of SIB1 in PDSCH

38.213 [13]

Slot # of Coreset 0 (Multiplexing pattern =2,3)

- One slot
- Periodicity equals the periodicity of the SS block
- Slot (n_c) and SFN provided in the table itself

Example

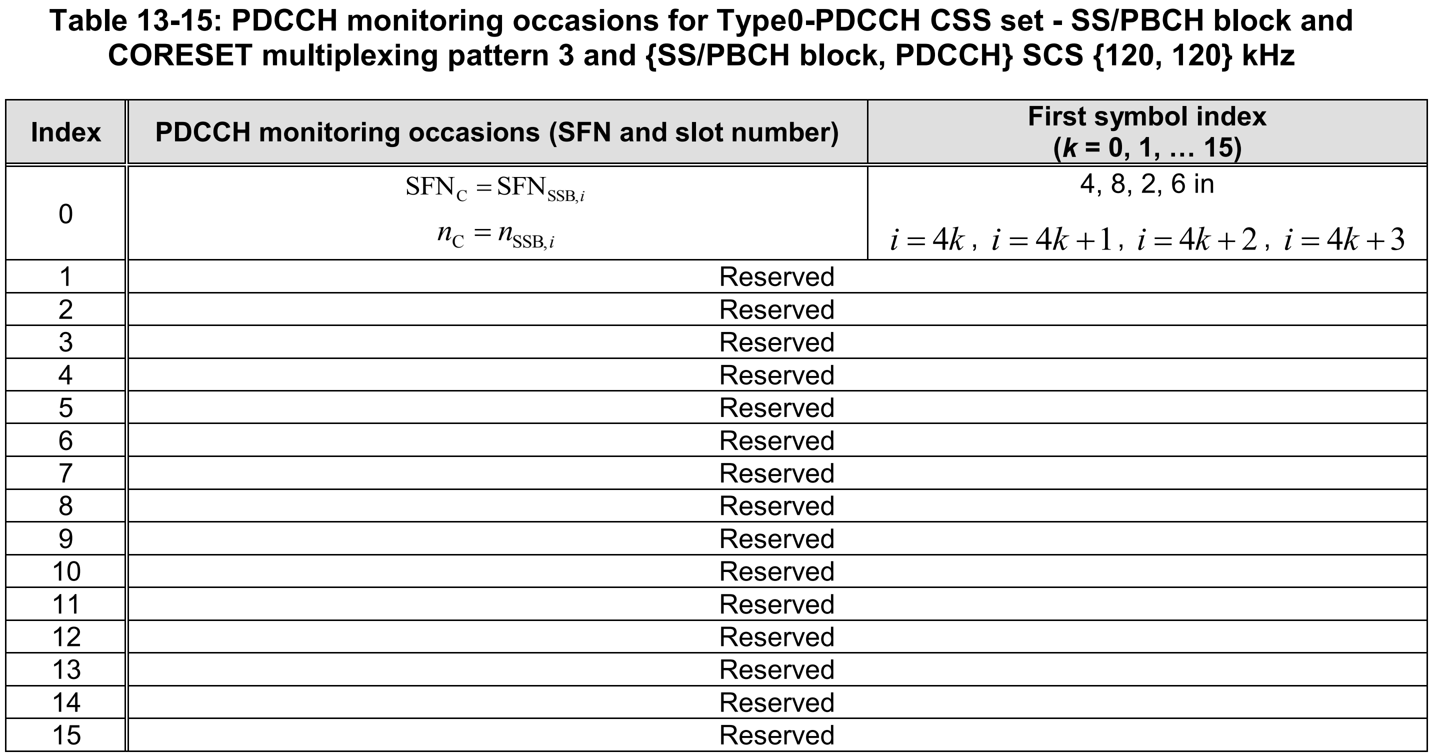


Table 13-15: PDCCH monitoring occasions for Type0-PDCCH CSS set - SS/PBCH block and CORESET multiplexing pattern 3 and {SS/PBCH block, PDCCH} SCS {120, 120} kHz

Index	PDCCH monitoring occasions (SFN and slot number)	First symbol index (*k* = 0, 1, … 15)
0	$\mathrm{SFN_C} = \mathrm{SFN}_{\mathrm{SSB},i}$ $n_\mathrm{C} = n_{\mathrm{SSB},i}$	4, 8, 2, 6 in $i = 4k$, $i = 4k+1$, $i = 4k+2$, $i = 4k+3$
1	Reserved	
2	Reserved	
3	Reserved	
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	
8	Reserved	
9	Reserved	
10	Reserved	
11	Reserved	
12	Reserved	
13	Reserved	
14	Reserved	
15	Reserved	

What information is there in SIB1?

- Cell specific SIB (System information block)
- Has information about PRACH
- Has information about other important System parameters (SI messages)
    - See next slides.

38.331 [5.2]

Periodicity

- DL-SCH channel
- SIB1 is transmitted with a periodicity of 160 ms.
  - Variable transmission repetition periodicity within 160 ms as specified in TS 38.213 [13], clause 13
- Transmitted in PDSCH
  - The scheduling of which is done in Coreset0.
    - Format 1-0 (more about it later, when we do control channel)

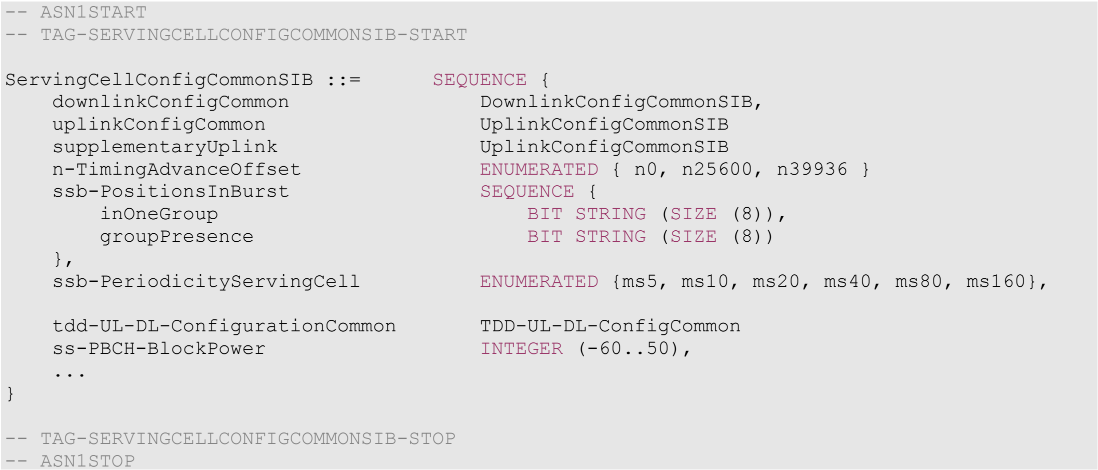
SIB1

- MIB and SIB1 are called the Minimum System Information
- SIB1 is in the PDSCH
  - Scheduled by DCI 1-0 in Coreset 0
- Transmitted every 20 ms in FR1
- Contains critical cell information and also information for other SIBs.

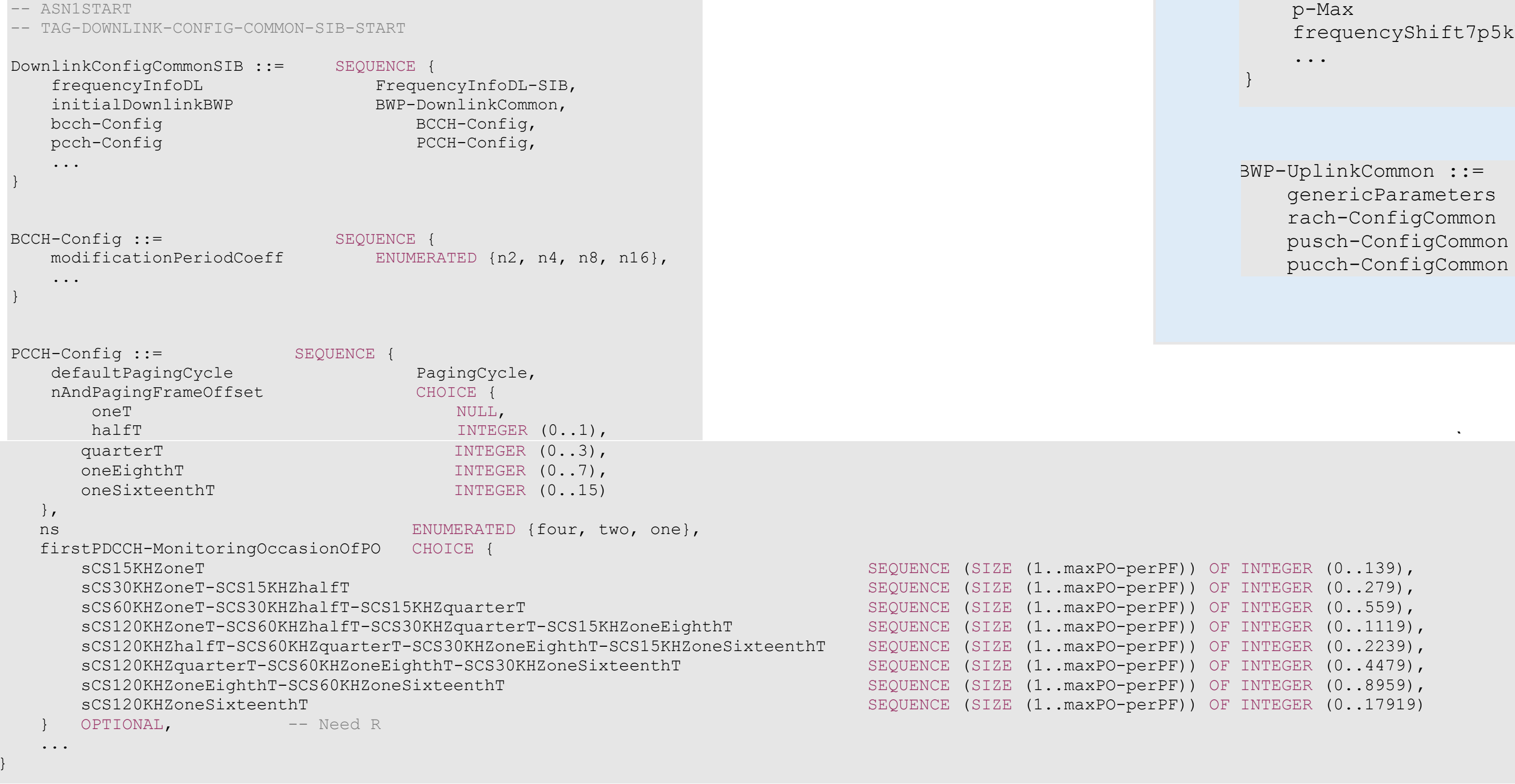
Contents of SIB1

- Cell selection information
- Cell access related information
- Connection establishment failure
- Si-scheduling info
- Serving cell Config Common
- IMS Emergency Support Flag
- eCall Over IMS Support Flag
- UE -Timer and Constants
- UAC-Barring Information
- Use Full Resume ID Flag

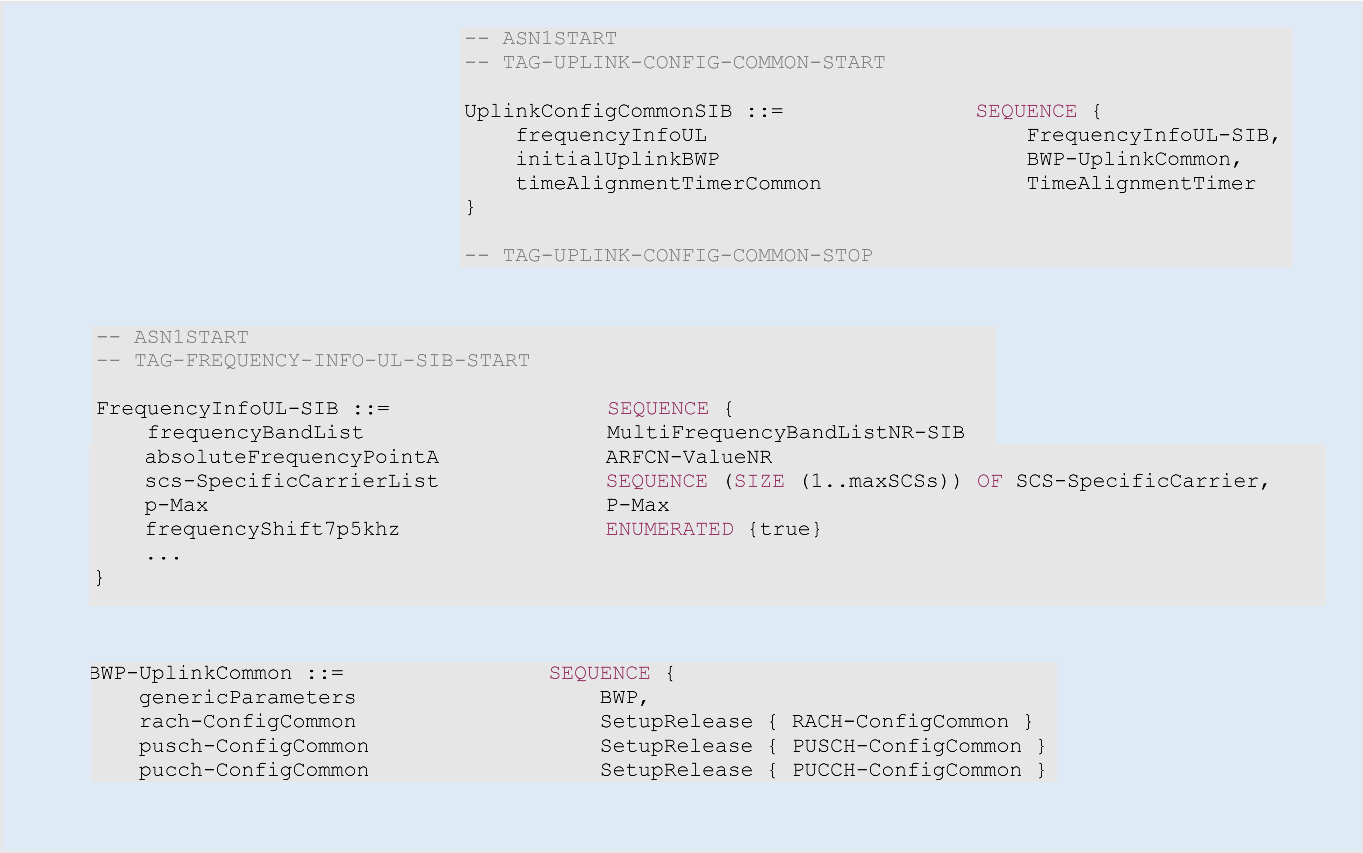
```
-- ASN1START
-- TAG-SIB1-START

SIB1 ::=        SEQUENCE {
    cellSelectionInfo                   SEQUENCE {
        q-RxLevMin                          Q-RxLevMin,
        q-RxLevMinOffset                    INTEGER (1..8)                                              OPTIONAL,   -- Need S
        q-RxLevMinSUL                       Q-RxLevMin                                                  OPTIONAL,   -- Need R
        q-QualMin                           Q-QualMin                                                   OPTIONAL,   -- Need S
        q-QualMinOffset                     INTEGER (1..8)                                              OPTIONAL    -- Need S
    }                                                                                                   OPTIONAL,   -- Cond Standalone
    cellAccessRelatedInfo               CellAccessRelatedInfo,
    connEstFailureControl               ConnEstFailureControl                                           OPTIONAL,   -- Need R
    si-SchedulingInfo                   SI-SchedulingInfo                                               OPTIONAL,   -- Need R
    servingCellConfigCommon             ServingCellConfigCommonSIB                                      OPTIONAL,   -- Need R
    ims-EmergencySupport                ENUMERATED {true}                                               OPTIONAL,   -- Need R
    eCallOverIMS-Support                ENUMERATED {true}                                               OPTIONAL,   -- Cond Absent
    ue-TimersAndConstants               UE-TimersAndConstants                                           OPTIONAL,   -- Need R

    uac-BarringInfo                     SEQUENCE {
        uac-BarringForCommon                UAC-BarringPerCatList                                       OPTIONAL,   -- Need S
        uac-BarringPerPLMN-List             UAC-BarringPerPLMN-List                                     OPTIONAL,   -- Need S
        uac-BarringInfoSetList              UAC-BarringInfoSetList,
        uac-AccessCategory1-SelectionAssistanceInfo CHOICE {
            plmnCommon                           UAC-AccessCategory1-SelectionAssistanceInfo,
            individualPLMNList                   SEQUENCE (SIZE (2..maxPLMN)) OF UAC-AccessCategory1-SelectionAssistanceInfo
        }                                                                                               OPTIONAL    -- Need S
    }                                                                                                   OPTIONAL,   -- Need R

    useFullResumeID                     ENUMERATED {true}                                               OPTIONAL,   -- Need N

    lateNonCriticalExtension            OCTET STRING                                                    OPTIONAL,
    nonCriticalExtension                SEQUENCE{}                                                      OPTIONAL
}

UAC-AccessCategory1-SelectionAssistanceInfo ::= ENUMERATED {a, b, c}

-- TAG-SIB1-STOP
-- ASN1STOP
```



```
-- ASN1START
-- TAG-SERVINGCELLCONFIGCOMMONSIB-START

ServingCellConfigCommonSIB ::=      SEQUENCE {
    downlinkConfigCommon                DownlinkConfigCommonSIB,
    uplinkConfigCommon                  UplinkConfigCommonSIB
    supplementaryUplink                 UplinkConfigCommonSIB
    n-TimingAdvanceOffset               ENUMERATED { n0, n25600, n39936 }
    ssb-PositionsInBurst                SEQUENCE {
        inOneGroup                          BIT STRING (SIZE (8)),
        groupPresence                       BIT STRING (SIZE (8))
    },
    ssb-PeriodicityServingCell          ENUMERATED {ms5, ms10, ms20, ms40, ms80, ms160},

    tdd-UL-DL-ConfigurationCommon       TDD-UL-DL-ConfigCommon
    ss-PBCH-BlockPower                  INTEGER (-60..50),
    ...
}

-- TAG-SERVINGCELLCONFIGCOMMONSIB-STOP
-- ASN1STOP
```



```
-- ASN1START
-- TAG-DOWNLINK-CONFIG-COMMON-SIB-START

DownlinkConfigCommonSIB ::=     SEQUENCE {
    frequencyInfoDL                 FrequencyInfoDL-SIB,
    initialDownlinkBWP              BWP-DownlinkCommon,
    bcch-Config                         BCCH-Config,
    pcch-Config                         PCCH-Config,
    ...
}


BCCH-Config ::=                 SEQUENCE {
    modificationPeriodCoeff         ENUMERATED {n2, n4, n8, n16},
    ...
}


PCCH-Config ::=             SEQUENCE {
    defaultPagingCycle                  PagingCycle,
    nAndPagingFrameOffset               CHOICE {
        oneT                                NULL,
        halfT                               INTEGER (0..1),
        quarterT                            INTEGER (0..3),
        oneEighthT                          INTEGER (0..7),
        oneSixteenthT                       INTEGER (0..15)
    },
    ns                                  ENUMERATED {four, two, one},
    firstPDCCH-MonitoringOccasionOfPO   CHOICE {
        sCS15KHZoneT                                                        SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..139),
        sCS30KHZoneT-SCS15KHZhalfT                                          SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..279),
        sCS60KHZoneT-SCS30KHZhalfT-SCS15KHZquarterT                         SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..559),
        sCS120KHZoneT-SCS60KHZhalfT-SCS30KHZquarterT-SCS15KHZoneEighthT     SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..1119),
        sCS120KHZhalfT-SCS60KHZquarterT-SCS30KHZoneEighthT-SCS15KHZoneSixteenthT SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..2239),
        sCS120KHZquarterT-SCS60KHZoneEighthT-SCS30KHZoneSixteenthT          SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..4479),
        sCS120KHZoneEighthT-SCS60KHZoneSixteenthT                           SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..8959),
        sCS120KHZoneSixteenthT                                              SEQUENCE (SIZE (1..maxPO-perPF)) OF INTEGER (0..17919)
    }   OPTIONAL,           -- Need R
    ...
}
```



```
-- ASN1START
-- TAG-UPLINK-CONFIG-COMMON-START

UplinkConfigCommonSIB ::=               SEQUENCE {
    frequencyInfoUL                         FrequencyInfoUL-SIB,
    initialUplinkBWP                        BWP-UplinkCommon,
    timeAlignmentTimerCommon                TimeAlignmentTimer
}

-- TAG-UPLINK-CONFIG-COMMON-STOP
```

```
-- ASN1START
-- TAG-FREQUENCY-INFO-UL-SIB-START

FrequencyInfoUL-SIB ::=                 SEQUENCE {
    frequencyBandList                       MultiFrequencyBandListNR-SIB
    absoluteFrequencyPointA                 ARFCN-ValueNR
    scs-SpecificCarrierList                 SEQUENCE (SIZE (1..maxSCSs)) OF SCS-SpecificCarrier,
    p-Max                                   P-Max
    frequencyShift7p5khz                    ENUMERATED {true}
    ...
}
```

```
BWP-UplinkCommon ::=                    SEQUENCE {
    genericParameters                       BWP,
    rach-ConfigCommon                       SetupRelease { RACH-ConfigCommon }
    pusch-ConfigCommon                      SetupRelease { PUSCH-ConfigCommon }
    pucch-ConfigCommon                      SetupRelease { PUCCH-ConfigCommon }
```

SIB1 messages

SIB1 field descriptions
cellSelectionInfo Parameters for cell selection related to the serving cell.
ims-EmergencySupport Indicates whether the cell supports IMS emergency bearer services for UEs in limited service mode. If absent, IMS emergency call is not supported by the network in the cell for UEs in limited service mode.
q-QualMin Parameter "$Q_{qualmin}$" in TS 38.304 [20], applicable for serving cell. If the field is not present, the UE applies the (default) value of negative infinity for $Q_{qualmin}$.
q-QualMinOffset Parameter "$Q_{qualminoffset}$" in TS 38.304 [20]. Actual value $Q_{qualminoffset}$ = field value [dB]. If the field is not present, the UE applies the (default) value of 0 dB for $Q_{qualminoffset}$. Affects the minimum required quality level in the cell.
q-RxLevMin Parameter "$Q_{rxlevmin}$" in TS 38.304 [20], applicable for serving cell.
q-RxLevMinOffset Parameter "$Q_{rxlevminoffset}$" in TS 38.304 [20]. Actual value $Q_{rxlevminoffset}$ = field value * 2 [dB]. If absent, the UE applies the (default) value of 0 dB for $Q_{rxlevminoffset}$. Affects the minimum required Rx level in the cell.
q-RxLevMinSUL Parameter "$Q_{rxlevminSUL}$" in TS 38.304 [20], applicable for serving cell
servingCellConfigCommon Configuration of the serving cell.
uac-AccessCategory1-SelectionAssistanceInfo Information used to determine whether Access Category 1 applies to the UE, as defined in TS 22.261 [25]. A UE compliant with this version of the specification shall ignore this field.
uac-BarringForCommon Common access control parameters for each access category. Common values are used for all PLMNs, unless overwritten by the PLMN specific configuration provided in *uac-BarringPerPLMN-List*. The parameters are specified by providing an index to the set of configurations (*uac-BarringInfoSetList*). UE behaviour upon absence of this field is specified in clause 5.3.14.2.
ue-TimersAndConstants Timer and constant values to be used by the UE.
useFullResumeID Indicates which resume identifier and Resume request message should be used. UE uses full I-RNTI and *RRCResumeRequest1* if the field is present, or short I-RNTI and *RRCResumeRequest* if the field is absent.

Cell selection information

38.304

- q-RxLevMin
  - Minimum required RX level in the cell (dBm).
    - Field: INTEGER (-70..-22)
    - The IE Q-RxLevMin is used to indicate for cell selection/ re-selection the required minimum received RSRP level in the (NR) cell.
    - Actual value $Q_{rxlevmin}$ = field value * 2 [dBm].
  - $Q_{rxlevminoffset}$
    - Offset to the required RX level
- q-QualMin (q-QualMinOffset)
  - Minimum required quality level in the cell (dB).

$\text{Srxlev} = Q_{rxlevmeas} – (Q_{rxlevmin} + Q_{rxlevminoffset}) – P_{compensation} - \text{Qoffset}_{temp}$

$\text{Squal} = Q_{qualmeas} – (Q_{qualmin} + Q_{qualminoffset}) – \text{Qoffset}_{temp}$

$Q_{rxlevmeas}$	Measured cell RX level value (RSRP)
$Q_{qualmeas}$	Measured cell quality value (RSRQ)

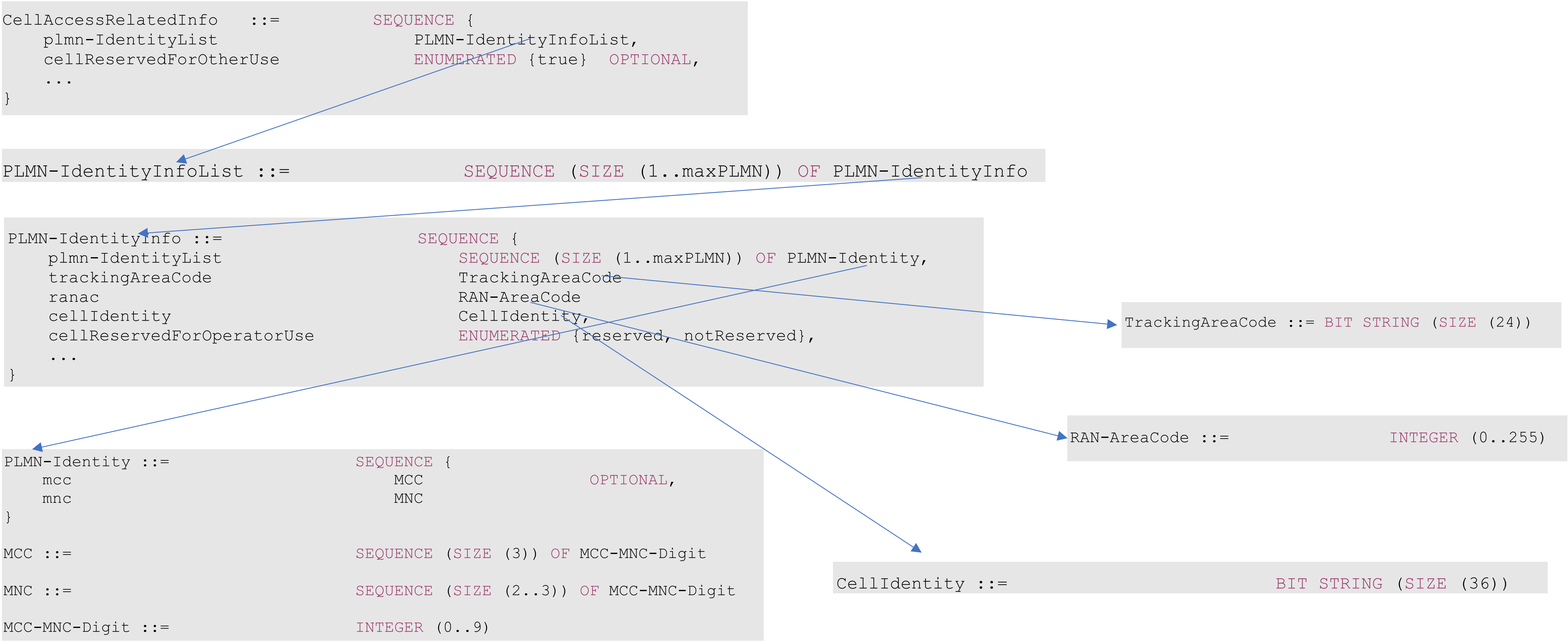
The cell selection criterion S is fulfilled when:

Srxlev > 0 AND Squal > 0

What is the difference between RSRP and RSRQ?

Will come to this slide later

CellAccessRelatedInfo



```
CellAccessRelatedInfo    ::=          SEQUENCE {
    plmn-IdentityList                     PLMN-IdentityInfoList,
    cellReservedForOtherUse               ENUMERATED {true}   OPTIONAL,
    ...
}
```

```
PLMN-IdentityInfoList ::=                 SEQUENCE (SIZE (1..maxPLMN)) OF PLMN-IdentityInfo
```

```
PLMN-IdentityInfo ::=                     SEQUENCE {
    plmn-IdentityList                         SEQUENCE (SIZE (1..maxPLMN)) OF PLMN-Identity,
    trackingAreaCode                          TrackingAreaCode
    ranac                                     RAN-AreaCode
    cellIdentity                              CellIdentity,
    cellReservedForOperatorUse                ENUMERATED {reserved, notReserved},
    ...
}
```

```
TrackingAreaCode ::= BIT STRING (SIZE (24))
```

```
RAN-AreaCode ::=                     INTEGER (0..255)
```

```
PLMN-Identity ::=                     SEQUENCE {
    mcc                                   MCC                   OPTIONAL,
    mnc                                   MNC
}

MCC ::=                               SEQUENCE (SIZE (3)) OF MCC-MNC-Digit

MNC ::=                               SEQUENCE (SIZE (2..3)) OF MCC-MNC-Digit

MCC-MNC-Digit ::=                     INTEGER (0..9)
```

```
CellIdentity ::=                               BIT STRING (SIZE (36))
```

https://itectec.com/spec/5g-nr-ue-cell-reservations-and-access-restrictions/

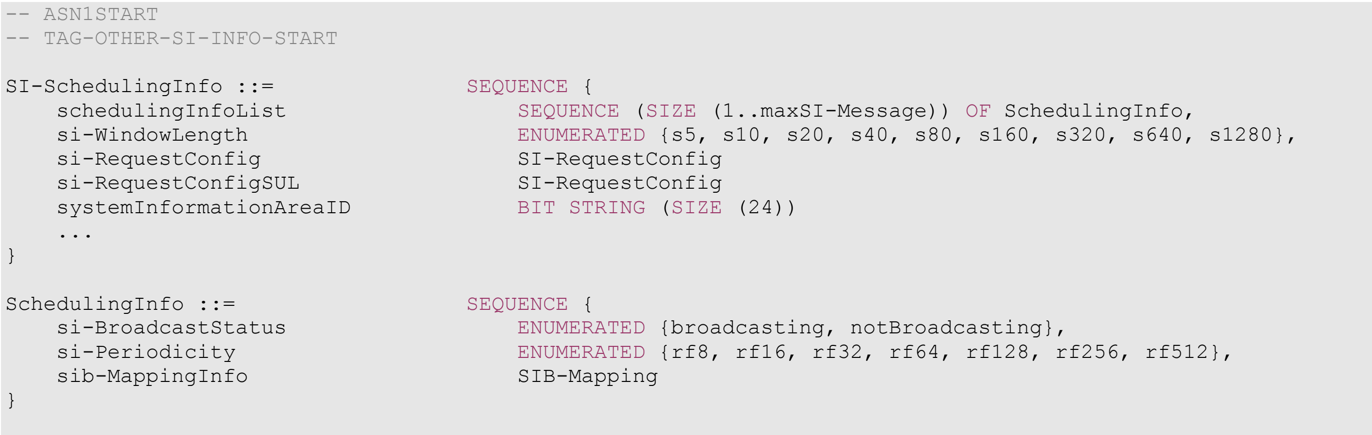
ConnEstFailureControl

```
ConnEstFailureControl ::=   SEQUENCE {
    connEstFailCount                      ENUMERATED {n1, n2, n3, n4},
    connEstFailOffsetValidity             ENUMERATED {s30, s60, s120, s240, s300, s420, s600, s900},
    connEstFailOffset                     INTEGER (0..15)
}
```

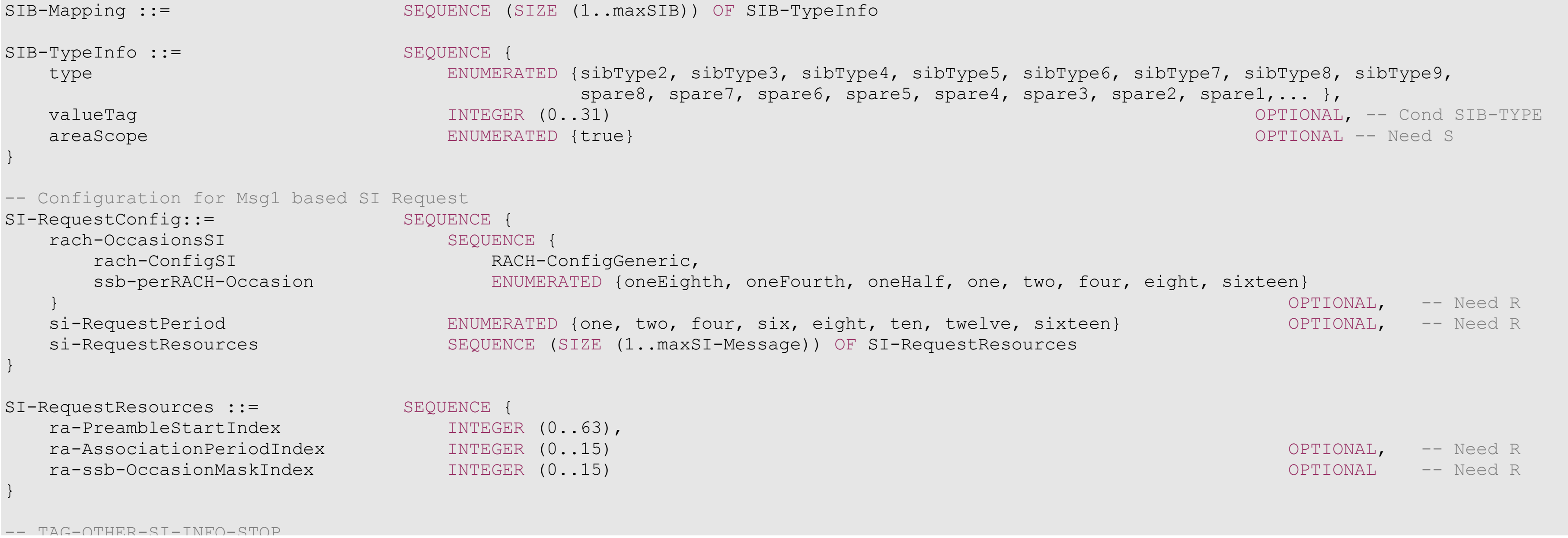
Handles re-connection timers and additional offsets for power controls

http://howltestuffworks.blogspot.com/2019/10/5g-nr-system-information-block-1-sib1.html

SI-SchedulingInfo



```
-- ASN1START
-- TAG-OTHER-SI-INFO-START

SI-SchedulingInfo ::=               SEQUENCE {
    schedulingInfoList                  SEQUENCE (SIZE (1..maxSI-Message)) OF SchedulingInfo,
    si-WindowLength                     ENUMERATED {s5, s10, s20, s40, s80, s160, s320, s640, s1280},
    si-RequestConfig                    SI-RequestConfig
    si-RequestConfigSUL                 SI-RequestConfig
    systemInformationAreaID             BIT STRING (SIZE (24))
    ...
}

SchedulingInfo ::=                  SEQUENCE {
    si-BroadcastStatus                  ENUMERATED {broadcasting, notBroadcasting},
    si-Periodicity                      ENUMERATED {rf8, rf16, rf32, rf64, rf128, rf256, rf512},
    sib-MappingInfo                     SIB-Mapping
}
```



```
SIB-Mapping ::=                     SEQUENCE (SIZE (1..maxSIB)) OF SIB-TypeInfo

SIB-TypeInfo ::=                    SEQUENCE {
    type                                ENUMERATED {sibType2, sibType3, sibType4, sibType5, sibType6, sibType7, sibType8, sibType9,
                                                    spare8, spare7, spare6, spare5, spare4, spare3, spare2, spare1,... },
    valueTag                            INTEGER (0..31)                                                          OPTIONAL, -- Cond SIB-TYPE
    areaScope                           ENUMERATED {true}                                                        OPTIONAL -- Need S
}

-- Configuration for Msg1 based SI Request
SI-RequestConfig::=                 SEQUENCE {
    rach-OccasionsSI                    SEQUENCE {
        rach-ConfigSI                       RACH-ConfigGeneric,
        ssb-perRACH-Occasion                ENUMERATED {oneEighth, oneFourth, oneHalf, one, two, four, eight, sixteen}
    }                                                                                                           OPTIONAL,   -- Need R
    si-RequestPeriod                    ENUMERATED {one, two, four, six, eight, ten, twelve, sixteen}               OPTIONAL,   -- Need R
    si-RequestResources                 SEQUENCE (SIZE (1..maxSI-Message)) OF SI-RequestResources
}

SI-RequestResources ::=             SEQUENCE {
    ra-PreambleStartIndex               INTEGER (0..63),
    ra-AssociationPeriodIndex           INTEGER (0..15)                                                             OPTIONAL,   -- Need R
    ra-ssb-OccasionMaskIndex            INTEGER (0..15)                                                             OPTIONAL    -- Need R
}

-- TAG-OTHER-SI-INFO-STOP
```

- Minimum System Information (Minimum SI)
    - MIB and SIB1
    - Sufficient for UE to lock on to cell
- Other system information
    - Consists of all SiB and information not in Minimum SI

Handles Scheduling other SIB messages (System Information)

http://howltestuffworks.blogspot.com/2019/10/5g-nr-system-information.html

ServingCellConfigCommon

All the important cell parameters are in this.

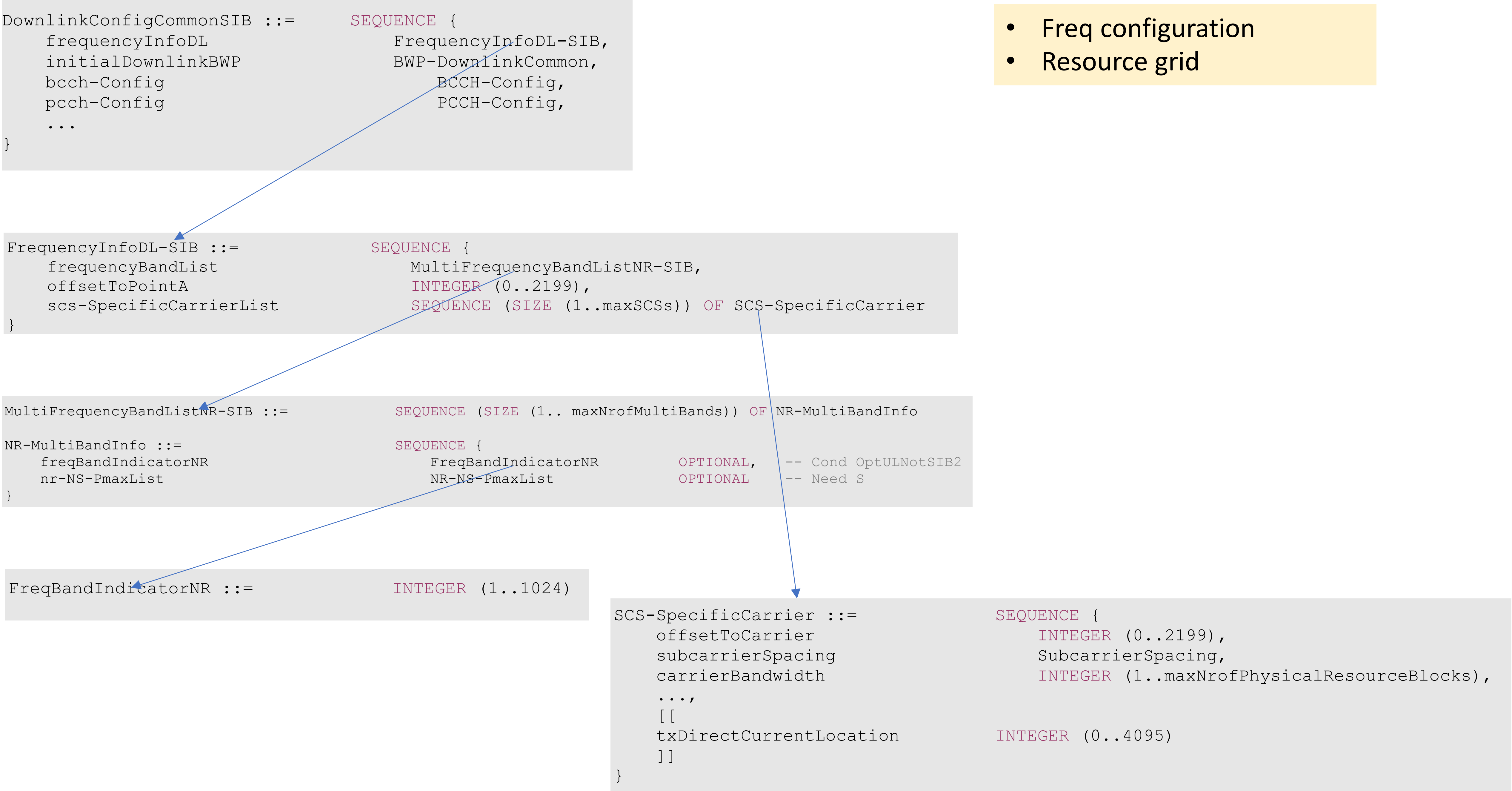
```
ServingCellConfigCommonSIB ::=      SEQUENCE {
    downlinkConfigCommon                DownlinkConfigCommonSIB,
    uplinkConfigCommon                  UplinkConfigCommonSIB
    supplementaryUplink                 UplinkConfigCommonSIB
    n-TimingAdvanceOffset               ENUMERATED { n0, n25600, n39936 }
    ssb-PositionsInBurst                SEQUENCE {
        inOneGroup                          BIT STRING (SIZE (8)),
        groupPresence                       BIT STRING (SIZE (8))
    },
    ssb-PeriodicityServingCell          ENUMERATED {ms5, ms10, ms20, ms40, ms80, ms160},

    tdd-UL-DL-ConfigurationCommon       TDD-UL-DL-ConfigCommon
    ss-PBCH-BlockPower                  INTEGER (-60..50),
    ...
}
```

```
TDD-UL-DL-ConfigCommon ::=          SEQUENCE {
    referenceSubcarrierSpacing          SubcarrierSpacing,
    pattern1                            TDD-UL-DL-Pattern,
    pattern2                            TDD-UL-DL-Pattern
    ...
}
```

```
TDD-UL-DL-Pattern ::=                   SEQUENCE {
    dl-UL-TransmissionPeriodicity           ENUMERATED {ms0p5, ms0p625, ms1, ms1p25, ms2, ms2p5, ms5, ms10},
    nrofDownlinkSlots                       INTEGER (0..maxNrofSlots),
    nrofDownlinkSymbols                     INTEGER (0..maxNrofSymbols-1),
    nrofUplinkSlots                         INTEGER (0..maxNrofSlots),
    nrofUplinkSymbols                       INTEGER (0..maxNrofSymbols-1),
    ...,
    [[
    dl-UL-TransmissionPeriodicity-v1530     ENUMERATED {ms3, ms4}
    ]]
}
```

- TDD configuration
    - Details in later class

- Freq configuration
- Resource grid



```
DownlinkConfigCommonSIB ::=     SEQUENCE {
    frequencyInfoDL                 FrequencyInfoDL-SIB,
    initialDownlinkBWP              BWP-DownlinkCommon,
    bcch-Config                         BCCH-Config,
    pcch-Config                         PCCH-Config,
    ...
}
```

```
FrequencyInfoDL-SIB ::=             SEQUENCE {
    frequencyBandList                   MultiFrequencyBandListNR-SIB,
    offsetToPointA                      INTEGER (0..2199),
    scs-SpecificCarrierList             SEQUENCE (SIZE (1..maxSCSs)) OF SCS-SpecificCarrier
}
```

```
MultiFrequencyBandListNR-SIB ::=            SEQUENCE (SIZE (1.. maxNrofMultiBands)) OF NR-MultiBandInfo

NR-MultiBandInfo ::=                        SEQUENCE {
    freqBandIndicatorNR                         FreqBandIndicatorNR         OPTIONAL,   -- Cond OptULNotSIB2
    nr-NS-PmaxList                              NR-NS-PmaxList              OPTIONAL    -- Need S
}
```

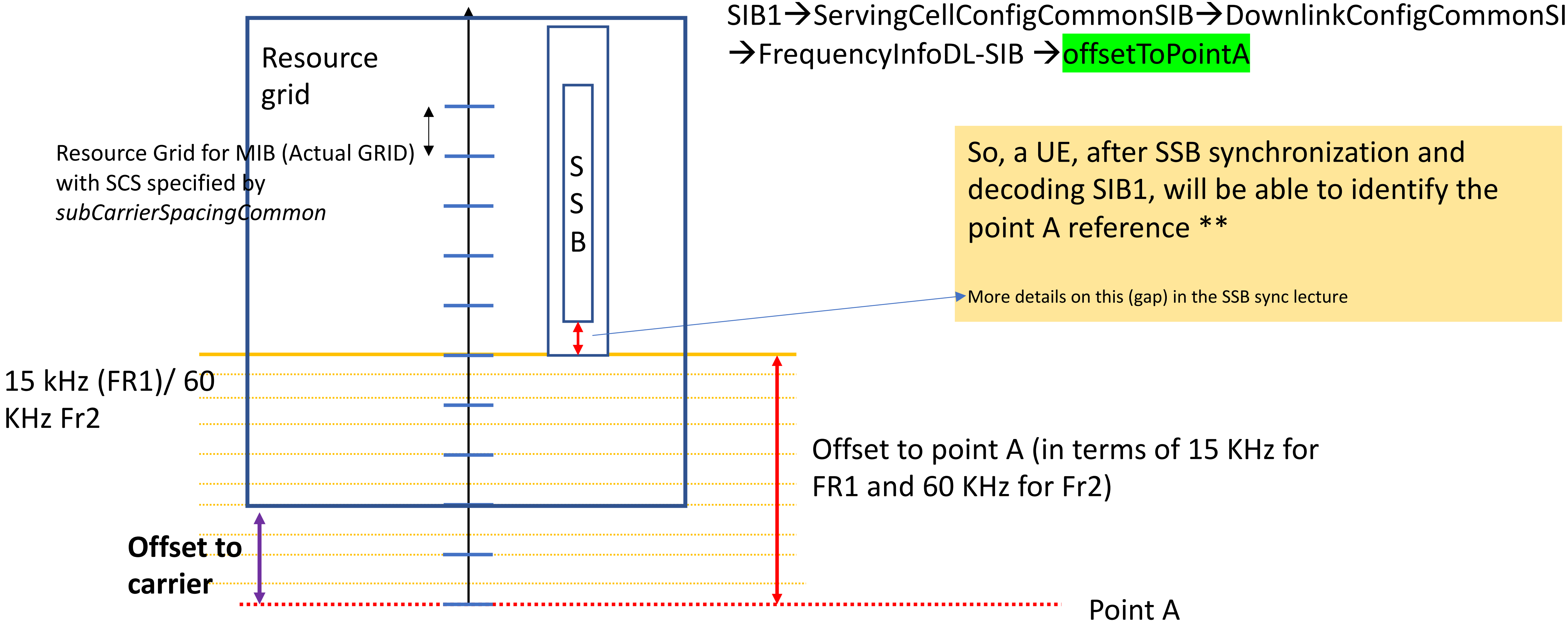
```
FreqBandIndicatorNR ::=             INTEGER (1..1024)
```

```
SCS-SpecificCarrier ::=             SEQUENCE {
    offsetToCarrier                     INTEGER (0..2199),
    subcarrierSpacing                   SubcarrierSpacing,
    carrierBandwidth                    INTEGER (1..maxNrofPhysicalResourceBlocks),
    ...,
    [[
    txDirectCurrentLocation         INTEGER (0..4095)
    ]]
}
```

Point A (Common reference point) and offset to carrier

38.211

```
FrequencyInfoDL-SIB ::=                SEQUENCE {
    frequencyBandList                      MultiFrequencyBandListNR-SIB,
    offsetToPointA                         INTEGER (0..2199),
    scs-SpecificCarrierList                SEQUENCE (SIZE (1..maxSCSs)) OF SCS-SpecificCarrier
}
```



SIB1→ServingCellConfigCommonSIB→DownlinkConfigCommonSIB →FrequencyInfoDL-SIB →offsetToPointA

Resource grid

Resource Grid for MIB (Actual GRID) with SCS specified by *subCarrierSpacingCommon*

So, a UE, after SSB synchronization and decoding SIB1, will be able to identify the point A reference **

More details on this (gap) in the SSB sync lecture

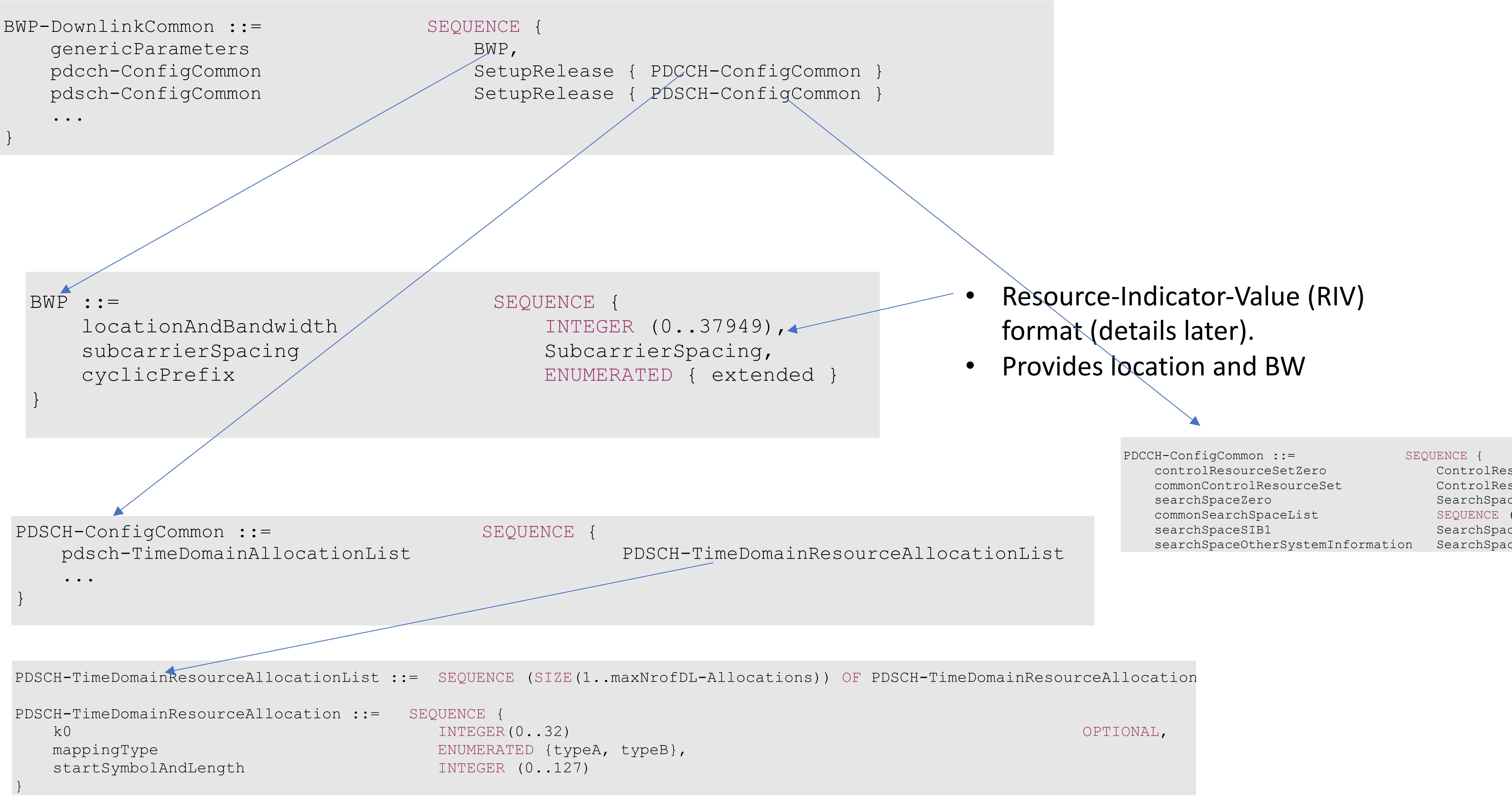
15 kHz (FR1)/ 60 KHz Fr2

Offset to point A (in terms of 15 KHz for FR1 and 60 KHz for Fr2)

Offset to carrier

Point A

**This is for Pcell. For other cells, absoluteFrequencyPointA is used and is specified in terms of ARFCN



```
BWP-DownlinkCommon ::=                   SEQUENCE {
    genericParameters                        BWP,
    pdcch-ConfigCommon                       SetupRelease { PDCCH-ConfigCommon }
    pdsch-ConfigCommon                       SetupRelease { PDSCH-ConfigCommon }
    ...
}
```

```
BWP ::=                                  SEQUENCE {
    locationAndBandwidth                     INTEGER (0..37949),
    subcarrierSpacing                        SubcarrierSpacing,
    cyclicPrefix                             ENUMERATED { extended }
}
```

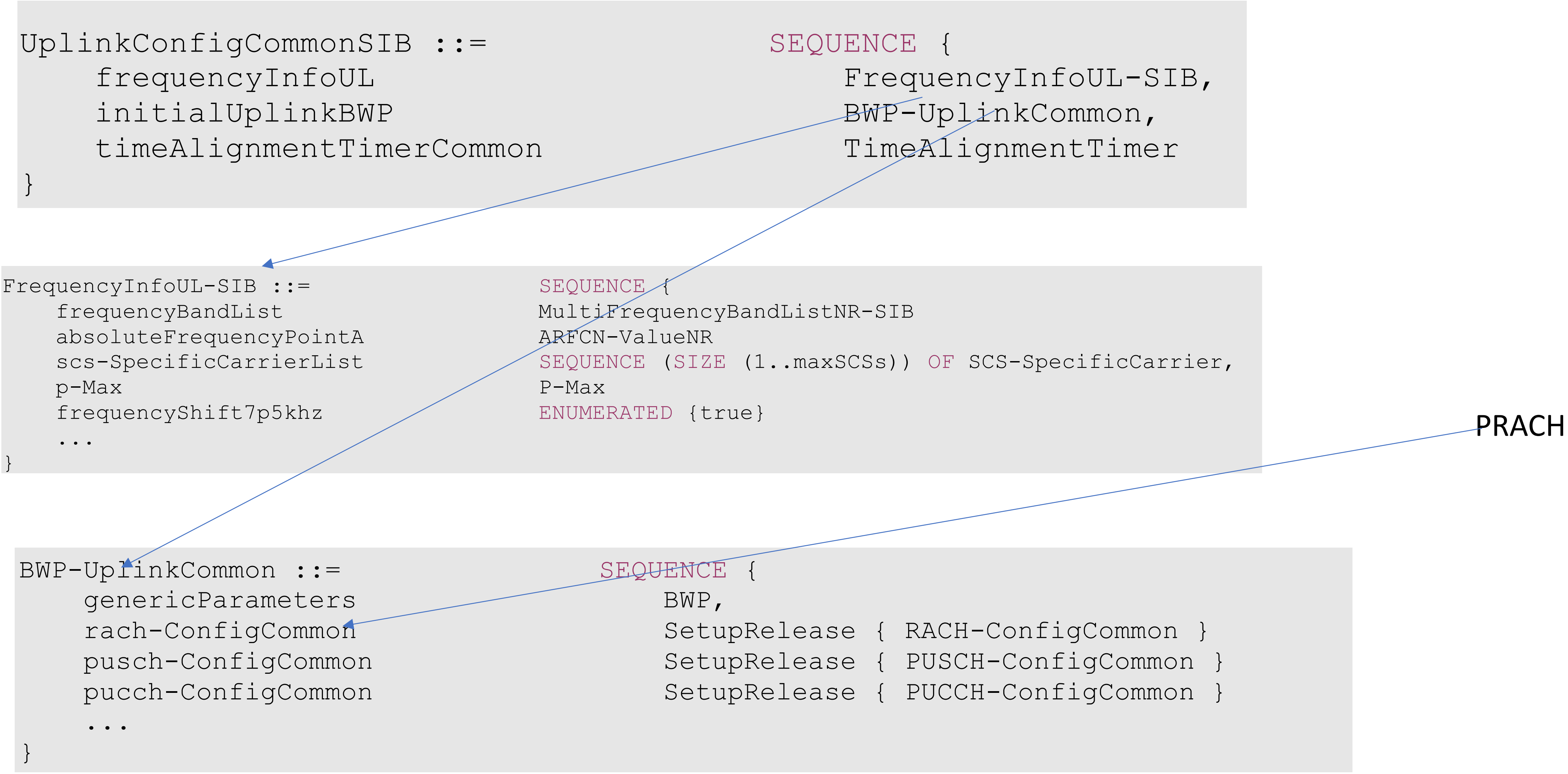
- Resource-Indicator-Value (RIV) format (details later).
- Provides location and BW

```
PDCCH-ConfigCommon ::=                   SEQUENCE {
    controlResourceSetZero                   ControlRes
    commonControlResourceSet                 ControlRes
    searchSpaceZero                          SearchSpac
    commonSearchSpaceList                    SEQUENCE (
    searchSpaceSIB1                          SearchSpac
    searchSpaceOtherSystemInformation        SearchSpac
```

```
PDSCH-ConfigCommon ::=                   SEQUENCE {
    pdsch-TimeDomainAllocationList               PDSCH-TimeDomainResourceAllocationList
    ...
}
```

```
PDSCH-TimeDomainResourceAllocationList ::=  SEQUENCE (SIZE(1..maxNrofDL-Allocations)) OF PDSCH-TimeDomainResourceAllocation

PDSCH-TimeDomainResourceAllocation ::=   SEQUENCE {
    k0                                       INTEGER(0..32)                                       OPTIONAL,
    mappingType                              ENUMERATED {typeA, typeB},
    startSymbolAndLength                     INTEGER (0..127)
}
```

UplinkConfigCommonSIB



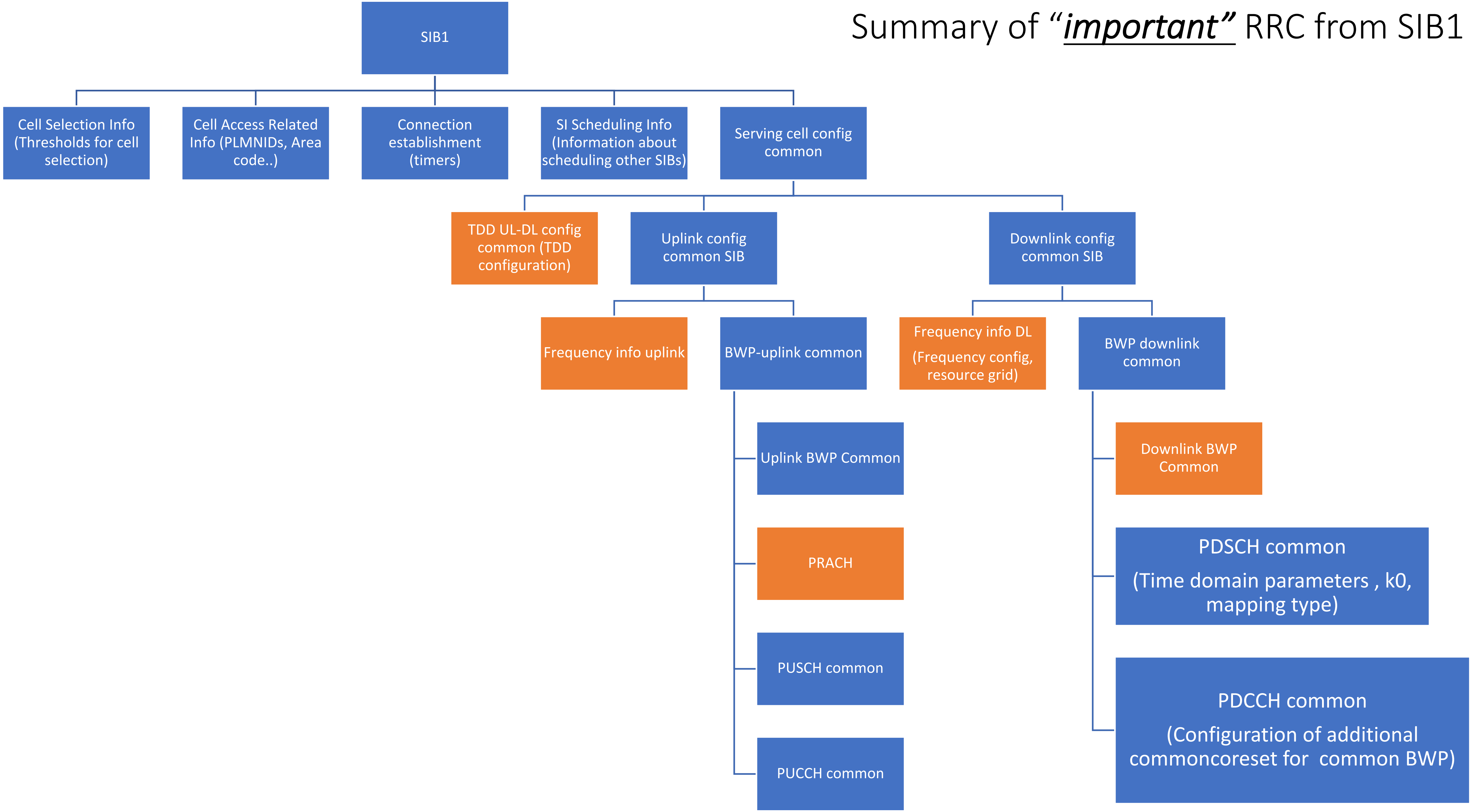
```
UplinkConfigCommonSIB ::=                   SEQUENCE {
    frequencyInfoUL                             FrequencyInfoUL-SIB,
    initialUplinkBWP                            BWP-UplinkCommon,
    timeAlignmentTimerCommon                    TimeAlignmentTimer
}
```

```
FrequencyInfoUL-SIB ::=                 SEQUENCE {
    frequencyBandList                   MultiFrequencyBandListNR-SIB
    absoluteFrequencyPointA             ARFCN-ValueNR
    scs-SpecificCarrierList             SEQUENCE (SIZE (1..maxSCSs)) OF SCS-SpecificCarrier,
    p-Max                               P-Max
    frequencyShift7p5khz                ENUMERATED {true}
    ...
}
```

PRACH

```
BWP-UplinkCommon ::=                    SEQUENCE {
    genericParameters                       BWP,
    rach-ConfigCommon                       SetupRelease { RACH-ConfigCommon }
    pusch-ConfigCommon                      SetupRelease { PUSCH-ConfigCommon }
    pucch-ConfigCommon                      SetupRelease { PUCCH-ConfigCommon }
    ...
}
```

Summary of “***important***” RRC from SIB1



- SIB1
  - Cell Selection Info (Thresholds for cell selection)
  - Cell Access Related Info (PLMNIDs, Area code..)
  - Connection establishment (timers)
  - SI Scheduling Info (Information about scheduling other SIBs)
  - Serving cell config common
    - TDD UL-DL config common (TDD configuration)
    - Uplink config common SIB
      - Frequency info uplink
      - BWP-uplink common
        - Uplink BWP Common
        - PRACH
        - PUSCH common
        - PUCCH common
    - Downlink config common SIB
      - Frequency info DL (Frequency config, resource grid)
      - BWP downlink common
        - Downlink BWP Common
        - PDSCH common (Time domain parameters , k0, mapping type)
        - PDCCH common (Configuration of additional commoncoreset for common BWP)

Questions

- How does the UE know if the system is TDD/FDD?
  - Does SIB1 have any details on this?