

NB-IoT 下行物理层技术

蜂窝式物联网技术大体分为两种，一种是 NB-IoT 技术，一种是非 NB-IoT 技术（例如 eMTC 等），这两种技术在物理层架构，协议标准上有所区别。

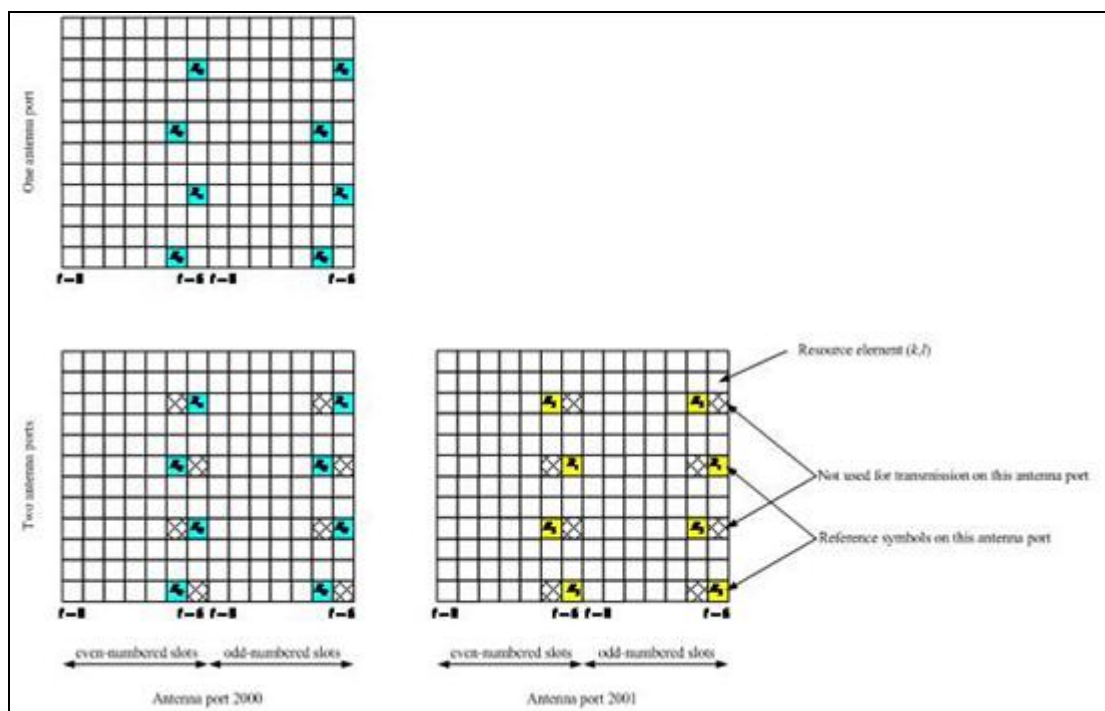
NB-IoT 技术提供了一种低功耗的网络接入方式。目前 NB-IoT 协议只支持 FDD(频分双工) 工作模式，载波带宽 180kHz，相当于 LTE 网络的一个 PRB 的带宽，子载波间隔可以是 3.75kHz 或者 15kHz。NB-IoT 与 Rel-8 定义的 LTE 网络技术，UE 是相对独立的，像跨系统移动性，切换，测量报告，GBR，载波聚合，双连接，CSFB 回落，物物通信等技术功能在 NB-IoT 是不支持的。

NB-IoT 与 LTE 大网之间的共存模式有三种，分别是 In-band(带内),Guard-band (保护带),Standalone (独立)。这也意味着 NB-IoT 如何组网，采取带内组网方式部署较容易，同厂家升级较快，但是对于 LTE 网内的资源调度会有一定影响。保护带组网方式相比带内频率效率更高，但是需要考虑和大网的干扰共存。独立组网方式与 LTE 大网可以完全分开，独立运维，但是需要额外的 FDD 频谱资源。

UE 通过小区同步，解读 MIB-NB 系统消息可以得知组网的模式。

```
hyperSFN-LSB-r13          BIT STRING (SIZE (2)),..
schedulingInfoSIB1-r13    INTEGER (0..15),..
systemInfoValueTag-r13    INTEGER (0..31),..
ab-Enabled-r13            BOOLEAN,..
operationModeInfo-r13     CHOICE {
  inband-SamePCI-r13       Inband-SamePCI-NB-r13,..
  inband-DifferentPCI-r13  Inband-DifferentPCI-NB-r13,..
  guardband-r13           Guardband-NB-r13,..
  standalone-r13          Standalone-NB-r13,..
},..
spare                     BIT STRING (SIZE (11))..
},..
ChannelRasterOffset-NB-r13 ::= ENUMERATED {khz-7dot5, khz-2dot5, khz2dot5, khz7dot5}..
Guardband-NB-r13 ::= SEQUENCE {
  rasterOffset-r13        ChannelRasterOffset-NB-r13,..
  spare                   BIT STRING (SIZE (3))..
},..
Inband-SamePCI-NB-r13 ::= SEQUENCE {
  eutra-CRS-SequenceInfo-r13  INTEGER (0..31) ..
},..
Inband-DifferentPCI-NB-r13 ::= SEQUENCE {
  eutra-NumCRS-Ports-r13      ENUMERATED {same, four},..
  rasterOffset-r13            ChannelRasterOffset-NB-r13,..
  spare                       BIT STRING (SIZE (2)) ..
},..
Standalone-NB-r13 ::= SEQUENCE {
  spare                      BIT STRING (SIZE (5)) ..
},..
-- ASN1STOP..
```

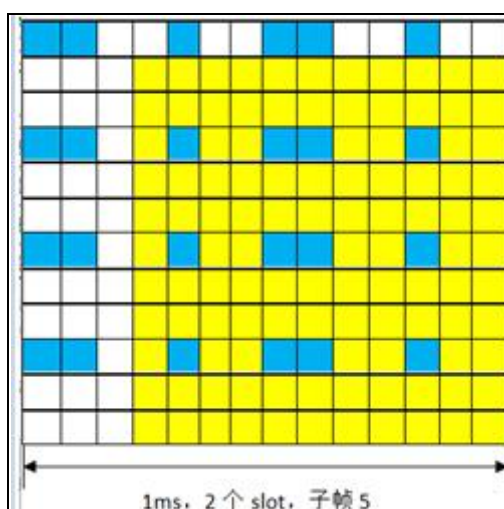
NRS (Narrowband Reference Signal) 如同 LTE 的 CRS，窄带参考信号也是 NB-IoT 里面重要的物理层信号，作为信道估计与网络覆盖评估的重要参考依据。在 UE 没有解读到 MIB-NB 里面的 *operationModeInfo* 字段时，UE 默认 NRS 窄带参考信号分别在子帧 0,4 和 9 (不包含 NSSS) 上进行传输。当 UE 解码 MIB-NB 中的 *operationModeInfo* 字段指示为 *guardband* 或者 *standalone* 模式后，在 UE 进一步解码 SIB1-NB 前，UE 默认 NRS 在子帧 0,1,3,4 和 9 (不包含 NSSS) 上进行传输。如果解码 SIB1-NB 后，UE 默认 NRS 在每个不含 NPSS 或者 NSSS 的 NB-IoT 下行子帧进行传输。当 UE 解码 MIB-NB 中的 *operationModeInfo* 字段指示为 *inband-SamePCI* 或者 *inband-DifferentPCI* 模式后，在 UE 解码 SIB1-NB 之前，UE 默认 NRS 在子帧 0,4,9 (不包含 NSSS) 上进行传输。当 UE 解码 SIB1-NB 之后，UE 默认在每个不含 NPSS 或者 NSSS 的 NB-IoT 的下行子帧进行传输。



单天线端口 NRS 位置 vs 双天线端口 NRS 位置 (注：NB-IoT 最多只支持下行双天线端口传输)

主同步信号 NPSS (Narrowband primary synchronization signal)

NB-IoT 的主同步信号仅作为小区下行同步使用。在 NB-IoT 中主同步信号传输的子帧是固定的，同时对应的天线端口号也是固定的，这也意味着在其他子帧传输的主同步信号的端口号并不一致。



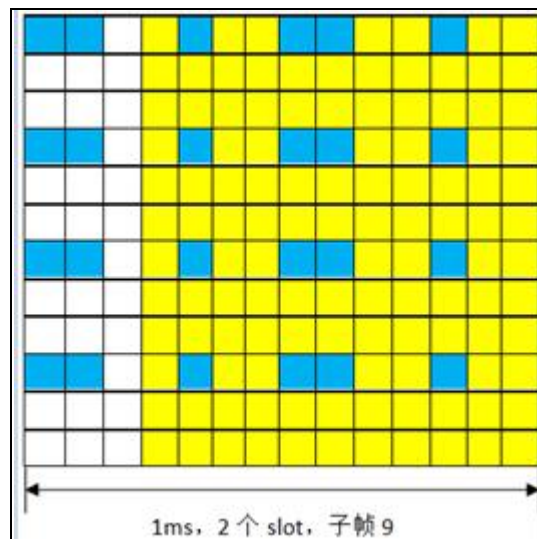
蓝色部分为 CRS 的位置，黄色部分为 NPSS 位置

值得注意的是，传输 NPSS 的 5 号子帧上没有 NRS 窄带参考信号，另外如果在带内组网模式下与 CRS 小区参考信号重叠，重叠部分不计作 NPSS，但是仍然作为 NPSS 符号的一个占位匹配项 (详见 36.211. R13 10.2.7.1.2)

辅同步信号 NSSS (Narrowband secondary synchronization signal)

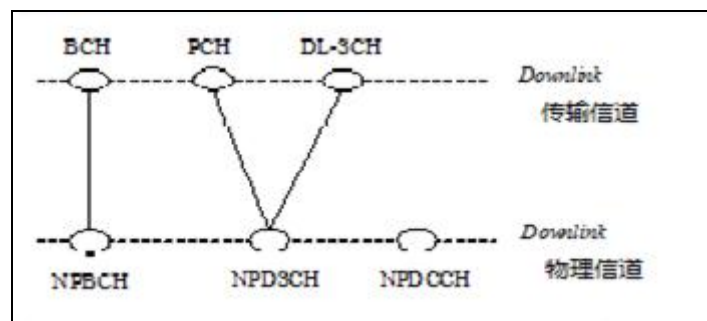
与 NPSS 位置部署原则大体一致，NSSS 部署在偶数无线帧的 9 号子帧上，从第 4 个 OFDM 符号开始，占满 12 个子载波。该 9 号子帧上没有 NRS 窄带参考信号，另外如果在带内组网模式下与 CRS 小区参考信号重叠，重叠部分不计作 NSSS，但是仍然作为 NSSS 符号的一个占位匹配项。

与 LTE 大网中 PCI 需要通过 PSS 和 SSS 联合确定不同，窄带物联网的物理层小区 ID 仅仅需要通过 NSSS 确定（依然是 504 个唯一标识），这意味着 NSSS 的编码序列有 504 组。



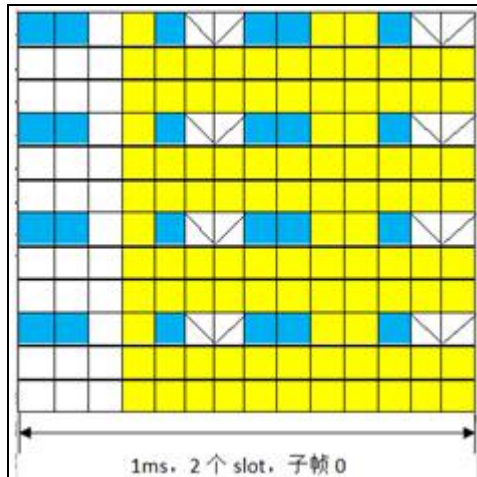
蓝色部分为 CRS 的位置，黄色部分为 NSSS 位置

通过 UE 角度看，NB-IoT 下行是半双工传输模式，子载波带宽间隔是固定的 15kHz，每一个 NB-IoT 载波只有一个资源块（resource block）。下行窄带参考信号被布置在每个时隙的最后两个 OFDM 符号中，每个下行窄带参考信号都对应一个天线端口，NB-IoT 天线端口是 1 个或者 2 个。物理层同样被分配了 504 个小区 ID，UE 需要确认 NB-IoT 的小区 ID 与 LTE 大网 PCI 是否一致，如果二者一致，那么对于同频的小区，UE 可以通过使用相同天线端口数的 LTE 大网小区的 CRS（小区级参考信号）来进行解调或者测量。UE 除了根据 NSSS（Narrowband Secondary Synchronization Signal）确定小区物理 ID 之外，还需要像 LTE 大网小区驻留流程一样，根据这两个同步信号进行下行同步，NPSS 的位置位于每个无线帧的第 6 子帧的前 11 个子载波，NSSS 的位置位于每个无线帧的第 10 子帧上的全部 12 个子载波。



NB-IoT 的下行物理信道与 LTE 大网的物理信道区别不大

NPBCH（Narrowband physical broadcast channel）以 64 个无线帧为循环，在 $\text{mod}64=0$ 的无线帧上的 0 号子帧进行传输，同样的内容在接下来连续的 7 个无线帧中的 0 号子帧进行重复传输，NPBCH 不可占用 0 号子帧的前三个 OFDM 符号，避免与 LTE 大网的 CRS 以及物理控制信道的碰撞。根据 3GPP 36.211 R13 定义，一个小区的 NPBCH 需要传输 1600 比特，采取 QPSK 调制，映射成 800 个调制符号，而每 8 个无线帧重复传输，64 个无线帧将这 800 个调制符号传完，意味着每 8 个无线帧重复传输 100 个调制符号，那么在这 8 个无线帧的每个 0 号子帧中需要传输这 100 个调制符号。这里进行一个简单的计算，一个 NB-IoT 子帧包含 $12 \times 7 \times 2 = 168$ 个 RE，扣掉前三个 OFDM 符号，再扣掉 NRS 占用的 RE，再扣掉 CRS 占用的 RE（假设为双端口发射），那么一共 $168 - 12 \times 3 - 4 \times 4 \times 2 = 100$ 个 RE，恰好对应 100 个 QPSK 调制符号，因此每个无线帧上的 0 号子帧恰好装满了 NPBCH 的符号。



蓝色部分为 CRS 的位置，斜线部分为 NRS 的位置，黄色部分为 NPBCH 位置

NPDCCH(Narrowband physical downlink control channel)，相比 LTE 下行较多的物理控制信道，NB-IoT 只有 NPDCCH 信道传递控制信息。窄带物理控制信道通过连续的一个或者聚合两个 NCCE (Narrowband control channel element) 的方式进行传输。一个 NCCE 占据 6 个连续的子载波，其中 NCCE0 占据 0~5 子载波，NCCE1 占据 6~11 子载波。每个 NPDCCH 是以 R 个连续的 NB-IoT 下行子载波进行重复传输的。NPDCCH 有三种。第一种是 Type1-NPDCCH 公共搜索空间，UE 通过检测该搜索空间获取寻呼消息。第二种是 Type2-NPDCCH 公共搜索空间，UE 通过检测该搜索空间获取随机接入响应消息（RAR）。第三种是 UE 专用 NPDCCH 搜索空间，UE 通过检测专属空间获取专属控制信息。

我们先计算一下 NPDCCH 的起始子帧位置，如果是 Type1-NPDCCH 公共搜寻空间模式，以 k0 为起始位置，这也是寻呼的起始位置。

这里有必要多说一下寻呼，寻呼消息是在寻呼帧（Paging Frame,PF）的寻呼子帧（Paging Occasion,PO）上发出的，因此 UE 需要周期性的监听这些位置。如果 $\text{defaultPagingCycle} = \text{rf256}, \text{NB} = \text{twoT}, \text{SFN mod T} = (\text{T div N}) * (\text{UE_ID mod N}), i_s = \text{floor}(\text{UE_ID}/\text{N}) \text{ mod } \text{Ns}, \text{UE_ID} = \text{IMSI mod } 4096 (\text{LTE UE_ID} = \text{IMSI mod } 1024)$,

例如 IMSI 为 460003313889448 经过计算 UE_ID 为 168，那么 PF 为 $\text{mod}256 = 168$ 的无线帧，PO 为 0 号子帧，那么 UE 就需要侦听无线帧为 168，子帧 0 上是否有 P-RNTI，并且以 256 无线帧为周期循环侦听 P-RNTI。

UE 还需侦听连续的 R-1 个子载波获得可靠的重复发送 NPDCCH，R 是根据 Rmax 和 DCI 子帧连续数共同决定。UE 如果没有把连续的 Rmax 通过获取小区系统消息块 SystemInformationBlockType2-NB 中的控制信息 radioResourceConfigCommon 中的参数 npdcch-NumRepetitionPaging 获取，该参数取值范围 { r1, r2, r4, r8, r16, r32, r64, r128, r256, r512, r1024, r2048 }

假设 Rmax 取值 64，DCI 子帧重复数取值为 3，查表（36.213 R13 表 16.6-2）可知对应 R 取值为 8，那么根据以上寻呼起始位置的计算，意味着 UE 需要周期侦听无线帧 $168 + 256n (n=0,1,2,3,...)$ ，子帧 0，同时连续重复 8 个子帧获取 NPDCCH 中的寻呼消息。值得一提的是，这里 DCI 子帧连续数并不是高层消息告知 UE 的，在这里 UE 采取盲检机制逐步尝试检测所有的 DCI 模式。如果没有检测到连续的控制信息，UE 会将已检测到的 NPDCCH 丢弃。从这点来看，NB-IoT 对于控制信道的解码可靠性还是极为看重的，要么不收，要么收全。

当然，在网络侧实际配置 NPDCCH 时需要与 NPBCH 的时隙错开，因此 UE 会尝试在非子帧 0 的其他子帧开始尝试检测 NPDCCH。NB-IoT 也可以采取多载波的方式进行数据传输，网络侧需要将 NPSS, NSSS, NPBCH 与 UE 专属 NPDCCH

分别配置在不同的载波。NPDCCH 在子帧中的起始位置 NPDCCHStart 取决于 SIB1-NB 里的 *eutraControlRegionSize* 参数设置,对于 Type2-NPDCCH 和 UE 专属 NPDCCH 的起始位置确定方式与 Type1 有所不同,具体细节可参考 36.213 R13 16.6。

NPDSCH(Narrowband physical downlink shared channel)

NB-IoT 对于 NPDSCH 的传输稳定性极为关注,通过重复传递同一 NPDSCH 的方式确保传输的质量,这也是 NB-IoT 宣称的强化覆盖技术手段之一。NPDSCH 可以承载 BCCH,例如承载系统消息,也可以承载一般的用户数据传输。对应这两种承载,传输信号加扰的方式有所不同。同时,子帧重复传输的模式也有所不同。

承载 NPDSCH 的子帧以及占位有一定规则,NPDSCH 的子帧不可以与 NPBCH,NPSS 或者 NSSS 的子帧复用。另外,承载子帧中 NRS 和 CRS 的位置既不作为 NPDSCH,也不作为符号匹配。

在收到传输 NPDCCH 以及 DCI 的最后一个子帧 n 后,UE 尝试在 $n+5$ 子帧为其实之后的 N 个连续下行子帧(不含承载系统消息的子帧)进行对应 NPDSCH 的解码。这 N 个连续下行子帧的取决于两个因素,也就是 $N = N_{\text{rep}} * N_{\text{SF}}$,一个是 N_{rep} ,意味着每一个 NPDSCH 子帧总共重复传输的次数, N_{SF} 意味着待传输数据需要占用的子帧数量,这两个因素都是根据对应的 DCI 解码得出的,在协议中可以查表得出对应关系(36.213 R13 16.4.1.3)。根据不同的 DCI(N_1, N_2)格式,需要注意的是在 UE 预期的 $n+5$ 子帧以及实际传输 NPDSCH 的起始子帧之间存在调度延迟,如果是 N_2 格式,该调度延迟为 0。如果是 N_1 格式,可以根据 DCI 的延迟指示 Idelay 和 NPDCCH 的最大重传 R_{max} 依据协议规定(36.213 R13 表 16.4.1-1)共同确定调度延迟。协议规定,UE 在 NPUSCH 上传数据之后的三个下行子帧之内不认为网络会传输 NPDSCH 数据,另外一种在物理层体现延迟传输 NPDSCH 的技术是设置 GAP,GAP 的长度由系统消息中的公共资源配置参数决定的,这也为多用户的数据错峰传输预留了空间。

DL-GapConfig-NB information element	
<pre>-- ASN1START { DL-GapConfig-NB-r13 ::= SEQUENCE { dl-GapThreshold-r13 ENUMERATED {n32, n64, n128, n256}, dl-GapPeriodicity-r13 ENUMERATED {sf64, sf128, sf256, sf512}, dl-GapDurationCoeff-r13 ENUMERATED {oneEighth, oneFourth, threeEighth, oneHalf}, } -- ASN1STOP</pre>	

NPDSCH 承载系统消息和承载非系统消息数据的物理层流程以及帧结构有所不同。承载非系统消息数据的 NPDSCH 每个子帧先重复发送,直到 $N = N_{\text{rep}} * N_{\text{SF}}$ 个子帧都传输完。而承载系统消息的 NPDSCH 先以 N_{SF} 个子帧传输完,在循环重复,直到 $N = N_{\text{rep}} * N_{\text{SF}}$ 个子帧都传输完。这两种传输方式占用资源的方式相似,之所以在重复传输机制上有所差异,可能主要还是考虑 UE 对于系统消息响应的及时程度。对于承载非系统消息数据的 NPDSCH 是通过对应 NPDCCH 加扰的 P-RNTI,临时 C-RNTI 或者 C-RNTI 进行解码的,同时 NPDSCH 持续占用的子帧情况也是通过解码 DCI 予以明确的。而与之不同的是,承载系统消息的 NPDSCH 起始无线帧以及重复传输占用子帧情况是通过解码小区 ID 和 MIB-NB 消息中的 *schedulingInfoSIB1* 参数获得的,当然这样承载系统消息的 NPDSCH 是通过 SI-RNTI 进行符号加扰的。SIB1-NB 是在子帧 4 进行传输的。对于在子帧内具体的起始位置则取决于组网方式,如果 NPDSCH 承载 SIB1-NB 并且是带内组网模式,则从第 4 个 OFDM 符号开始(避开前三个 OFDM 符号),其他组网模式从第一个 OFDM 符号(0 号 OFDM 符号)。如果 NPDSCH 承载其他信息,说明此时已经正确解码了 SIB1-NB,那么通过解读 SIB1-NB 中的 *eutraControlRegionSize* 参数(这是可选参数)来获取起始位置,如果该参数没有出现,那么从 0 号 OFDM 符号开始传输。

SystemInformationBlockType1-NB message			
-- ASN1START			
SystemInformationBlockType1-NB ::= SEQUENCE {			
hyperSFN-MSB-r13	BIT STRING (SIZE (8)),		
cellAccessRelatedInfo-r13	SEQUENCE {		
plmn-IdentityList-r13	PIMN-IdentityList-NB-r13,		
trackingAreaCode-r13	TrackingAreaCode,		
cellIdentity-r13	CellIdentity,		
cellBarred-r13	ENUMERATED {barred, notBarred},		
intraFreqReselection-r13	ENUMERATED {allowed, notAllowed},		
},			
cellSelectionInfo-r13	SEQUENCE {		
q-RxLevMin-r13	Q-RxLevMin,		
q-QualMin-r13	Q-QualMin-r9,		
},			
p-Max-r13	P-Max	OPTIONAL,	-- Need OP.
freqBandIndicator-r13	FreqBandIndicator-NB-r13,		
freqBandInfo-r13	NS-PmaxList-NB-r13	OPTIONAL,	-- Need OR.
multiBandInfoList-r13	MultiBandInfoList-NB-r13	OPTIONAL,	-- Need OR.
downlinkBitmap-r13	DL-Bitmap-NB-r13	OPTIONAL,	-- Need OR.
eutraControlRegionSize-r13	ENUMERATED {n1, n2, n3}	OPTIONAL,	-- Need OR.
maxCRS-Powerised-r13	ENUMERATED {dB-8, dB-10, dB-12, dB-14, dB-16, dB-18, dB-20, dB-22, dB-24, dB-26, dB-28, dB-30, dB-32, dB-34, dB-36, dB-38, dB-40, dB-42, dB-44, dB-46, dB-48, dB-50, dB-52, dB-54, dB-56, dB-58, dB-60, dB-62, dB-64, dB-66, dB-68, dB-70, dB-72, dB-74, dB-76, dB-78, dB-80, dB-82, dB-84, dB-86, dB-88, dB-90, dB-92, dB-94, dB-96, dB-98, dB-100}		
}			
-- ASN1END			

除了承载系统消息以及非系统消息(一般用户数据,寻呼信令等),NPDSCH还承载对上行信道NPUSCH的ACK/NACK消息,UE在NPUSCH传完子帧之后的第4个子帧进行侦听。

通过对于整个NB-IoT下行物理层结构以及流程的了解,NB-IoT利用了延迟以及重传帧结构设计保障了数据传输的稳定性以及可靠性,提升了“软性”的覆盖,同时也考量了与LTE大网的兼容共存,这可能是由于NB-IoT业务定位导向(设计NB-IoT的初衷据说就是为了查水表)进行调整设计的,这也再次指明了技术标准的一个发展方向,即不以技术本身的指标为考量,反而更多的以契合应用需求为准绳。

微信扫描下方二维码,免费加入【5G俱乐部】,还赠送整套:5G前沿、NB-IoT、4G+(VoLTE)资料。

