

NB-IoT 上行物理层技术

相比 LTE 的上行物理信道，NB-IoT 的上行物理信道可谓简化了很多，因此一些流程机制也改变很多。由于不需要在上行信道中传输 CSI 或者 SR，因此在上行信道结构设计中也不需要专门保留上行控制共享信道。NB-IoT 上行信道包含两种物理信道，一个是窄带物理上行共享信道（NPUSCH），另外一个窄带物理随机接入信道（NPRACH），控制信息可以通过 NPUSCH 复用传输，这意味着 NPUSCH 不仅承载上行数据业务，同时也肩负了类似 LTE 中 PUCCH 承载一些上行反馈信息的功能。另外，由于没有了上行资源调度的概念，同时为了简化帧结构，作为全频段信道估计用的 Sounding Reference Signal(SRS)也被省略掉了，上行物理信号只保留了窄带解调参考信号，这样不仅简化了物理层流程，同时也将有限的带宽资源尽可能预留给了数据传输。

NPUSCH (Narrowband Physical uplink shared channel)

上行传输有两种模式，一种是 single-tone,另一种是 multi-tone。对于 single-tone 传输模式，可以有两种子载波间隔 3.75kHz 和 15kHz，资源块在这里并没有定义，这意味着并不以资源块作为基本调度单位。如果子载波间隔是 15kHz,那么上行包含连续 12 个子载波，如果子载波间隔是 3.75kHz，那么上行包含连续 48 个子载波。我们知道，对于通过 OFDM 调制的数据信道，如果在同样的带宽下，子载波间隔越小，相干带宽越大，那么数据传输抗多径干扰的效果越好，数据传输的效率更高，当然，考虑到通过 IFFT 的计算效率，子载波也不能设置的无限小。同时，也要考虑与周围 LTE 大网的频带兼容性，选取更小的子载波也需要考虑与 15kHz 的兼容性。当上行采取 single tone 3.75kHz 模式传输数据时，物理层帧结构最小单位为基本时长 2ms 时隙，该时隙与 FDD LTE 帧保持对齐。每个时隙包含 7 个 OFDM 符号，每个符号包含 8448 个 Ts(时域采样)，其中这 8448 个 Ts 含有 256Ts 个循环校验前缀（这意味着 IFFT 的计算点数是 8448-256=8192 个，恰好是 2048 (15kHz) 的 4 倍），剩下的时域长度（2304Ts）作为保护带宽。single-tone 和 multi-tone 的 15kHz 模式与 FDD LTE 的帧结构是一致的，最小单位是时长为 0.5ms 的时隙。而区别在于 NB-IoT 没有调度资源块，single-tone 以 12 个连续子载波进行传输，multi-tone 可以分别按照 3,6,12 个连续子载波分组进行数据传输。

相比 LTE 中以 PRB 对进行基本资源调度单位，NB-IoT 的上行共享物理信道 NPUSCH 的资源单位是以灵活的时频资源组合进行调度的，调度的基本单位称作资源单位（Resource Unit）。NPUSCH 有两种传输格式，两种传输格式对应的资源单位不同，传输的内容也不一样。NPUSCH 格式 1 用来承载上行共享传输信道 UL-SCH，传输用户数据或者信令，UL-SCH 传输块可以通过一个或者几个物理资源单位进行调度发送。所占资源单位包含 single-tone 和 multi-tone 两种格式。其中

single-tone 3.75kHz 32ms, 15kHz 8ms;

multi-tone 15kHz 3 子载波 4ms,6 子载波 2ms,12 子载波 1ms。

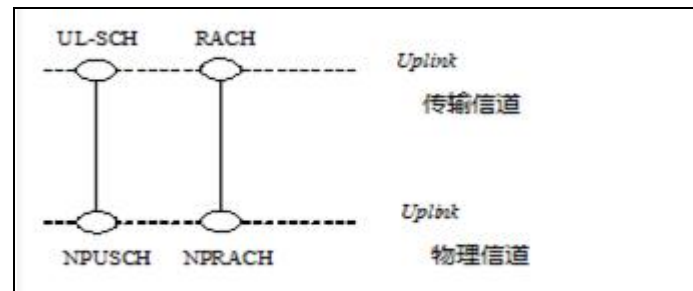
NPUSCH 格式 2 用来承载上行控制信息（物理层），例如 ACK/NAK 应答。根据 3.75kHz 8ms 或者 15kHz 2ms 分别进行调度发送的。

NPUSCH格式	子载波间隔	频域子载波数	时域time slot数量	时域持续时间	time slot长度
1	3.75kHz	1	16	32ms	2ms
		1	16	8ms	0.5ms
		3	8	4ms	0.5ms
		6	4	2ms	0.5ms
		12	2	1ms	0.5ms
2	3.75kHz	1	4	8ms	2ms
	15kHz	1	4	2ms	0.5ms

NPUSCH 信道基本调度资源单位（Resource Unit）

NB-IoT 没有特定的上行控制信道，控制信息也复用在上行共享信道（NPUSCH）中发送。所谓的控制信息指的是与 NPDSCH 对应的 ACK/NAK 的消息，并不像 LTE 大网那样还需要传输表征信道条件的 CSI 以及申请调度资源

源的 SR (Scheduling Request)。



NB-IoT 上行物理信道进行了简化

NPUSCH 目前只支持天线单端口，NPUSCH 可以包含一个或者多个 RU。这个分配的 RU 资源单位数量由 NPDDCH 承载的针对 NPUSCH 的 DCI 格式 N0 (format N0) 来指明。这个 DCI 格式 N0 包含分配给 RU 的连续子载波数量 n_{sc} 、分配的 RU 数量 N_{RU} ，重复发送的次数 N_{rep} 。UE 通过解读 DCI 格式 N0 获取相关 NPUSCH 上行传输的时间起点以及所占用的时频资源，上行共享信道子载波间隔与解码随机接入 grant 指示 Msg3 发送采用的子载波间隔保持一致。另外 NPUSCH 上行具体的对应取值在协议中有明确的定义（见 36.213 R13 16.5.1.1&16.5.1.2）。在子载波上映射的 NPUSCH 符号应该与上行参考信号错开。在映射了 N_{slots} 个时隙后，为

了提升上行软覆盖，保证数据传输质量，这 N_{slots} 个时隙需要被重复 N_{rep} 次。具体的计算公式如下（ 36.211 R13 10.1.3.6 ）

$$M_{identical}^{NPUSCH} = \begin{cases} \min \left(M_{rep}^{NPUSCH} / 2, 4 \right) & N_{sc}^{RU} > 1 \\ 1 & N_{sc}^{RU} = 1 \end{cases}$$

$$N_{slots} = \begin{cases} 1 & \Delta f = 3.75 \text{ kHz} \\ 2 & \Delta f = 15 \text{ kHz} \end{cases}$$

我们进行一点简单的计算。

对于 NPUSCH 格式 1 中子载波间隔 3.75kHz, RU 频域子载波数为 1 的情况，查表计算得出每个传输的时隙不需要重复。

这样 NPUSCH 的待发符号会映射满一个 RU (1 个子载波，8 个时隙，持续 32ms)，之后再重复 $M_{rep}^{NPUSCH} - 1$ 次；

对于 NPUSCH 格式 2 中子载波间隔 15kHz, RU 频域子载波数为 1 的情况，查表计算出每 2 个时隙需要被重复发送，而

RU 内部重复次数 $M_{identical}^{NPUSCH}$ 为 1，其实意味着和前一种情况一致了。NPUSCH 的待发信号映射满一个 RU (1 个子载波，4 个时隙，持续 2ms)，之后再重复 $M_{rep}^{NPUSCH} - 1$ 次；

对于 NPUSCH 格式 1 中子载波间隔 15kHz，RU 频域子载波数为 6 的情况，查表计算出每 2 个时隙需要被重复发送，

假设通过解码 DCI 获得 M_{rep}^{NPUSCH} 的值为 4，那么经计算 $M_{identical}^{NPUSCH}$ 为 2，那么实际情况是在该 RU 持续的 4 个时隙内，NPUSCH 符号先映射满 2 个时隙，然后 RU 内部一重复，这种映射方式直到 NPUSCH 符号被完全发送完，之后 NPUSCH 重复 3 次，也就是说每映射 2 个时隙的 NPUSCH 符号，实际总共需要 16 个时隙重来保障上行数据接受的可靠性。

通过这些例子的简单计算，我们可以摸清 NPUSCH 映射传输的一些规律，NPUSCH 采取“内部切片重传”与“外部整体重传”的机制保证上行信道数据的可靠性。对于格式 2 承载的一些控制信息，由于数据量较小，就没有采

取内部分割切片的方式，而是数据 NPUSCH 承载的控制信息传完以后再重复传输保证质量。NPUSCH 在传输过程中需要与 NPRACH 错开，NPRACH 优先程度较高，如果与 NPRACH 时隙重叠，NPUSCH 需要延迟一定的时隙传输（36.211 R13 10.1.3.6）。在传输完 NPUSCH 或者 NPUSCH 与 NPRACH 交叠需要延迟 256ms 传输，需要在传输完 NPUSCH 或者 NPRACH 之后加一个 40ms 的保护间隔，而被延迟的 NPUSCH 与 40ms 保护间隔交叠的数据部分则认为是保护带的一部分了，换言之，这部分上传数据被废弃掉了。在 NPUSCH 的上行信道配置中还同时考虑了与 LTE 上行参考信号 SRS 的兼容问题，这里通过 SIB2-NB 里面的 *NPUSCH-ConfigCommon-NB* 信息块中的 *npusch-AllSymbols* 和 *srs-SubframeConfig* 参数共同控制，如果 *npusch-AllSymbols* 设置为 false，那么 SRS 对应的位置记作 NPUSCH 的符号映射，但是并不传输，如果 *npusch-AllSymbols* 设置为 true，那么所有的 NPUSCH 符号都被传输。对于需要兼容 SRS 进行匹配的 NPUSCH，意味着一定程度上的信息损失，这也是与 LTE 采取带内模式组网时需要考虑的。

NB-IoT 上行共享信道具有功控机制，通过“半动态”调整上行发射功率使得信息能够成功在基站侧被解码。所以说上行功控的机制属于“半动态”调整（这里与 LTE 功控机制比较类似），是由于在功控过程中，目标期望功率在小区级是不变的，UE 通过接入小区或者切换至新小区通过重配消息获取，功控中进行调整的部分只是路损补偿。UE 需要检测 NPDCCH 中的 UL grant 以确定上行的传输内容（NPUSCH 格式 1,2 或者 Msg3），不同内容路损的补偿的调整系数有所不同，同时上行期望功率的计算也有差异，具体计算公式可以参见 36.213 R13 16.2.1.1.1。上行功控以时隙作为基本调度单位，值得注意的是在如果 NPUSCH 的 RU 重传次数大于 2，那么意

味着此时 NB-IoT 进行深度覆盖受限环境，上行信道不进行功控，采取最大功率发射，该值不超过 UE 的实际最大发射功率能力，对于 class3UE 最大发射功率能力是 23dBm，class5UE 最大发射功率能力 20dBm。

$$P_{\text{CMAX},c}(i) [\text{dBm}]$$

DMRS(Demodulation reference signal)

不同格式的 RU 对应产生不同的解调参考信号。主要按照 $N_{\text{sc}}^{\text{RU}} = 1$ （一个 RU 包含的子载波数量）和

$$N_{\text{sc}}^{\text{RU}} > 1$$

两类来计算。另外 NPUSCH 两种格式的解调参考信号也不一样，格式 1 每个 NPUSCH 传输时隙包含一个解调参考信号，而格式 2 每个传输时隙则包含 3 个解调参考信号，这种设计可能源于承载控制信息的 NPUSCH 的 RU 中空闲位置较多，而且分配给控制信息的 RU 时域资源相对较少，因此每个传输时隙通过稍多的解调参考信号予以进行上行控制信息的解调保障。对于包含不同子载波的 RU 而言（当然我们也可以按照 single-tone, multi-tone 分类）需要保证每个子载波至少一个 DMRS 参考信号以确定信道质量，同时 DMRS 的功率与所在 NPUSCH 信道的功率保持一致。对于 multi-tone 中如何生成参考信号，既可以通过解读系统消息 SIB2-NB 中的 *NPUSCH-ConfigCommon-NB* 信息块中的参数（可选）获取，也可以根据小区 ID 通过既定公式计算获取（36.211 R13 36.211）。解调参考信号可以通过序列组跳变（Group hopping）的方式避免不同小区间上行符号的干扰。序列组跳变并不改变 DMRS 参考信号在不同子帧的位置，而是通过编码方式的变化改变

DMRS 参考信号本身。对于 $N_{\text{sc}}^{\text{RU}} = 1$ 的 RU，RU 内部的每个时隙中的序列组跳变是一样的，而对于

$$N_{\text{sc}}^{\text{RU}} > 1$$

的 RU，RU 内部每个偶数时隙的序列组的计算方式就要重新变化一次。DMRS 映射到物理资源的原则是确保 RU 内每个时隙的每个子载波至少一个参考信号，这个也很好理解，通俗的说就说保证每个时隙上的子载波能够被正确解调，同时又不由于过多的分配 DMRS 导致资源消耗过多，物理层设计的时候也进行了相应

的权衡。当然在物理资源映射分配上格式 1 与格式 2 的 DMRS 还是有些差异。格式 1 在每个时隙每个子载波上只分配 1 个 DMRS 参考信号，格式 2 在每个时隙每个子载波上分配 3 个 DMRS 参考信号。

NPUSCH格式	每时隙上的OFDM符号位	
	$\Delta f=3.75\text{kHz}$	$\Delta f=15\text{kHz}$
1	4	3
2	0, 1, 2	2, 3, 4

NB-IoT 上行 SC-FDMA 基带信号对于单载波 RU 模式需要区分 BPSK，QPSK 模式,即基于不同的调制方式和不同的时隙位置进行相位偏置，这一点与 LTE 是不同的，LTE 上行的 SC-FDMA 主要是由于考虑到终端上行的 PAPR 问题采取在 IFFT 前加 DFT 变换，同时分配给用户频域资源中不同子载波功率是一致的，这样 PAPR 问题得到了有效的缓解。而对于 NB-IoT 而言，对于 single-tone 的这种单载波传输的方式，功率谱密度更高，对带外旁瓣泄露更加敏感，另外相比 multi-tone 传输方式，单 DFT 抽头抑制 PAPR 效果相对较弱，因此通过基于不同调制方式数据的相位偏置可以进行相应的削峰处理，同时又不会像简单 clipping 技术一样使得频域旁瓣产生泄漏，产生带外干扰。

NPRACH(Narrowband physical random access channel)

窄带随机接入信道顾名思义就是传输随机接入请求的。随机接入过程是 UE 从空闲态获取专用信道资源转变为连接态的重要方法手段。在 NB-IoT 中没有了同步状态下的 SR 流程对于调度资源的申请，NB-IoT 主要靠随机接入流程申请调度资源。随机接入使用的 3.75kHz 子载波间隔，同时采取在单载波跳频符号组的方式发送不同循环前缀的 preamble。随机接入符号组如图所示，它由 5 个相同的 OFDM 符号与循环前缀拼接而成。随机接入前导序列只在前面加循环前缀，而不是在每个 OFDM 符号前都加（如 NB-IoT 的 NPUSCH 上行共享信道），主要原因是由于其并不是多载波调制，因此不用通过 CP 保持子载波之间的正交性，节省下 CP 的资源可以承载更多的前导码信息，基站侧通过检测最强径的方式确认随机接入前导码。随机接入前导码包含两种格式，两种格式的循环前缀不一样。



一个前导码 (preamble) 包含了 4 个符号组，同时被连续传输 N_{rep}^{NPRACH} 。通过一系列的时频资源参数配置，随机接入前导码占据预先分配的时频资源进行传输。UE 通过解读 SIB2-NB 消息获取这些预配置参数。


```

NPRACH-ParametersList-NB-r13 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF NPRACH-Parameters-NB-r13
NPRACH-ParametersList-NB-v1330 ::= SEQUENCE (SIZE (1.. maxNPRACH-Resources-NB-r13)) OF NPRACH-Parameters-NB-v1330
NPRACH-Parameters-NB-r13 ::= SEQUENCE {
    nprach-Periodicity-r13 ENUMERATED {ms40, ms80, ms160, ms240, ms320, ms640, ms1280, ms2560},
    nprach-StartTime-r13 ENUMERATED {ms8, ms16, ms32, ms64, ms128, ms256, ms512, ms1024},
    nprach-SubcarrierOffset-r13 ENUMERATED {n0, n12, n24, n36, n48, n60, n72, n84, n96, n108, n120, n132, n144, n156, n168, n180, n192, n204, n216, n228, n240, n252, n264, n276, n288, n300, n312, n324, n336, n348, n360, n372, n384, n396, n408, n420, n432, n444, n456, n468, n480},
    nprach-NumSubcarriers-r13 ENUMERATED {n12, n24, n36, n48},
    nprach-SubcarrierMSG3-RangeStart-r13 ENUMERATED {zero, oneThird, twoThird, one},
    maxNumPreambleAttemptCE-r13 ENUMERATED {n3, n4, n5, n6, n7, n8, n10, spare1},
    numRepetitionsPerPreambleAttempt-r13 ENUMERATED {n1, n2, n4, n8, n16, n32, n64, n128},
    npdcch-NumRepetitions-RA-r13 ENUMERATED {r1, r2, r4, r8, r16, r32, r64, r128, r256, r512, r1024, r2048, spare4, spare3, spare2, spare1},
    npdcch-StartSF-CSS-RA-r13 ENUMERATED {v1dot5, v2, v4, v8, v16, v32, v64, v128, v256, v512, v1024, v2048},
    npdcch-Offset-RA-r13 ENUMERATED {zero, oneEighth, oneFourth, threeEighth}
}
NPRACH-Parameters-NB-v1330 ::= SEQUENCE {
    nprach-NumCBRA-StartSubcarriers-r13 ENUMERATED {n8, n10, n11, n12, n20, n22, n23, n24, n32, n34, n35, n36, n40, n44, n46, n48}
}
RSRP-ThresholdsNPRACH-InfoList-NB-r13 ::= SEQUENCE (SIZE (1..2)) OF RSRP-Range
-- ASN1STOP

```

如何通过这些配置参数确定前导码的起始位置？为了避免枯燥的参数解读与描述，我们通过简单的计算来说明。

起始

假设 $nprach-Periodicity=1280ms$ ，那么发起随机接入的无线帧号应该是 0,128,256....(128 的整数倍)，当然随着这个取值越大，随机接入延迟越大，但是这对于物联网 NB-IoT 来说并不太敏感，基于抄水表的物联网终端更需要保证的是数据传递准确性，对于延迟可以进行一定的容忍。 $nprach-StartTime$ 决定了具体的起始时刻，假设 $nprach-StartTime=8$ ，那么前导码可以在上述无线帧的第 4 号时隙上发送（ $8ms/2ms=4$ ）。这两组参数搭配取值也有一定的潜规则，如果 $nprach-Periodicity$ 取值过小， $nprach-StartTime$ 取值过大，建议可以进行适当的调整。

重复

一个前导码占用 4 个符号组，假设 $numRepetitionsPerPreambleAttempt=128$ （最大值），意味前导码需要被重复传递 128 次，这样传输前导码实际占用时间为 $4*128*(T_{CP}+T_{SEQ})T_s$ （时间单位），而协议规定，每传输 $4*64(T_{CP}+T_{SEQ})T_s$ ，需要加入 $40*30720T_s$ 间隔（36.211 R13 10.1.6.1），假设采取前导码格式 0 进行传输，那么传输前导码实际占用时间为 796.8ms，相比 LTE 的随机接入，这是一个相当大的时间长度，物联网终端随机接入需要保证用户的上行同步请求被正确解码，而对于接入时延来讲依然不那么敏感。

频域位置

分配给 preamble 的频域资源不能超过频域最大子载波数，即

$nprach-SubcarrierOffset+nprach-NumSubcarriers \leq 48$ ，超过 48 意味着参数配置无效。这两个参数，决定了每个符号（注：我们这里并没有用 OFDM 符号这个词，由于随机接入前导码并没有采取 OFDM 调制技术，只是占用了 OFDM 符号的位置而已）中 NPRACH 的起始位置，NPRACH 采取在不同的符号的不同单子载波跳频，但是有一个限制条件，就是在起始位置以上的 12 个子载波内进行跳频，具体的跳频位置计算比较复杂，可参见（36.311 R13 10.1.6.1）

$nprach-NumCBRA-StartSubcarriers$ 和 $nprach-SubcarrierMSG3-RangeStart$ 这两个参数决定了随机过程竞争阶段的起始子帧位置，如果 $nprach-SubcarrierMSG3-RangeStart$ 取值为 1/3 或者 2/3，那么指示 UE 网络侧支持 multi-tone 方式的 msg3 传输。

基带导频信号生成

基带导频信号的生成和每个符号组跳频的偏置相关，是个复信号，具体计算公式参见 36.311 R13 10.1.6.2。

随机接入过程

UE 在发起非同步随机接入之前，需要通过高层获取 NPRACH 的信道参数配置。在物理层的角度来看，随机接入过程包含发送随机接入前导码和接收随机接入响应两个流程。其余的消息，比如竞争解决及响应 (msg3, msg4)，认为在共享信道传输，因此不认为是物理层的随机接入过程。

过程 1: 发送随机接入前导码 (发送 Msg1)

随机接入信道为每个连续的前导码符号占用一个子载波。层 1 的随机过程是由高层的接入请求触发的，随机接入的发射目标功率 (随机信道受高层控制有功率抬升机制)，对应的 RA-RNTI 和 NPRACH 资源分配也是由高层决定的。

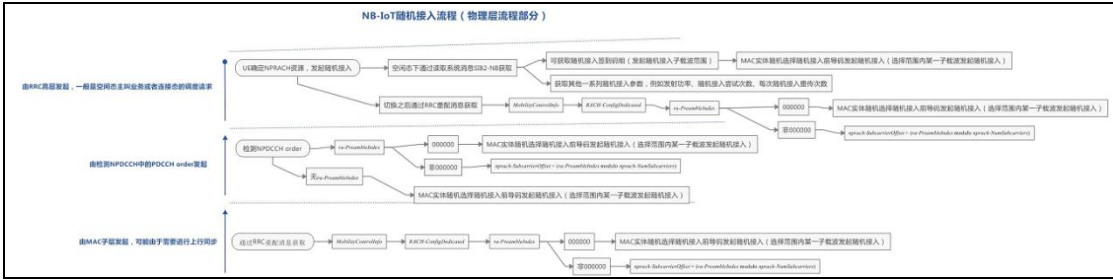
过程 2: 接收随机接入响应获取 uplink grant (解码 Msg2, RAR)

UE 通过 RA-RNTI 解码下行 NPDCCH 获取被对应 RA-RNTI 加扰的 DCI，通过 DCI 获取对应 DL-SCH 资源传输块，将资源块传递高层，高层解析资源块，并向物理层指明 Nr-bit 的上行授权 (uplink grant)。Nr=15，这 15bit 包含了如下的相关信息 (从左至右)

〈MSB〉→15-bit UL Grant (窄带随机接入响应资源预留)→〈LSB〉						
上行子载波间隔 Δf (1bit)	为竞争解决 (Msg3) 分配的子载波 I_{sc} (6bit)	调度时延 T_{delay} ，在检测到随机接入 grant 下行子帧之后的 k_0 发起上行竞争请求 (2bit)	Msg3 重传次数 N_{rep} (3bit)	通过 NPUSCH 传输 Msg3 的 MCS 索引 (3bit)		
0=3.75kHz	$I_{sc}=I_{sc}$ ，取值 0~47，而 48, 49, ..., 63 作为预留 (3.75kHz)	$T_{delay}=0, k_0=12$	$I_{rep}=0, N_{rep}=1$	000, pi/2 BPSK ($\Delta f=3.75kHz$ 或 $\Delta f=15kHz$ 并 $I_{sc}=0, 1, \dots, 11$)	QPSK ($\Delta f=15kHz$ 并 $I_{sc}>11$)	占用 4 个 RU，传输块 (TBS) 88bit
1=15kHz	$I_{sc}=I_{sc}$ ，当 I_{sc} 取值 0~11 (15kHz)	$T_{delay}=1, k_0=16$	$I_{rep}=0, N_{rep}=2$	001, pi/4 QPSK ($\Delta f=3.75kHz$ 或 $\Delta f=15kHz$ 并 $I_{sc}=0, 1, \dots, 11$)	QPSK ($\Delta f=15kHz$ 并 $I_{sc}>11$)	占用 3 个 RU，传输块 (TBS) 88bit
	$I_{sc}=3(I_{sc}-12)+\{0, 1, 2\}$ ，当 I_{sc} 取值 12~15 (15kHz)	$T_{delay}=2, k_0=32$	$I_{rep}=0, N_{rep}=4$	010, pi/4 QPSK ($\Delta f=3.75kHz$ 或 $\Delta f=15kHz$ 并 $I_{sc}=0, 1, \dots, 11$)	QPSK ($\Delta f=15kHz$ 并 $I_{sc}>11$)	占用 1 个 RU，传输块 (TBS) 88bit
	$I_{sc}=6(I_{sc}-16)+\{0, 1, 2, 3, 4, 5\}$ ，当 I_{sc} 取值 16~17 (15kHz)	$T_{delay}=3, k_0=64$	$I_{rep}=0, N_{rep}=8$	预留		
	$I_{sc}=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ ，当 I_{sc} 取值 18 (15kHz)		$I_{rep}=0, N_{rep}=16$			
	预留，当 I_{sc} 取值 19~63 (15kHz)		$I_{rep}=0, N_{rep}=32$			
			$I_{rep}=0, N_{rep}=64$			
			$I_{rep}=0, N_{rep}=128$			

通过解读 NPDCCH 中 DCI 获得随机接入响应资源预留，规定了 Msg3 发送占用的资源以及调制方式

为了更直观的说明物理层随机接入过程，我们用流程图的方式进行了整理



微信扫码以下二维码，免费加入【5G 俱乐部】，还赠送整套：5G 前沿、NB-IoT、4G+ (VoLTE) 资料。

