

# 5G随机接入流程以及相关参数

讲师：捻叶成剑

# 什么是随机接入

我们之前在PRACH物理信道中简单介绍过，在这里，在进一步的系统的讲解随机接入（本节课需要复习PRACH）本质上，随机接入，核心目的就是2个，一个是上行同步，另外一个上行资源授权（ULGrant）。

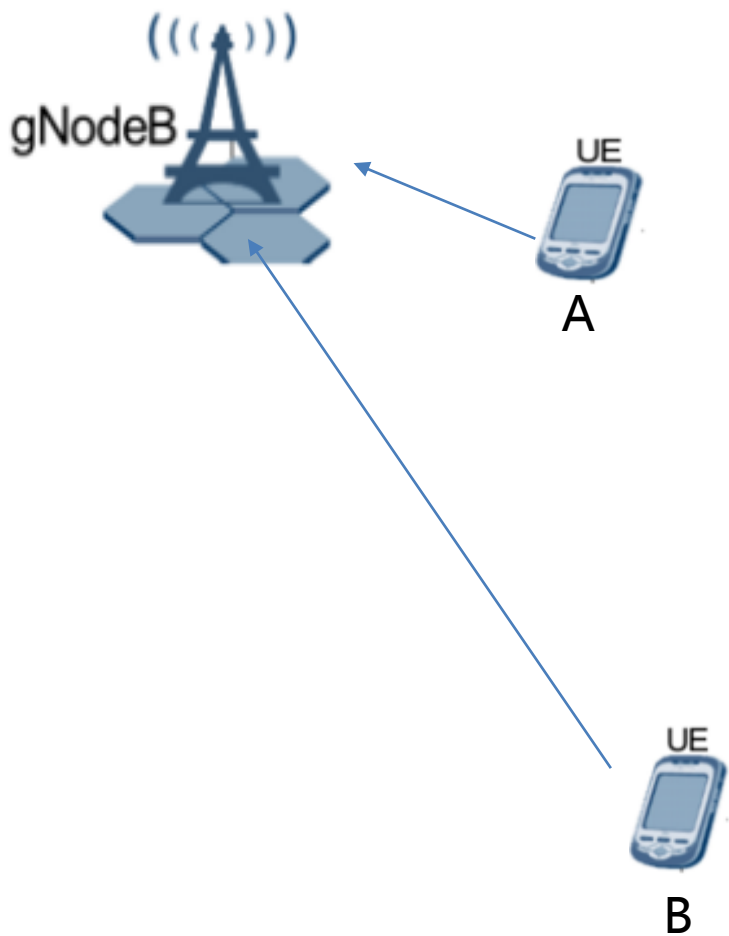
UE空闲态（RRC idle）或者处于上行失去同步状态，UE是只可以接收特定的消息（系统消息，寻呼消息等），无法接收用户数据（上网，打电话），也无法给基站发送消息（数据）的。

UE想要跟基站收发数据，首先需要进行上行同步，让UE与别的UE的发送信息节奏统一，同时，也告诉基站给自己分配上行的资源（PUSCH信道资源），上行资源分配之后，UE就可以通过信令来告诉基站自己想干啥（分很多种情况，比如建立RRC连接，或者有数据要发送等）。

在整个过程中，当然也会触发下行资源的分配（PDSCH信道资源），因为基站也有信息发给UE。

最粗略的理解，也可以认为，随机接入，就是让UE和基站之间建立上下行连接的首要步骤。

# 上行同步



↑  
上行同步

UE在这个时间点上，信号到达基站要是同步到达

对于B手机，需要一个时间提前量，好让A和B发送的信息到达基站的时间是一样的。  
这个时间提前量，叫做TA（Timing Advance）时间定时调整量

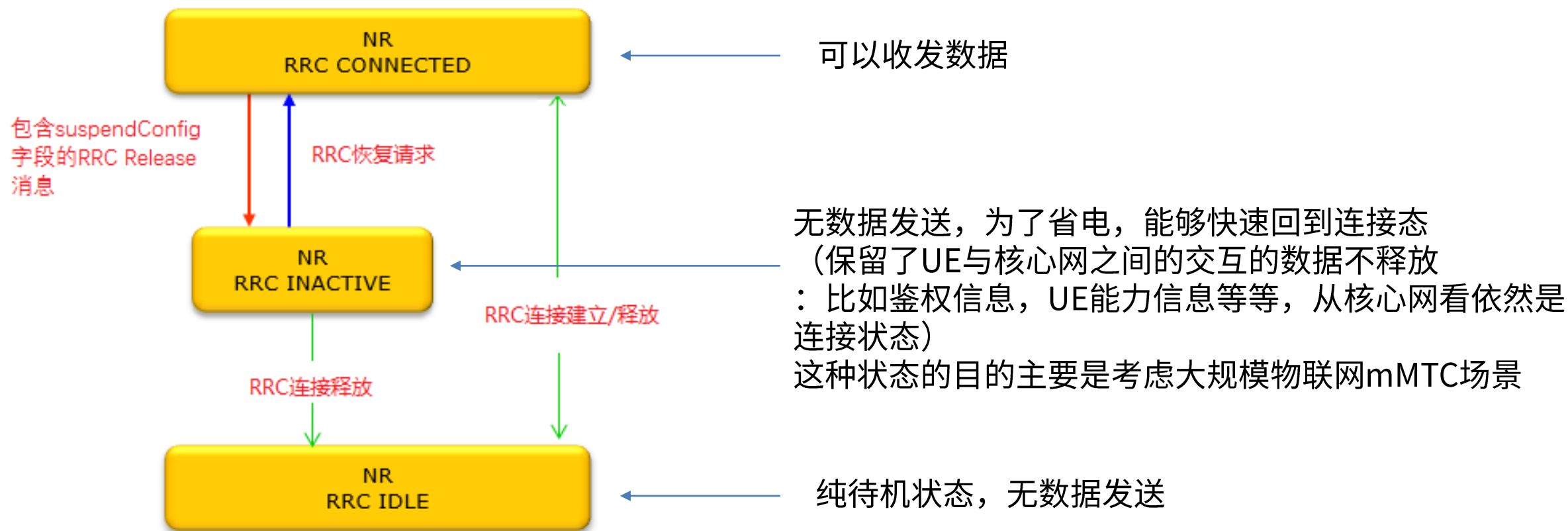
# 随机接入的触发场景

依据3GPP协议[TS38.300]，随机接入主要发生在以下几种场景：

- Initial access from RRC\_IDLE: 从idle状态发起初始RRC建立
- RRC Connection Re-establishment procedure; RRC连接重建流程
- DL or UL data arrival during RRC\_CONNECTED when UL synchronisation status is "non-synchronised"; RRC\_CONNECTED态下，上行或下行数据到达时，此时UE上行处于失步状态
- UL data arrival during RRC\_CONNECTED when there are no PUCCH resources for SR available;  
RRC\_CONNECTED态下，上行数据到达，此时UE没有用于SR的PUCCH资源时；
- SR failure; SR失败：通过随机接入过程重新获得PUCCH资源；
- Request by RRC upon synchronous reconfiguration (e.g. handover); 切换
- Transition from RRC\_INACTIVE; RRC\_INACTIVE态下的接入：UE会从RRC\_INACTIVE态到RRC\_CONNECTED态
- To establish time alignment for a secondary TAG; 在SCell添加时建立时间对齐；载波聚合场景下
- Request for other SI (see clause 7.3); 请求其他SI：UE处于RRC\_IDLE态和RRC\_CONNECTED态下时，通过随机接入过程请求其他SI（小区选择和重选里面讲过OSI可以通过请求的方式获取，也可以周期性发送）
- Beam failure recovery. 波束恢复；当UE PHY层检测到波束失步时，会通知UE MAC发起RA

# RRC状态介绍（补充）

5G的RRC存在3种状态，既：RRC IDLE（空闲态），RRC CONNECTED（连接态），RRC INACTIVE（不活动态）



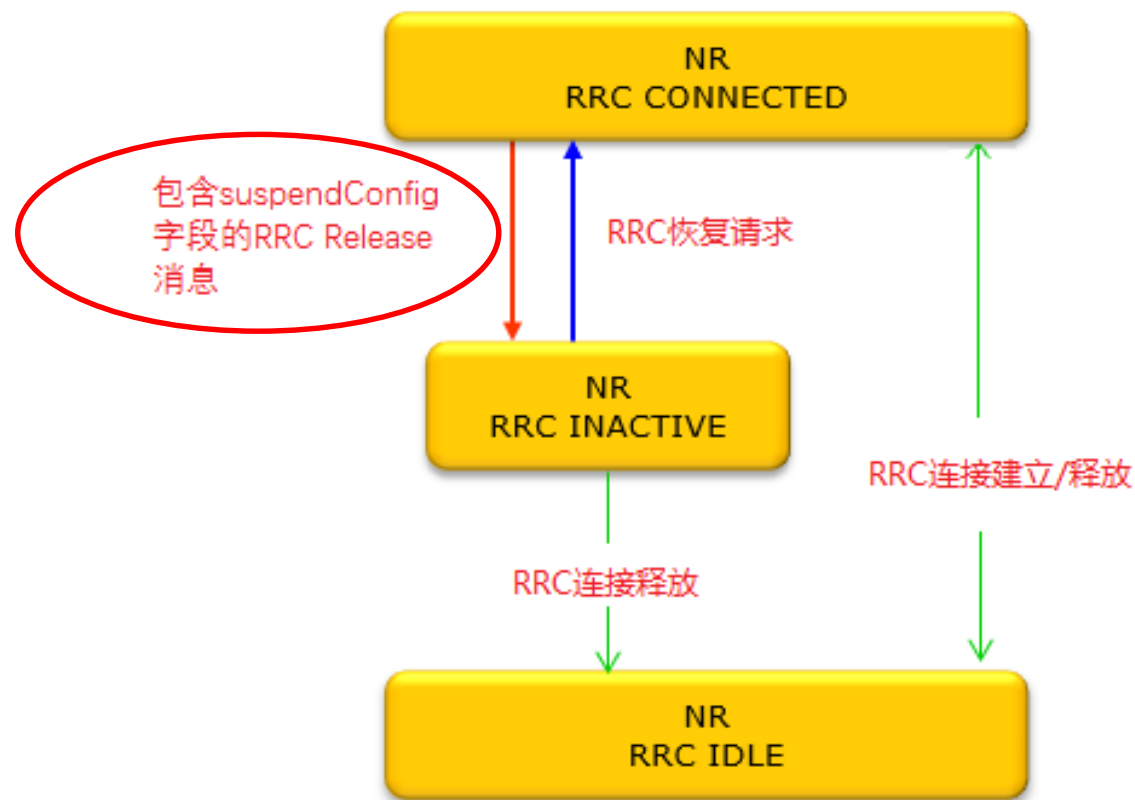


# RRC状态介绍（补充）

包含suspendConfig（状态挂起配置）字段的RRC Release消息

```
RRCRelease ::= SEQUENCE {  
    rrc-TransactionIdentifier      RRC-TransactionIdentifier,  
    criticalExtensions CHOICE {  
        rrcRelease                 RRCRelease-IEs,  
        criticalExtensionsFuture   SEQUENCE {}  
    }  
}  
  
RRCRelease-IEs ::= SEQUENCE {  
    redirectedCarrierInfo          RedirectedCarrierInfo OPTIONAL, -- Need N  
    cellReselectionPriorities     CellReselectionPriorities OPTIONAL, -- Need R  
    suspendConfig                 SuspendConfig OPTIONAL, -- Need R  
    deprioritisationReq           SEQUENCE {  
        deprioritisationType      ENUMERATED {frequency, nr},  
        deprioritisationTimer     ENUMERATED {min5, min10, min15, min30}  
    } OPTIONAL, -- Need N  
    lateNonCriticalExtension OCTET STRING OPTIONAL,  
    nonCriticalExtension RRCRelease-v1540-IEs OPTIONAL  
}
```

```
SuspendConfig ::= SEQUENCE {  
    fullI-RNTI          I-RNTI-Value,  
    shortI-RNTI         ShortI-RNTI-Value,  
    ran-PagingCycle     PagingCycle,  
    ran-NotificationAreaInfo RAN-NotificationAreaInfo OPTIONAL, -- Need M  
    t380                PeriodicRNAU-TimerValue OPTIONAL, -- Need R  
    nextHopChainingCount NextHopChainingCount,  
    ...  
}
```



# RRC状态介绍（补充）

通过RRC resume request可以将UE的RRC状态从不活动状态转移到连接状态

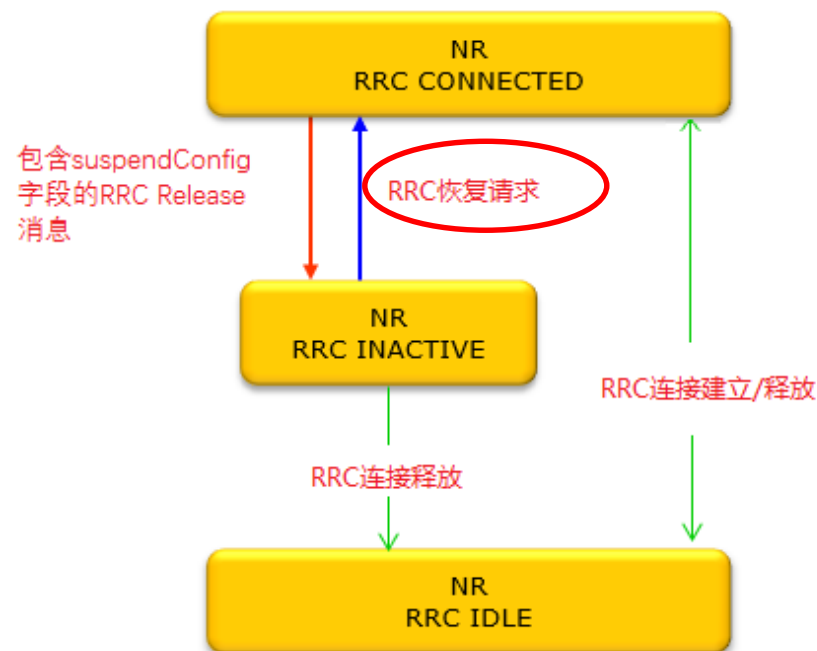
```
RRCResumeRequest1-IEs ::= SEQUENCE {
    resumeIdentity           ShortI-RNTI-Value,
    resumeMAC-I             BIT STRING (SIZE (16)),
    resumeCause             ResumeCause,
    spare                   BIT STRING (SIZE (1))
}
```

```
RRCResumeRequest1 ::= SEQUENCE {
    rrcResumeRequest1      RRCResumeRequest1-IEs
}
```

```
RRCResumeRequest1-IEs ::= SEQUENCE {
    resumeIdentity          I-RNTI-Value,
    resumeMAC-I            BIT STRING (SIZE (16)),
    resumeCause            ResumeCause,
    spare                 BIT STRING (SIZE (1))
}
```

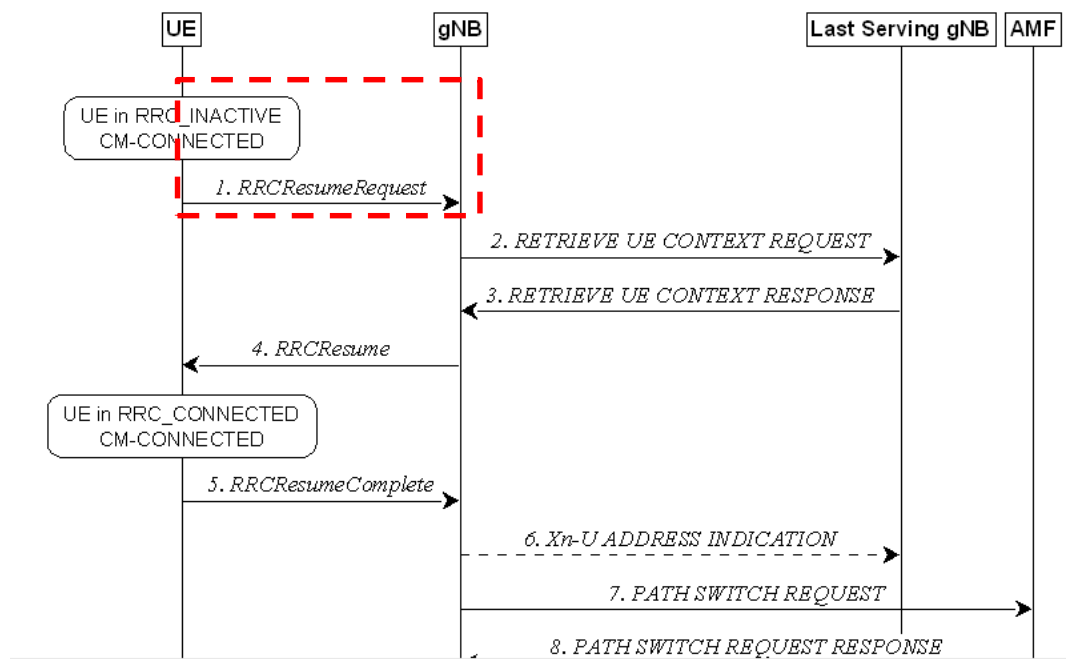
```
ResumeCause ::= ENUMERATED {
    emergency,
    highPriorityAccess,
    mt-Access,
    mo-Signalling,
    mo-Data,
    mo-VoiceCall,
    mo-VideoCall,
    mo-SMS,
    rna-Update,
    mps-PriorityAccess,
    mcs-PriorityAccess,
    spare1, spare2, spare3, spare4, spare5
}
```

emergency	-----	紧急业务
highPriorityAccess	-----	高级接入
mt-Access	-----	被叫接入
mo-Signalling	-----	主叫信令
mo-Data	-----	主叫数据
mo-VoiceCall	-----	主叫音频通话
mo-VideoCall	-----	主叫视频通话
mo-SMS	-----	主叫短信
rna-Update	-----	跟踪区更新
mps-PriorityAccess	-----	mps接入（多媒体优先服务：视频，音频，图片）
mcs-PriorityAccess	-----	mcs接入（关键任务服务）

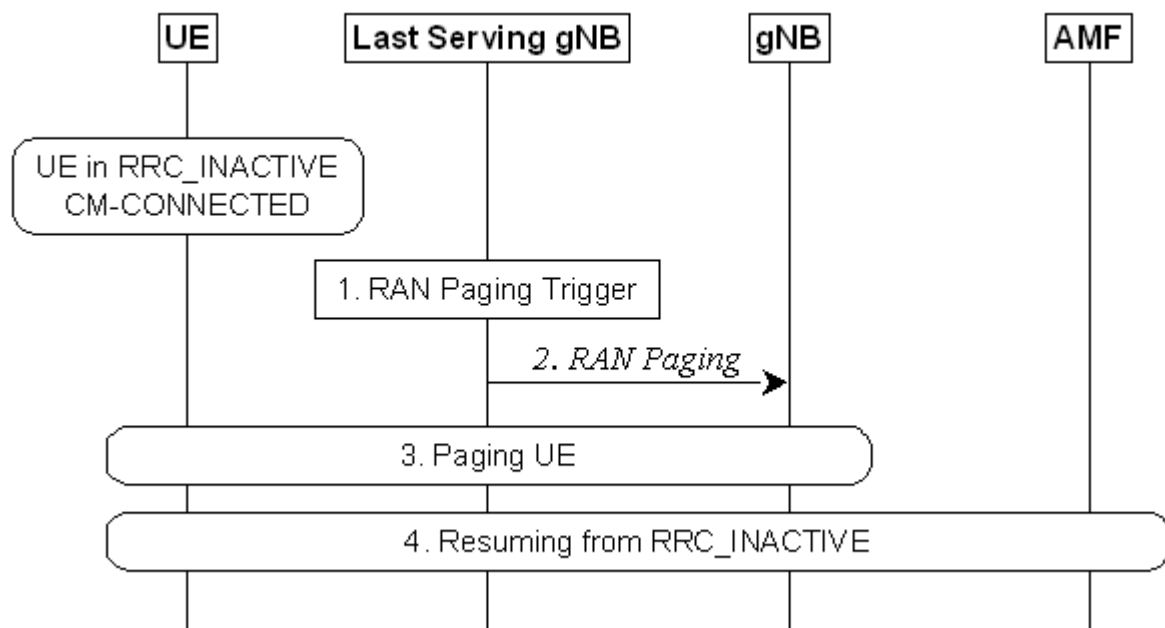


# RRC状态介绍（补充）

RRC resume request恢复请求可以通过UE触发，也可以通过网络触发



UE触发恢复请求



网络侧触发恢复请求



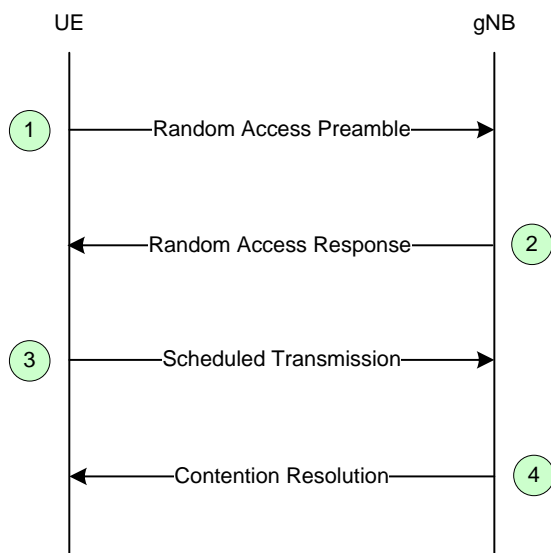
# 随机接入的分类

随机接入分为**基于竞争的随机接入**和**非竞争的随机接入**。

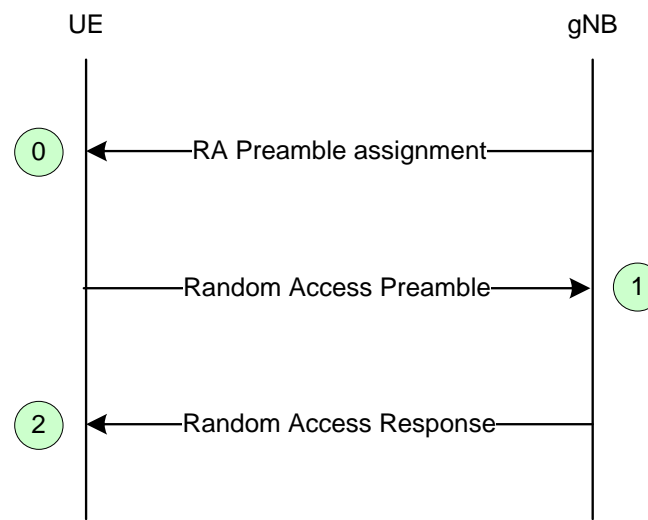
那么，这里面有两个问题，“谁”去竞争？“竞争”的是啥？

答案是：**UE竞争，竞争的是preamble码**！随机接入最重要的第一步就是发**前导码preamble**（总共64个）。

因此，竞争的随机接入，就是UE需要竞争随机接入码，如果发送的随机接入码与别的UE冲突，此时只有一个UE能够在那一时刻接入，另外一个就失败。而非竞争的随机接入，就是UE不需要与别的UE去抢码，由系统指定给UE一个码，然后用这个码来接入，不与别的UE冲突。



基于竞争的随机接入



基于非竞争的随机接入

# 随机接入场景的分类

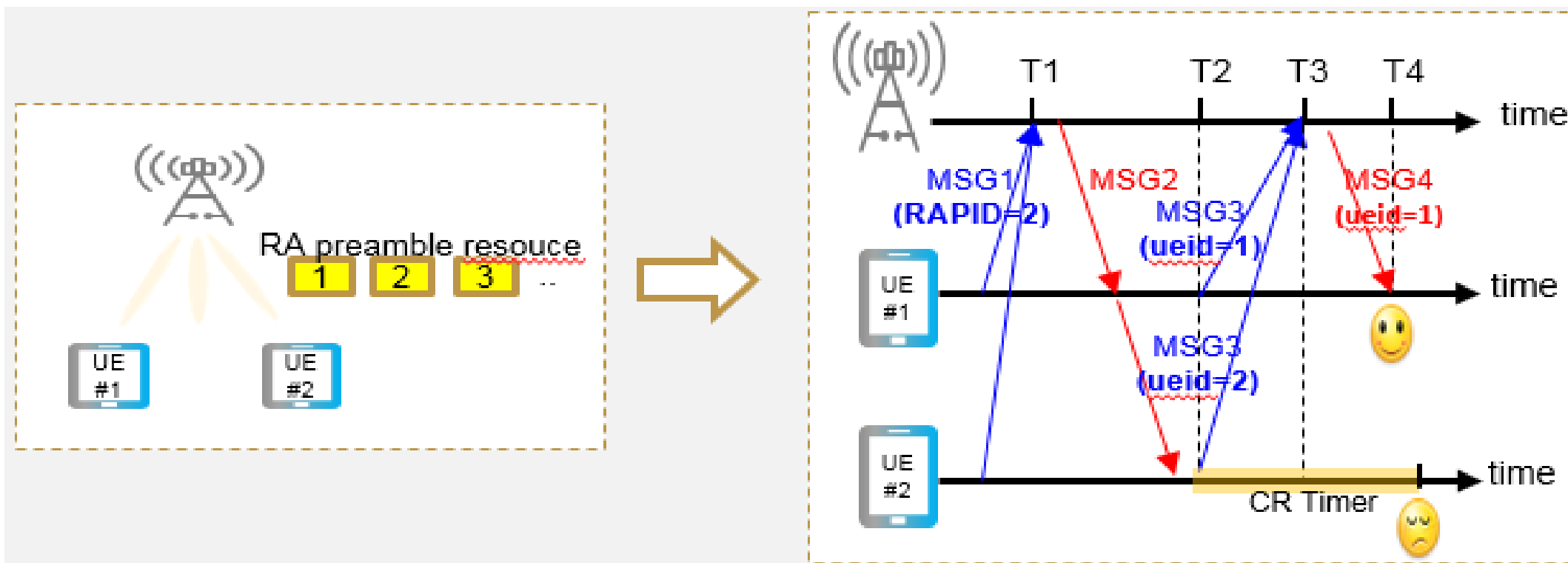
序号	触发场景	场景描述	竞争机制
1	初始RRC连接建立	当UE从空闲态转到连接态时，UE会发起RA。	基于竞争的RA
2	RRC连接重建	当无线链路失步后，UE需要重新建立RRC连接时，UE会发起RA。	基于竞争的RA
3	RRC_INACTIVE态用户状态迁移	当UE从RRC_INACTIVE态转到连接态时，UE会发起RA。	基于竞争的RA
4	切换（包括SA和NSA的DC）	当UE进行切换时，UE会在目标小区发起RA。	基于非竞争的RA，但是在： 1.gNB专用前导用完时或者未获取SSB测量结果时，会使用基于竞争的RA； 2.gNB给UE分配的专用Preamble所在的波束不满足UE最低接入信号门限时，UE会回退到基于竞争的RA
5	上行失步态UE下行数据到达	当gNB检测到UE处于上行失步态且下行数据需要传输时，指示UE发起RA。	基于非竞争的RA，但是在gNB专用前导用完时，会使用基于竞争的RA。
6	上行失步态UE上行数据到达	当UE处于上行失步态且有上行数据需要传输时，UE将发起RA。	基于竞争的RA
7	订阅ODOSI	订阅ODOSI	MSG1方式：基于非竞争的RA MSG3方式：基于竞争的RA
8	波束失败恢复	UE物理层检测到波束失败恢复	基于非竞争的RA，但是在gNB专用前导用完时，会使用基于竞争的RA。
9	Secondary TAG上行同步	载波聚合情况下，在SCell添加时建立时间对齐	基于非竞争的RA
10	SR失败	通过随机接入过程重新获得PUCCH资源；	基于竞争的RA



## 基于竞争的随机接入

# 基于竞争的随机接入

基于竞争的RA过程中，接入的结果具有随机性，并不能保证100%成功；接入前导由UE选择，不同UE产生前导可能冲突，gNB需要通过竞争机制解决不同UE的接入。竞争冲突过程示意图如下：



# 基于竞争的随机接入流程

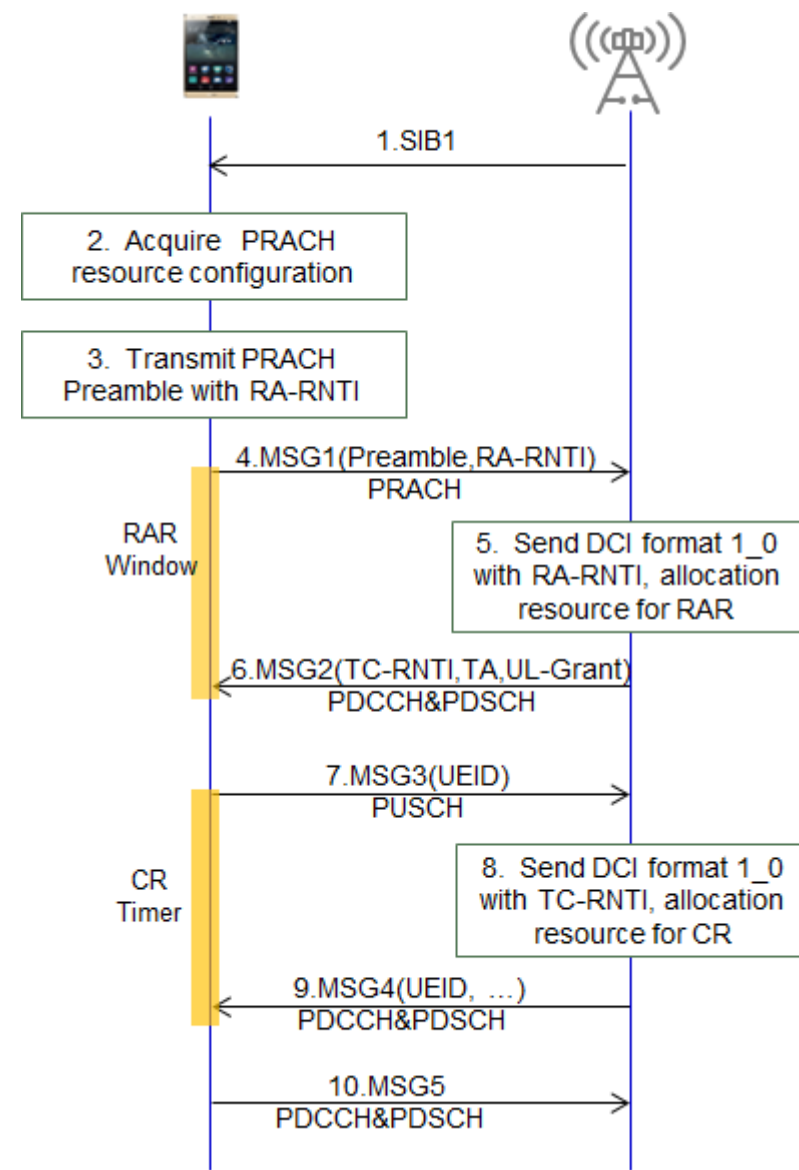
- 1、基站发送SIB1
- 2、UE读取SIB1中的RACH配置，获取PRACH资源，随机选择Preamble码
- 4：通过PRACH信道传输preamble码（MSG1）（计算RA-RNTI）
- 6、通过PDSCH发送随机接入响应（MSG2）携带TA，TC-RNTI，UL-grant
- 7、通过PUSCH信道发送msg3（携带UEID） TC-RNTI加扰
- 8、竞争解决MSG4（携带UEID），TC-RNTI加扰

TC-RNTI :临时小区无线网络临时标识

TA：定时提前指示

UL-grant：上行授权

随机接入的信令，在L1的信令里面能够看到，不在层3信令当中



# SIB1中的PRACH配置

```
RACH-ConfigCommon ::= SEQUENCE {
  rach-ConfigGeneric          RACH-ConfigGeneric,
  totalNumberOfRA-Preambles   INTEGER (1..63)          OPTIONAL,  -- Need S
  ssb-perRACH-OccasionAndCB-PreamblesPerSSB CHOICE {
    oneEighth  ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
    oneFourth  ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
    oneHalf    ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
    one        ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32,n36,n40,n44,n48,n52,n56,n60,n64},
    two        ENUMERATED {n4,n8,n12,n16,n20,n24,n28,n32},
    four       INTEGER (1..16),
    eight      INTEGER (1..8),
    sixteen    INTEGER (1..4)
  } OPTIONAL,  -- Need M

  groupBconfigured SEQUENCE {
    ra-Msg3SizeGroupA ENUMERATED {b56, b144, b208, b256, b282, b480, b640, b800, b1000,
                                   spare7, spare6, spare5, spare4, spare3, spare2, spare1},
    messagePowerOffsetGroupB ENUMERATED {minusinfinity, dB0, dB5, dB8, dB10, dB12,
                                           dB15, dB18},
    numberOfRA-PreamblesGroupA INTEGER (1..64)
  } OPTIONAL,  -- Need R

  ra-ContentionResolutionTimer ENUMERATED {sf8, sf16, sf24, sf32, sf40, sf48, sf56, sf64},
  rsrp-ThresholdSSB
  rsrp-ThresholdSSB-SUL
  prach-RootSequenceIndex
    1839
    1139
  },
  msg1-subcarrierspacing SU
  restrictedSetConfig EN
  msg3-transformPrecoding EN
  ...
}
```

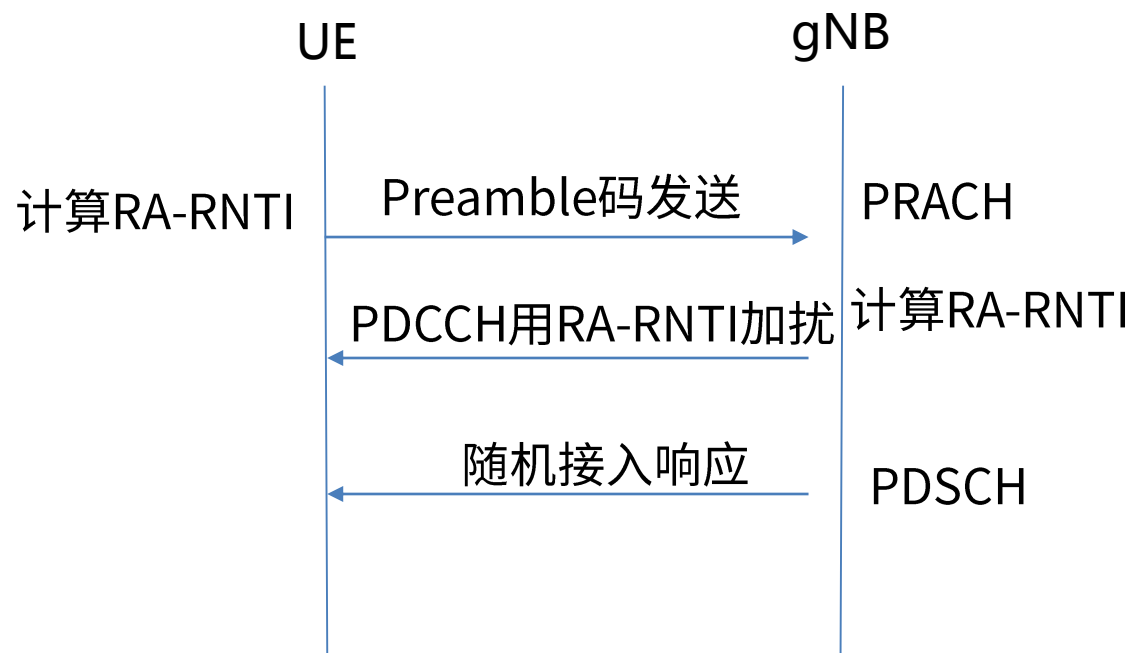
```
RACH-ConfigGeneric ::= SEQUENCE {
  prach-ConfigurationIndex INTEGER (0..255),
  msg1-FDM                  ENUMERATED {one, two, four, eight},
  msg1-FrequencyStart      INTEGER (0..maxNrofPhysicalResourceBlocks-1),
  zeroCorrelationZoneConfig INTEGER(0..15),
  preambleReceivedTargetPower INTEGER (-200..-74),
  preambleTransMax          ENUMERATED {n3,n4,n5,n6,n7,n8,n10,n20,n50,n100,n200},
  powerRampingStep          ENUMERATED {dB0, dB2, dB4, dB6},
  ra-ResponseWindow         ENUMERATED {s11, s12, s14, s18, s110, s120, s140, s180}
}
```



# MSG1 : 发送preamble并计算RA-RNTI

UE发送preamble码，并且计算出RA-RNTI。某个UE发送的preamble时频位置是确定的，gNodeB在解码preamble时，也获得了该preamble的时频位置，进而知道了RAR中需要使用的RA-RNTI。随后，使用这个RA-RNTI对PDCCH进行加扰。接下来UE使用这个RA-RNTI对PDCCH进行解扰，如果解扰成功，则说明接入被响应，在依据PDCCH上的指示去PDSCH上读取RA Response消息。

这里面，不能完全避免不同UE的冲突，同一个小区，有两个或者多个终端在相同的时频资源（RA-RNTI）上选择了相同的preamble发送，概率虽小，但也是可能的。对于这种冲突还需要后面的Msg3和Msg4去解决



流程

RNTI对PDCCH的加扰和解扰，  
我在PDCCH物理信道中讲过，可以去复习一下。

# 计算RA-RNTI算法

RA-RNTI的计算在协议38.321中给出

$$\text{RA-RNTI} = 1 + s\_id + 14 \times t\_id + 14 \times 80 \times f\_id + 14 \times 80 \times 8 \times ul\_carrier\_id$$

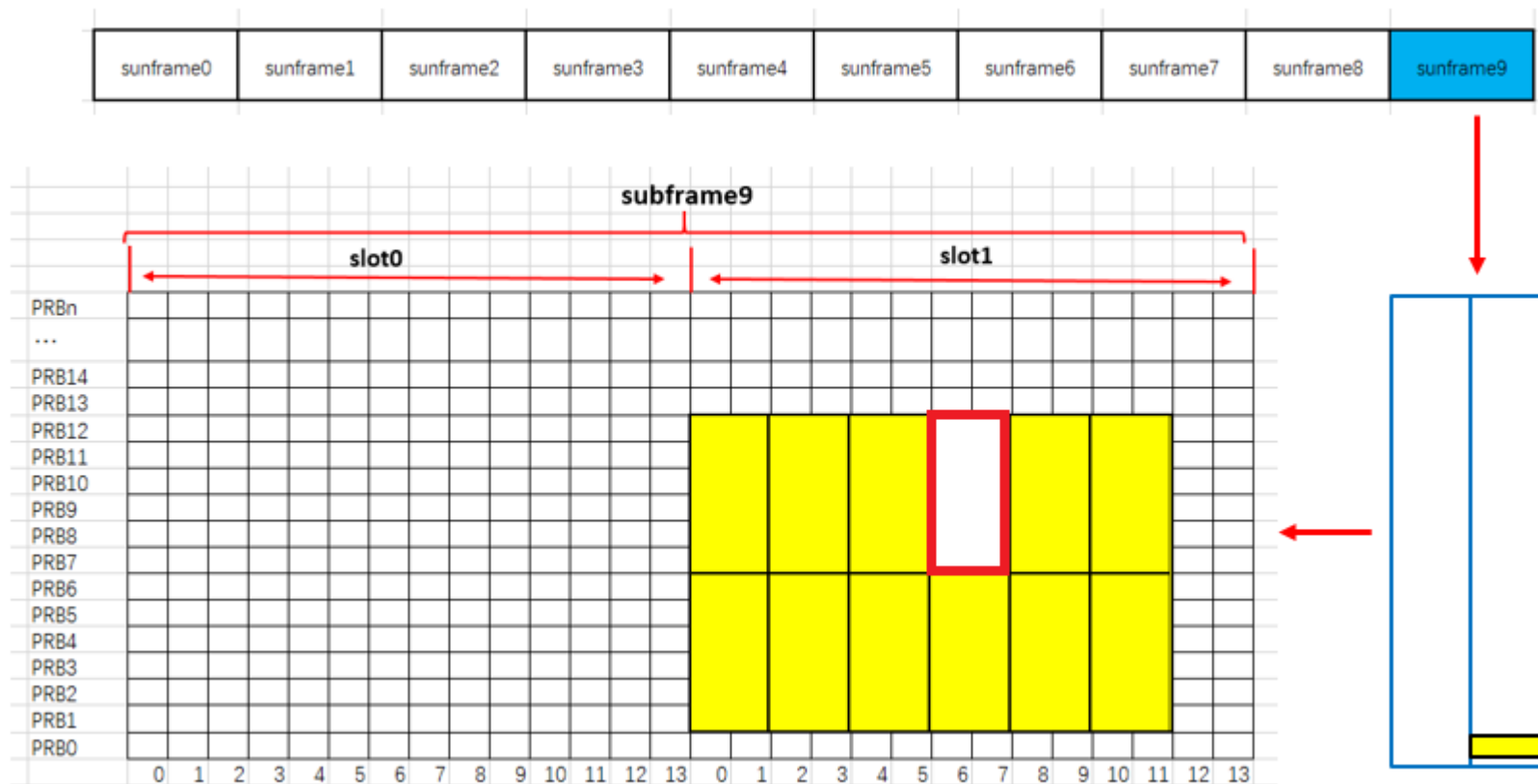
s\_id：PRACH的时频资源第一个符号

t\_id：PRACH的时频资源的第一个时隙

f\_id：PRACH时频资源的频域编号

ul\_carrier\_id 上行载波(取值：0 for 主载波, 1 for SUL载波).

# 计算RA-RNTI举例



资源部分，我们在PRACH物理信道中讲过，忘记的同学可以回去看！

$$\text{RA-RNTI} = 1 + s\_id + 14 \times t\_id + 14 \times 80 \times f\_id + 14 \times 80 \times 8 \times ul\_carrier\_id$$

$$\text{RA-RNTI} = 1 + 0 + 14 \times 19 + 14 \times 80 \times 1 + 0 = 1 + 266 + 1120 = 1387$$

# preamble码的分组

我们知道preamble码有64个，在随机接入的时候，可以随机选择一个，进行发送。然而，preamble码还可以进行分组，比如分成groupA和groupB。

如果满足下列条件，则选择group B，否则选择group A：

1. 该RA过程还没有传输过Msg3；
2. 高层参数配置了group B；
3. Msg3的大小（加上MAC头和MAC控制单元等）大于ra-Msg3SizeGroupA
4. 路损小于下式计算结果：

$$PCMAX - preambleReceivedTargetPower - msg3-DeltaPreamble - messagePowerOffsetGroupB$$

其中：

- PCMAX是可以配置的UE最大发送功率
- preambleReceivedTargetPower是基站目标接收功率
- msg3-DeltaPreamble是Msg3和preamble发送时的功率偏移
- messagePowerOffsetGroupB是选择preamble的功率门限。
- 如果已传输过Msg3，则选择第一次传输Msg3相同的group。

## MSG2 : Random Access Response随机接入响应

当UE通过RA-RNTI解扰了PDCCH并且通过PDSCH解码了RAR，这时候，并不能说明UE成功接收了随机接入响应，需要该RAR中的preamble index（序号）与UE发送的preamble index相同时，才认为成功接收了RAR，此时UE就可以停止监听RAR了。

RAR里面的内容，主要看MAC层的数据

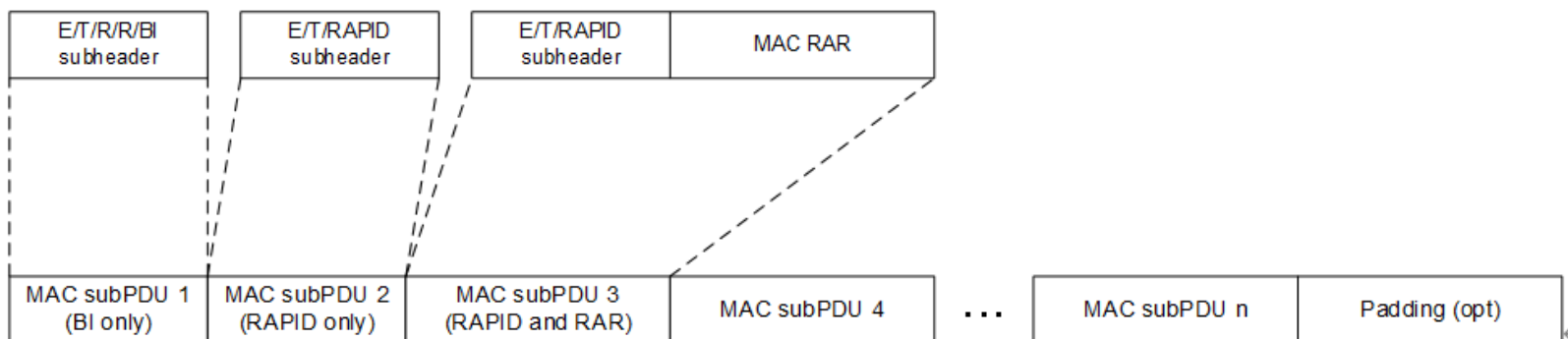


Figure 6.1.5-3: Example of MAC PDU consisting of MAC RARs

RAR MAC PDU结构当中，具有RAPID的MAC子头（相当于快递包裹上面，打了一个标签，MAC PDU是包裹，子头相当于标签）

RAPID为Random Access Preamble Identifier的简称，为gNB在检测preamble时所得到的preamble index。

# MSG2 : Random Access Response随机接入响应

RAR MAC PDU 里面的主体内容，包含有3个部分：

- TA (Timing Advance) 定时提前调整量----基站根据UE发送的MSG1估计出来的
- UL-grant: 上行资源授权（配置PUSCH，发送MSG3）
- TC-RNTI: 临时小区无线网络临时标识（为了MSG3加扰）

Table 8.2-1: Random Access Response Grant Content field size

RAR grant field	Number of bits
Frequency hopping flag	1
PUSCH frequency resource allocation	14
PUSCH time resource allocation	4
MCS	4
TPC command for PUSCH	3
CSI request	1

PUSCH资源配置

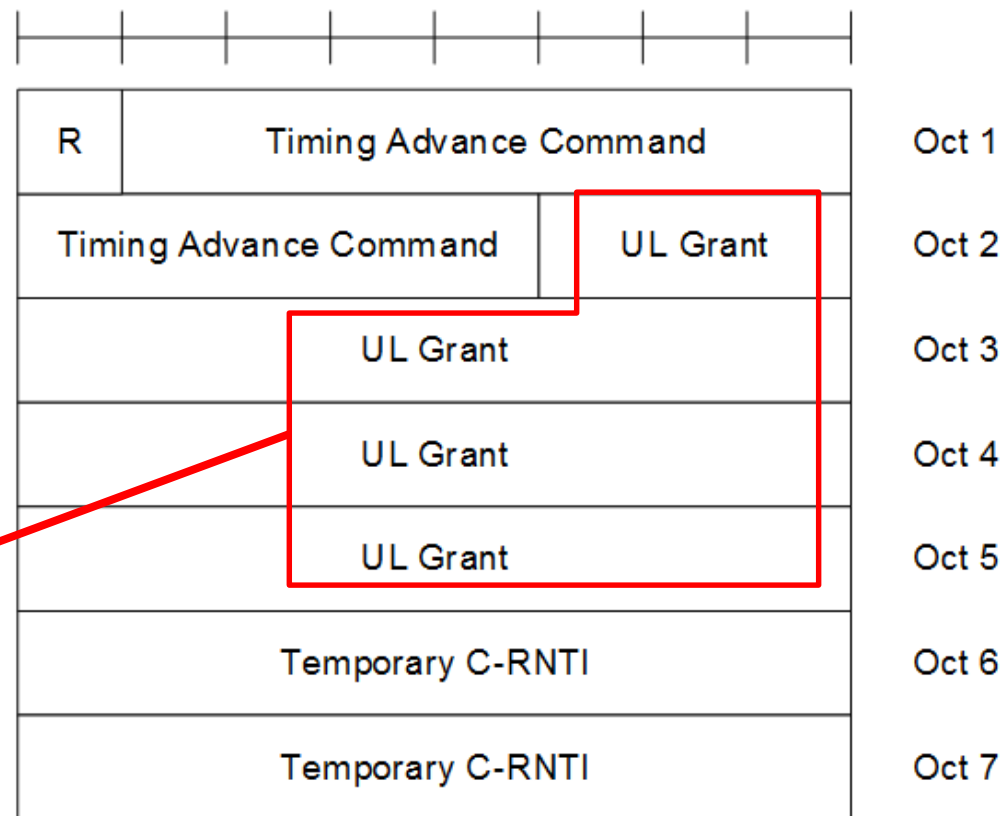


Figure 6.2.3-1: MAC RAR



## MSG3 : Scheduled Transmission调度传输

竞争的随机接入第三步，通常称为MSG3传输， MSG3传输的内容取决于随机接入的场景  
(MSG3使用TC-RNTI加扰)

Msg3中需要包含一个重要信息：每个UE唯一的标志（UEID），该标志将用于步骤四的冲突解决。

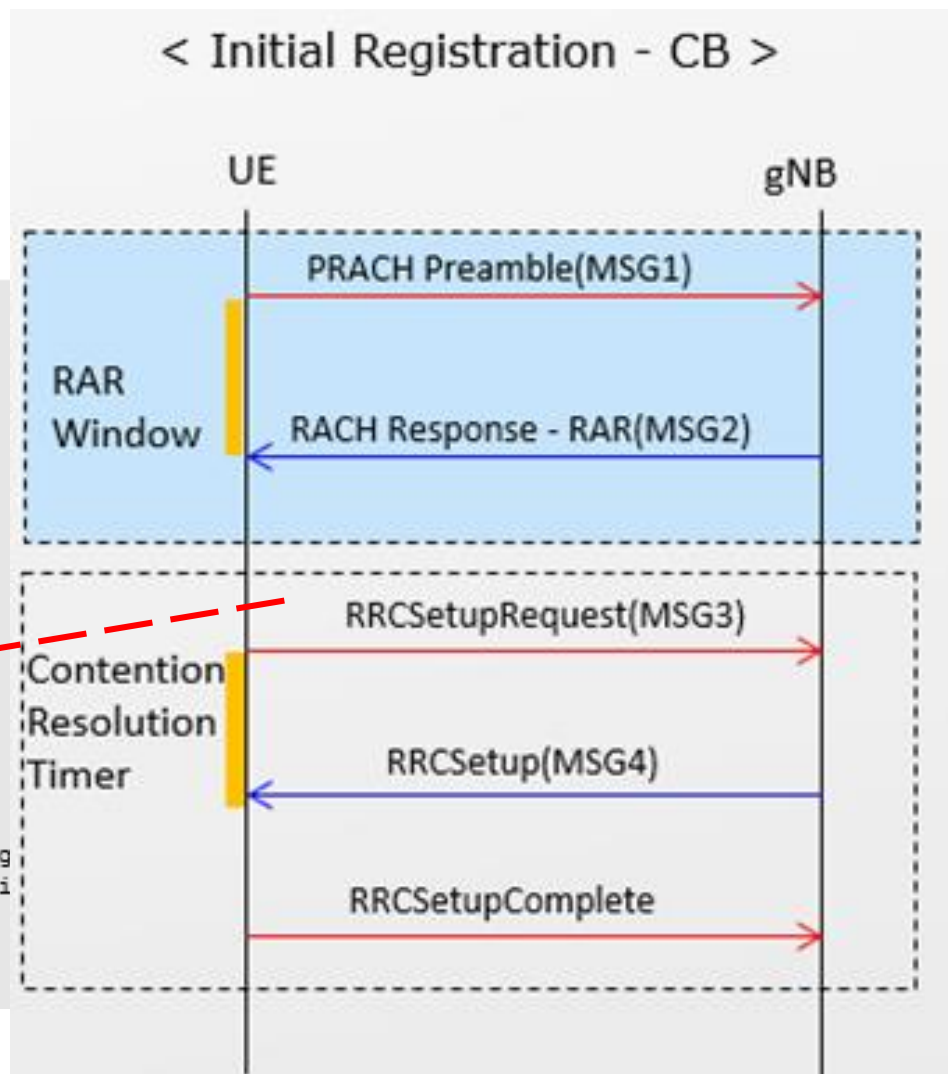
- 如果UE处于RRC IDLE状态，UE标识是ng-5G-S-TMSI-part1或随机数（39位）
- 如果UE处于RRC INACTIVE状态，UE标识是resume Identity（24位或40位）
- 当UE处于RRC\_CONNECTED态但上行不同步时，UE有自己的C-RNTI，然后使用C-RNTI尝试对PDCCH进行解扰，如果解扰成功，UE认为竞争解决，随机接入成功。

# MSG3 : Scheduled Transmission

以RRC连接建立场景为例，RRC连接建立请求对应的就是MSG3在请求当中，可以使用ng-5G-S-TMSI-Part1或者randomValue作为UE的标识

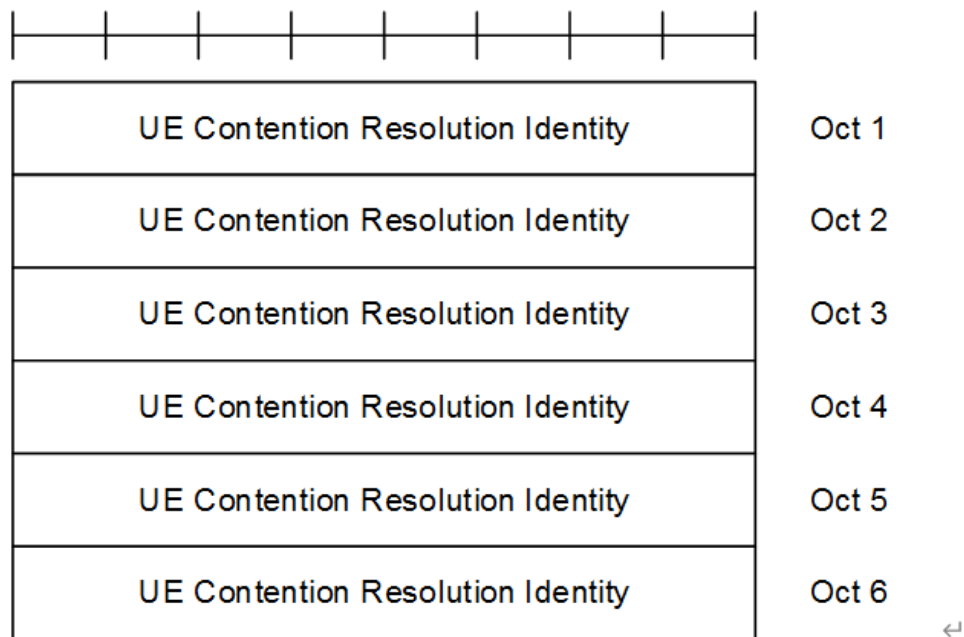
*RRCSetupRequest message*

```
-- ASN1START
-- TAG-RRCSETUPREQUEST-START
RRCSetupRequest ::=
    SEQUENCE {
        rrcSetupRequest
    }
RRCSetupRequest-IEs ::=
    SEQUENCE {
        ue-Identity
        establishmentCause
        spare
    }
InitialUE-Identity ::=
    CHOICE {
        ng-5G-S-TMSI-Part1 BIT STRING (SIZE (39)),
        randomValue BIT STRING (SIZE (39))
    }
EstablishmentCause ::=
    ENUMERATED {
        emergency, highPriorityAccess, mt-Access, mo-Signalling
        mo-Data, mo-VoiceCall, mo-VideoCall, mo-SMS, mps-Priori
        spare6, spare5, spare4, spare3, spare2, spare1
    }
-- TAG-RRCSETUPREQUEST-STOP
-- ASN1STOP
```



# MSG4 : Contention Resolution 竞争解决

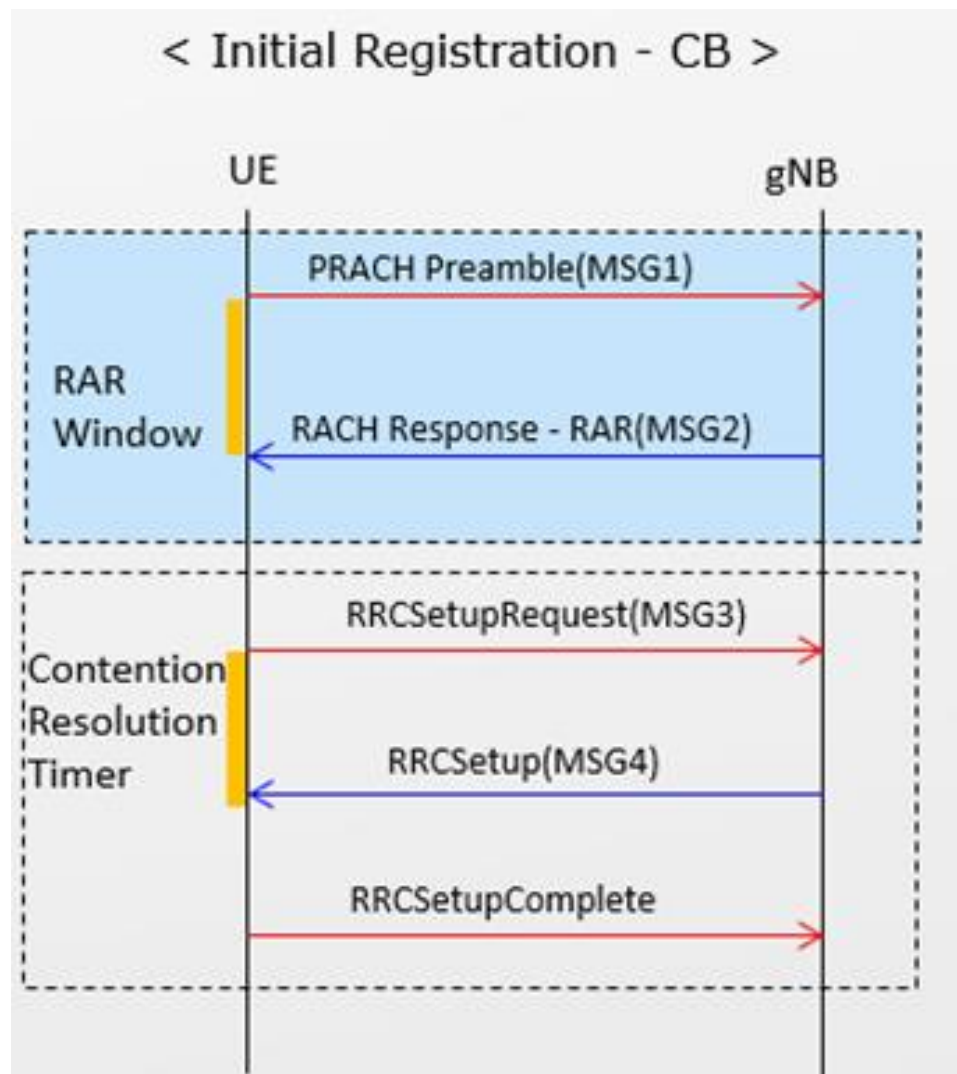
基站在MSG4通过UE Contention Resolution Identity MAC CE（控制单元）（48位）携带UE标识（如果长度超过48位，只取前48位），如果MSG4和MSG3携带的UE标识相同，UE认为竞争解决，随机接入成功。而其它没有在冲突解决中胜出的UE将重新发起随机接入。



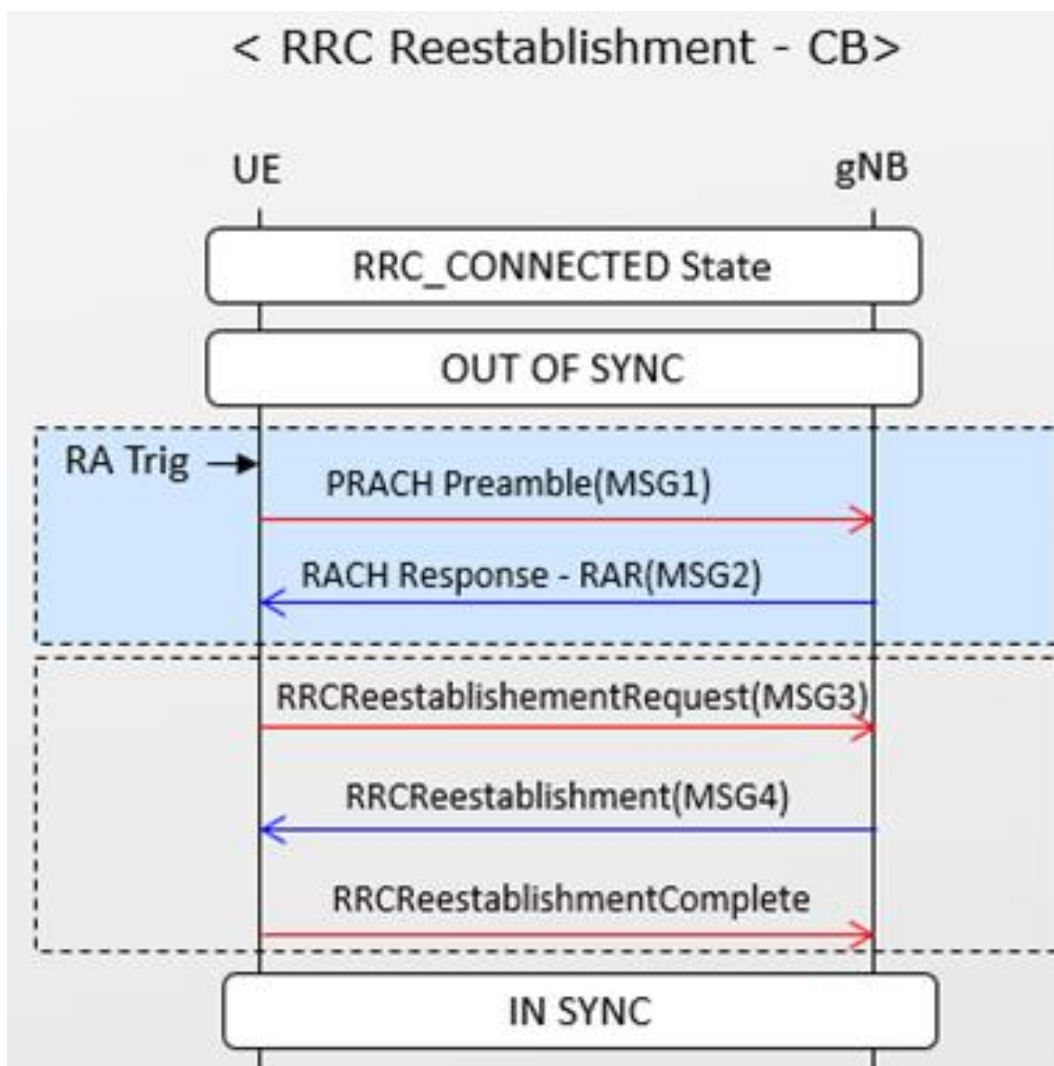
《3GPP TS 38.321》

Figure 6.1.3.3-1: UE Contention Resolution Identity MAC CE

# 基于竞争的随机接入一场景



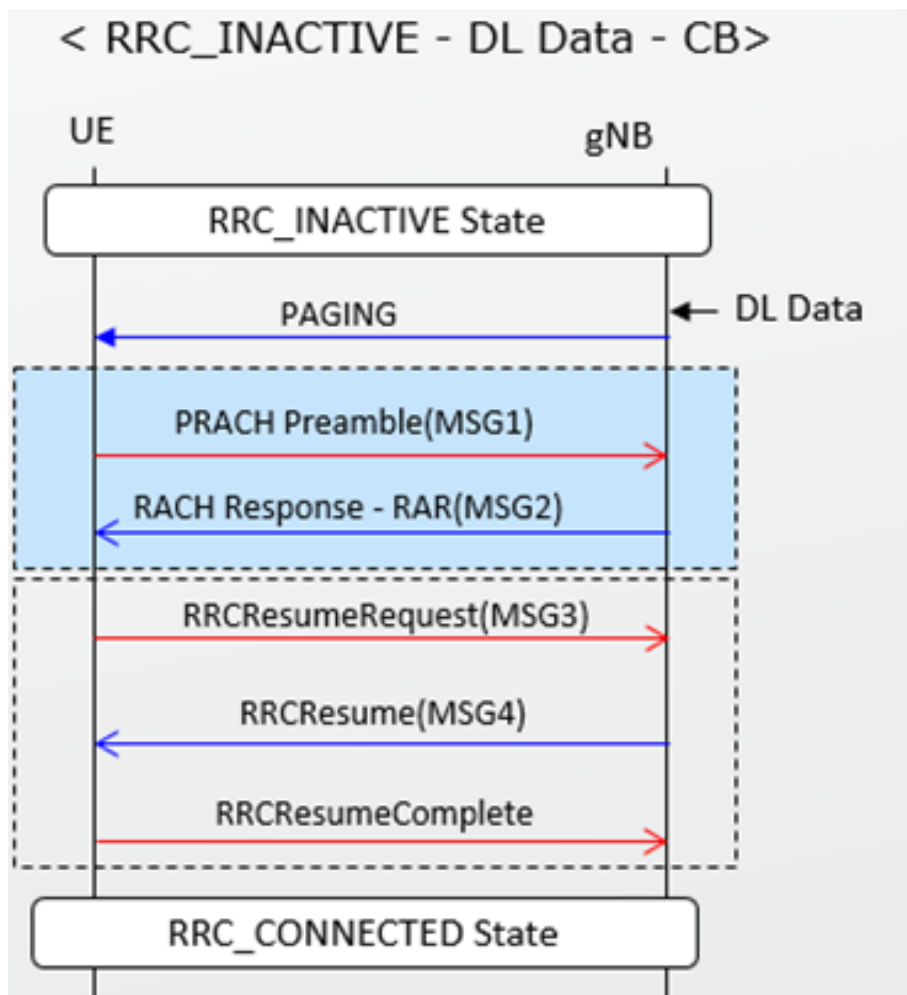
初始接入场景



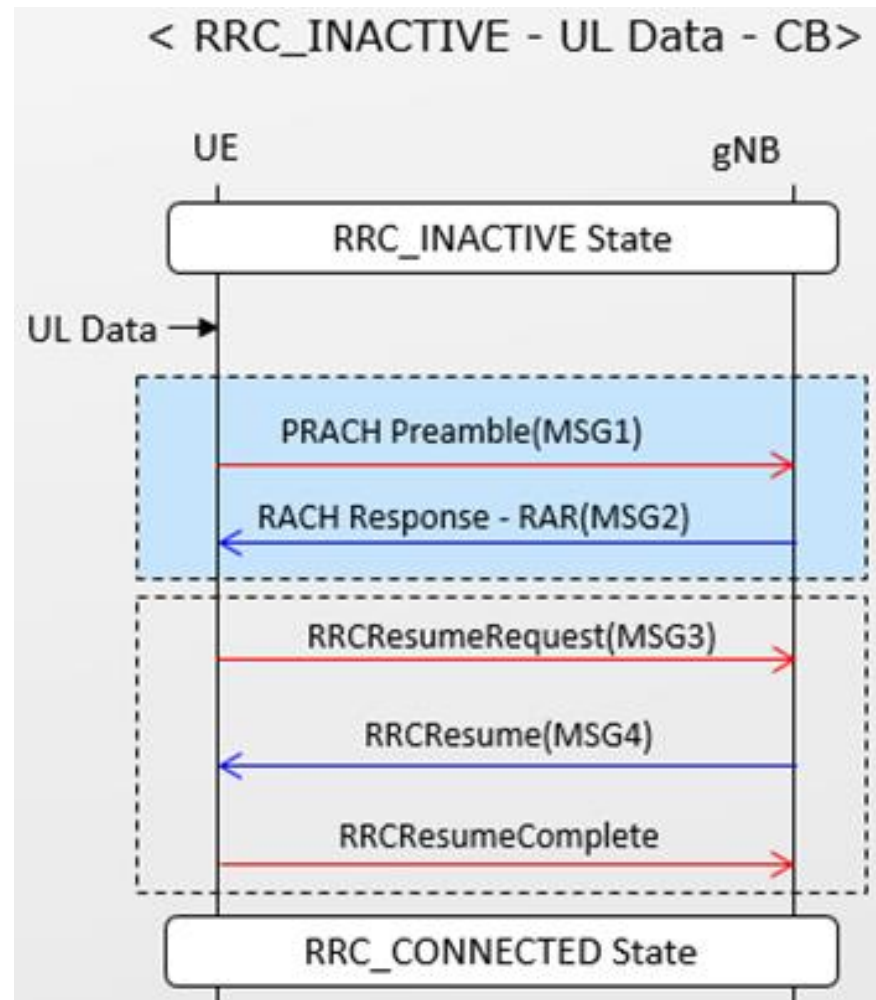
RRC连接重建

# 基于竞争的随机接入一场景

INACTIVE状态转移CONNECTED状态

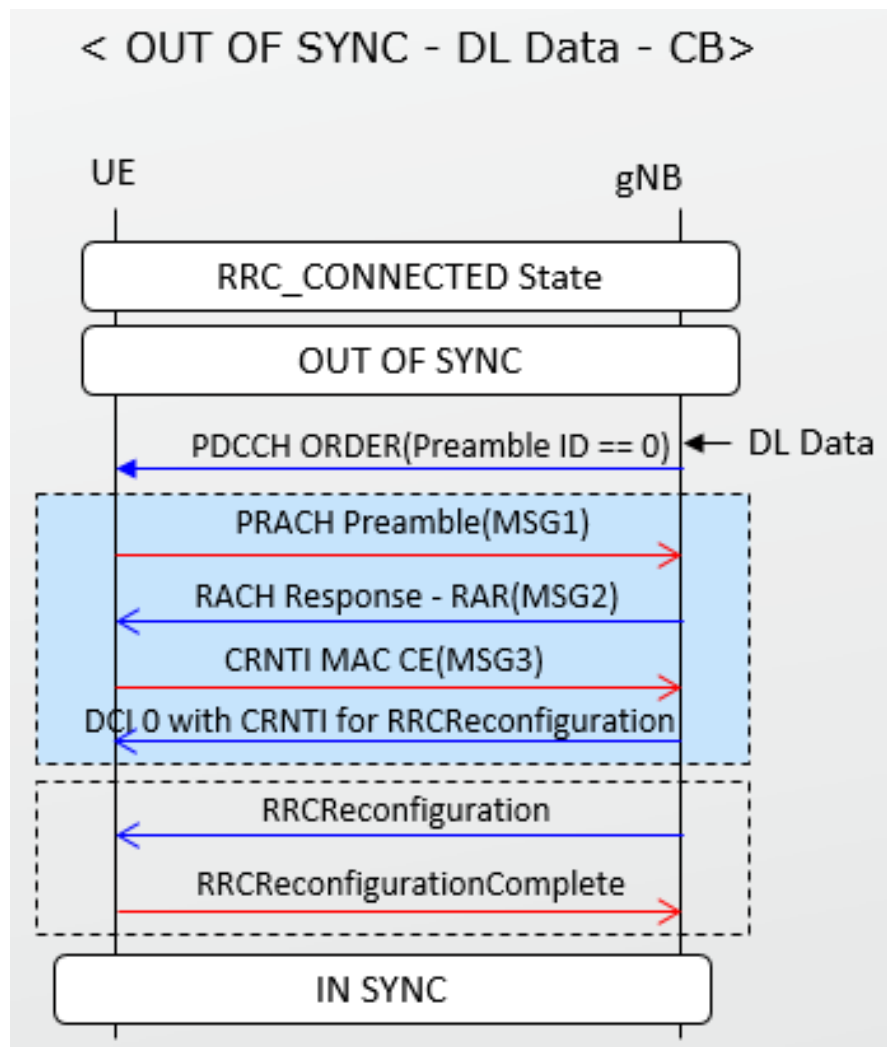


不活动状态--下行数据到达

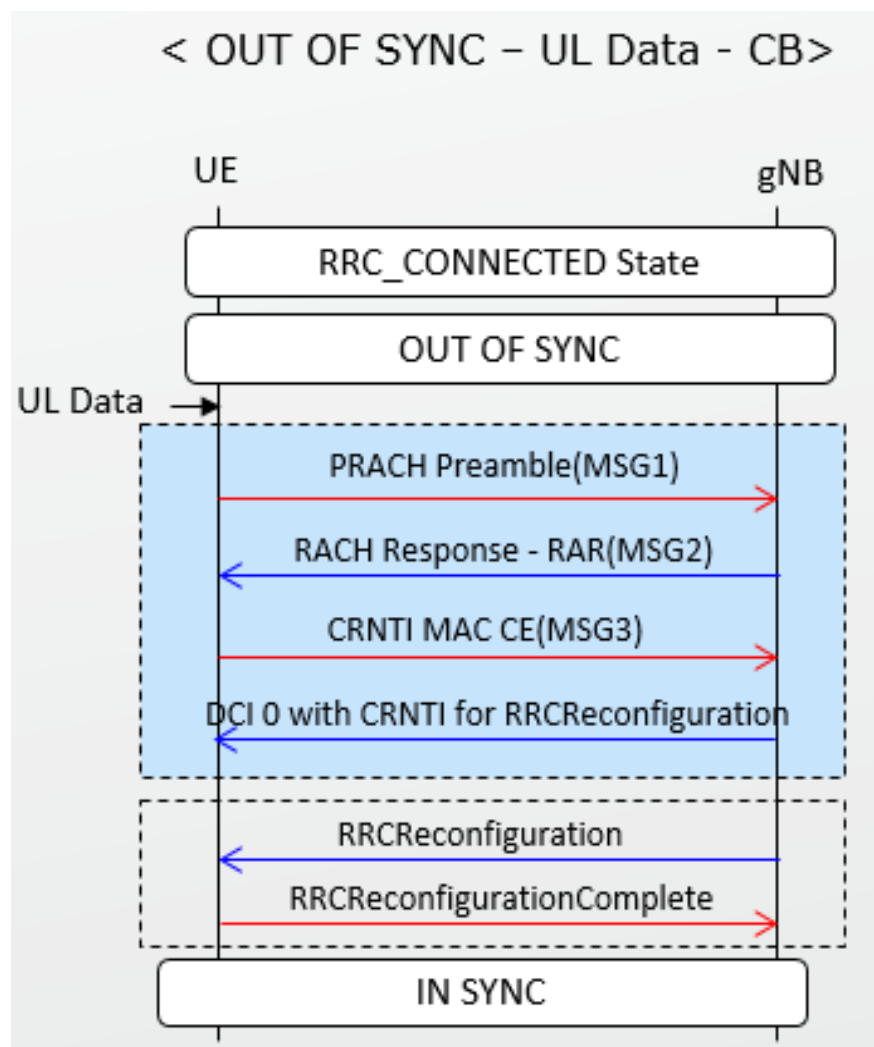


不活动状态--上行数据到达

# 基于竞争的随机接入一场景



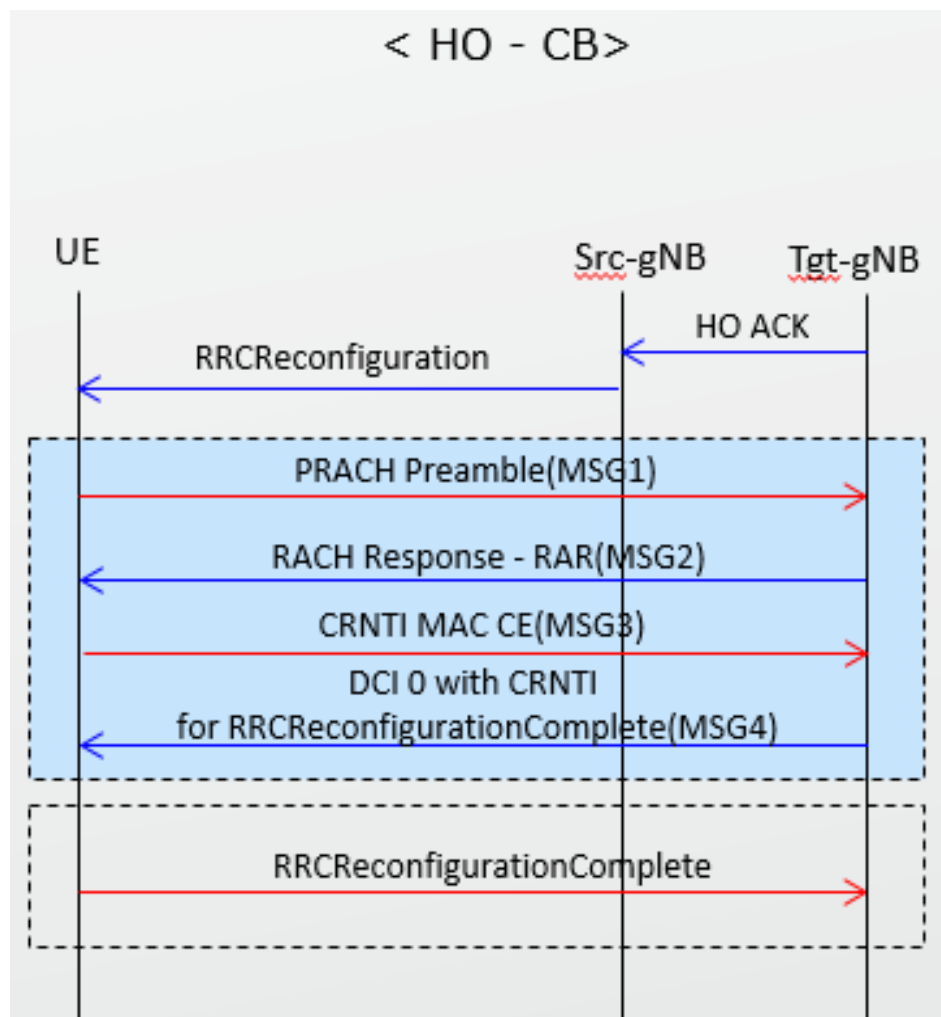
上行失步状态--下行数据到达



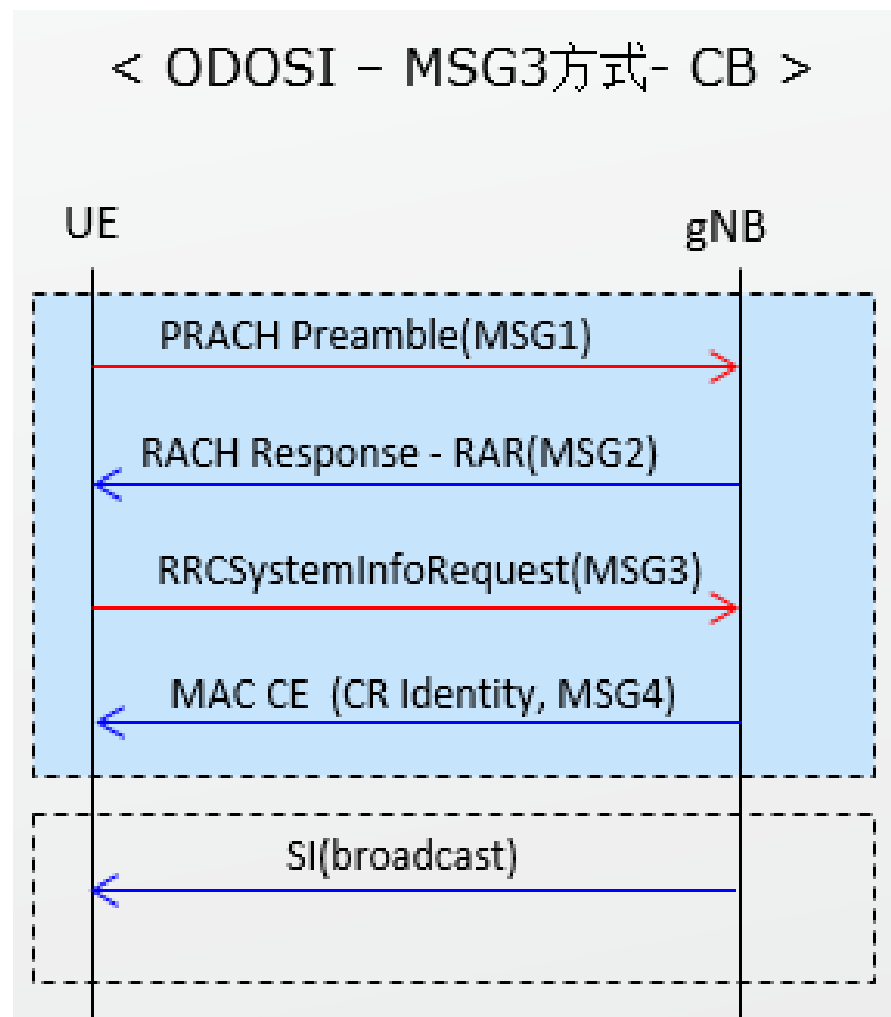
上行失步状态--上行数据到达



# 基于竞争的随机接入一场景



切换



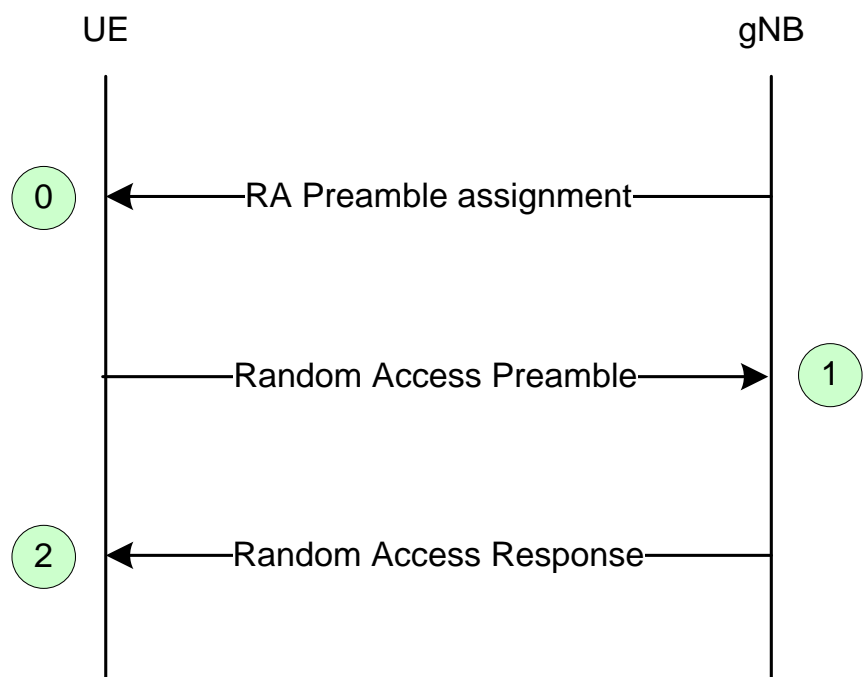
MSG3方式获取其他系统消息



## 基于非竞争的随机接入

# 基于非竞争的随机接入

在基于非竞争的RA过程中，gNB为UE分配专用的RACH（Random Access Channel）资源进行接入，但当专用的RACH资源不足时，gNB会指示UE发起基于竞争的RA。



第一步：RACH Preamble (PRACH) Assignment随机接入指派  
通过PDCCH order或者RRC Reconfiguration指派preamble

第二步：发送preamble码

第三步： Random Access Response 随机接入响应

\*第二步和第三步和竞争的随机接入是一样的，这里主要说下第一步

# PDCCH order

先看一下PDCCH order方式指派Preamble

我们知道PDCCH是传输DCI的，这里面使用DCI1-0来配置preamble。我们之前学习PDCCH时候，学习过，DCI1-0正常情况下，是配置下行的PDSCH物理信道资源的。而DCI1-0，当使用C-RNTI加扰时，并且当Frequency domain assignment field字段全为比特1时，并不是来调度PDSCH，而是来指示UE需要发送随机接入，分配的preamble index，发送载波指示等信息在DCI中指示。

< DCI format 1_0 with CRC scrambled by C-RNTI >		
Field (Item)	Bits	Reference
Identifier for DCI formats	1	Always set to 1, meaning this is for DL
Frequency domain resource assignment	Variable	Variable with DL BWP $N_{RB}$ $\left\lceil \log_2(N_{RB}^{DL,BWP}(N_{RB}^{DL,BWP} + 1)/2) \right\rceil$
Time domain resource assignment	4	Carries the row index of the items in <a href="#">pdsch_allocationList in RRC</a>
VRB-to-PRB mapping	1	According to 38.212 Table 7.3.1.1.2-33 0 : Non-Interleaved 1 : Interleaved
Modulation and coding scheme	5	<a href="#">38.214 - Table 5.1.3.1-1: MCS index table 1 for PDSCH</a> <a href="#">38.214 - Table 5.1.3.1-2: MCS index table 2 for PDSCH</a>
New data indicator	1	
Redundancy version	2	
HARQ process number	4	
Downlink assignment index	2	
TPC command for scheduled PUCCH	2	
PUCCH resource indicator	3	See <a href="#">here</a> , <a href="#">here</a>
PDSCH-to-HARQ_feedback timing indicator	3	maps to $k1=\{1,2,3,4,5,6,7,8\}$

左边这是正常格式DCI1-0的内容，但是一旦Frequency domain assignment field字段全为比特1时这个DCI1-0中的内容，就会变成跟preamble有关。

# PDCCH order

3GPP TS 38.212中定义了用于preamble指配的DCI1-0里面的内容：

If the CRC of the DCI format 1\_0 is scrambled by C-RNTI and the "Frequency domain resource assignment" field are of all ones, the DCI format 1\_0 is for random access procedure initiated by a PDCCH order, with all remaining fields set as follows:↵

- **Random Access Preamble index** – 6 bits according to *ra-PreambleIndex* in Subclause 5.1.2 of [8, TS38.321]↵
- **UL/SUL indicator** – 1 bit. If the value of the "Random Access Preamble index" is not all zeros and if the UE is configured with *supplementaryUplink* in *ServingCellConfig* in the cell, this field indicates which UL carrier in the cell to transmit the PRACH according to Table 7.3.1.1.1-1; otherwise, this field is reserved↵
- **SS/PBCH index** – 6 bits. If the value of the "Random Access Preamble index" is not all zeros, this field indicates the SS/PBCH that shall be used to determine the RACH occasion for the PRACH transmission; otherwise, this field is reserved. ↵
- **PRACH Mask index** – 4 bits. If the value of the "Random Access Preamble index" is not all zeros, this field indicates the RACH occasion associated with the SS/PBCH indicated by "SS/PBCH index" for the PRACH transmission, according to Subclause 5.1.1 of [8, TS38.321]; otherwise, this field is reserved↵
- **Reserved bits** – 10 bits↵

指配的preamble

# RRC Reconfiguration 指配 preamble

RACH-config Dedicated (RACH专用配置)

CFRA (Contention Free Random Access) : 非竞争随机接入

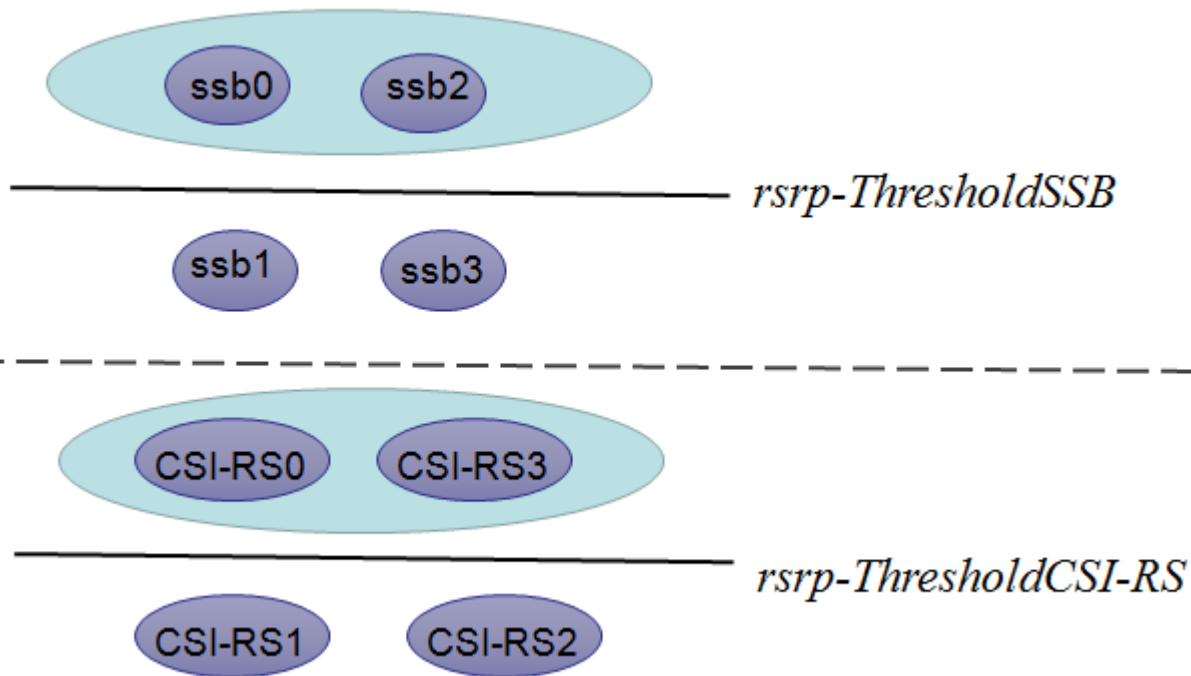
```
RACH-ConfigDedicated ::= SEQUENCE {  
    cfra CFRA OPTIONAL, -- Need S  
    ra-Prioritization RA-Prioritization OPTIONAL, -- Need N  
    ...  
}  
  
CFRA ::= SEQUENCE {  
    occasions SEQUENCE {  
        rach-ConfigGeneric RACH-ConfigGeneric,  
        ssb-perRACH-Occasion ENUMERATED {oneEighth, oneFourth, oneHalf, one, two, four, eight, sixteen} OPTIONAL, -- Cond SSB-CFRA  
    } OPTIONAL, -- Need S  
    resources CHOICE {  
        ssb SEQUENCE {  
            ssb-ResourceList SEQUENCE (SIZE(1..maxRA-SSB-Resources)) OF CFRA-SSB-Resource,  
            ra-ssb-OccasionMaskIndex INTEGER (0..15)  
        },  
        csirs SEQUENCE {  
            csirs-ResourceList SEQUENCE (SIZE(1..maxRA-CSIRS-Resources)) OF CFRA-CSIRS-Resource,  
            rsrp-ThresholdCSI-RS RSRP-Range  
        }  
    },  
    ...  
    [[  
        totalNumberOfRA-Preambles-v1530 INTEGER (1..63) OPTIONAL -- Cond Occasions  
    ]]  
}  
  
CFRA-SSB-Resource ::= SEQUENCE {  
    ssb SSB-Index,  
    ra-PreambleIndex INTEGER (0..63),  
    ...  
}  
  
CFRA-CSIRS-Resource ::= SEQUENCE {  
    csi-RS CSI-RS-Index,  
    ra-OccasionList SEQUENCE (SIZE(1..maxRA-OccasionsPerCSIRS)) OF INTEGER (0..maxRA-Occasions-1),  
    ra-PreambleIndex INTEGER (0..63),  
    ...  
}
```



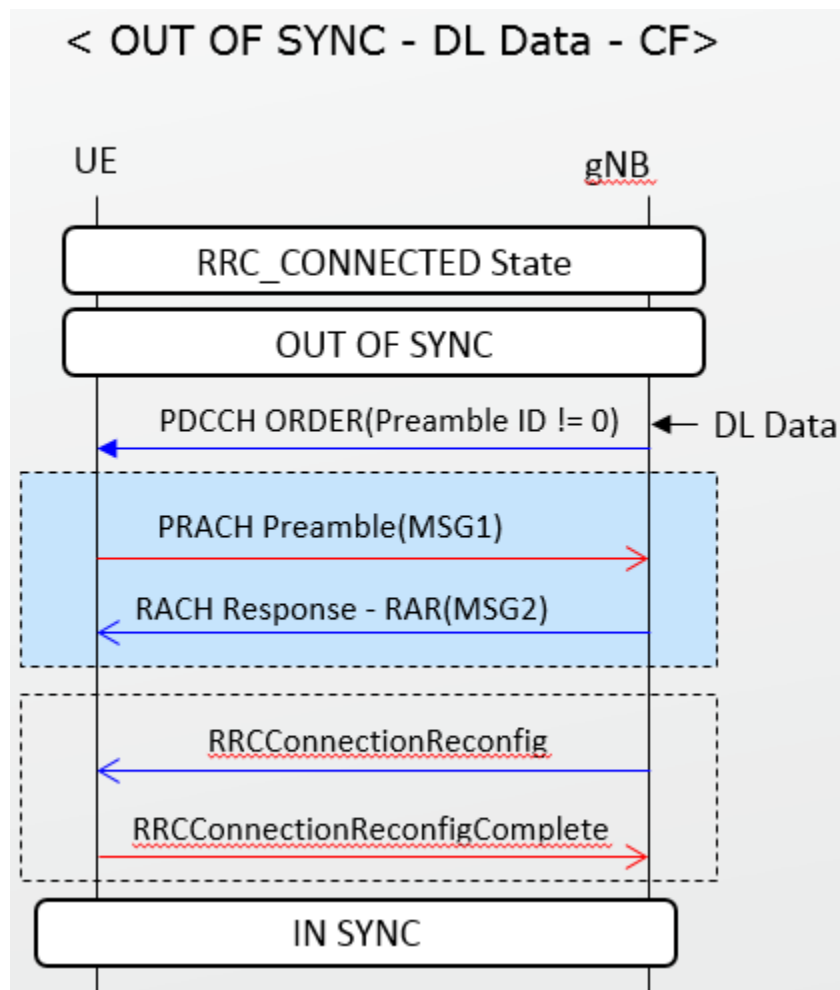
# RRC Reconfiguration指配preamble

SSB/CSI-RS与PreambleIndex有配对关系，因为基站也不清楚对于UE来说哪个是最强的Beam波束，选取SSB/CSI-RS测量值大于门限的SSB/CSI-RS，也即确定了preamble index

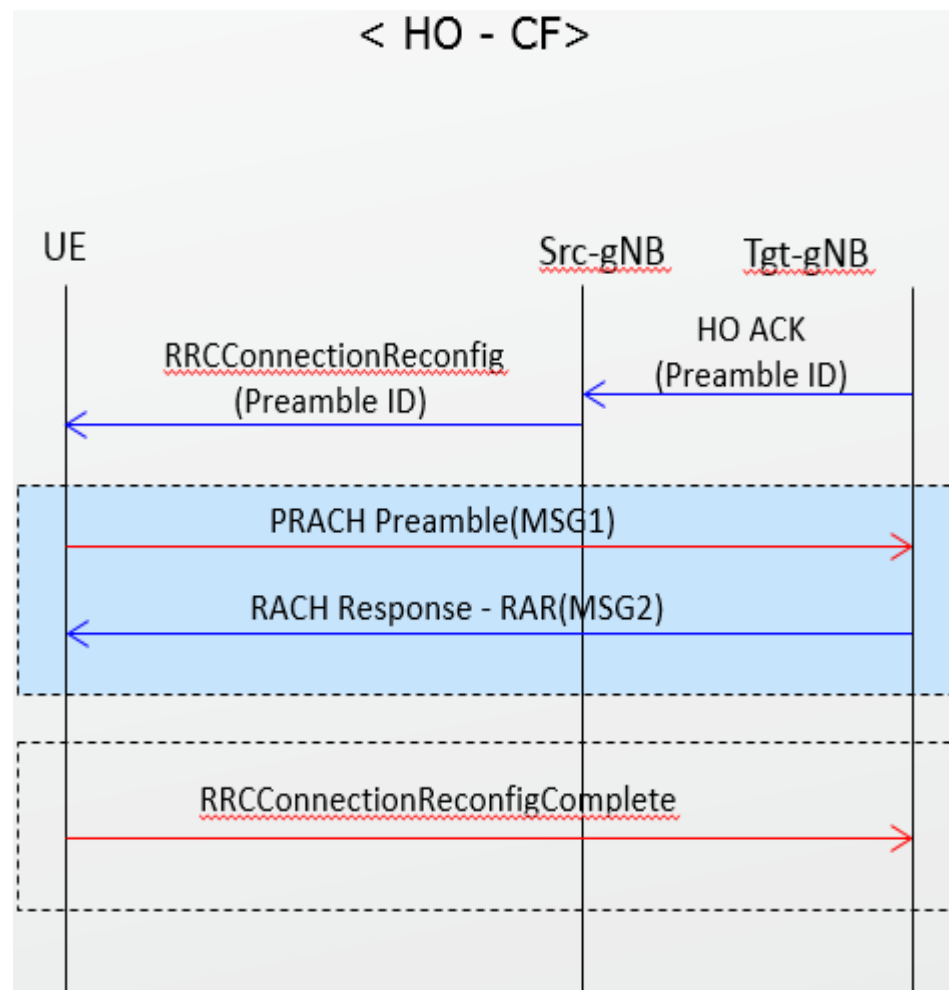
```
CFRA-SSB-Resource ::= SEQUENCE {  
    ssb SSB-Index,  
    ra-PreambleIndex INTEGER (0..63),  
    ...  
}  
  
CFRA-CSI-RS-Resource ::= SEQUENCE {  
    csi-RS CSI-RS-Index,  
    ra-OccasionList SEQUENCE (SIZE (1..maxRA-OccasionsPerCSI-RS)) OF INTEGER (0..maxRA-Occasions-1),  
    ra-PreambleIndex INTEGER (0..63),  
    ...  
}
```



# 基于非竞争的随机接入一场景

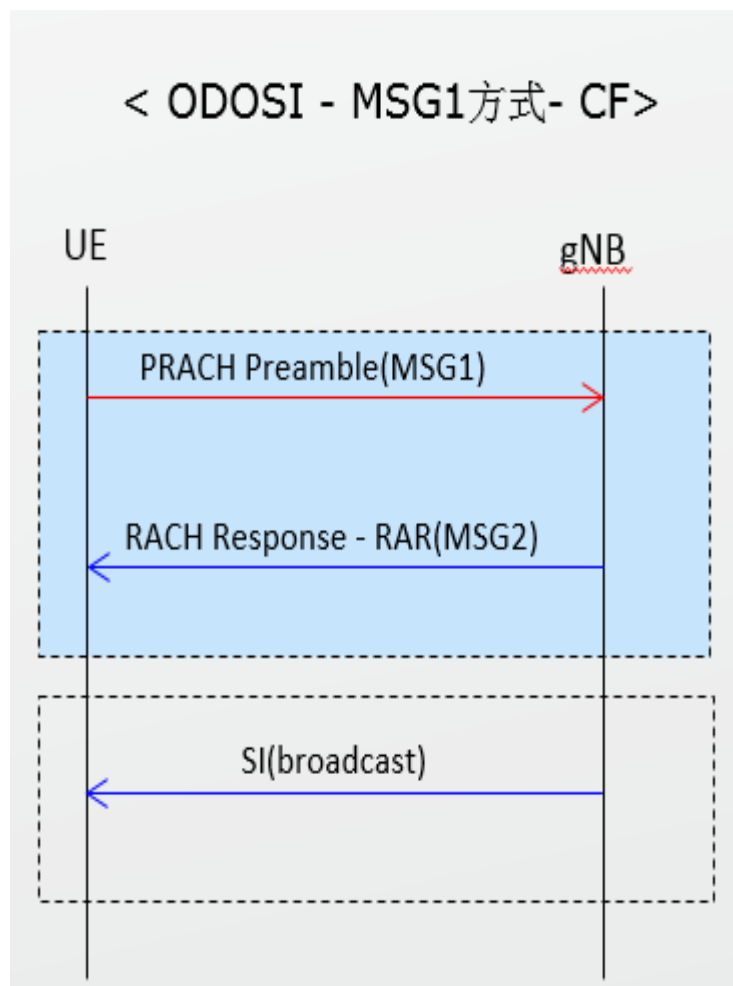


失步-下行数据到达



切换

# 基于非竞争的随机接入一场景



MSG1方式获取其他系统消息



## 5G随机接入重点参数

# 1、PRACH配置索引

## PRACH配置索引

参数ID: Prach Configuration Index

当参数取值为0~255时,小区的PRACH配置索引等于该参数的值;当参数取值为65535时,小区的PRACH配置索引由系统自动生成

这个参数主要是配置PRACH的Preamble格式以及发送preamble的时域资源

## UE如何知道使用哪种preamble格式?

在SIB1中RACH-ConfigGeneric (通用配置) -> prach-ConfigurationIndex (配置列表) 指示

```
RACH-ConfigGeneric ::= SEQUENCE {  
  prach-ConfigurationIndex INTEGER (0..255),  
  msg1-FDM ENUMERATED {one, two, four, eight},  
  msg1-FrequencyStart INTEGER (0..maxNrofPhysicalResourceBlocks-1),  
  zeroCorrelationZoneConfig INTEGER (0..15),  
  preambleReceivedTargetPower INTEGER (-200..-74),  
  preambleTransMax ENUMERATED {n3, n4, n5, n6, n7, n8, n10, n20, n50, n100, n200},  
  powerRampingStep ENUMERATED {dB0, dB2, dB4, dB6},  
  ra-ResponseWindow ENUMERATED {s11, s12, s14, s18, s110, s120, s140, s180}  
}
```

取值范围: 0-255

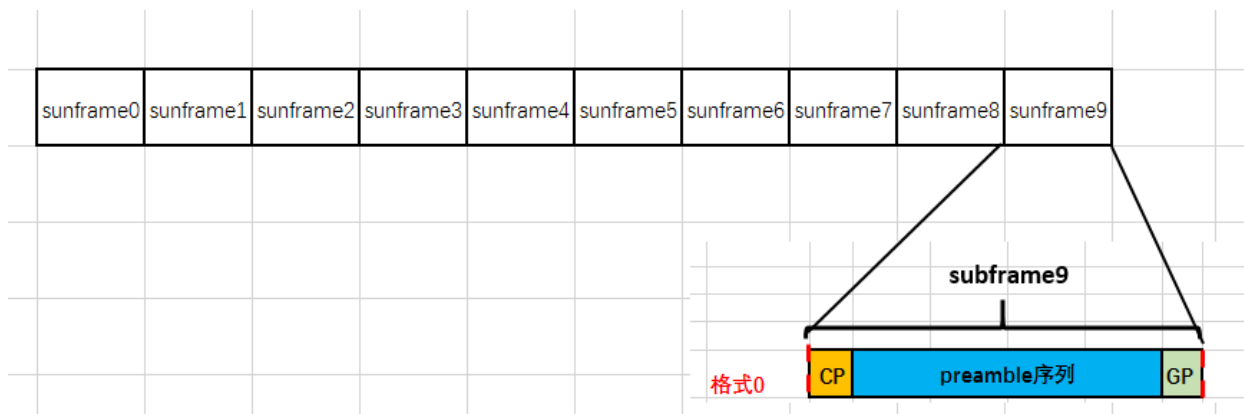
前面我们讲PRACH物理信道的时候, 讲过这个

根据小区不同的频域和双工模式, 38.211的第6.3.3节中给出了prach-ConfigurationIndex数值所对应的表格。

# 1、PRACH配置索引

比如Prach Configuration Index=4

< 38.211 v15.3.0-Table 6.3.3.2-3: Random access configurations for FR1 and unpaired spectrum >								
PRACH Configuration Index	Preamble format	$n_{SFN} \bmod x = y$		Subframe number	Starting symbol	Number of PRACH slots within a subframe	$N_t^{RA, slot}$ , number of time-domain PRACH occasions within a PRACH slot	$N_{dur}^{RA}$ , PRACH duration
		x	y					
4	0	2	1	9	0	-	-	0



查询NR DU小区PRACH配置

NR DU小区标识 = 0  
PRACH配置索引 = 65535  
选择SSB的RSRP门限 = 0  
竞争随机接入前导比例(%) = 255  
前导最大传输次数 = 10次  
PRACH门限提升比例(%) = 0  
窄波束试探门限 = -108  
随机接入算法优化开关 = 非竞争随机接入优化开关:关  
PRACH虚警检测算法开关:关  
PRACH虚警检测距离门限(米) = 100000  
(结果个数 = 1)

## 2、根序列逻辑索引

### 根序列逻辑索引

参数ID: Logical Root Sequence Index

邻区和服务小区配置的根序列如果一致，那么如果服务小区和邻区同时接入的时候，可能导致根序列冲突导致接入失败。

Preamble码是通过根序列（ZC序列）+循环位移得到的。

正常情况下，根序列设置不一样，循环位移NCs设置一样

如果两个邻区的根序列相同，那么通过相同的循环位移生成的preamble码，就会一样，这样两个相邻小区的preamble码相同的情况下，UE发起接入的时候，可能会产生冲突。进而导致接入失败

根序列逻辑索引

5  
4  
3  
6  
8  
9  
10

根序列逻辑索引，每个小区都不一样



### 3、基于逻辑根序列的循环移位参数

#### 基于逻辑根序列的循环移位参数

参数ID: zeroCorrelationZoneConfig

zeroCorrelationZoneConfig 用来设置Ncs循环位移参数，以便通过循环位移生成preamble序列

现网小区循环位移都设置成一样的。默认值为6

Table 6.3.3.1-5:  $N_{CS}$  for preamble formats with  $\Delta f^{RA} = 1.25$  kHz.

zeroCorrelationZoneConfig	$N_{CS}$ value		
	Unrestricted set	Restricted set type A	Restricted set type B
0	0	15	15
1	13	18	18
2	15	22	22
3	18	26	26
4	22	32	32
5	26	38	38
6	32	46	46
7	38	55	55
8	46	68	68
9	59	82	82
10	76	100	100
11	93	128	118
12	119	158	137
13	167	202	-
14	279	237	-
15	419	-	-

## 4、前导初始接收目标功率

### 前导初始接收目标功率

参数ID: PreambleInitRxTargetPwr

参数MO: NRDUCELLULPCCONFIG

该参数表示前导初始接收目标功率。这个参数用于计算UE发送preamble的发射功率，该参数设置越高，会提高前导的发射功率，减少发送前导的次数，但是会给邻小区带来干扰。

现网这个参数，设置值是-50dbm，步长是2dbm，实际值就是-100dbm

前导初始接收目标功率 (2毫瓦分贝)

-50

-50

$$P_{PRACH} = P_{PRACH,target,f,c} + PL_{b,f,c} = (\text{preambleReceivedTargetPower} + DELTA\_PREAMBLE + (PREAMBLE\_POWER\_RAMPING\_COUNTER - 1) \times PREAMBLE\_POWER\_RAMPING\_STEP) + (referenceSignalPower - \text{higher layer filtered RSRP})$$

## 5、前导最大传输次数

### 前导最大传输次数

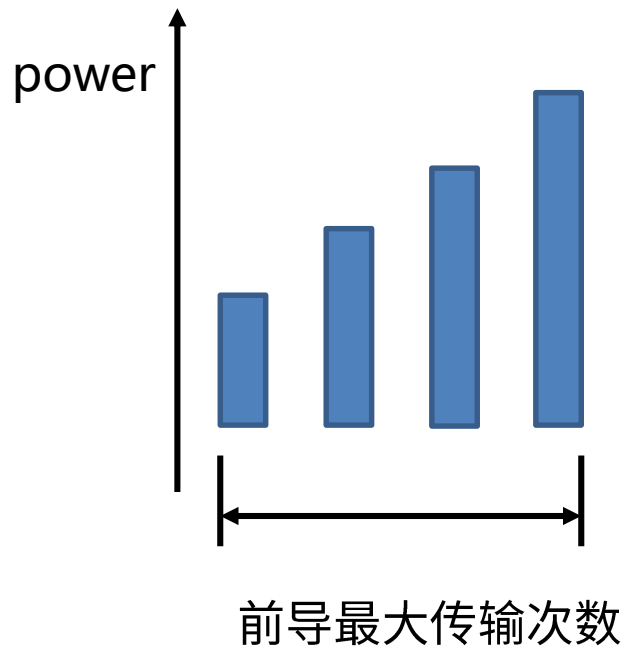
参数ID: MaxPreambleTransCnt

参数表示preamble最大发送次数。当UE发送随机接入前导后，未收到响应，则会把发射功率加上功率攀升步长进行再次尝试，直到前缀发送次数达到最大传输次数。

取值范围: {n3, n4, n5, n6, n7, n8, n10, n20, n50, n100, n200}

#### 查询NR DU小区PRACH配置

```
NR DU小区标识 = 0
PRACH配置索引 = 65535
选择SSB的RSRP门限 = 0
竞争随机接入前导比例(%) = 255
前导最大传输次数 = 10次
PRACH门限提升比例(%) = 0
```



$$P_{PRACH} = P_{PRACH,target,f,c} + PL_{b,f,c} = (preambleReceivedTargetPower + DELTA\_PREAMBLE + (PREAMBLE\_POWER\_RAMPING\_COUNTER - 1) \times PREAMBLE\_POWER\_RAMPING\_STEP) + (referenceSignalPower - higher\ layer\ filtered\ RSRP)$$

限制这个值的取值

## 6、Preamble的功率攀升步长

### Preamble的功率攀升步长

参数ID：powerRampingStep

取值：{dB0, dB2, dB4, dB6}

当UE发送随机接入前缀后，未收到响应，则会把发射功率加上powerRampingStep进行再次尝试，直到前缀发送次数达到前导最大传输次数

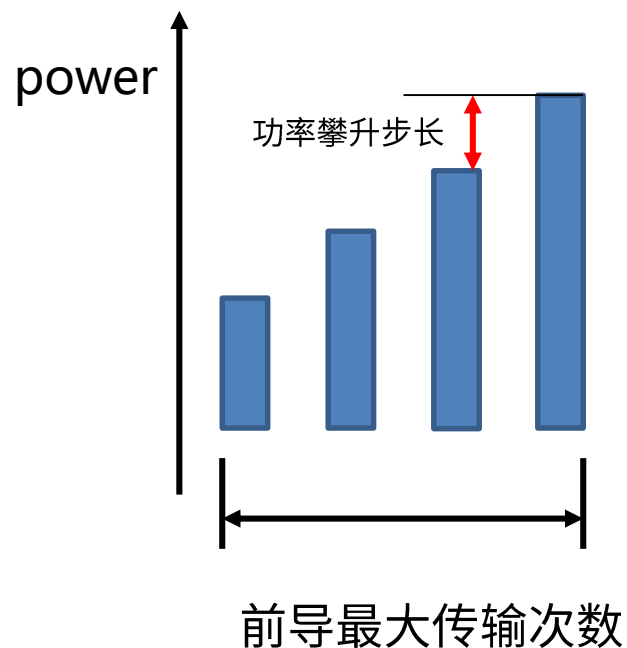
现网这个参数，设置值是4db，步长是1db，实际值就是4db

功率攀升步长

4 dB

4 dB

$$P_{PRACH} = P_{PRACH,target,f,c} + PL_{b,f,c} = ( preambleReceivedTargetPower + DELTA\_PREAMBLE + (PREAMBLE\_POWER\_RAMPING\_COUNTER - 1) \times PREAMBLE\_POWER\_RAMPING\_STEP ) + (referenceSignalPower - higher\ layer\ filtered\ RSRP)$$



\_\_\_\_\_



**Figure 6.2.3-1: MAC RAR**↵

## 7、Msg2中的功率调整值

### Msg2中的功率调整值

参数ID: TpcValueInMsg2

该参数影响MSG3的发射功率。该参数设置的越小，Msg3的发射功率越低，对邻区的干扰越小；  
该参数设置的越大，Msg3的发射功率越高，对邻区的干扰越大。

现网这个参数，设置值是6db，步长是1db，实际值就是6db

Msg2中的功率调整值

6 dB  
6 dB  
6 dB

$$P_{\text{PUSCH},b,f,c}(i,j,q_d,l) = \min \left\{ \begin{array}{l} P_{\text{CMAX},f,c}(i), \\ P_{\text{O\_PUSCH},b,f,c}(j) + 10 \log_{10}(2^{\mu} \cdot M_{\text{RB},b,f,c}^{\text{PUSCH}}(i)) + \alpha_{b,f,c}(j) \cdot PL_{b,f,c}(q_d) + \Delta_{\text{TF},b,f,c}(i) + f_{b,f,c}(i,l) \end{array} \right\} \quad [\text{dBm}]^{\leftarrow}$$

$$f_{b,f,c}(0,l) = \Delta P_{\text{rampup},b,f,c} + \delta_{\text{msg2},b,f,c}$$

## 8、Msg3相对于前导的功率偏置

Msg3相对于前导的功率偏置

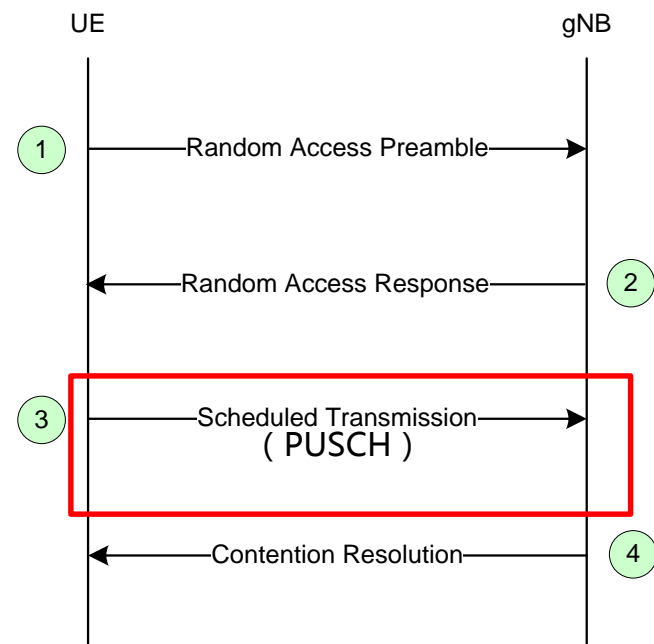
参数ID: DeltaPreambleMsg3

该参数参与计算MSG3的发射功率，该参数设置的越大，Msg3的发射功率越高，Msg3被基站成功接收的概率越大，对邻区的干扰越大。

现网这个参数，设置值是4，步长是2db，实际值就是8db

Msg3相对于前导的功率偏置 (2分贝)

4  
4



$$P_{\text{PUSCH},b,f,c}(i,j,q_d,l) = \min \left\{ P_{\text{CMAX},f,c}(i), P_{\text{O\_PUSCH},b,f,c}(j) + 10 \log_{10} (2^{\mu} \cdot M_{\text{RB},b,f,c}^{\text{PUSCH}}(i)) + \alpha_{b,f,c}(j) \cdot PL_{b,f,c}(q_d) + \Delta_{\text{TF},b,f,c}(i) + f_{b,f,c}(i,l) \right\} \quad [\text{dBm}]^{\leftarrow}$$

$$P_{\text{O\_NOMINAL\_PUSCH},f,c}(0) = P_{\text{O\_PRE}} + \Delta_{\text{PREAMBLE\_Msg3}}$$



感谢观看