浙江大学
Zhejiang University

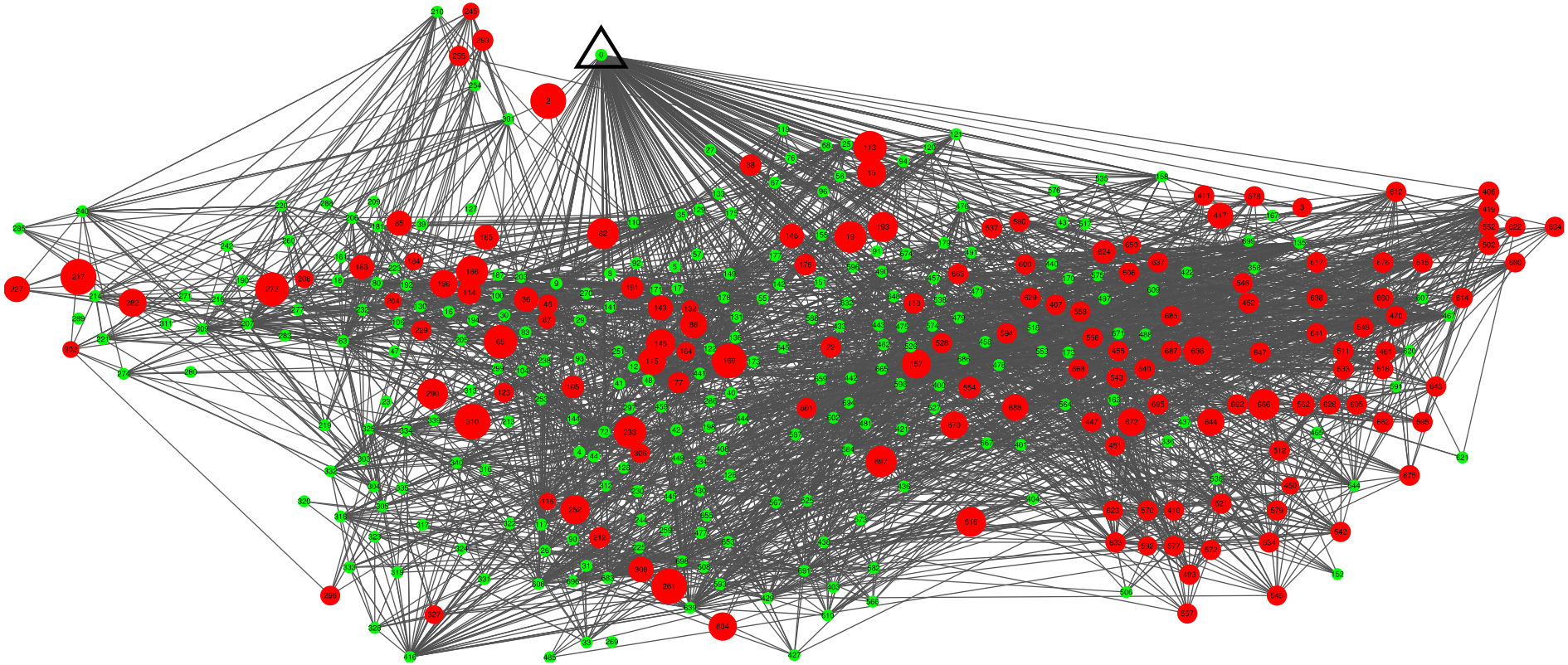
第六章 低功耗广域网

目录

- 概述
- 关键特性
- NB-IoT
- LoRaWAN
- 研究进展
- 总结和展望

概述

- 低功耗短距离通信：多跳→长距离传输
  - 问题：通信不可靠、能量黑洞（energy hole）
  - Ex：GreenOrbs、CitySee



概述

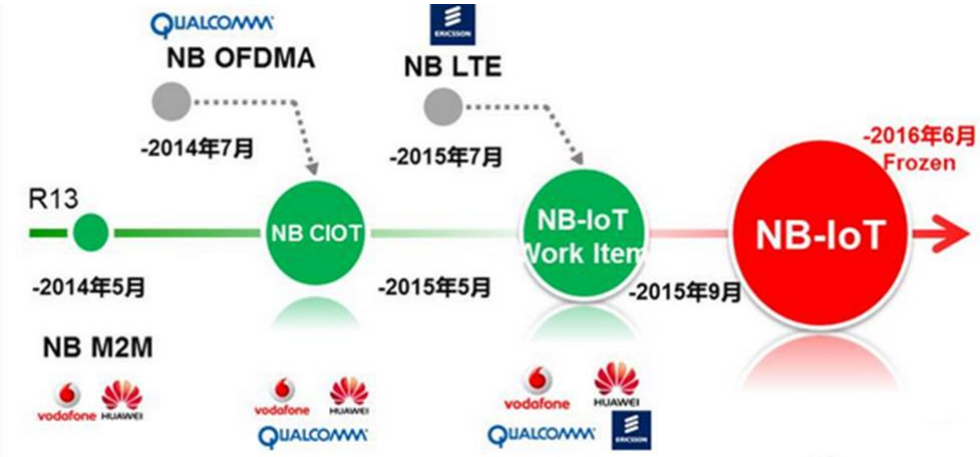
- **低功耗长距离通信/LPWAN：可以实现低功耗长距离通信**
  - e.g. NB-IoT单跳通信距离可以达到10km~30km
- **LPWAN近年来受到广泛关注**
  - 2017年6月19日工信部发布了《工业和信息化部办公厅关于全面推进移动物联网（NB-IoT）建设发展的通知》
  - 中国移动、电信、华为大力推进NB-IoT
  - 阿里联合联通大力推进LoRa, 2018年云栖大会阿里获得LoRa芯片IP授权。
- **LPWAN的典型技术**
  - NB-IoT: Narrow Band Internet of Things，窄带物联网技术
  - LoRaWAN： Long Range (LoRa), 长距离广域网技术
  - SigFox：法国的一家全球网络运行商建立的广域网通信公司
  - eMTC (LTE enhanced MTC): 3GPP对LTE裁剪后的低功耗广域网技术
  - EC-GSM：重用了GSM的广域网技术

物联网传输技术

5G、LTE-V等
LPWA（低功率广覆盖）
（eMTC）
Wi-Fi
UWB
Bluetooth
ZigBee
NB-IoT，LoRa，Sigfox
网络性能要求
覆盖要求
短距离
长距离
吴老司扯通信

NB-IoT

- NB-IoT (Narrowband Internet-of-Things)是由通信行业联盟和标准化组织3GPP 提出并标准化的低功耗广域网技术。



- 2014年，英国的沃达丰(vodafone) 和华为一起提出了NB-M2M解决方案。
- 2014年7月，美国的高通(qualcomm)基于现有的LTE技术提出了NB-OFDMA。最终融合成第一个满足低功耗、提升室内覆盖性能、支持大规模设备连接的统一方案，称为NB-CIoT。
- 2015年7月，以爱立信等公司为首的阵营基于现有的LTE技术，提出了NB-LTE。
- 接着，NB-CIoT和NB-LTE被融合起来，最终在2016年6月形成了现在的NB-IoT技术标准。

LoRaWAN

- 2013年8月，Semtech公司发布一款新型的基于chirp扩频调制的、用于进行超长距低功耗通信的芯片—LoRa（Long Range）
- 2015年3月， Semtech公司牵头成立了LoRa联盟，目前拥有超过290多家成员，中国有35名成员，如阿里巴巴和腾讯等。其他还包括各类运营商、系统、软件、芯片、模组、云服务和应用厂商等。
- 2015年6月，LoRa联盟提出了基于LoRa的MAC协议LoRaWAN v1.0。LoRaWAN定义了LoRa通信的网络架构，LoRa设备与基站之间通信的模式等。

LoRa Alliance™

SEMTECH

其他LPWAN

- **Sigfox**
  - Sigfox是由法国的一家全球网络运行商与2009年投资成立的一家公司。 Sigfox采用超窄带技术(100Hz)实现极低功率覆盖大范围区域的目的。目前，Sigfox 已经实现了全球30 多个国家的网络覆盖，但，国内目前还没有可用的Sigfox频段。
- **eMTC**
  - eMTC (enhanced Machine-Type Communication)由3GPP组织于2016年6月标准化。 3GPP对LTE协议进行裁剪，舍弃大部分复杂功能，形成适用于低功耗、低成本设备联网的新的eMTC协议。与NB-IoT/LoRa/Sigfox不同的是，eMTC 支持移动性和基站定位，可实现漫游和无缝切换，并且支持语音业务。
- **EC-GSM**
  - EC-GSM，即扩展覆盖GSM技术（Extended Coverage-GSM）。EC-GSM作为GSM向物联网演进的技术，重用了GSM物理层设计，与传统的GSM终端可以共载波部署。运用此技术，现有已布建的GSM（2G）基地台也能更适合用于物联网应用，相同的GSM基地台，可以拥有更大的通讯覆盖面积，支援更多GSM终端节点的通讯。

不同无线通信技术的比较

	WiFi	BLE	ZigBee Pro	SIGFOX	LoRa	LTE-M/(eMTC)	EC-GSM	NB-IOT	5G
覆盖范围	17-30+米	1-10+米	1-100+米	<12千米 160 dB	<10千米 157 dB	<10千米 156 dB	<15千米 164 dB	<15千米 164 dB	<12千米 160 dB
频谱带宽	2.4G 802.11	2.4G 802.15.1	2.4G 802.15.4	未授权 900MHz 100Hz	未授权 900MHz <500kHz	授权的 7-900MHz 1.4MHz共享频段	授权的 8-900MHz共享频段	授权的 7-900MHz 200kHz共享频段	授权的 7-900MHz共享频段
速率	150Mbps+	1Mbps	250kbps	<100bps	<10bps	<1Mbps	10kbps	<50kbps	<1Mpbs
终端价格	4$ (2016)	4$ (2016)	3$ (2016)	4$ (2016) 2.64$ (2020)	4$ (2016) 2.64$ (2020)	5$ (2015) 3.3$ (2020)	4.5$ (2015) 2.97$ (2020)	4$ (2015) 2-3$ (2020)	<2$
网络重构	无	无	无	大	大	小	中（LTE复用）	小-中	需5G网络

关键特性

- ①距离
- ②功耗
- ③数据率
- ④频段(授权与非授权)
- ⑤基站部署/公私网的使用
- ⑥非技术特征

关键特性：距离

- **增加距离的关键技术**
  - 增加功率谱密度(Power Spectrum Density)：单位频谱内的发送能量强度。密度越高，传输距离越远。
  - 重传：连续重传，融合多次接收的数据进行解码，提升解码率
  - 多接收天线：多根天线接收的信号融合，提升解码概率
  - 扩频：通过增加发送数据的冗余，提高数据解调的成功率
- **NB-IoT使用的技术**
  - 增加功率谱密度、重传、多接收天线
- **LoRa使用的技术**
  - 扩频

关键特性：距离

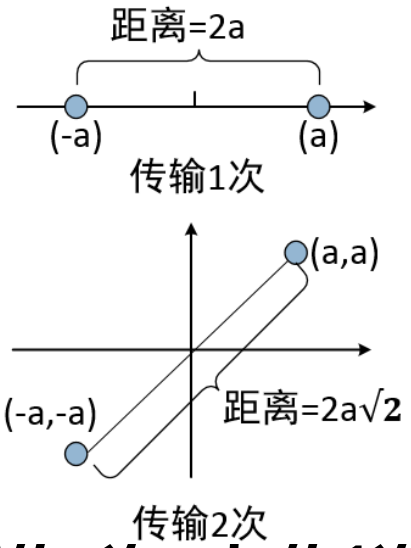
- NB-IoT提升距离的技术: 提升功率谱密度(7dB)
- PSD (Power Spectral Density) = 功率/带宽, PSD越大意味着单位频谱上发送信号强度增强

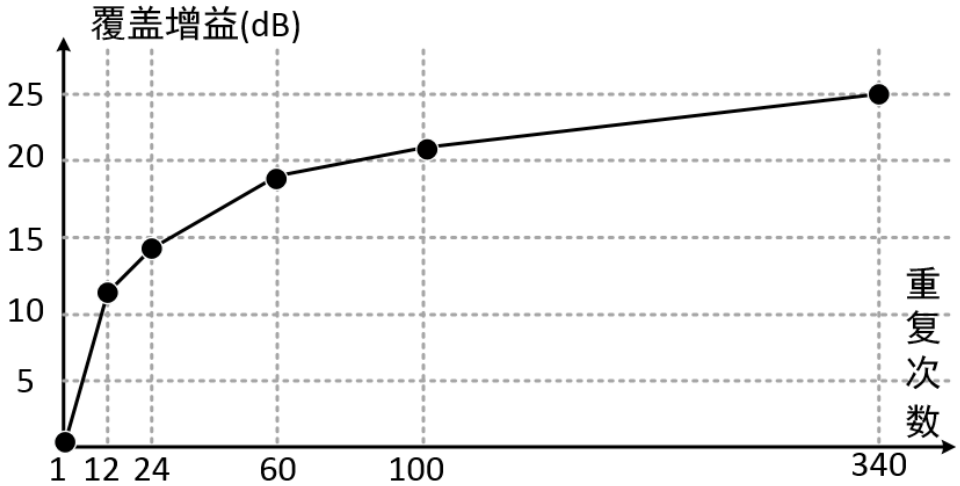
制式	功率(mW)	使用带宽(KHz)	功率谱密度=功率/带宽(mW/KHz)	比值(与GSM相比)
GSM	2000	200	10	--
NB-IoT	200	3.75	53.33	5.33
LoRaWAN	1000	125	8	0.8

- 因此，相比GSM, NB-IoT的PSD提升10*log(5.3)=7dB
- NB-IoT上行传输有两种带宽3.75KHz和15KHz可供选择，带宽越小，功率谱密度越大，覆盖增益越大。
- 相比GSM，LoRaWAN并没有在PSD上有增益，相反，有所下降-0.96dB。

关键特性：距离

- NB-IoT提升距离的技术: 重传（12dB）
- NB-IoT无论什么信道都可以重传，且上行最大128次，下行最大2048次。(信号强度增益)

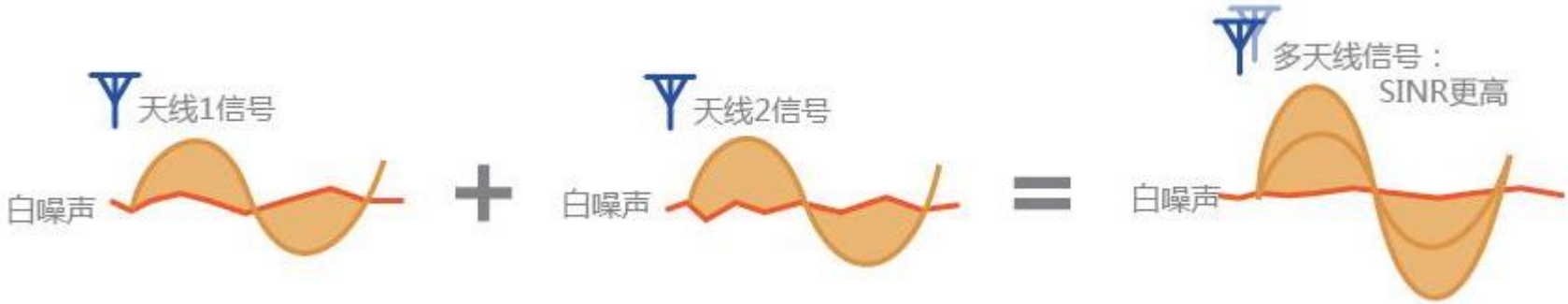




- 重传n次，与传1次相比，数据的欧式距离为$\sqrt{n}$倍；相当于传输1次，功率增加n倍
- 重传次数越多，速率降低，误码率降低。上行最大重复传输128次，理论上增益20dB，10*lg(N), N是次数
  - 而考虑到传输效率，系统不会无限制重传，一般取重传增益为12dB。

关键特性：距离

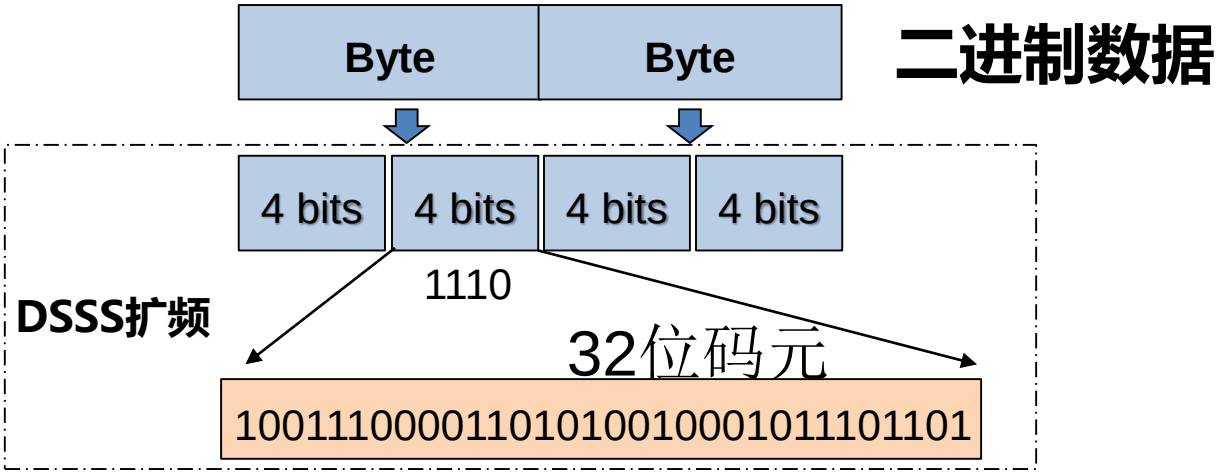
- **NB-IoT提升距离的技术: 多天线增益（0~3dB)**
- **多天线增益 (0-3db)**
  - 多根接收天线会融合接收到的信号，增强SINR
  - 理论来说，1TmR的系统比1T1R有10lg(m)的增益
  - 1T: 1 根发送天线; mR: m 根接收天线



  - 对于1T2R来说，比1T1R会有3db的增益

关键特性：距离

- LoRa提升距离的技术：CSS扩频
- 常见的扩频技术：DSSS(直接序列扩频), CSS (线性频变扩频)， FHSS (跳频扩频),。
- 扩频技术本质上是通过增加数据/带宽冗余性来提升抗干扰的能力，增加解码成功率
  - 比如，下述的DSSS扩频技术中，4 bits数据扩频成了32位的码元。



关键特性：功耗

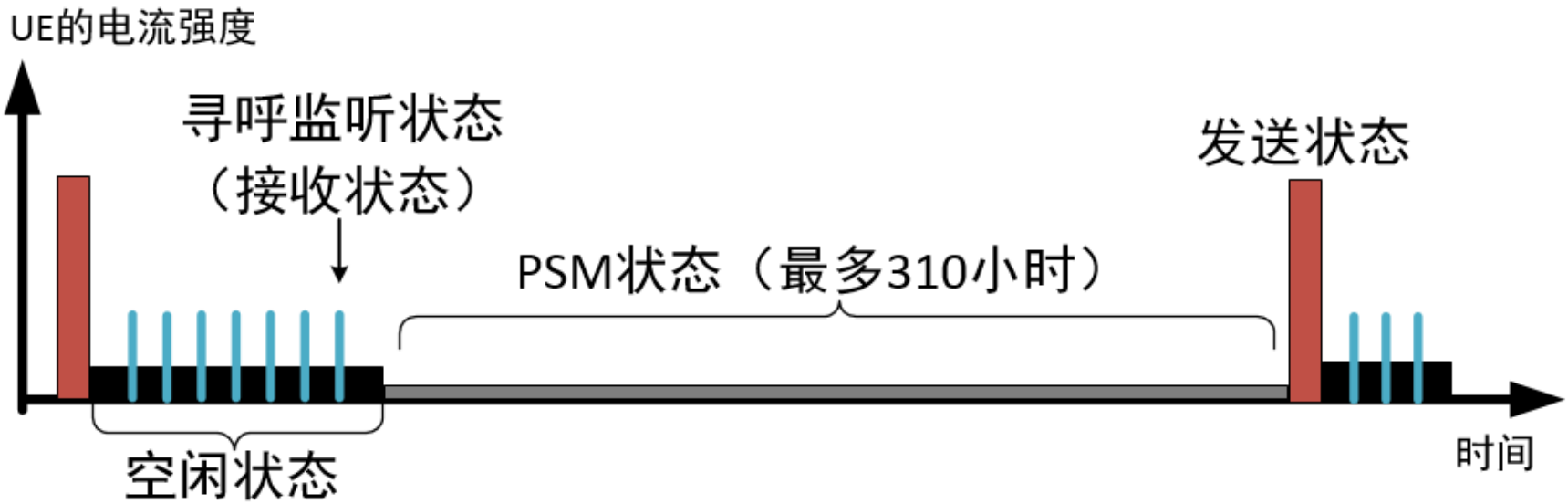
- 常见的状态：发送、接收、空闲、睡眠
- 较低的芯片功耗状态

	发送状态	空闲状态	接收状态	睡眠状态
NB-IoT (3.6V)	+26dBm,120mA	1mA	50mA	5uA
LoRa (3.3V)	+20dBm, 120mA	1.5uA	12mA	0.2uA

- NB-IoT低功耗技术
  - PSM(Power Saving Mode)机制
  - eDRX(extended Discontinuous Reception)增长的休眠时间
- LoRaWAN低功耗技术
  - 低功耗MAC协议(Class A, Class B)

关键特性：功耗

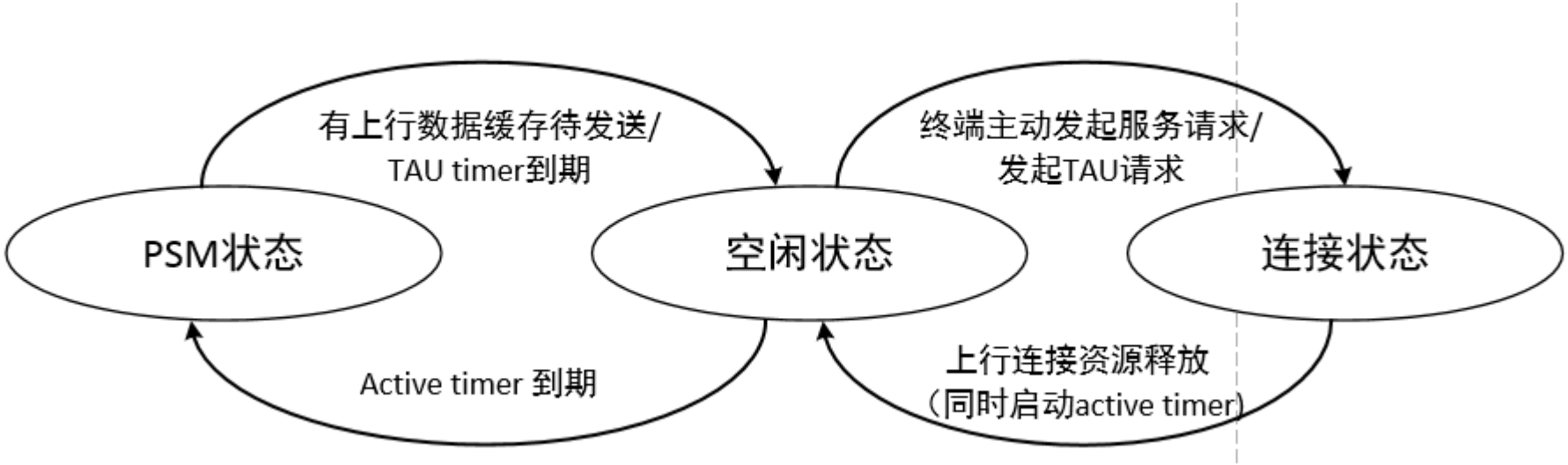
- **NB-IoT的PSM机制下，不同的功耗状态**
  - PSM: 深度睡眠状态，UE 不能被访问
  - Idle: 做周期性寻呼，UE 可以被访问，因为UE会周期性醒来



- **红色：发送状态 (120mA)**
- **黑色：空闲状态 (1mA)**
- **灰色：PSM状态 (睡眠状态)(5uA)**
- **蓝色：接收状态 (50mA)**

关键特性：功耗

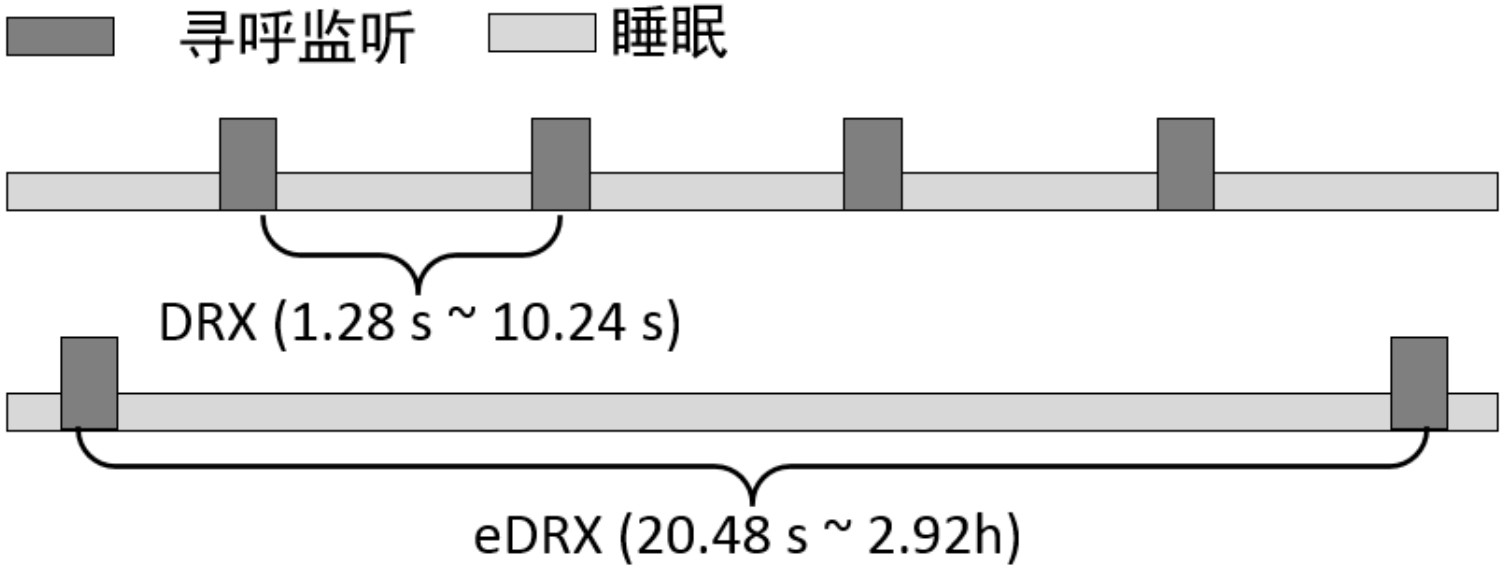
- NB-IoT低功耗技术: PSM 机制状态转换



- TAU: (Tracking Area Update), 当终端在空闲状态的寻呼监听中发现自己所处的tracking area id 变化时，需要向基站发起TAU请求，更新自己的相关参数。

关键特性：功耗

- **NB-IoT低功耗技术: eDRX机制**
  - DRX = Discontinuous Reception，非连续接收
  - eDRX = extended Discontinuous Reception，增长的非连续接收
- **DRX默认寻呼周期为1.28s, eDRX最长可达2.91h (在下行数据发送频率小时，通过核心网和终端的协商配合，终端跳过大部分的寻呼监听，从而达到省电的目的)**

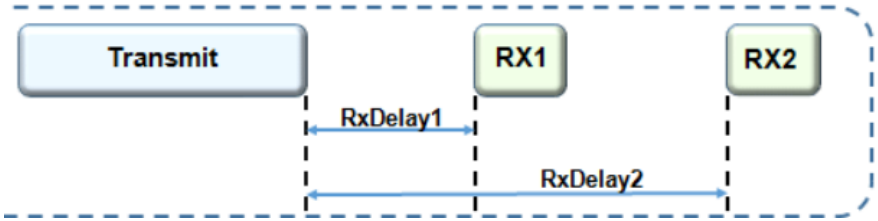


关键特性：功耗

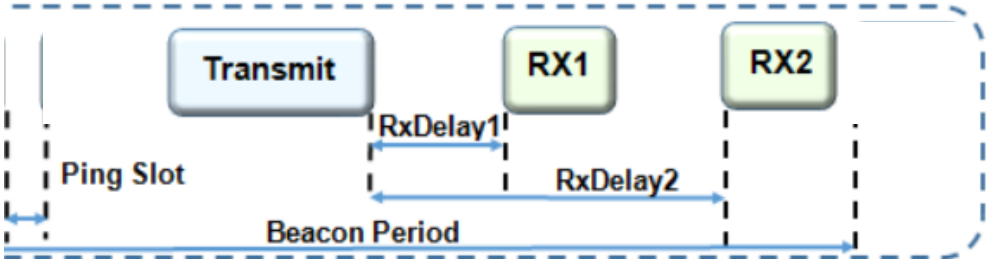
- **NB-IoT低功耗技术总结**
  - PSM和eDRX都是属于3GPP协议中的技术，且对NB-IoT和eMTC都是适用的
  - 相比较而言，eDRX（1mA, idle）的省电效果差些，但是实时性好些。(PSM, 5uA)
  - 如果终端每天发送一次200byte报文，5瓦时电池寿命可达12.8年。
  - 中国移动目前的策略是PSM为必选项，目前现网试点中主要开的也是PSM技术。

关键特性：功耗

- **LoRaWAN低功耗技术: 低功耗MAC技术**
- **LoRaWAN定义了三种MAC类型（A,B,C），其中两种是低功耗的MAC（A，B）。**
  - Class A,设备初始化. (RxDelay1>= 1s, RxDelay2=1s)
  - Class B,设备初始化 + 监听窗口
  - 严格的duty cycle控制。
    - *比如，**LoRa**设备在某个信道上发送了**0.5s** 长的数据后，由于该信道**1%** **duty cycle**的限制，该设备需等待**49.5s**之后才能再次在同一个信道上发送数据。*



Class A



Class B

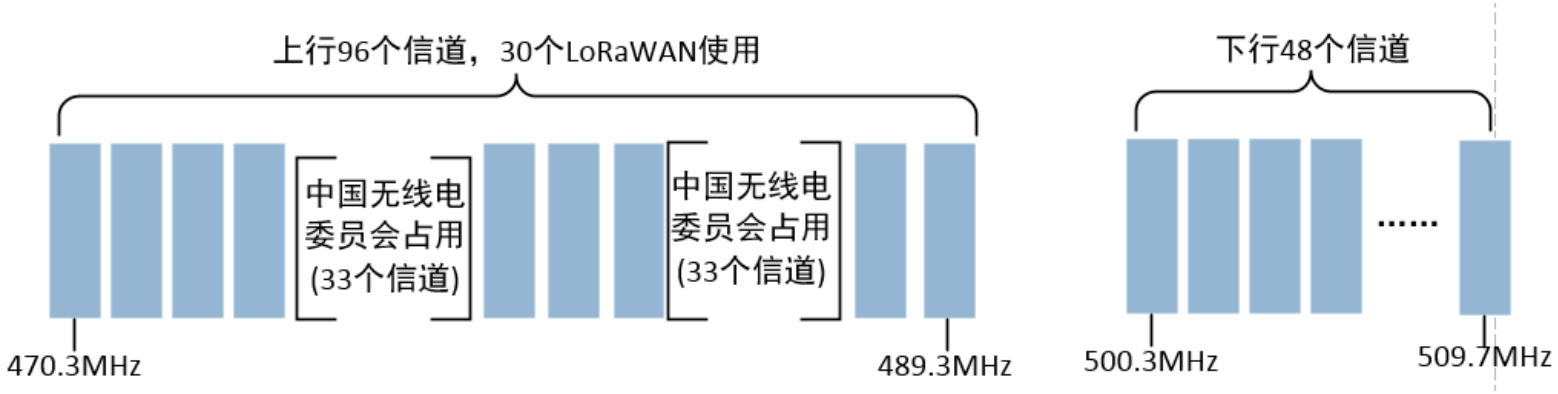
关键特性：数据率

- **LPWAN适用于数据传输延迟不敏感的场景，比如远程抄表，因此，其数据率相对较低。**
- **最大的物理层数据率**
  - NB-IoT上行为250kbps, 下行为 226.7kbps.
  - LoRaWAN上行和下行一样，最高为5.470kbps.
- **实际的数据率将会更低**
  - NB-IoT规定了上行和下行的不同类型数据包的传输间隔，比如，基站发完下行数据后， UE至少间隔12ms 才能反馈ACK/NACK数据包。因此，实际使用中，NB-IoT的下行一般为27.2kbps，而上行一般为62.5kbps.
  - LoRaWAN 对信道使用有严格的duty cycle限制。比如, 1%的信道使用限制，数据率则只能达到54.7bps。

关键特性：频段分配与使用

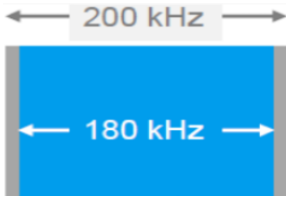
❑ LPWAN中使用了不同的频段类型 (授权与非授权)

- LoRaWAN使用了非授权频段, 分配给中国的是上行470.3 ~ 489.3MHz (共30个)，下行500.3~509.7MHz(48个信道), 带宽都是200kHz。实际上LoRa调制使用的带宽为125kHz，存在保护带



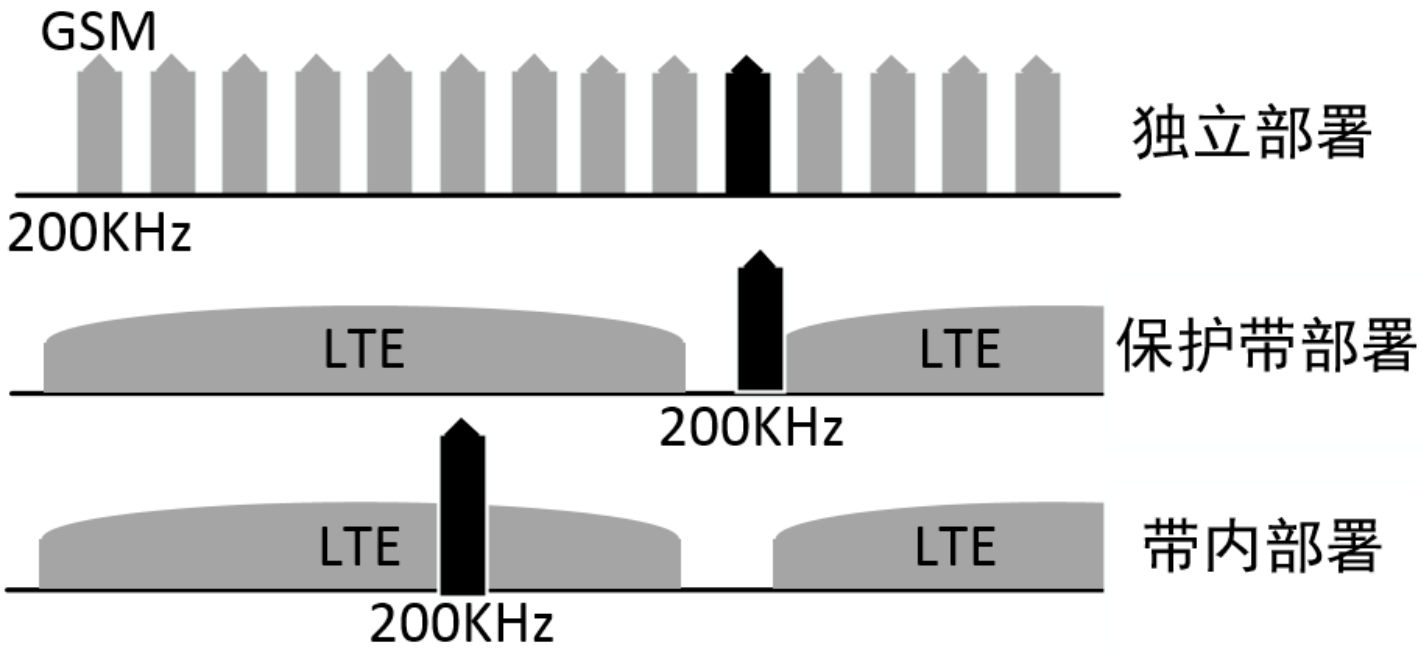
- NB-IoT使用了授权频段
  - *系统带宽为200kHz, 传输带宽为180kHz*

上行频率	下行频率	运营商
824~849MHz	869~894MHz	中国电信
880~915MHz	925~960MHz	中国移动、中国联通



关键特性：频段分配与使用

- **独立部署**：利用LTE系统未使用的200kHz带宽。通常使用GSM的一个信道（一个GSM带宽200KHz>NB-IoT 180KHz带宽)
- **保护带部署**：利用LTE边缘保护频带中未使用的180KHz带宽的资源块
- **带内部署**：占用LTE的1个PRB资源



关键特性：频段分配与使用

- **带内部署**：占用LTE的1个PRB，但不是任意PRB (续).
  - LTE 1.4M带宽下是不支持IB方式的NB的
  - 无论哪种带宽采用IB部署方式，都避开了中间的6个PRB，因为这里是要用来传LTE的同步信号和MIB消息的
  - 除了中间的6个PRB不能部署NB-IoT外，还有一些PRB也是不能用来部署NB-IoT的，e.g. NB的中心频点要满足100kHz 的栅格。因为UE仅在100kHz的倍数点上搜索

Table 3-1: Allowed LTE PRB indices for cell connection in NB-IoT in-band operation

LTE system bandwidth	3 MHz	5 MHz	10 MHz	15 MHz	20 MHz
LTE PRB indices for NB-IoT synchronization	2, 12	2, 7, 17, 22	4, 9, 14, 19, 30, 35, 40, 45	2, 7, 12, 17, 22, 27, 32, 42, 47, 52, 57, 62, 67, 72	4, 9, 14, 19, 24, 29, 34, 39, 44, 55, 60, 65, 70, 75, 80, 85, 90, 95

关键特性：基站部署/公私网的使用

- **低功耗广域网的网络架构**

- **基站**
  - 是LPWAN终端实现互通互联的关键
  - 应用层服务访问、控制LPWAN终端的关键
- **基站部署/公私网使用方式**
  - 复用现有的基站，比如LTE基站。部署效率高；但覆盖深度存在盲区，比如楼宇地下室。
  - 自行搭建基站。部署效率低，可以根据需求做到深度覆盖。
- **NB-IoT的部署方式：复用LTE基站，使用公网**
- **LoRaWAN的部署方式：自行搭建基站，使用私网**

关键特性：非技术特性

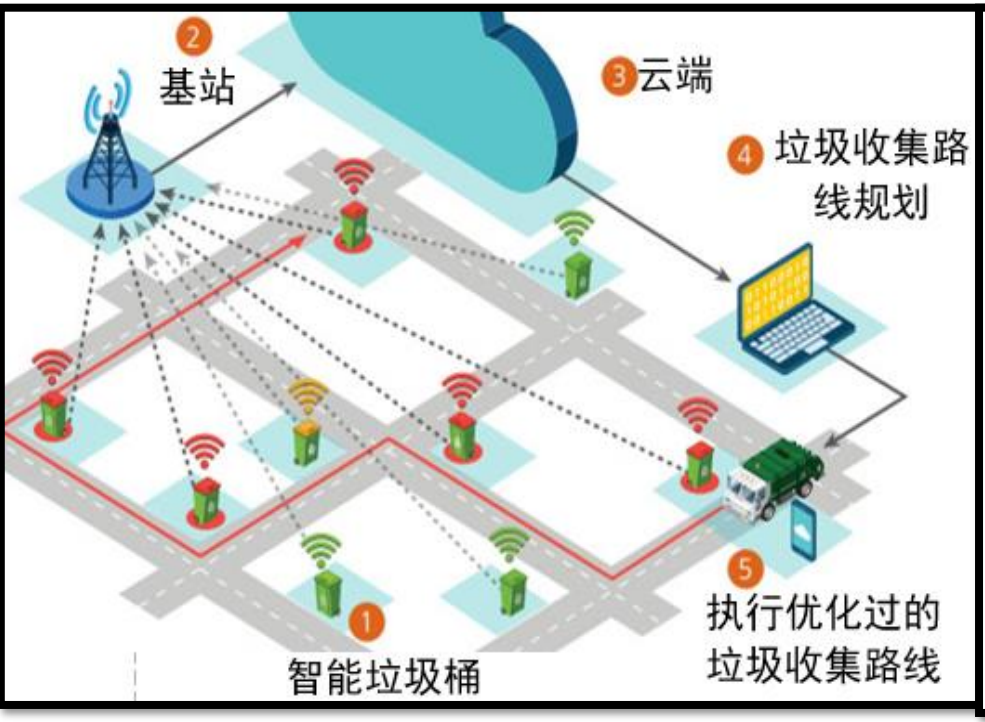
- **协议开放性**
  - NB-IoT闭源，LoRaWAN开源。
    - ***LoRaWAN 代码: https://github.com/Lora-net***
  - LoRaWAN比NB-IoT更受到研究界的喜爱，LoRa相关的论文频现计算机网络顶级会议
  - [sigcomm 18]PLoRa, [sigcomm 17]Chior, …
- **生态支持**
  - NB-IoT受到中国移动、联通和电信的支持。基于已有的基站部署节点的通信，市场规模容易推动和提升。
  - LoRaWAN由LoRa 联盟支持。目前拥有超过290多家成员，中国有35名成员，如阿里巴巴和腾讯等。2018年9月19日的云栖大会上，阿里利用飞在天上的飞艇和地下基站共同搭建了一个”阿里云天空物联网”

NB-IoT

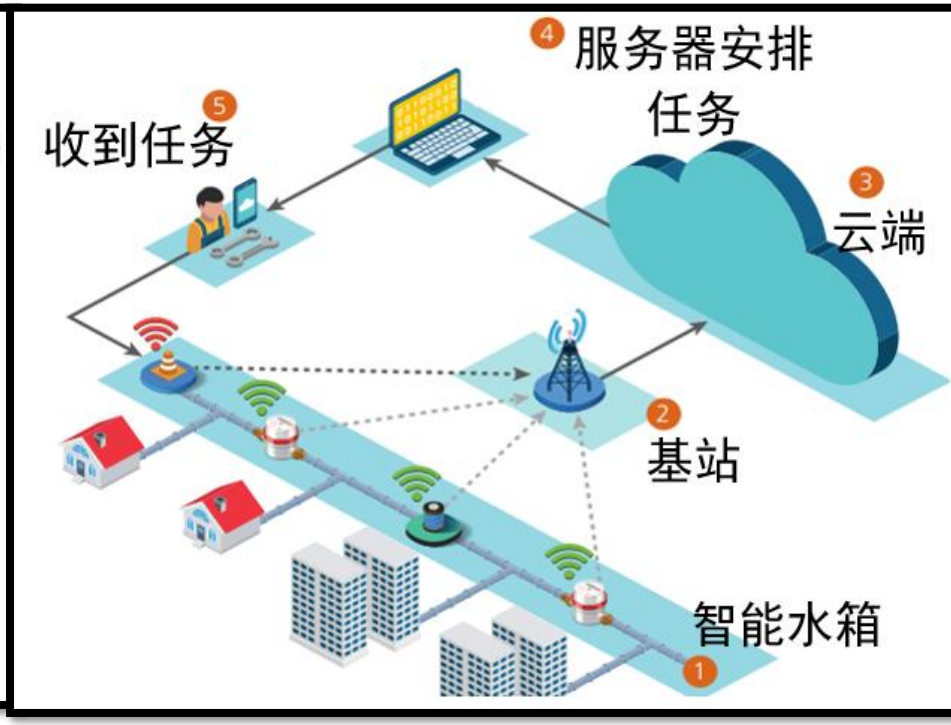
- **使用场景**
- **网络架构**
- **物理层**
  - 下行：资源分配、物理信号、物理信道
  - 上行：资源分配、物理信号、物理信道
  - 多址接入方式：OFDM（下行）、SC-FDMA（上行）
- **链路层**
  - 编码技术：TBCC（下行）、Turbo code（上行）
  - 随机接入过程
- **网络层和传输层**
  - IP及UDP
- **应用案例**

NB-IoT使用场景

- **广域物联网通信的典型场景**
  - 城市范围的垃圾桶使用状态检测
  - 城市范围的家庭水箱泄露检测



城市范围智能垃圾桶

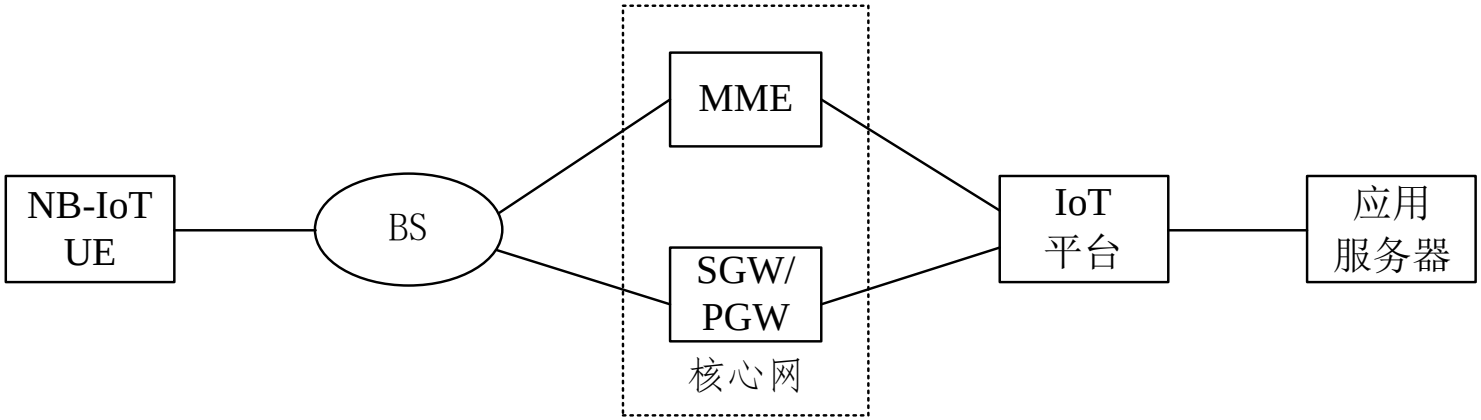


城市范围智能水箱

NB-IoT网络架构

整体网络架构

- NB-IoT UE：用户终端设备（User Equipment）
- BS：基站（Base Station），主要承担UE接入处理，小区管理等功能。
- IoT核心网：承担UE移动性、安全性等管理功能，并将IoT业务相关数据转发到IoT平台进行处理。
  - 移动性管理实体 (MME):NB-IoT协议参数设置（如协商包头压缩算法）和连接切换管理
  - 服务网关SGW：转发速率控制
  - 分组数据网关PGW：数据转发



NB-IoT网络架构

整体网络架构

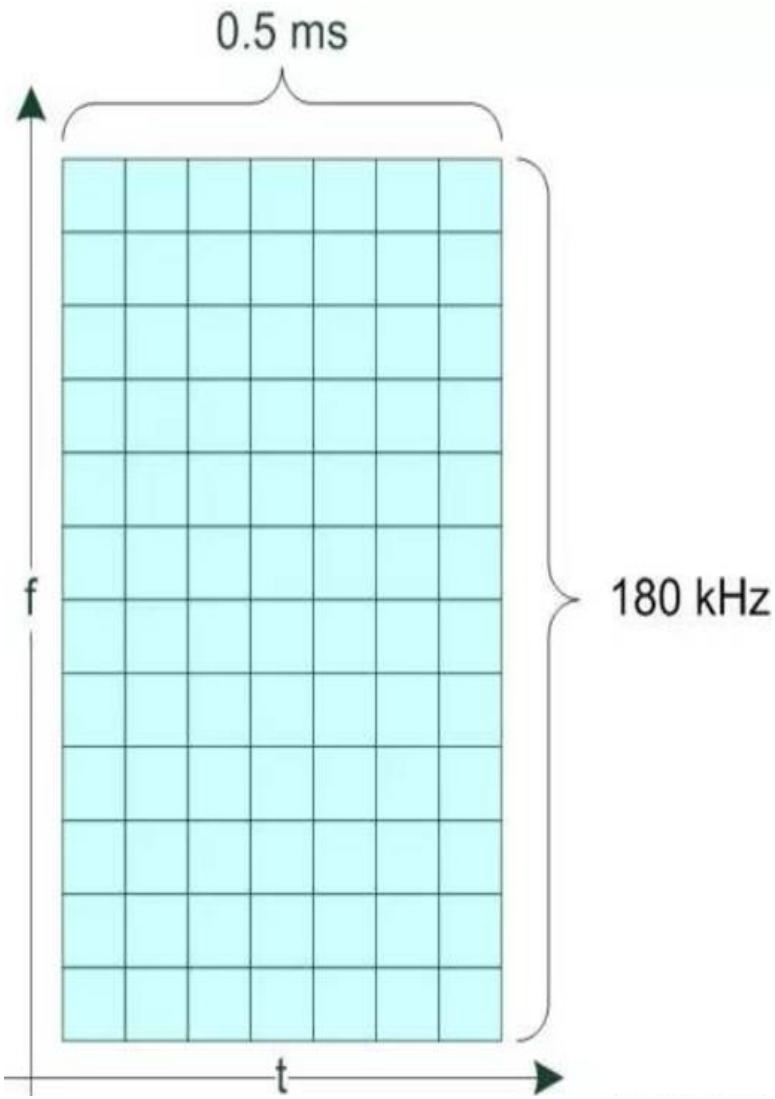
- IoT平台：汇聚从各种接入网得到的IoT数据，并根据不同类型转发至相应的业务应用器进行处理。
- 应用服务器：是IoT数据的最终汇聚点，根据客户的需求进行数据处理等操作。
- 说明：
  - NB-IoT UE通过无线空口连接到BS，而BS通常可以通过有线方式连接到核心网。当前很多工作主要分析和优化NB-IoT无线空口协议（如：空口承载、空口数据交互及空口加密等）的性能
  - NB-IoT基站可以依赖现有移动运营商的基站（GSM或LTE）进行部署（节省建网成本），也可以独立建站部署。
  - 用户NB-IoT应用服务器主要由移动通信运营商及相关协议制定方提供（如华为）

NB-IoT物理层：概述

- **下行**
  - 带宽180kHz, 分为12个子载波（subcarrier），子载波间隔（subcarrier spacing）为15kHz
  - OFDM技术
- **上行**
  - 带宽180kHz, 子载波间隔有15kHz, 3.75kHz两种
  - SC-FDMA技术

NB-IoT物理层：下行资源分配

- **频域上**：下行带宽180kHz, 分为12个子载波（subcarrier），子载波间隔（subcarrier spacing）为15kHz
- **时域上**：一个时隙（slot）长度为0.5ms，每个时隙中有7个符号（symbol）
- **子帧**(subframe): 基本调度单元, = 1 ms/2 slots (每个slot上有7个symbol)
- **无线帧**(frame) 包含 10个子帧 = 10ms



NB-IoT物理层：下行

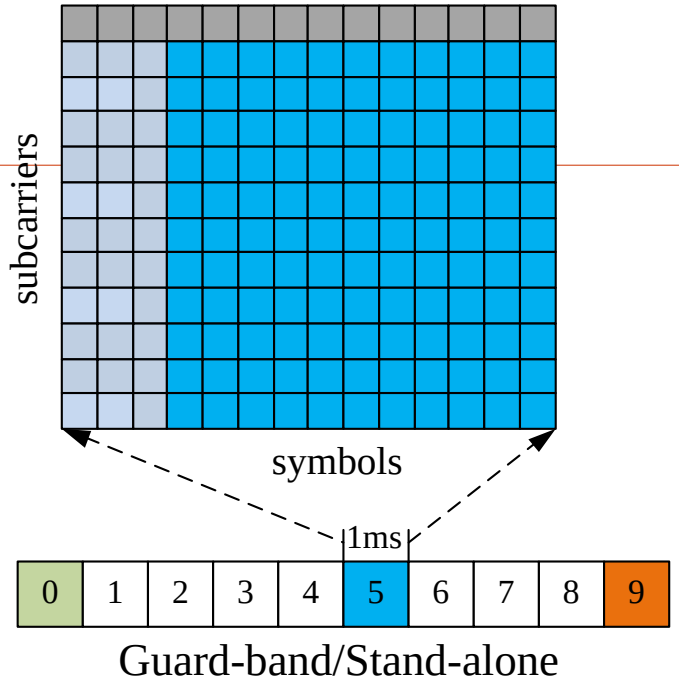
- **下行物理信号：**
  - NPSS: Narrowband Primary Sync Signal
  - NSSS: Narrowband Secondary Sync Signal
  - NRS: Narrowband Reference Signal
- **下行物理信道：**
  - NPBCH: Narrowband PHY Broadcast Channel
  - NPDCCH: Narrowband PHY DL Control Channel
  - NPDSCH: Narrowband PHY DL Shared Channel
- 注：信号不承载具体的信息bit，而信道是要传data的

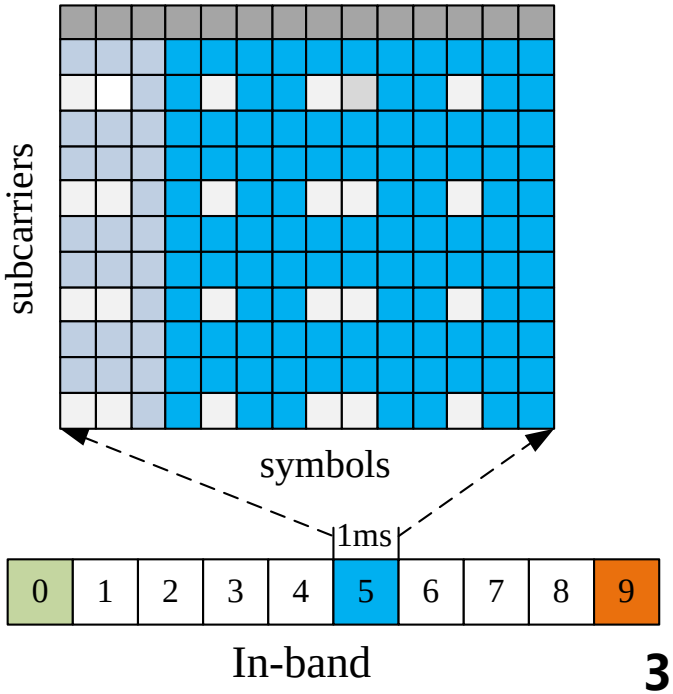
	Subframe number									
	0	1	2	3	4	5	6	7	8	9
Even frame	NPBCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPSS	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NSSS

	Subframe number									
	0	1	2	3	4	5	6	7	8	9
Odd frame	NPBCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPSS	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH	NPDCCH or NPDSCH

下行

- **下行同步信号 (NPSS & NSSS)**
  - 用于NB-IoT UE进行小区搜索(cell search)
    - ***Time and Frequency Synchronization***
    - ***Cell ID detection***
- **NPSS**
  - 用于完成时间和频域的同步
  - 在每个无线帧的子帧#5上发送，发送周期10ms
  - 具体而言，NPSS使用最后11个OFDM符号，每个符号占据(lowest)11个subcarrier; 在IB部署下，避免与LTE CRS碰撞，避开相应RE





下行: ZC序列

- **Zadoff-Chu Sequence:**

$$x_q(m) = e^{-j\frac{\pi q m(m+1)}{N_{ZC}^{RS}}}, \quad 0 \le m \le N_{ZC}^{RS} - 1$$

- **N_{zc}：**代表需要生成的序列的长度，=11 in NB-IoT. 因为是复数，所以有I、Q两个支路。
- **q：**代表是根序列的索引，生成任何一个ZC序列都需要赋一个根索引值, =5 in NB-IoT。
- **m：**代表这个序列的第m+1个(m从0开始)序列点的值。

ZC序列特点

- **ZC序列能在4G及以后的系统中大量使用，是因为这个序列具有很多好的特性：**
  - This sequence has a constant amplitude. 如果在坐标轴上表示就是分布在一个圆上，幅度就是圆的半径，是恒定不变的（这里即为1）。
  - Zero Autocorrelation. 也就是说这个序列的自相关性很好。如果你用这个公式生成了一个序列，同时进行循环移位N位 (N can be 1,2,....,size of sequence -1). 如果你对原序列和循环移位N后的序列进行相关性运算，你会惊奇的发现结果几乎是0.

相关性

- 两个实数序列$\{a_k\}$ ($|a_k|=1, k=0,\ldots,n$)，$\{b_k\}$($|b_k|=1, k=0,\ldots,n$)的相关性:

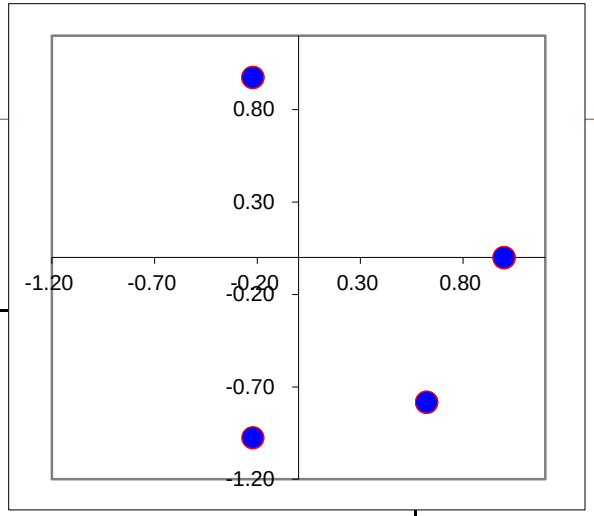
$$\frac{1}{n}\cdot\sum_{k=0}^{n}(a_k\cdot b_k)$$

- 两个复数序列$\{a_k+b_ki\}$($|a_k+b_ki|=1, k=0,\ldots,n$), $\{c_k+d_ki\}$($|c_k+d_ki|=1, k=0,\ldots,n$)的相关性:

$$\frac{1}{n}\cdot\sum_{k=0}^{n}(a_k+b_ki)\cdot(c_k-d_ki)=$$

$$\frac{1}{n}\sum_{k=0}^{n}[(a_kc_k+b_kd_k)+(b_kc_k-akdk)i]$$

- 说明：-1代表完全负相关，1代表完全正相关，0代表不相关



	q	5				
	N_{ZC}^{RS}	7				
	Shift	1				
根序列			循环移位数			
m	Zadoff(I)	Zadoff(Q)	Zadoff(I)-s1	Zadoff(Q)-s1	Corr-(I)	Corr-(Q)
0	1.00	0.00	1.00	0.00	1.00	0.00
1	-0.22	0.97	1.00	0.00	-0.22	0.97
2	0.62	-0.78	-0.22	0.97	-0.90	-0.43
3	-0.22	-0.97	0.62	-0.78	0.62	-0.78
4	0.62	-0.78	-0.22	-0.97	0.62	0.78
5	-0.22	0.97	0.62	-0.78	-0.90	0.43
6	1.00	0.00	-0.22	0.97	-0.22	-0.97
					0.00	0.00

序列长度

$$x_q(m) = e^{-j\frac{\pi q m(m+1)}{N_{ZC}^{RS}}},$$

生成公式

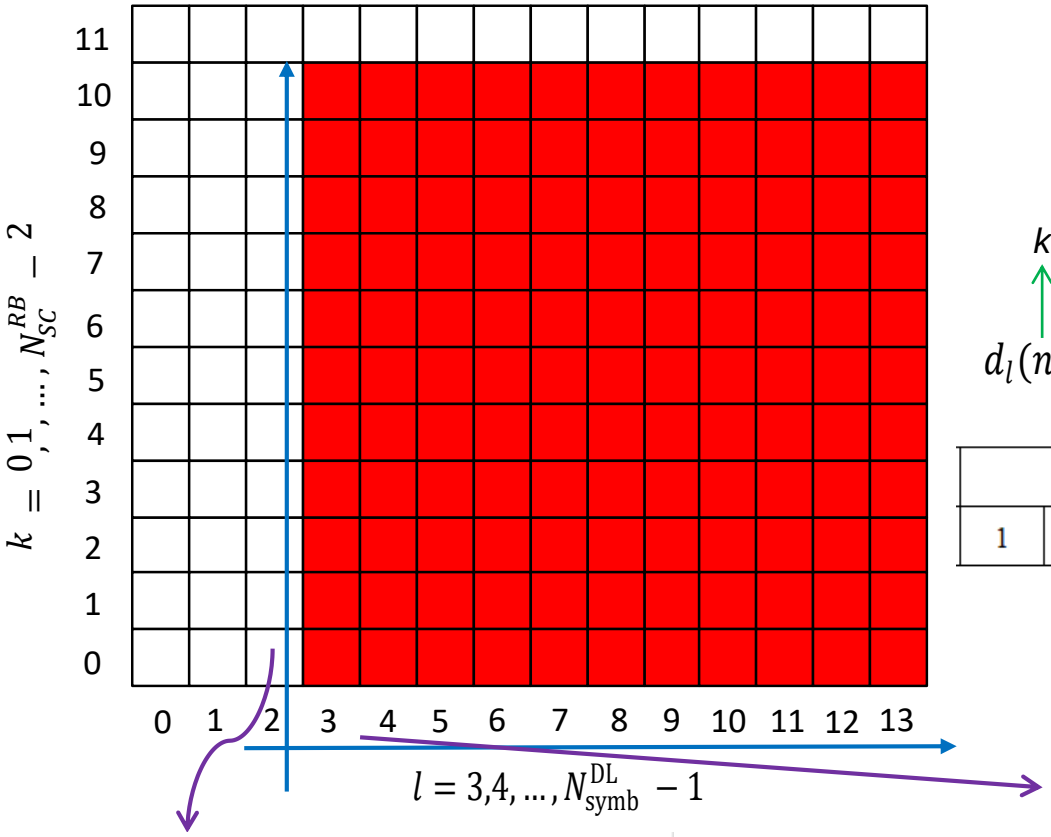
生成的I、Q支路序列值

思考题

- 在上述案例中，我们考虑ZC序列移位3位，计算序列第2（m=1）和第5个(m=4)位置上I、Q两路的相关性?

	q	5	$x_q(m) = e^{-j\frac{\pi q m(m+1)}{N_{ZC}^{RS}}}$, $0 \le m \le N_{ZC}^{RS} - 1$			
	N_{ZC}^{RS}	7				
	Shift	3				
m	Zadoff(I)	Zadoff(Q)	Zadoff(I)-s3	Zadoff(Q)-s3	Corr-(I)	Corr-(Q)
0	1.00	0.00				
1	-0.22	0.97				
2	0.62	-0.78				
3	-0.22	-0.97				
4	0.62	-0.78				
5	-0.22	0.97				
6	1.00	0.00				

NPSS资源映射



$$d_l(n) = s(l) \cdot e^{-j\frac{\pi u n\,(n+1)}{11}}, \quad n = 0,1,\ldots,10$$

k

S(3),…,S(13)										
1	1	1	1	-1	-1	1	1	1	-1	1

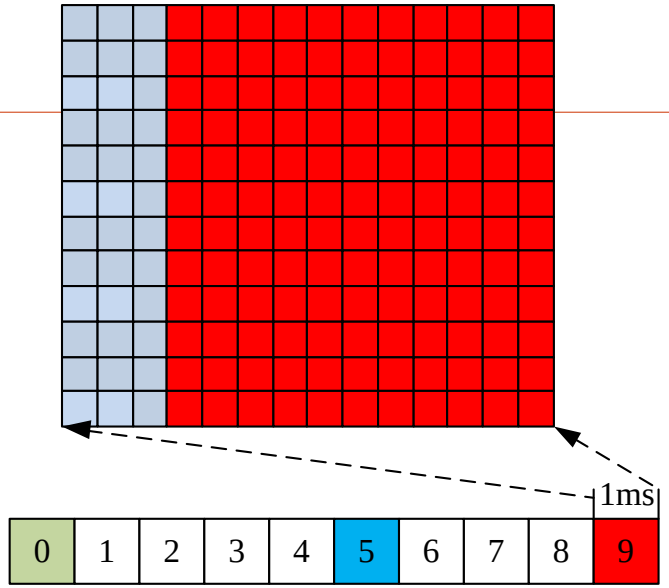
先频域映射，从子载波 0 开始到 10, 11 个序列值正好一一对应

再从第 3 个 symbol 开始时域映射，重复到最后一个符号

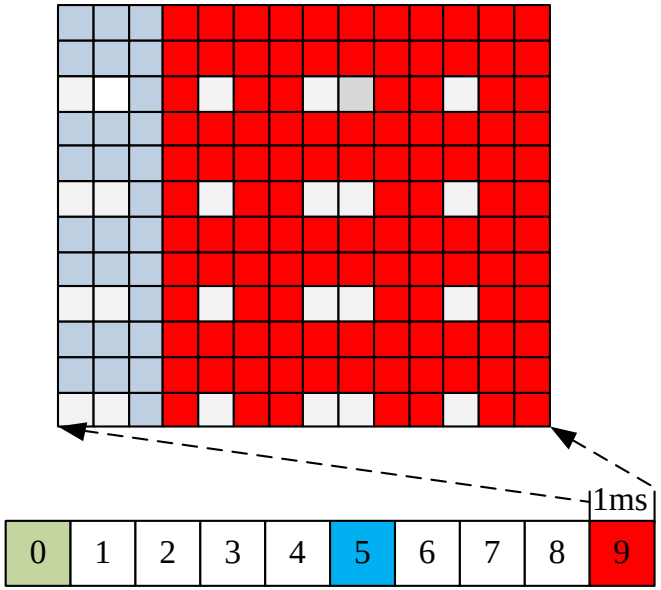
- 每个symbol上承载一个ZC序列 (n/k为纵轴)
- 不同symbol上承载不同的掩码S(l) (l为横轴)

下行

- **NSSS**
  - 携带小区ID(PCI)信息，范围0-503; 80ms内帧定时信息
  - 在偶数无线帧的子帧#9上发送，发送周期20ms
  - NSSS占用11(symbol)*12(subcarrier)=132 RE
  - 先生成131的根底序列, 再循环拓展成132长ZC序列，正好填充132 RE



Guard-band/Stand-alone

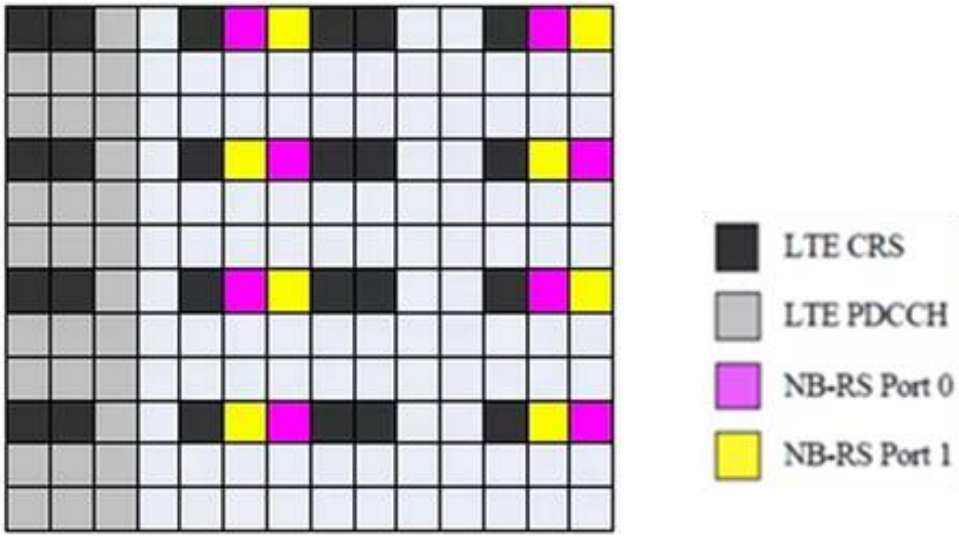


In-band

NRS

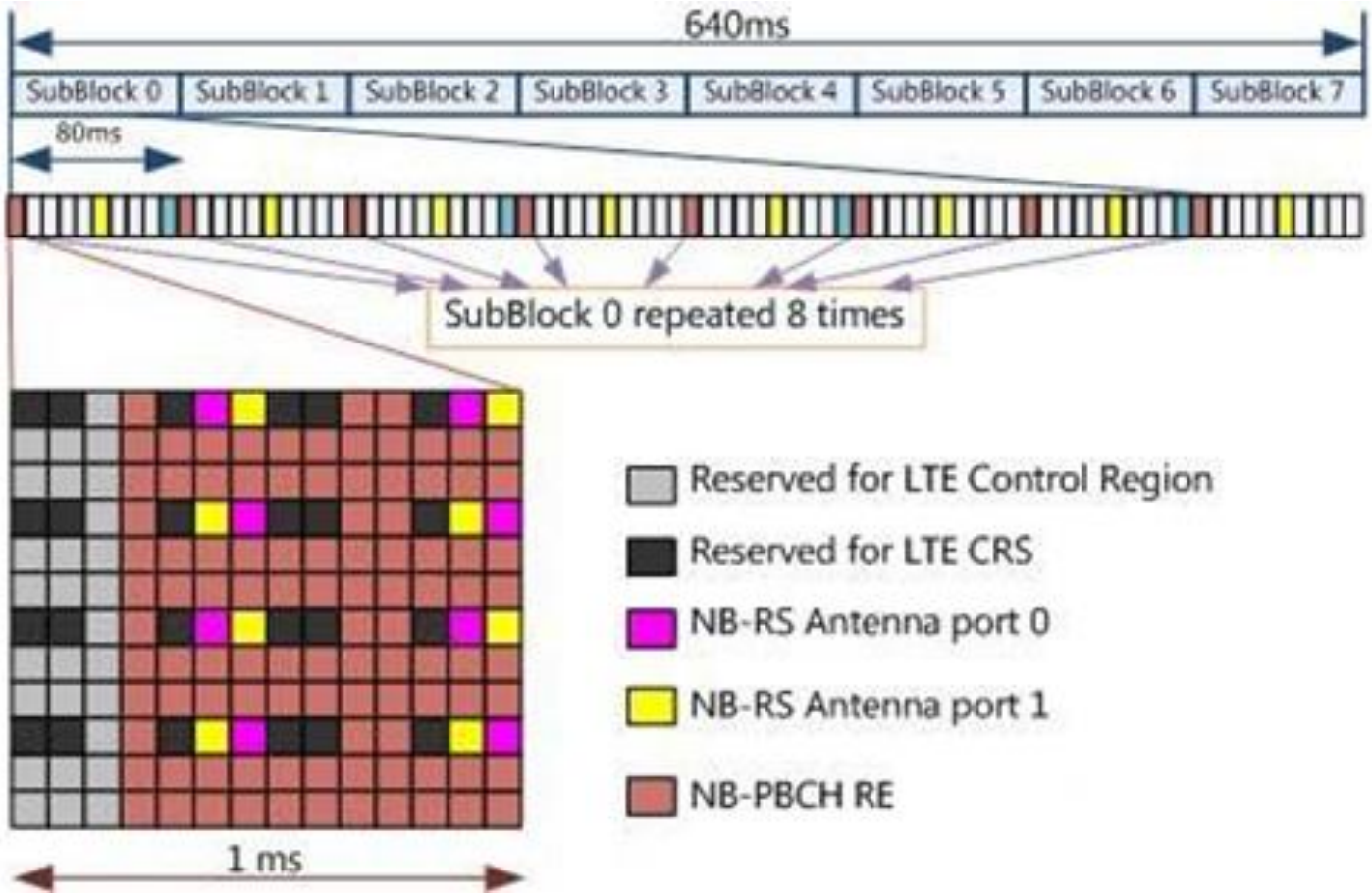
- **NRS**
  - 主要作用是下行信道质量测量的估计
  - 下行NPBCH, NPDCCH, NPDSCH信道无论有无数据传输，所有下行子帧(除了NPSS, NSSS)都要传输NRS
  - 每个子帧每个天线端口使用8个RE, 最多支持两个天线端口
- **NRS作用**
  - 上行功率控制，通过测量结果计算路损，从而设置终端发射功率
  - 终端判断所处覆盖等级，从而确定NPRACH的发送格式等

NRS资源映射



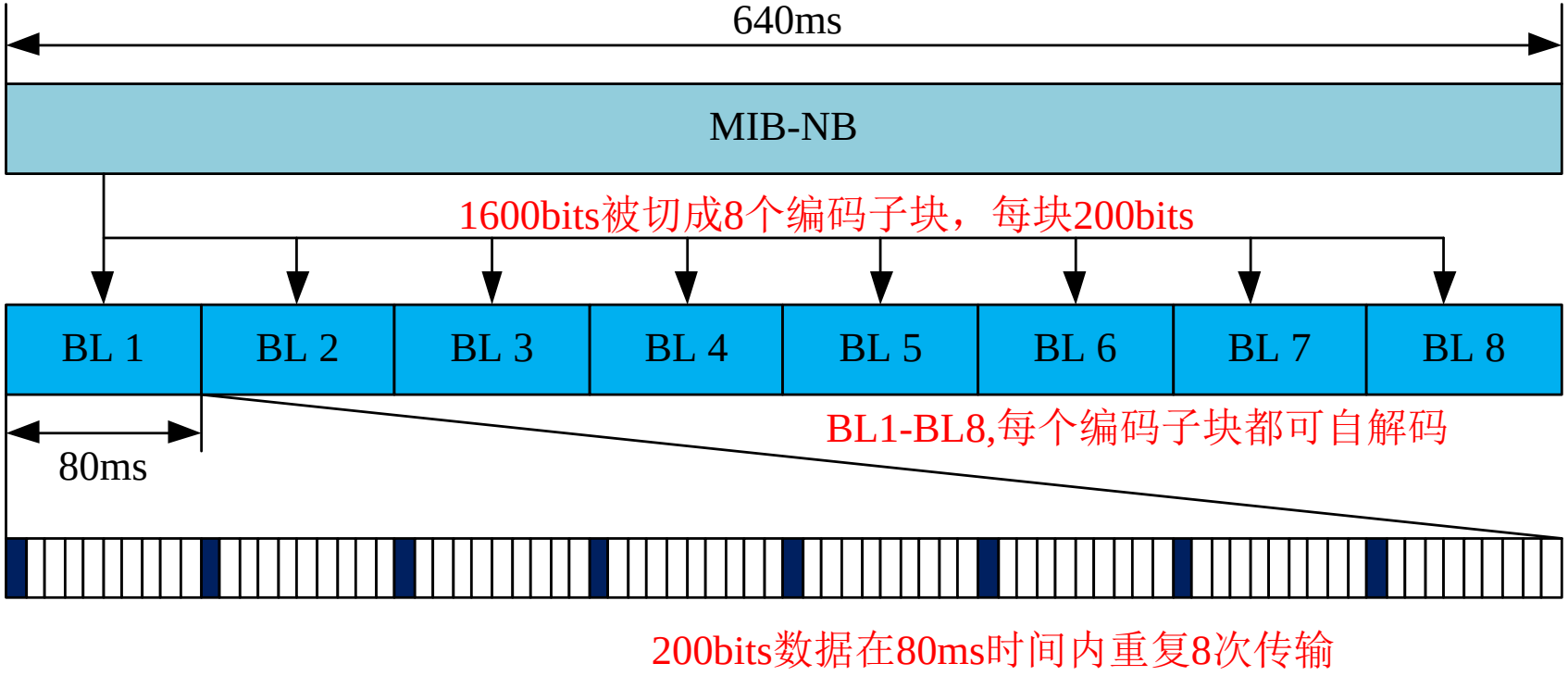
NPBCH

- **NPBCH**
  - 携带master information block (MIB，包括部署方式指示及系统消息SIB1-NB调度信息等)
  - 在每个无线帧的子帧#0上发送
  - N



NPBCH

- **分段**：34bits加扰后的1600bits被分为8个大小为200bits的编码子块
- **调制**：每个编码子块使用QPSK调制
- **资源映射**：每个编码子块调制后的符号在80ms时间内重复8次传输



NPDCCH

NPDCCH

- NPDCCH中承载的是DCI (Downlink Control Information)，包含一个或多个UE上的资源分配和其他的控制信息。UE需要首先解调NPDCCH中的DCI，然后才可以获取在相应的资源位置上解调属于UE自己的NPDSCH（包括广播消息，寻呼，UE的数据等）。

DCI Format	Size (bit)	作用
N0	23	上行NPUSCH调度，包括上行子载波指示I_{sc}(6 bits)、上行数据传输资源分配、调制编码方式以及上行数据传输确认信息等
N1	23	下行NPDSCH调度及随机接入下行响应，包括随机接入前导起始子载波指示以及随机接入重复次数等
N2	15	寻呼调度及系统消息更新指示，包括下行资源分配、DCI重复次数等

6-bit subcarrier indicator I_{SC}

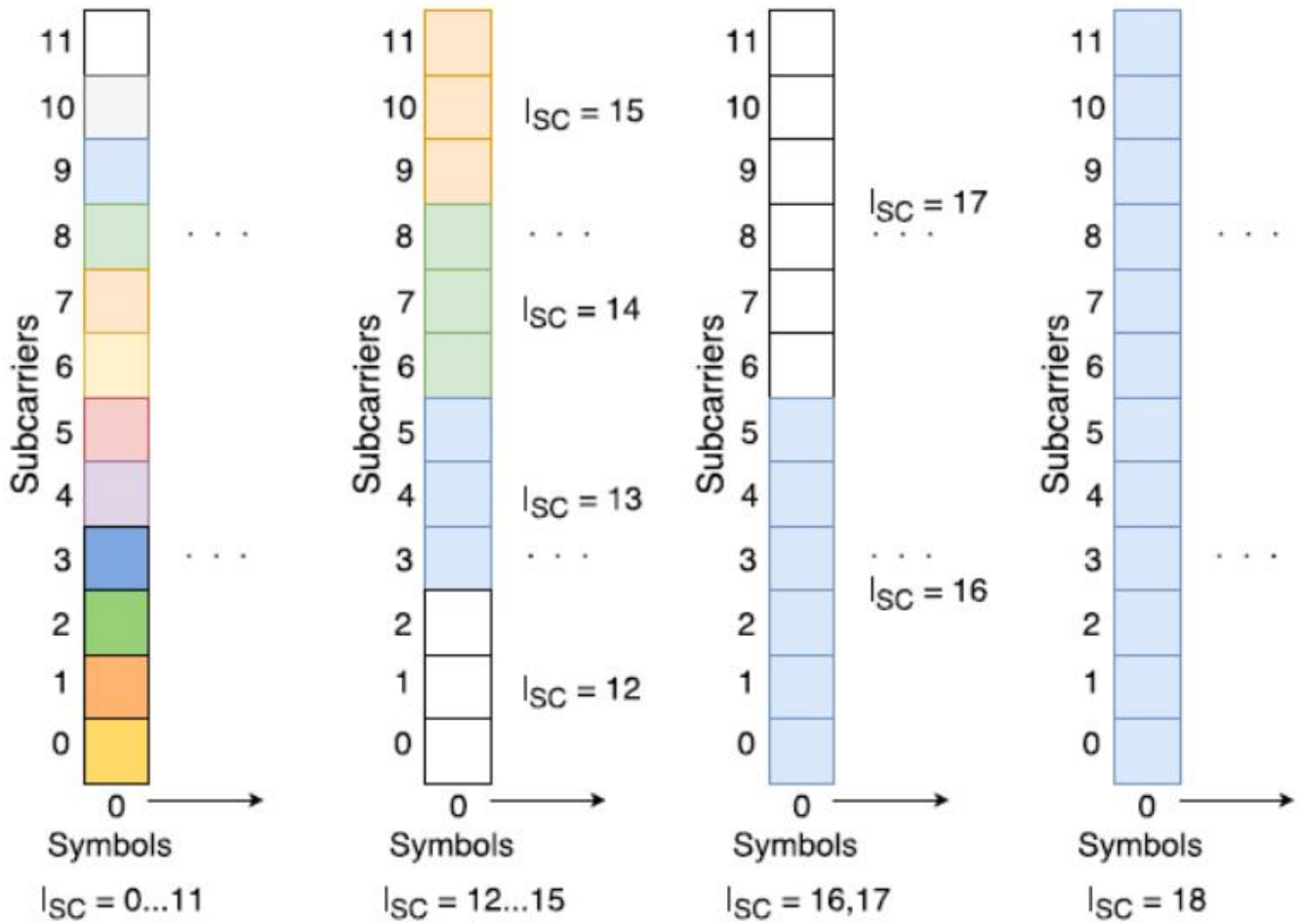
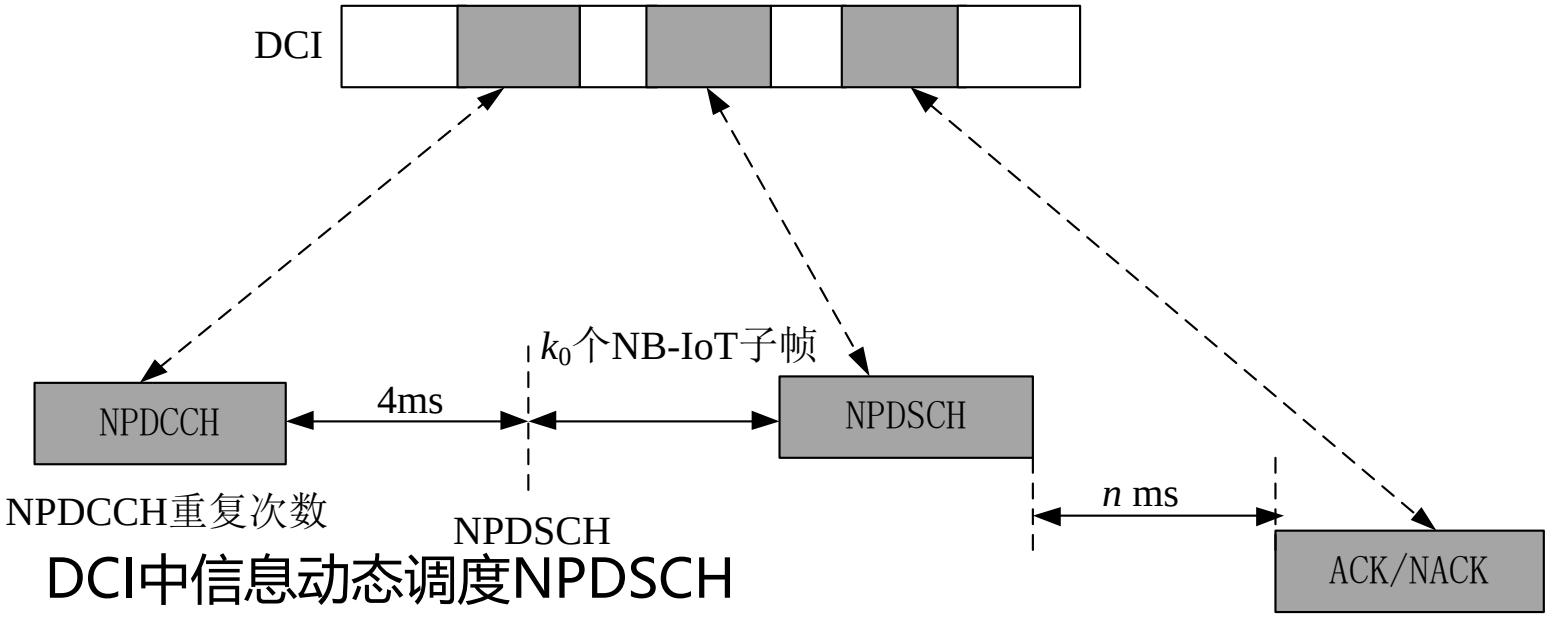


Figure 3.21: Subcarriers indicated by I_{SC}. For values 0-11 only that subcarrier is to be used. Values 12-15 indicate one of the four quarters of the subcarriers. 16 and 17 indicate either the top or bottom half of the subframe. 18 indicates all the subcarriers. Values 19-63 are reserved.

一些说明

- **DCI中包括一些重要的控制信息**
  - 下行上行数据信道的调度信息
    - NPDCCH end到NPDSCH至少4个subframe
    - NPDCCH end到NPUSCH至少8个subframe
  - 上行数据信道的HARQ ACK
  - Paging indication
  - RAR (random access response) 调度信息



DCI中信息动态调度NPDSCH

NPDSCH

- **NPDSCH**
  - 携带下行业务数据和系统消息(SIB1-NB，包括小区接入和选择信息)，包括单播业务数据, Paging message, sys info, RAR message
  - 子帧结构和NPDCCH一样
  - maximum tranport block size (TBS)为680bits
  - 一个TB依据其MCS可能需要N_{SF}个subframe传输, 同时可能重复传输N_{REP}次，因此总共需N_{SF}*N_{REP}个SFs传输。
    - *N_{SF}和N_{REP}都在DCI中指示.*
  - 下行业务数据没有自动ACK。如需要，则在DCI中会明确指示，ACK在NPUSCH format 2中发送 (相应的timing和subcarrier也在DCI中指示)

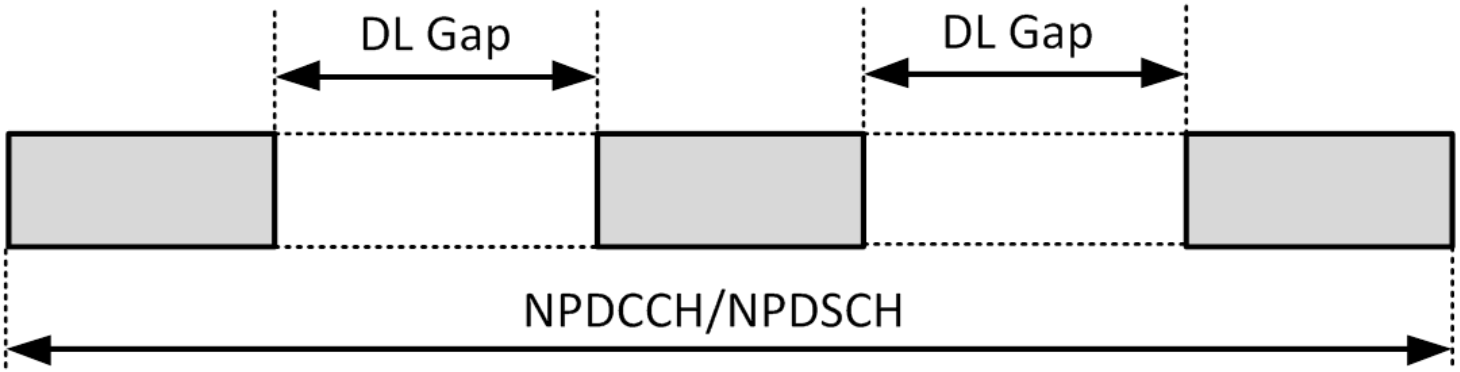
NPDSCH处理过程总结

- 选择TBS (承载上层有效数据)
- **附加CRC**: 通过24bits CRC进行错误检测
- **信道编码**：通过TBCC，将一个TB映射到N_{SF}个SFs
- **速率匹配**： => $N_{SF}*N_{REP}$ SFs?
- **加扰**
- **调制**：采纳只支持QPSK的调制方式
- **资源映射**：将调制后的符号映射到每个SF中RE

- **BLER**: block error rate
  - 例如将680bits TB分配到3个SFs (最小数目), 无重复; 如UE在3个SFs时间内正确解码(CRC通过), 称为一个sucessfulTransfer;
  - BLER = 1 - #sucessfulTransfers/#totalTransfers

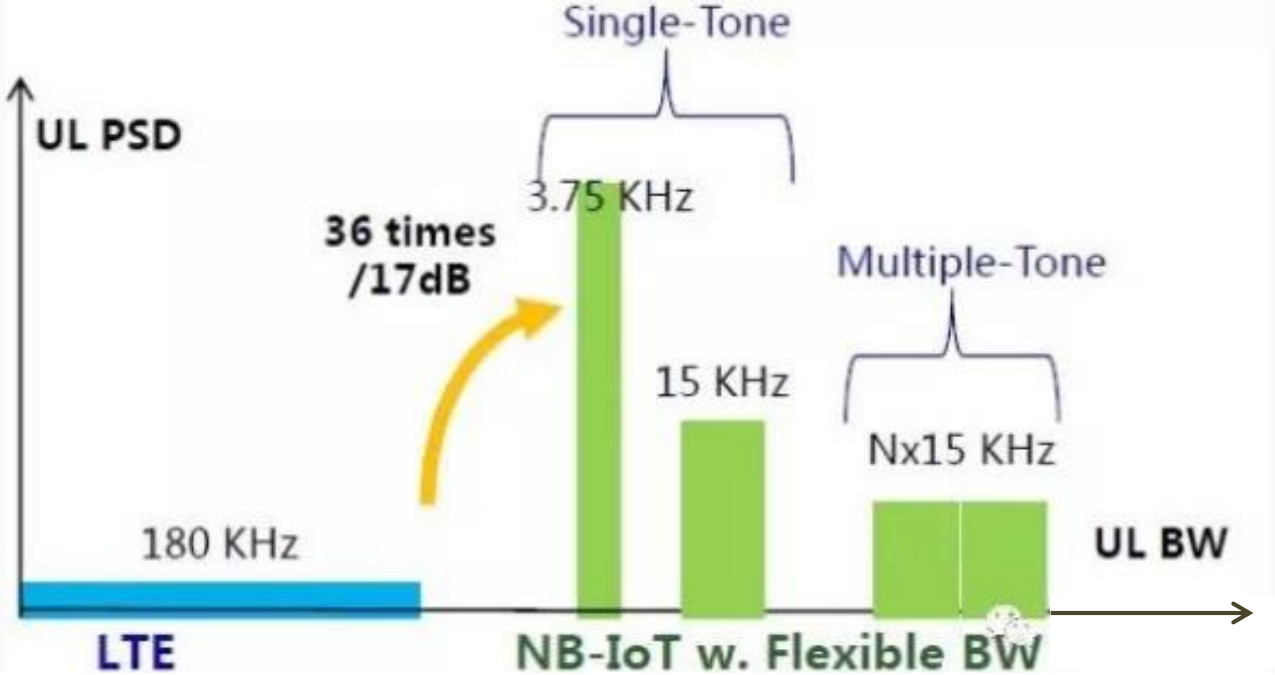
下行传输间隔

- 引入下行传输间隔DL Gap，避免覆盖信号弱的UE长时间占用下行资源 (NB-IoT下行本质是时分系统)
- 下行传输间隔参数由SIB配置，包括传输间隔周期 T_g={64,128,256,512}ms, 和传输间隔大小 ({1/8,1/4,3/8,1/2}*T_g)等
- 当下行传输NPDSCH/NPDCCH遇到DL Gap时，被延迟到DL Gap结束后



NB-IoT物理层：上行资源分配

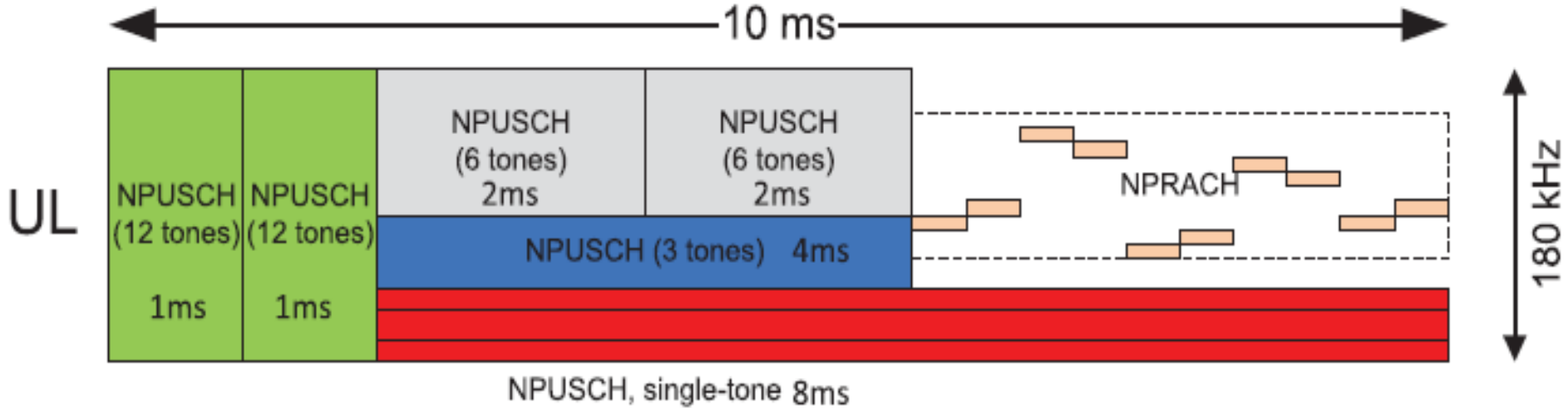
- **频域上**：上行带宽180kHz, 支持**两种子载波间隔**：
  - 12个subcarrier, 每个宽15kHz
  - 48个subcarrier, 每个宽3.75kHz
    两种模式：
  - single tone: 1 user用1个carrier; 支持15kHz, 3.75kHz
  - multi tone: 1 user用多个carrier; 支持15kHz



NB-IoT物理层：上行资源分配

- **时域上**：基本资源单位为一个时隙（slot）
  - 对于15kHz子载波间隔：1 slot = 0.5ms
  - 对于3.75kHz子载波间隔：1 slot = 2ms，记为NB-slot，以示区别
  - 注：频域上子载波间隔3.75K是15K的1/4，而时域上时隙2ms正好是0.5ms的4倍，两者其实是等效的。
- **无线帧**(frame) = 10ms = 5 NB-slot
- **RU** (resource unit): 时域、频域两个域的资源组合，**是资源分配的基本单位 (smallest unit to map a TB)**
  - 例如当子载波间隔为15kHz时，定义了以下4种RU: ①频域12个子载波，时域1ms; ②频域6个，时域2ms; ③频域3个，时域4ms; ④频域1个，时域8ms

NPUSCH RU


10 ms
UL
NPUSCH (12 tones) 1ms
NPUSCH (12 tones) 1ms
NPUSCH (6 tones) 2ms
NPUSCH (6 tones) 2ms
NPUSCH (3 tones) 4ms
NPRACH
180 kHz
NPUSCH, single-tone 8ms

NB-IoT物理层：上行RU

NPUSCH format	子载波间隔	子载波个数	每RU slot数	每slot持续时长 (ms)	每RU持续时长 (ms)	调制方式
format 1 (普通数据)	3.75kHz	1	16	2	32	π/2-BPSK
	15kHz	1	16	0.5	8	π/4-QPSK
		3	8		4	QPSK
		6	4		2	
		12	2		1	
format 2 (UCI, Uplink control information)	3.75kHz	1	4	2	8	π/2-BPSK
	15kHz	1	4	0.5	2	π/4-QPSK

NPUSCH根据用途被划分为了 Format 1和Format 2. 其中Format 1主要用来传普通数据，而Format 2资源主要用来传控制信息。

NB-IoT物理层：上行

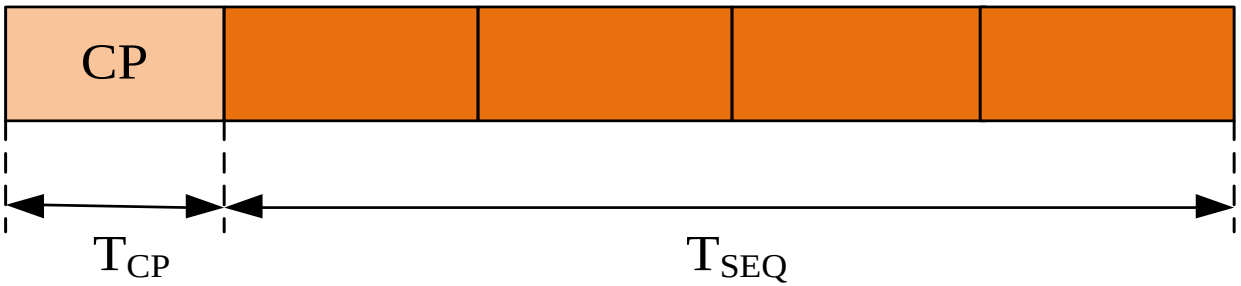
- **上行物理信道**
  - NPUSCH: Narrowband PHY UL Shared Channel
    - Format 1：用于传输上行信号的数据
    - Format 2：用于传输上行控制信息（UCI），目前限于HARQ ACK
  - NPRACH: Narrowband PHY Random Access Channel
- **上行物理信号**
  - DMRS: Demodulation Reference Signal

NPUSCH

- **NPUSCH format 1**
  - 用于传输上行数据
  - 使用turbo code
  - 最大传输块TBS=1000bits
  - 支持多载波传输方式, 如. 使用3,6,12个 子载波
  - 上行传输授权由DCI指定
- **NPUSCH format 2**
  - 用于传输上行控制信息
  - 使用repetition code
  - 只支持单载波传输

NPRACH

- 用于终端设备的随机接入过程，随机接入过程是用户设备从空闲态获取专用信道资源转变为连接态的重要手段（获取终端与基站间的上行同步）
- NPRACH preamble 的基本单位是 4 symbol groups
- 1 symbol group 由 1 CP（cyclic prefix）和 5 symbols组成
- 两种 CP lengths格式, 66.67us 和266.7us
- 5个symbols (相同内容) 持续传输时间T_{SEQ}=1.333ms
- 每个preamble中的symbol groups是连续传输的
- Preamble中的symbol group是通过调频方式传输的，且只支持单载波传输（3.75kHz）



NPRACH

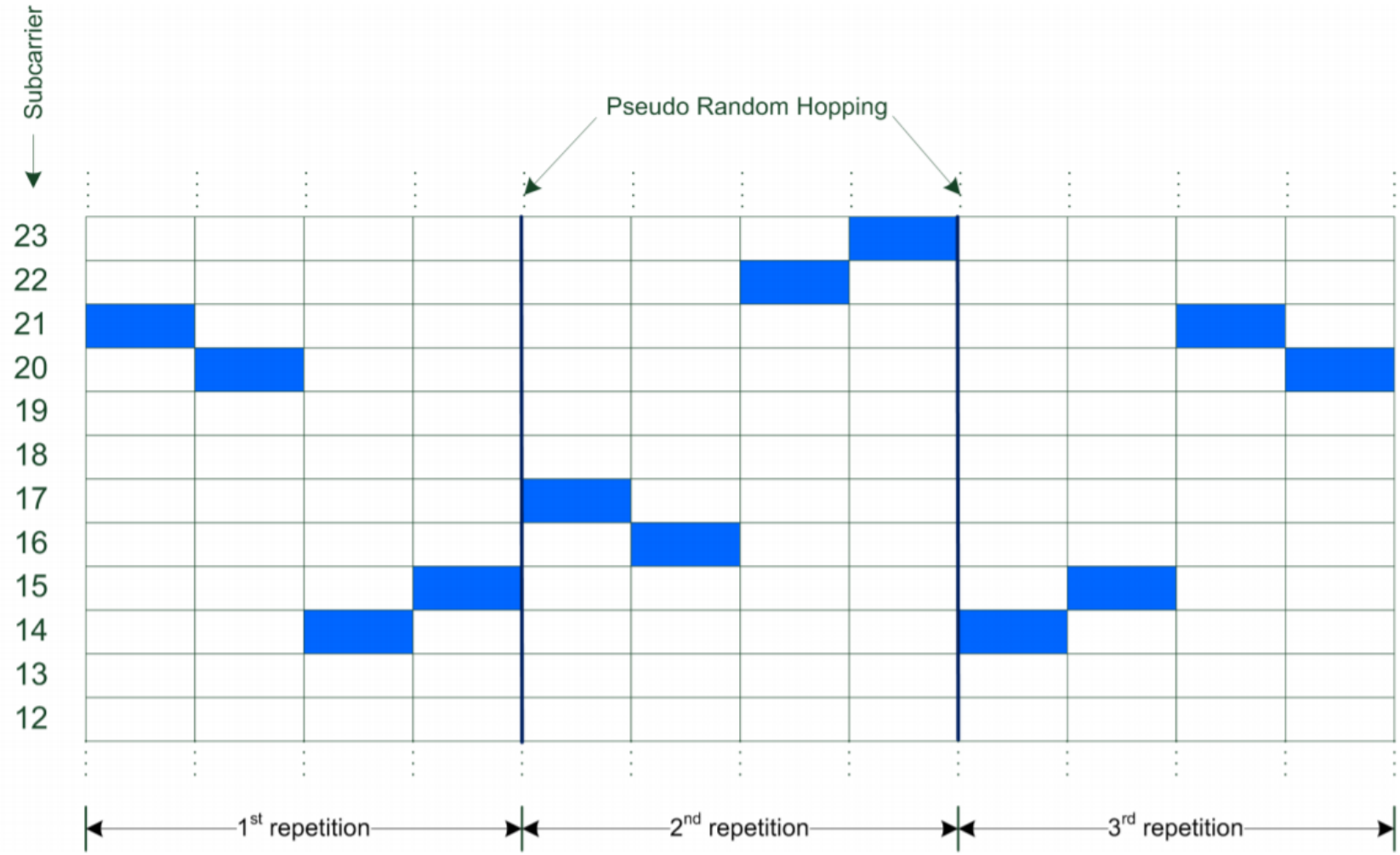


Figure 3-15: Preamble sequence in a frequency range between subcarrier 12 and 23

关键性能分析：最大数据传输率

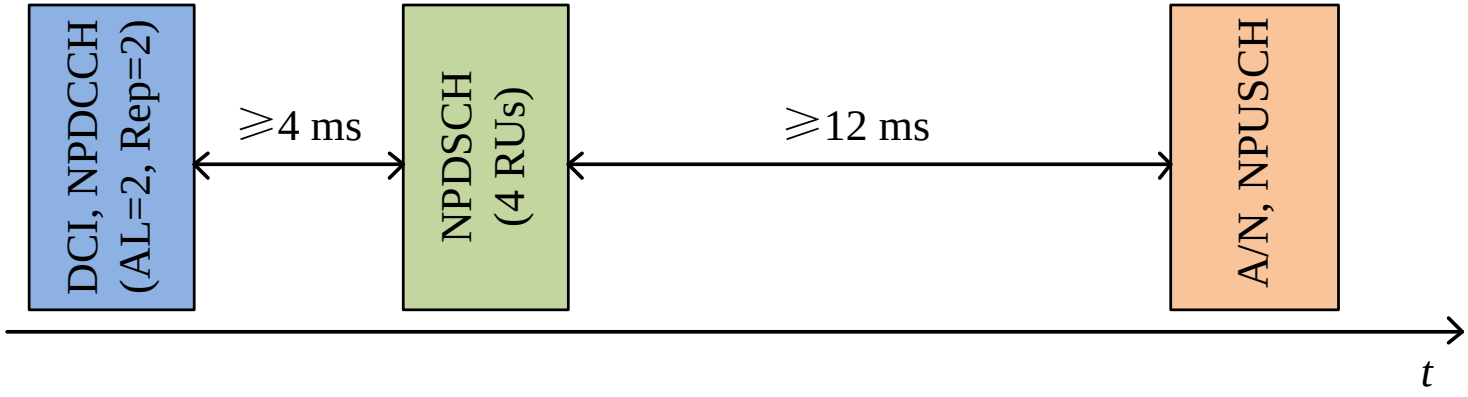
- **下行**
  - NPDSCH支持最大TBS=680bits
  - 最小需3SFs(=3ms)传输 [B2, p226]
  - => 226.7kb/s peak layer 1 data rate

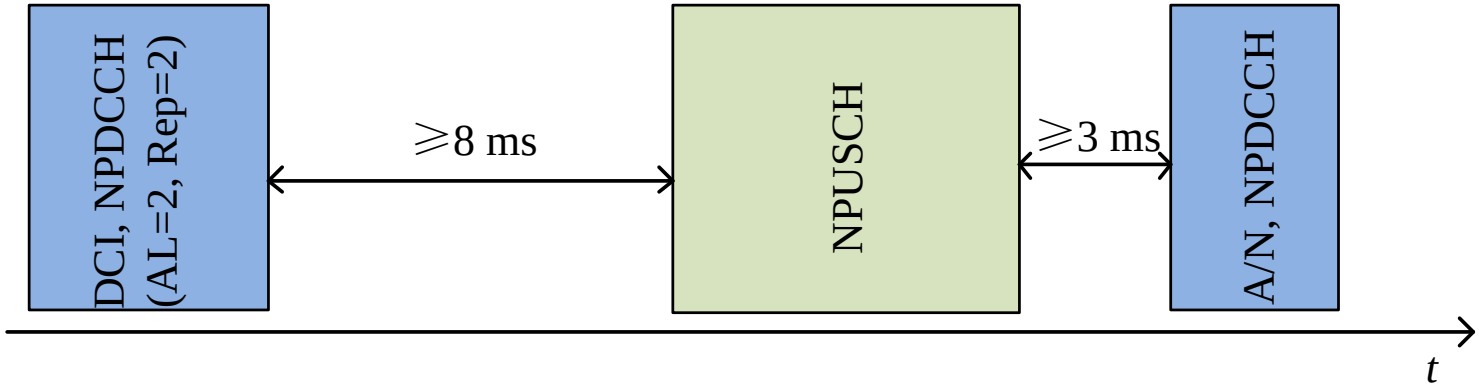
- **上行**
  - NPUSCH支持最大TBS=1000bits
  - 最小需4SFs(=4ms)传输 [B2, p236]
  - => 250kb/s peak layer 1 data rate

- **注**：上述计算没考虑在DCI, NPDSCH/NPUSCH, HARQ之间的时间间隔

调度和HARQ

- DCI调度下行数据及其ACK
- DCI调度上行数据及其ACK





关键性能分析：最大数据传输率 (考虑timing)

- **下行：** 680/25=**27.2kb/s** peak layer 1 data rate
- **上行：** 1000/16=**62.5kb/s** peak layer 1 data rate

Subframe Count	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Grant DL Data#1 **NPDCCH** to DL Data#1 **NPDSCH**	N1	1	2	3	4	680																				
DL Data#1 **NPDSCH** to **NPUSCH** ACK#1						680			1	2	3	4	5	6	7	8	9	10	11	12	A/N	A/N	Switch			
NPUSCH ACK#1 to NPDCCH						680														Switch	A/N	A/N	1	2	3	Gr

NPDSCH: 680 TBS across 3 SF

A/N across 2 SF (15 kHz, single tone)

DCI: Aggregation Layer = 1 (best condition for peak rate, 69 encoded bits, 14 OFDM symbols and 6 SC sufficient)

DCI: Repetitions = 1 (best condition for peak rate, no repetitions needed)

Subframe Count	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Grant UL Data#1 **NPDCCH** to UL Data#1 **NPUSCH**	N0	1	2	3	4	5	6	7	8	1000				Switch			
UL Data#1 **NPUSCH** to **NPDCCH**									Switch	1000				1	2	3	Gr

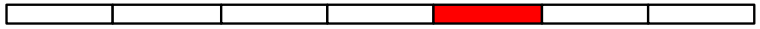
NPUSCH: 1000 TBS across 4 SF

DCI: Aggregation Layer = 1 (best condition for peak rate, 69 encoded bits, 14 OFDM symbols and 6 SC sufficient)

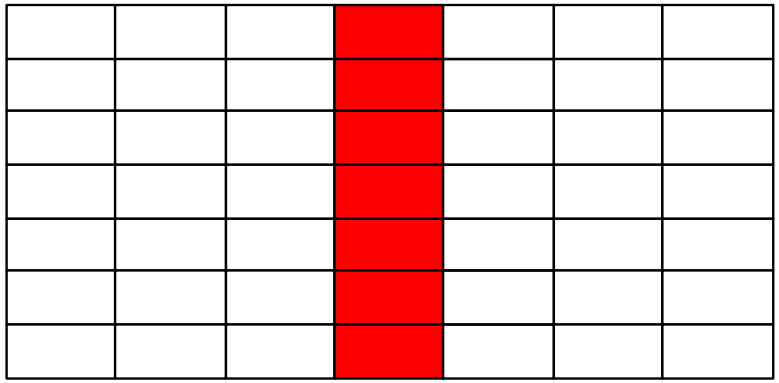
DCI: Repetitions = 1 (best condition for peak rate, no repetitions needed)

DMRS

- **DMRS**
  - 用于信道估计
  - 复用数据传输信道资源（NPUSCH）
  - 只能在包含有业务数据的RUs中传输
- 根据NPUSCH格式，DMRS每时隙传输1个或者3个SC-FDMA符号
  - NPUSCH format 1: DMRS occupies 1 of 7 symbols/slot

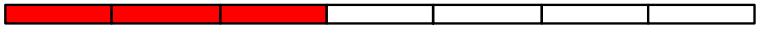


3.75kHz Subcarrier Spacing



15kHz Subcarrier Spacing

  - NPUSCH format 2: DMRS occupies 3 of 7 symbols/slot

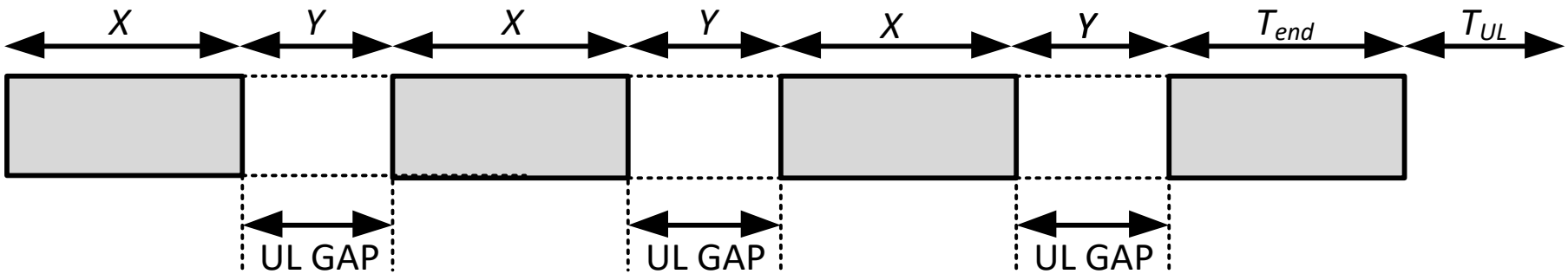


3.75kHz Subcarrier Spacing

15kHz Subcarrier Spacing

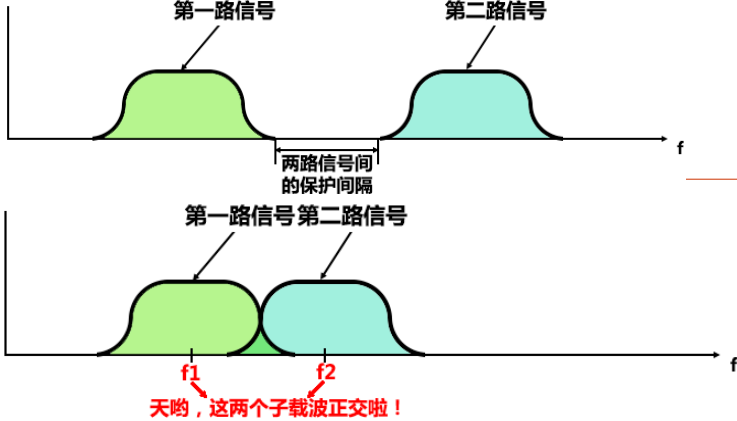
上行传输间隔

- 引入上行传输间隔，防止上行传输太长，影响下行的同步
- 每传输X时间后，引入Y时间的间隔
  - 若最后一次传输时间T_{end}=X, 则其后同样引入Y, UE不会在这个时间内进行下行数据接收

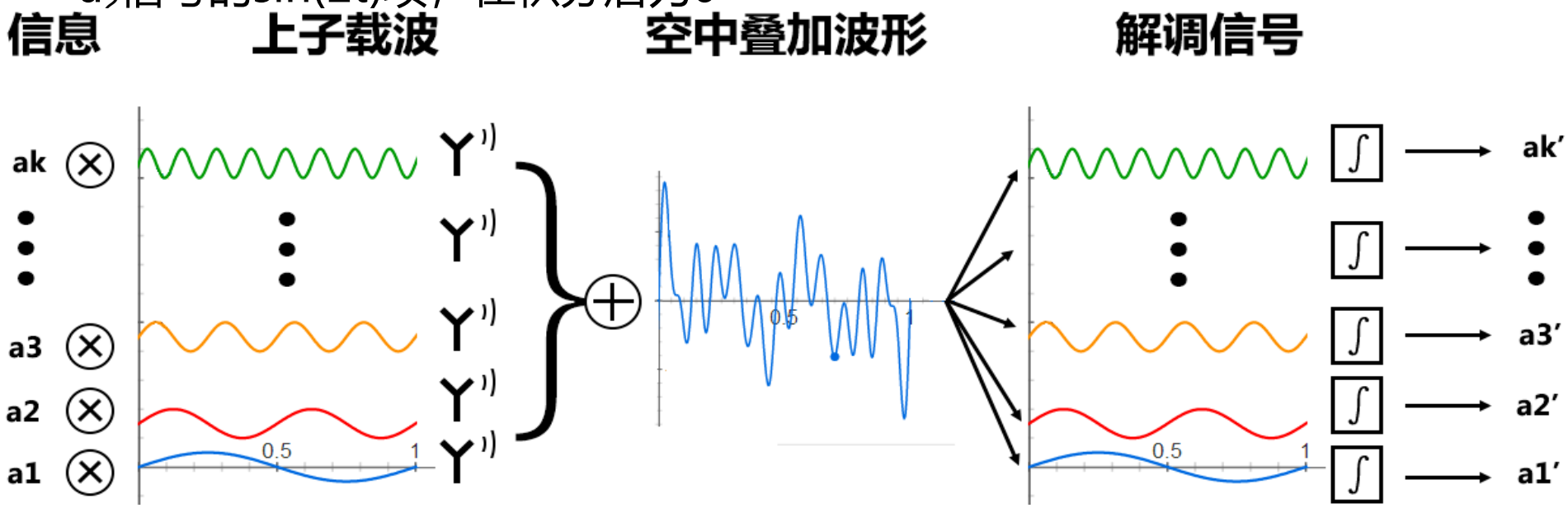


- 对于NPUSCH, X=256ms, Y=40ms
- 对于NPRACH, X=64 * preamble length
  - NPRACH format 0, X=64*5.6ms=358.4ms
  - NPRACH format 1, X=64*6.4ms=409.6ms

OFDM

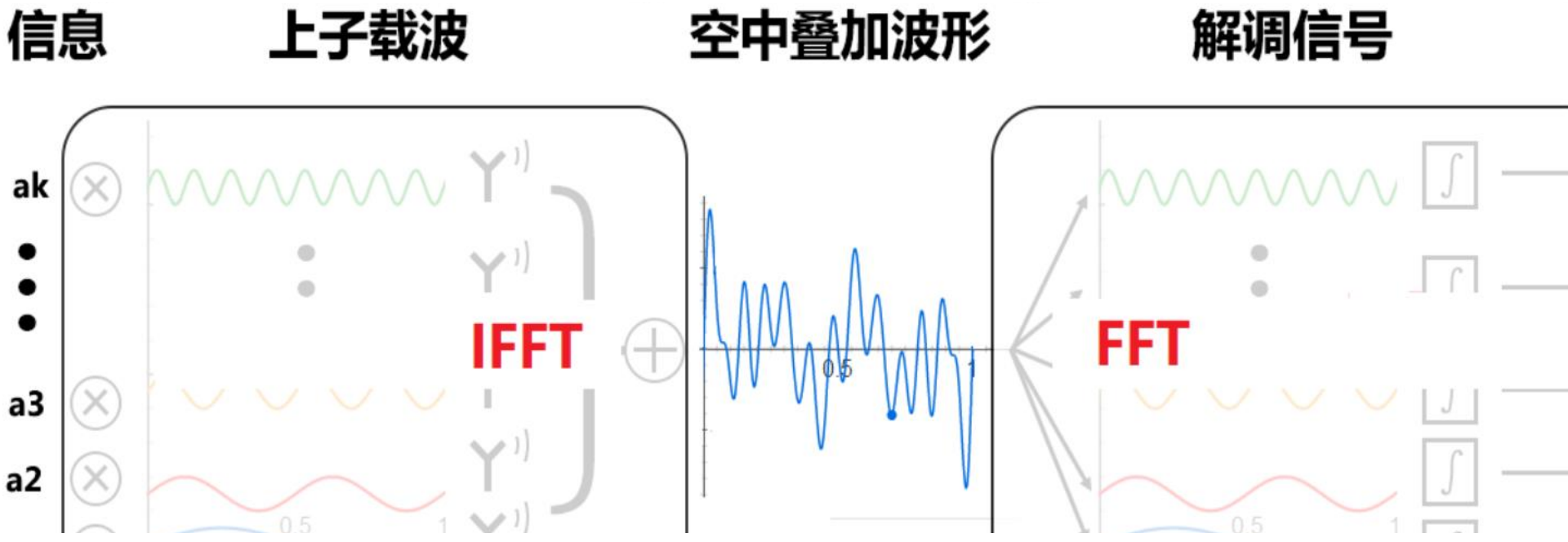


- **FDM vs. OFDM**
- 为什么靠这么近而没有干扰?
  - 利用载波的正交性
- {sin(2π·Δf·kt)}, {cos(2π·Δf·kt)} (k=1,2,...)中任何两个载波两两正交
- 假设每个子载波携带信息a_k, b_k (调制在子载波上的幅度上)
- $f(t) = \sum a_k \cdot \sin(2\pi \cdot \Delta f \cdot kt) + \sum b_k \cdot \cos(2\pi \cdot \Delta f \cdot kt) = \sum F_k \cdot e^{(j \cdot 2\pi \cdot \Delta f \cdot kt)}$
- 接收端收到叠加信号f(t)后，再在每个子载波上作相乘后积分，就可以解出每个子载波承载的信号。e.g. 接收信号乘sin(t)，积分解码出a_1, f(t)中传送a_2信号的sin(2t)项，在积分后为0



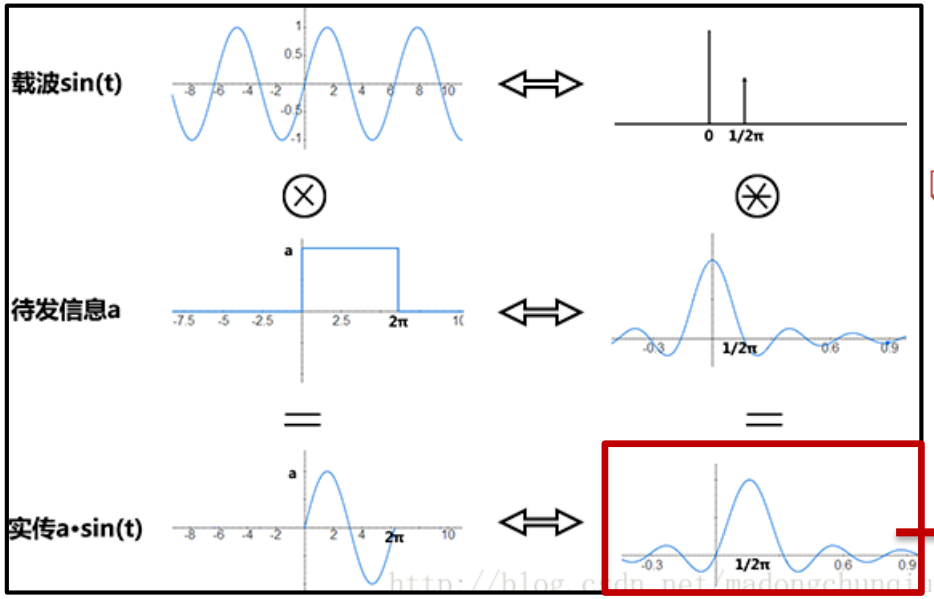
IFFT & FFT

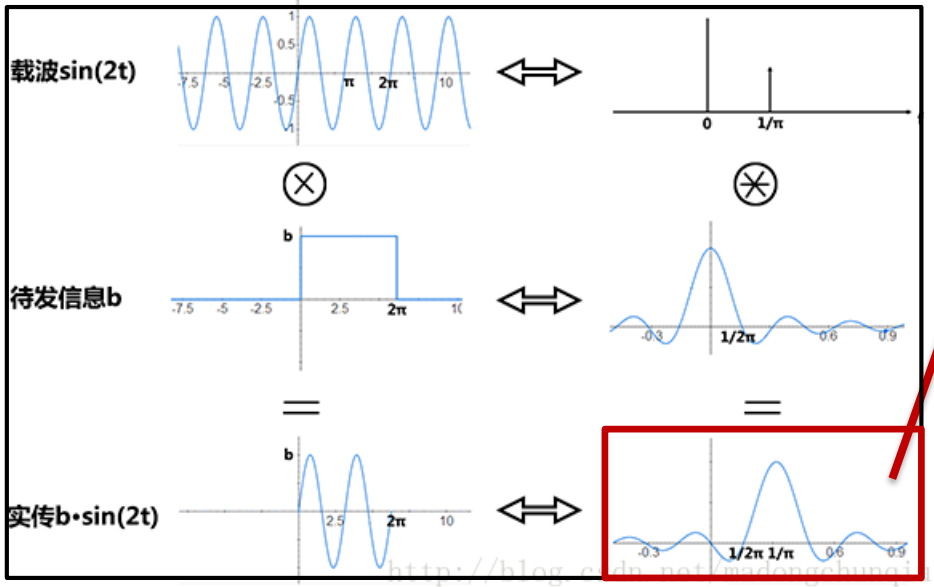
- **IFFT** ≈ IDFT -> $f_n = 1/N \cdot \sum F_k \cdot e^{(j \cdot 2\pi \cdot k \cdot n/N)}$, $n \in [1,N]$, **频域 -> 时域**
  - 从N个子载波信号（频域）直接算出空中叠加信号（时域）
  - 在一个OFDM symbol的时长T内，用N个子载波各自发送一个信号F_k（$k \in [1,N]$），等效于直接在时域上连续发送N个信号f_n（$n \in [1,N]$），每个信号发送T/N的时长。
- **FFT**, **时域 -> 频域**
  - 相比传统使用积分方法来去除其余正交子载波，可以一次把N个信号全算出来

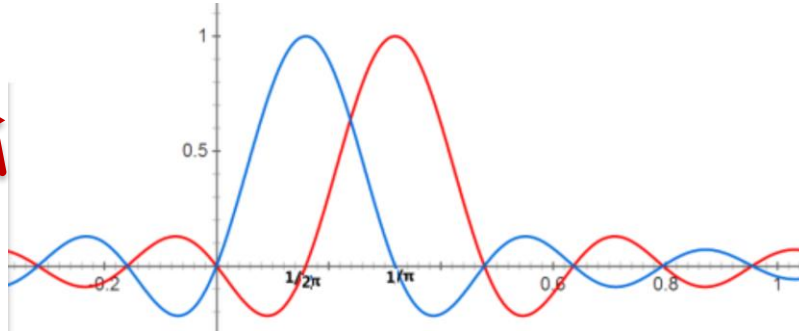


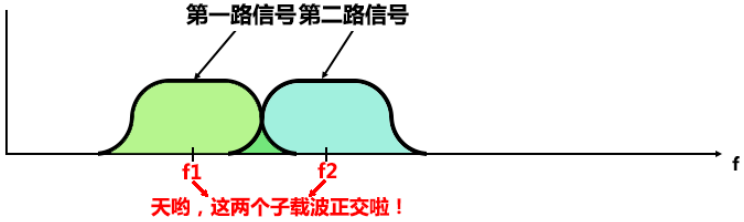
从频域看叠加信号

- sin(t)是正弦波，代表单一频率，频谱是一个冲激
- 限制在[0,2π]内的sin(t)信号，相当于无限长的sin(t)信号乘以一个[0,2π]上的门信号（矩形脉冲，频谱为sinc=sin(x)/x信号），其频谱为两者频谱的卷积, 相当于对sinc信号的搬移
- 基带信号在传输前，一般会通过脉冲成型滤波器，因此最终变为右下图所示信号。作用是有效的限制带宽外部的信号，最大限度的利用带宽，减少子载波间的各路信号的相互干扰



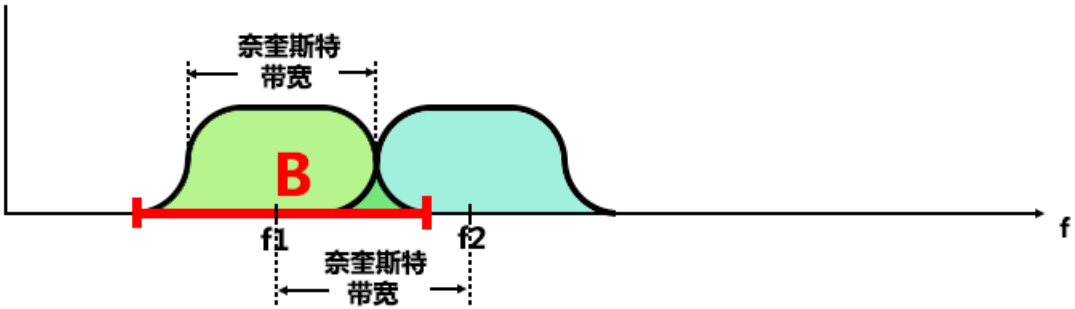






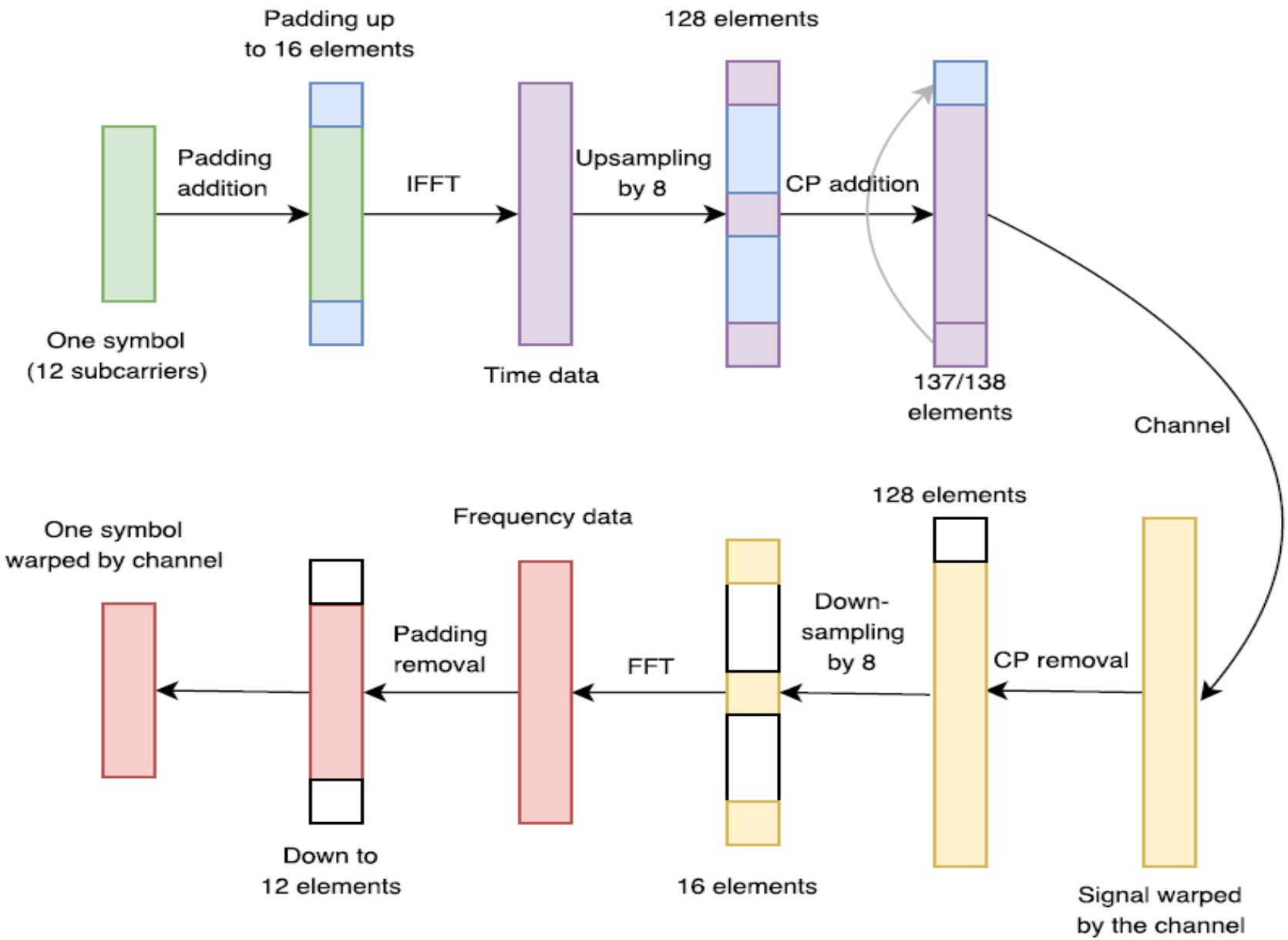
理想信道的频带利用率

- OFDM中两个子载波到底可以有多近? 奈奎斯特带宽
  - 对于理想低通信道，奈奎斯特带宽W=1/(2T), T: 一个symbol传输时间
  - 对于理想带通信道，奈奎斯特带宽W=1/T



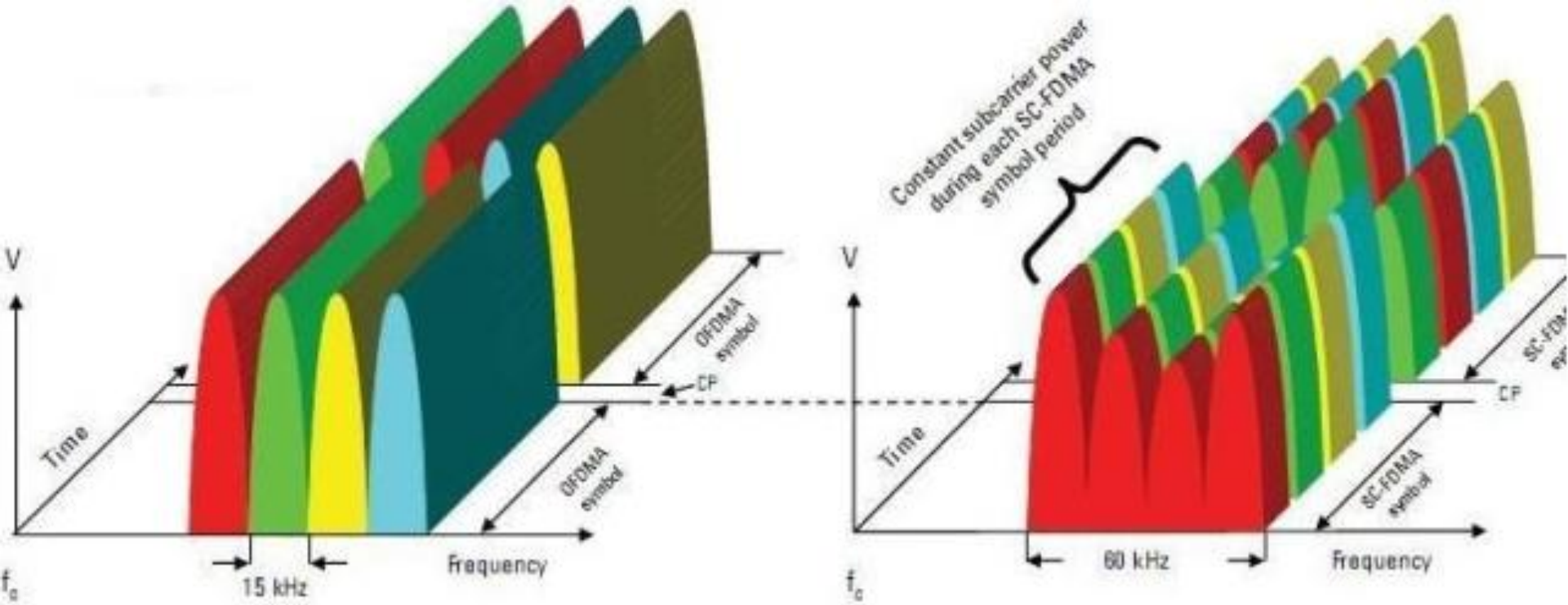
- 根据奈奎斯特第一准则，可以简单认为：**在理想情况下，信道的频带利用率为2Baud/Hz**；**OFDM达到了理想信道的频带利用率**。

NB-IoT OFDM调制/解调



SC-FDMA：单载波频分多址

- **OFDM（左图）**
  - 将频谱分成4个正交的子载波（每个15kHz），在一个symbol period里经过IFFT操作同时将4个symbol发送出去

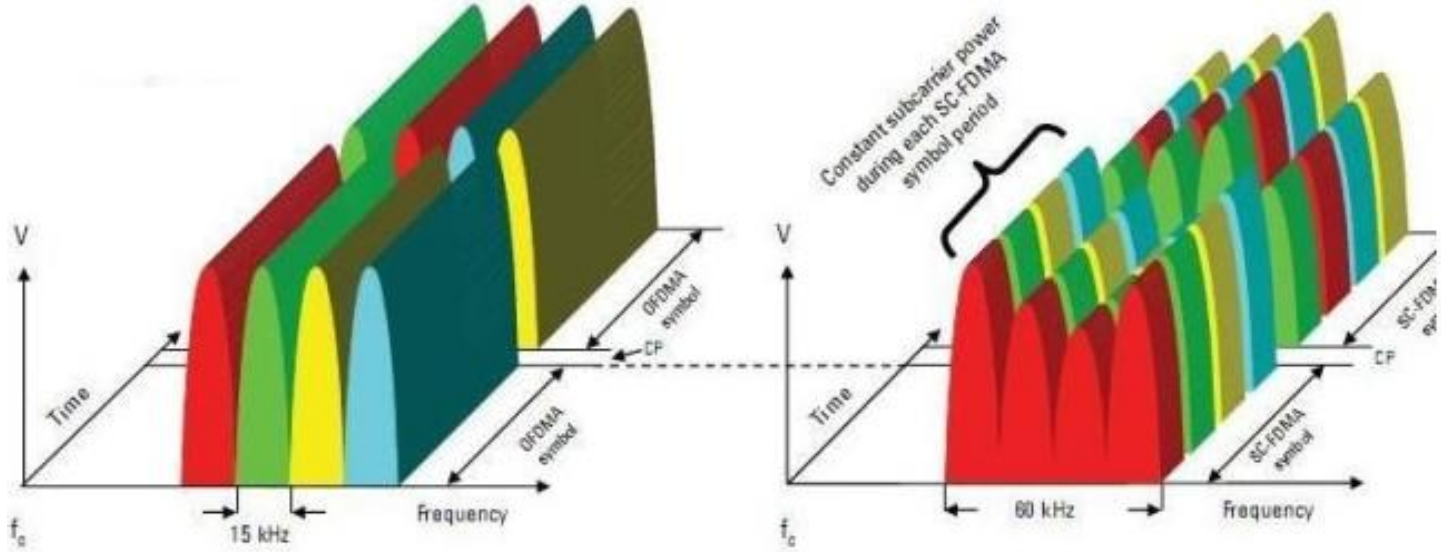


OFDMA
Data symbols occupy 15 kHz for one OFDMA symbol period

SC-FDMA
Data symbols occupy M*15 kHz for 1/M SC-FDMA symbol periods

SC-FDMA：单载波频分多址

- **SC-FDMA（右图）**
  - 将频谱分成4个正交的子载波（每个15kHz）。对4个symbol首先进行FFT，然后进行IFFT，然后在一个symbol period里，一个接一个传输symbol
  - 上述操作等效于把一个symbol放到4个子载波上同时发送，即single carrier
  - 相比于OFDM，SC-FDMA具有更低的峰均比

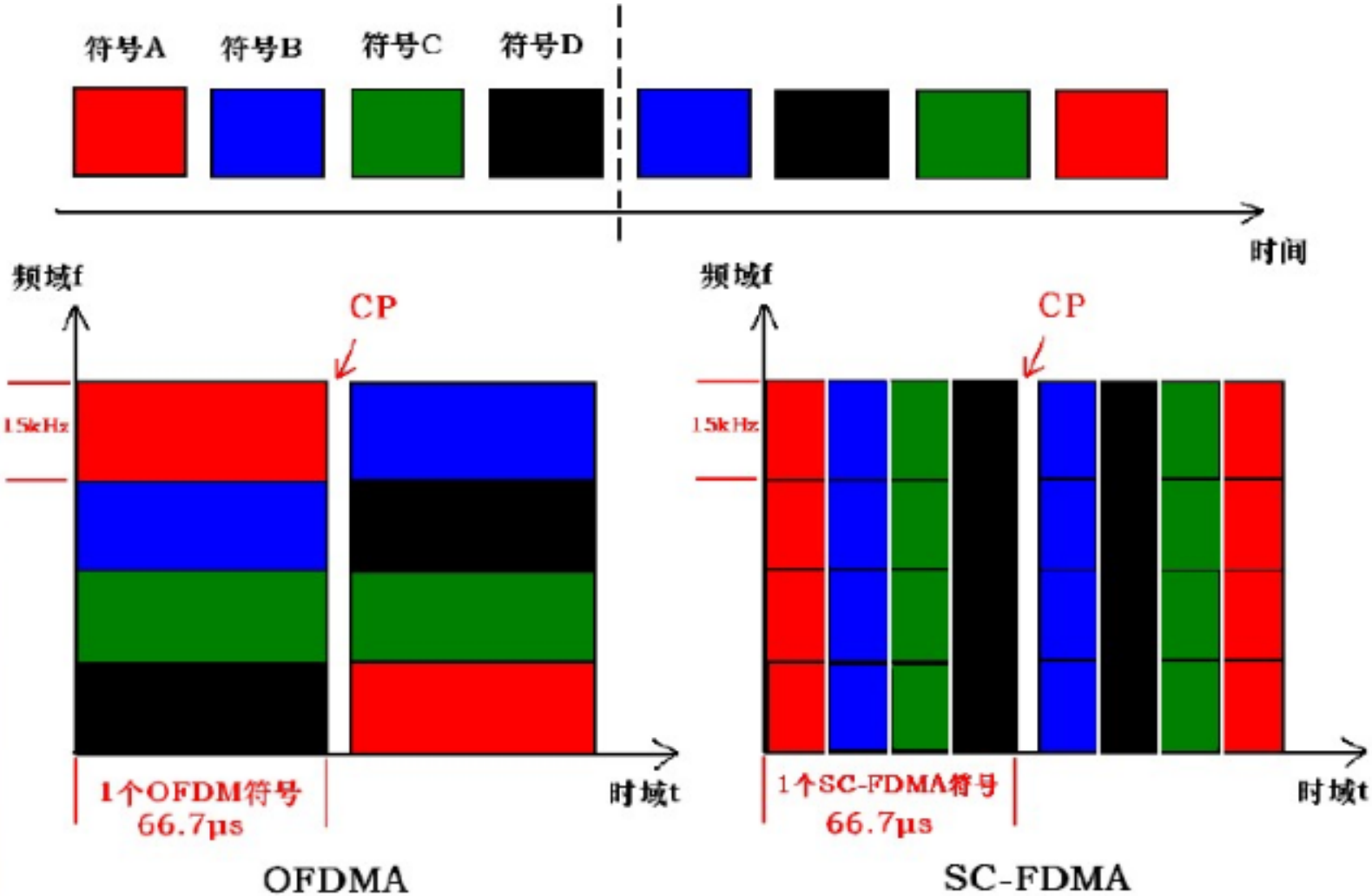


OFDMA
Data symbols occupy 15 kHz for one OFDMA symbol period

SC-FDMA
Data symbols occupy M*15 kHz for 1/M SC-FDMA symbol periods

SC-FDMA：单载波频分多址

- 用户数Q=1，子载波数M=4，symbol数N=4，两个符号周期



上行下行总结

	物理信道	作用及性质
下行	NPSS	用于时间和频率同步，在每个无线帧的子帧#5上发送，发送周期10ms
	NSSS	携带小区ID信息，在偶数无线帧的子帧#9上发送，发送周期20ms
	NRS	用于下行信道质量估计
	NPBCH	承载主系统消息块（MIB-NB，包括部署方式指示等），在每个无线帧的子帧#0上发送
	NPDCCH	承载DCI，包含一个或多个UE上的资源分配和其他控制信息
	NPDSCH	携带下行业务数据和系统消息，包括单播业务数据, 寻呼消息和随机接入响应消息
上行	DMRS	用于信道估计，根据NPUSCH格式，DMRS每时隙传输1个或者3个SC-FDMA符号
	NPRACH	用于设备随机接入过程，采用跳频传输方式
	NPUSCH	用于传输上行数据和上行控制信息，支持多载波传输方式

链路层

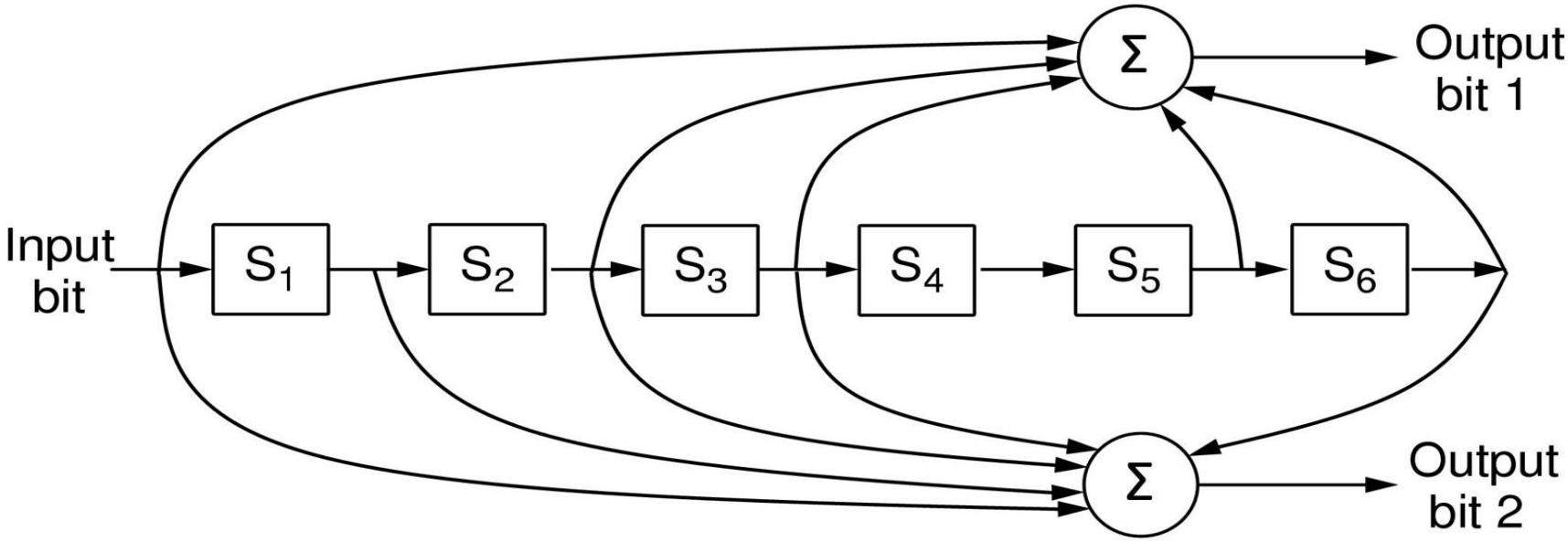
- **编码技术**
  - 下行：TBCC (Tail Bitting Convolutional Coding)
  - 上行：Turbo code
- **随机接入过程**

编码细节：Convolutional Code

- An encoder processes a sequence of input bits and generates a sequence of output bits.
- The output depends on the current and previous input bits. That is, the encoder has memory. #previous input bits is called constraint length.
- **Convolutional code is characterized by its coding rate and constraint length**

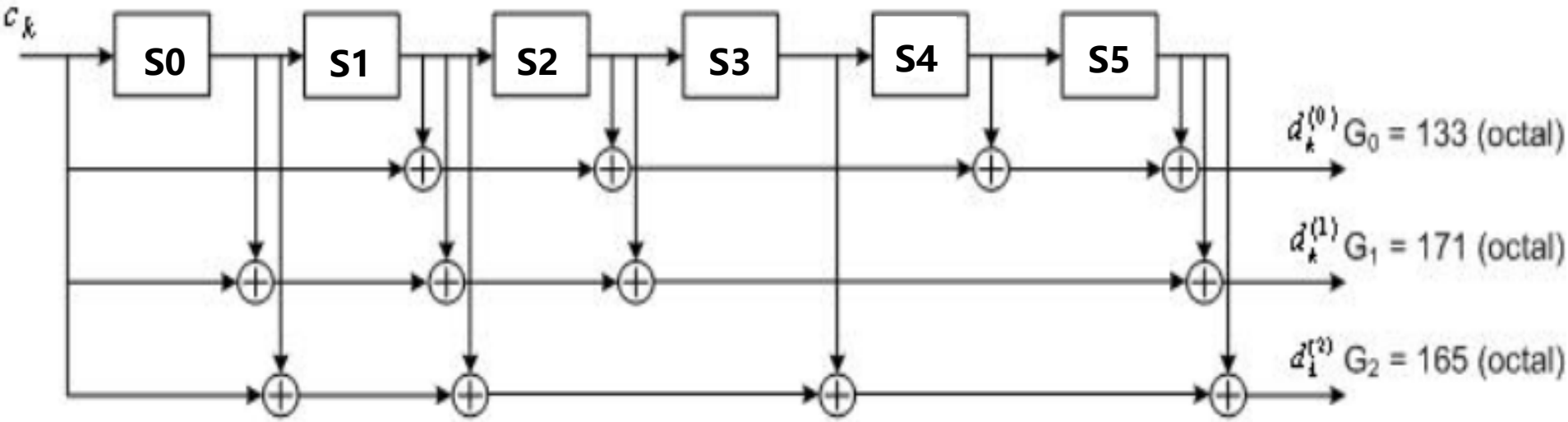
Convolutional code example

- Constraint length k=#registers+1=7, r=1/2
- Input bits: 111; Initial internal state $s_1..s_6$=000000
- Take the 1st input bit: output=11, state=100000
- Take the 2nd input bit: output=10, state=110000
- Take the 3rd input bit: output=01, state=111000



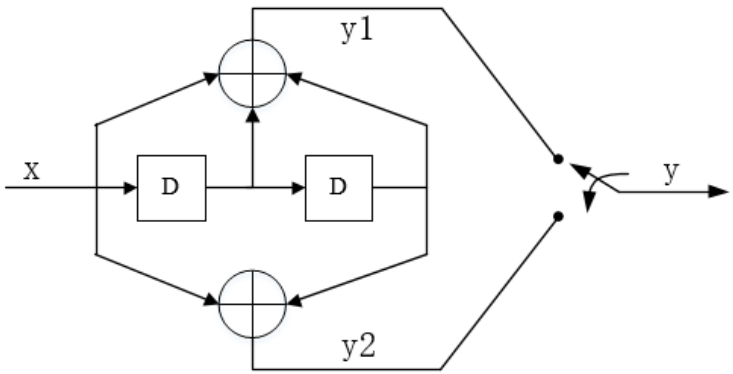
TBCC

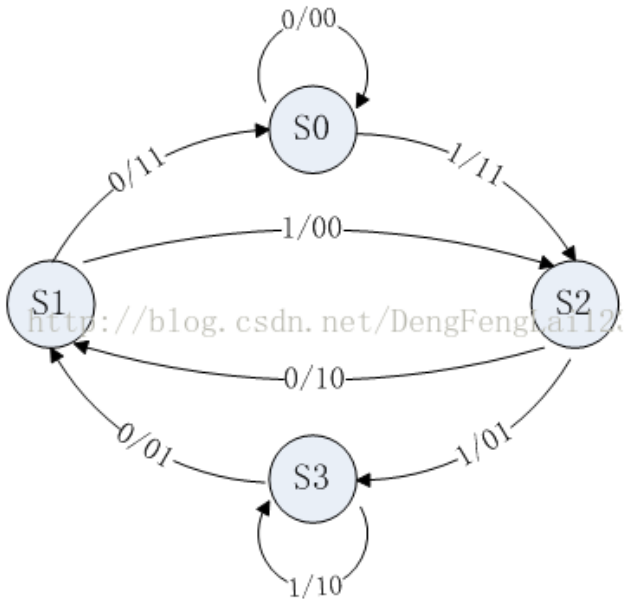
- **输入块:** $c_0c_1,\ldots,c_{k-1}$ 共k位
- **初始化:** $S_0S_1,\ldots,S_5=c_{k-1},c_{k-2},\ldots,c_{k-6}$
- 当最后的比特C_{k-1}进行编码完成后，寄存器又回到了初始状态，这种编码称为**咬尾卷积码** (Tail Bitting Convolutional Code, **TBCC**)。
- NB-IoT下行采用TBCC



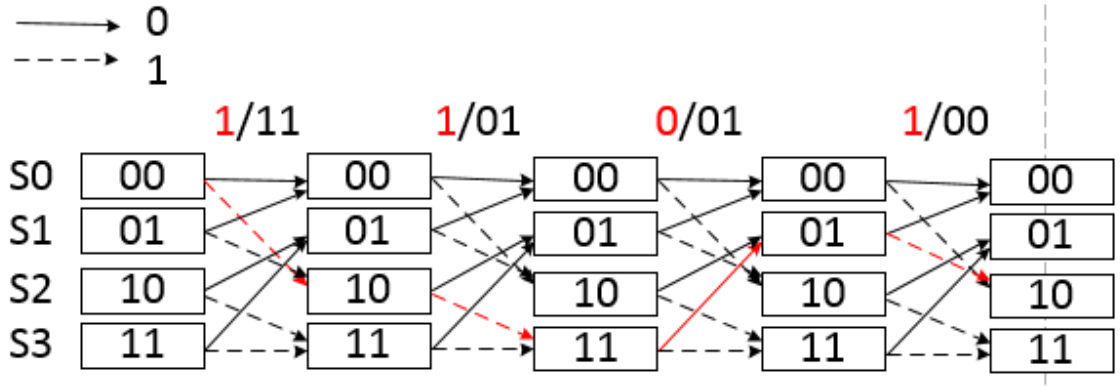
Trellis diagram

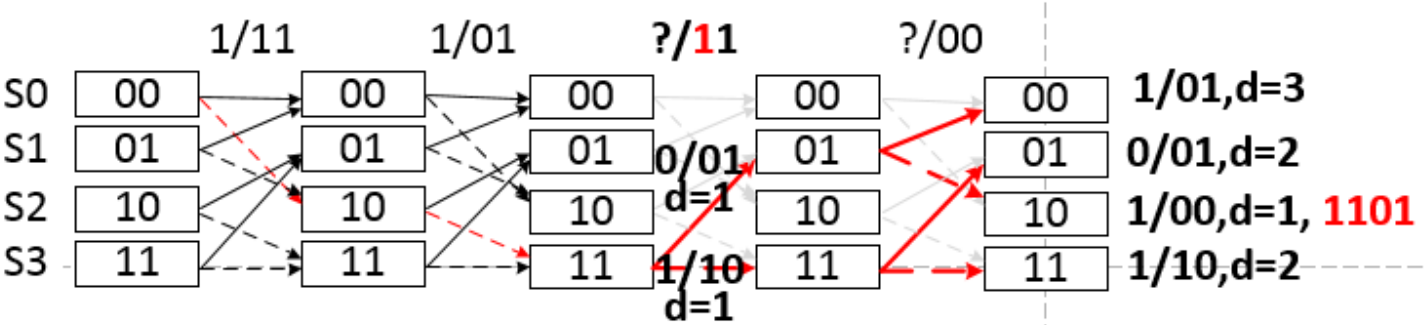
- **输入数据：** u = [1 1 0 1]
  - 初始状态: 00
- **编码：** V = [11 01 01 00]
- **正确接收：** R = [11 01 01 00] (上)
- **错误接收：** R = [11 01 11 00] (下)





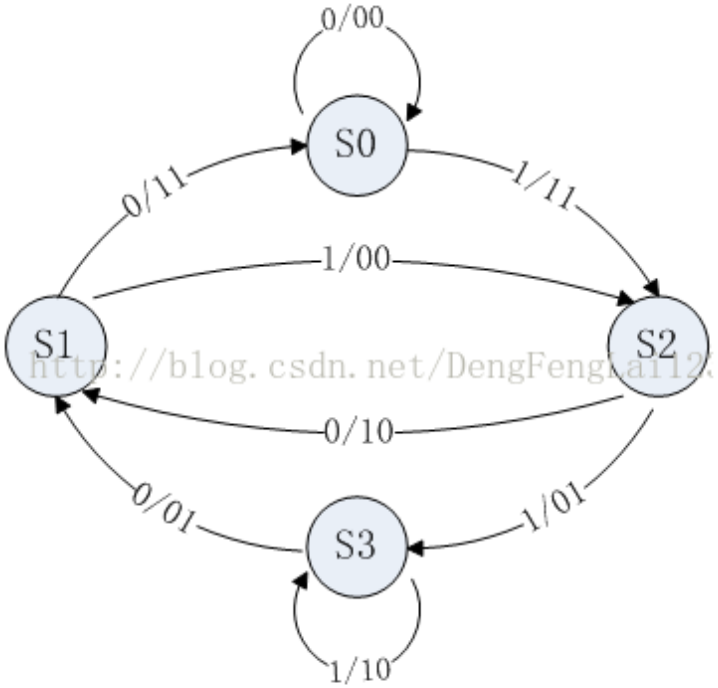
http://blog.csdn.net/DengFengJia112



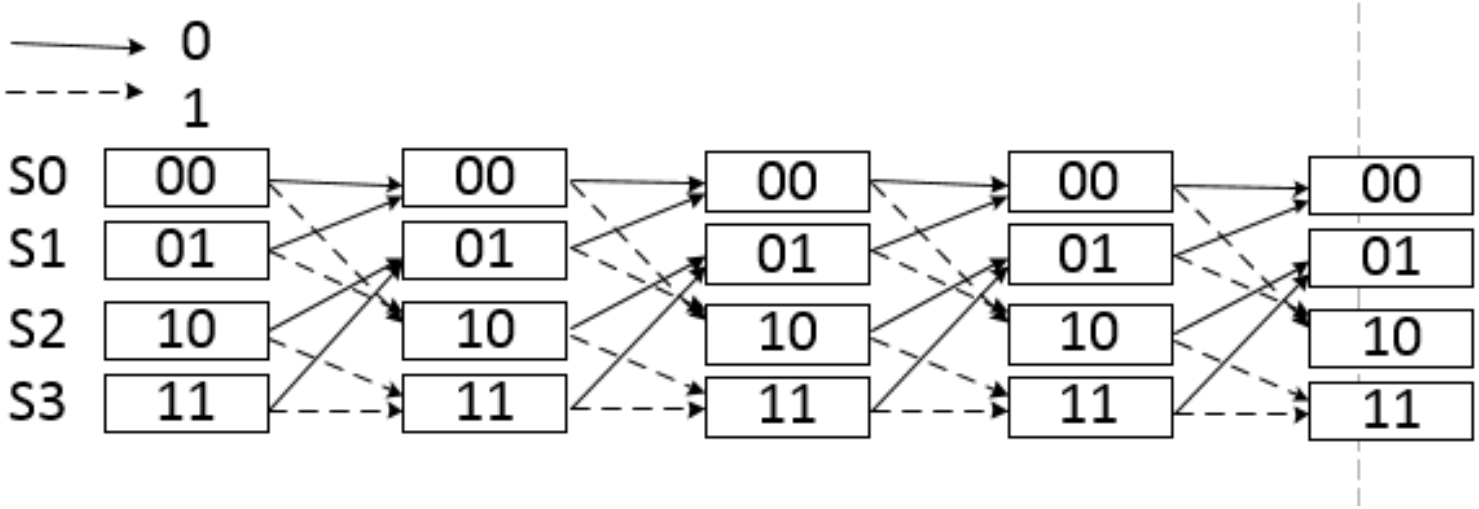


思考题

- **给定接收数据:** R = [**00 10 01 01**]
  - 初始状态: 00
- **问：原始数据是多少**

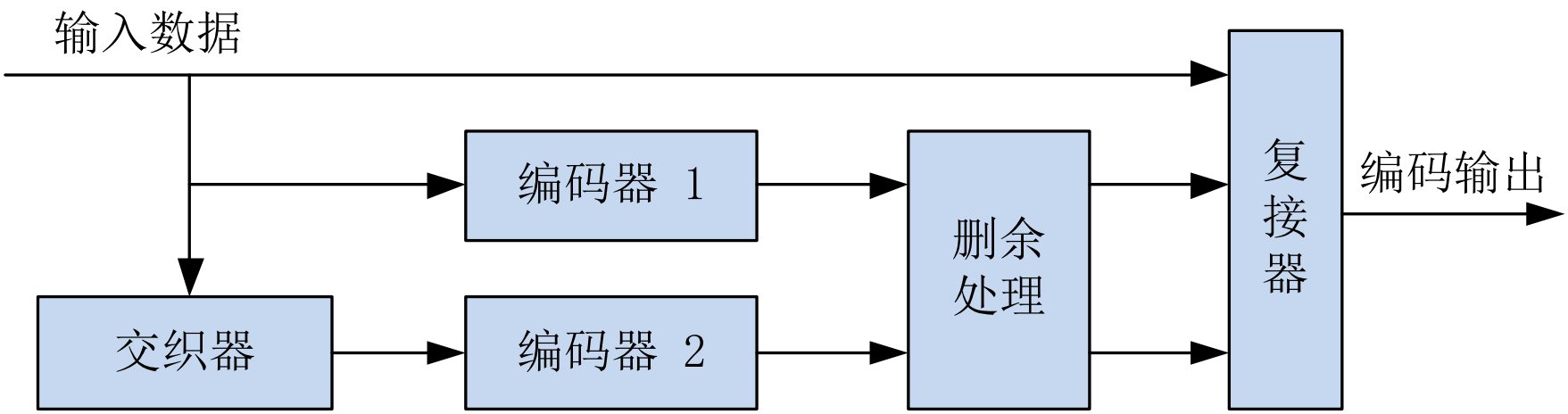


http://blog.csdn.net/DengFengkai12



Turbo code

- **Turbo码又称为并行级联卷积码**
- **由两到多个卷积码编码器级联而成，每个子编码器的输入由交织器分开**
  - 子编码器：产生系统信息和校验信息
  - 交织器：将原始数据按照给定的规则重新排序输出
  - 删余处理：通过删除冗余的校验信息来调整码率
  - 复接器：连接各子编码器的输出序列



Turbo code的特点

- **广泛应用于现有的LTE系统**
  - Turbo code充分利用了Shannon信道编码定理的基本条件，从而获得了接近Shannon理论极限的性能。
- **更好的纠错能力**
  - 通过交织处理，使码重分布合理，降低数据序列的相关性，增大输出码字的最小汉明距，实现随机编码。相比TBCC，有更好的纠错能力。
- **译码较为复杂，译码时延大**
  - 需使用迭代译码方式。相比TBCC，对接收端的译码能力有更高的要求，更适用于长数据包的发送。

NB-IoT随机接入过程

❑ 竞争随机接入过程的四个步骤

- RRC(Radio Resource Control, 无线资源控制)：包含对无线资源进行分配并发送相关控制信令，RRC消息承载了建立、修改和释放链路层和物理层协议实体所需的全部参数。

NB-IoT随机接入过程

- **竞争随机接入过程**
  - Msg1：随机接入前导发送
    - Preamble采用Single-tone方式传输，子载波间隔为3.75kHz，且默认通过调频发送
    - NPRACH（窄带物理随机接入信道）通过重复发送前导码获得覆盖增强，重发次数可选择为 {1,2,4,8,16,32,64,128}

NB-IoT随机接入过程

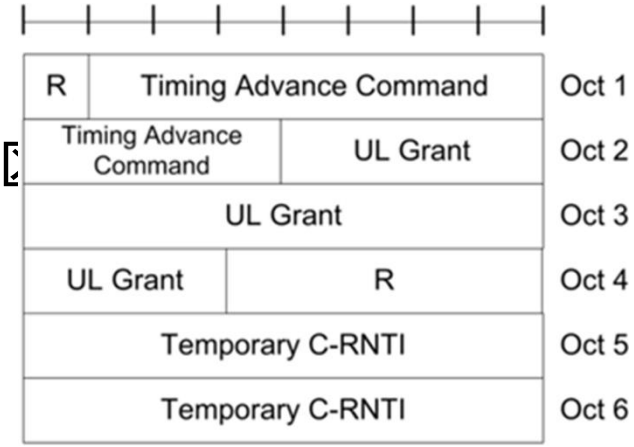
- **竞争随机接入过程**
  - Msg1：随机接入前导发送
    - *设置发射功率*
      - 根据基站BS在广播消息中指示的NPRACH期望接收功率和前导发送次数，进行NPRACH发送功率计算和功率调整过程：
      - TRANSMISSION_POWER = INITIAL_POWER + (TRANSMISSION_COUNTER-1) × POWER_RAMP_STEP

        其中，POWER_RAMP_STEP为功率递增的步长（取值范围与LTE系统相同：{0,2,4,6} dB）
      - 这个过程体现了随机接入在Msg1发送过程中所采用的开环功控的方式确定UE的发射功率，从而提高接入概率。

NB-IoT随机接入过程

❑ 竞争随机接入过程

- Msg2：随机接入响应（Random Access Response，RAR）
  - *在NPDSCH (Narrowband Physical Downlink Share Channel)上发送，其位置由NPDCCH (Narrowband Physical Downlink Control Channel)指示*
  - *携带的信息*
    - RA-preamble identifier
    - Timing alignment information（定时调整量）
    - Msg3传输的UL grant
    - Temporary C-RNTI (Cell-radio network temporary identifier, 小区无线网络临时标识)，是小区内UE的一个动态标识



NB-IoT随机接入过程

- **竞争随机接入过程**
  - Msg2：随机接入响应（Random Access Response，RAR）
    - 在NPDSCH (Narrowband Physical Downlink Share Channel)上发送，其子载波位置由NPDCCH (Narrowband Physical Downlink Control Channel)指示
    - 携带的信息
      - RA-preamble identifier
      - Timing alignment information（定时调整量）
      - Msg3传输的UL grant
      - Temporary C-RNTI
    - UE：发送前导后在一个时间窗口内等待RA response, 如果在时间窗内没有等到属于此UE的响应，认为本次接入失败，则重新发送preamble；否则，进入step3。

NB-IoT随机接入过程

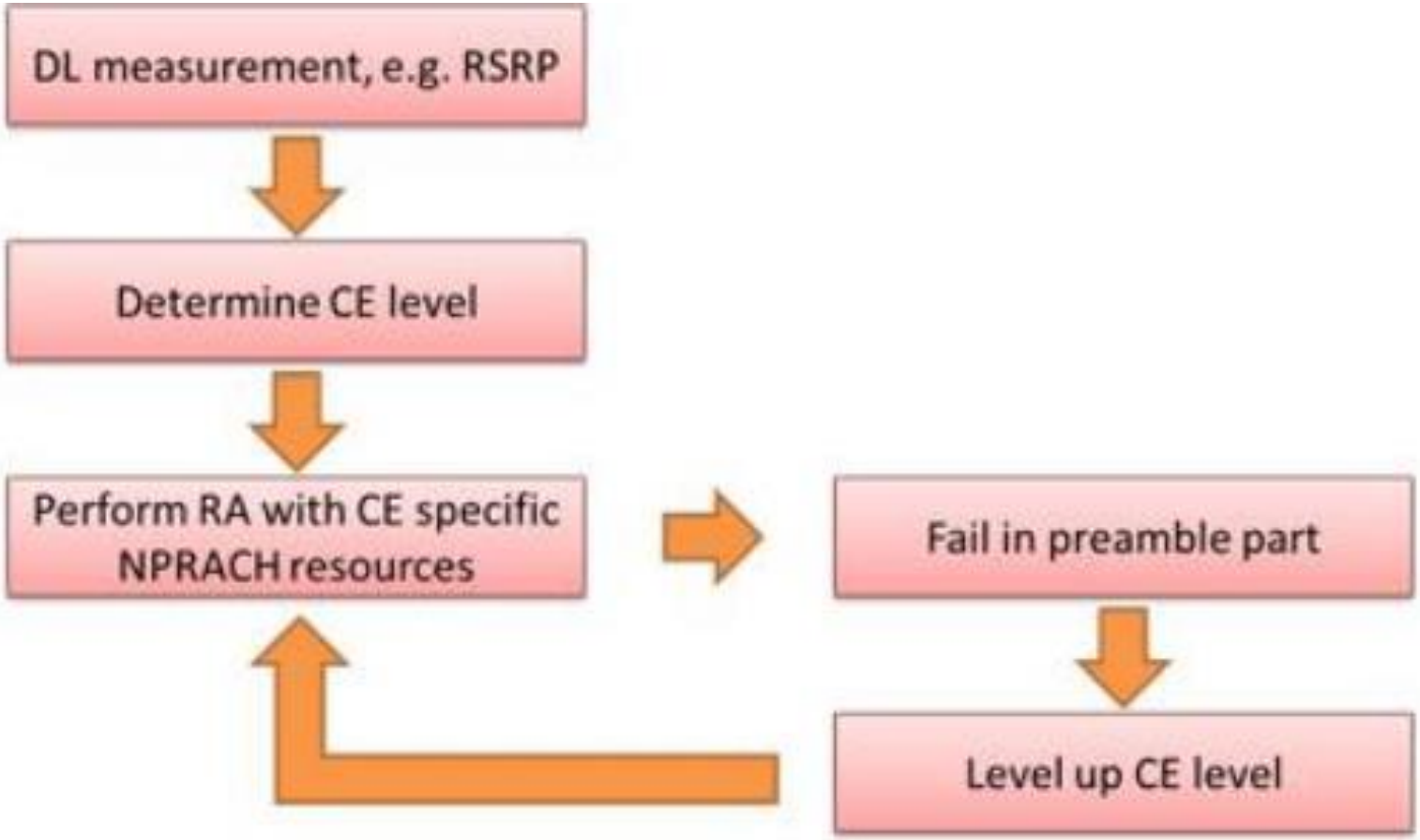
- **竞争随机接入过程**
  - Msg3：上行调度传输
    - 在NPDSCH (Narrowband Physical Downlink Share Channel)上发送，根据RAR响应授权，按指定的资源和格式在NPUSCH (Narrowband Physical Uplink Share Channel)上进行Msg3发送
    - 根据基站BS指示的相关功控参数、路损估计、NPUSCH资源占用等，计算Msg3的开环发射功率（与Msg1类似）
    - 携带的信息
      - 初始接入：传输RRC连接请求
      - RRC连接重建立：传输RRC连接重建请求
      - 其他情况至少传输UE的C-RNTI
    - UE：发送Msg3之后，启动竞争解决定时器等待Msg4

NB-IoT随机接入过程

- **竞争随机接入过程**
  - Msg4：竞争解决
    - BS在MAC层进行竞争检测，并通过在NPDCCH上发送C-RNTI或者在NPDSCH上发送竞争解决ID来指示UE
    - 内容包括
      - UE ID (C-RNTI)
      - 上行信道资源分配情况等
    - UE：在竞争解决定时器超时前，一直监听NPDCCH信道
      - 若在定时器内，检测到自己的C-RNTI，则认为竞争解决成功，并通知上层，上行同步过程结束，等待基站调度，发送上行数据
      - 若在定时器到时，还未收到自己的C-RNTI，UE将认为本次竞争解决失败。失败后，如果UE的随机接入尝试次数小于最大尝试次数，在随机退避一段时间后，重新进行一次随机接入尝试。否则，通知上层随机接入失败。

随机接入过程

- 覆盖增强等级(CE level, coverage enhancement level)估计、传输失败(RAR missing)处理



网络层和传输层

- **NB-IoT 基于phy, mac层可以构建网络层协议**
  - 钛比科技STM32 NB-IoT 开发板提供了IP和UDP网络服务，可以通过AT command 调用
  - AT command其实是个接口标准

- **IP协议相关主要的AT command**
  - AT+NBAND=n; //设置模块工作频段。n=5,8,20分别对应850MHz、900MHz、800MHz
  - AT+CSCON? ; //查询模块连接状态。返回1代表连接状态，0代表IDLE状态。
  - AT+CGPADDR=? ; //查询模块IP地址

网络层和传输层

- **UDP协议相关主要的AT command**
  - AT+NSOCR= DGRAM,17,3000,1; //DGRAM指socket类型，17代表UDP协议, 3000为UDP端口号,1代表能接收下行数据
  - AT+NSOST=0,120.162.113.26,3000,1,40; //0代表UDP协议，120.162.113.26代表服务器IP, 3000为UDP端口号, 1指明发送数据长度为1byte，40为数据内容 (十六进制数，代表ASCCI的10）。最大支持发送数据为512bytes。

案例：NB-IoT上行数据传输

- **1. 通过AT command将节点模块(BC95)与网络服务器建立连接**
  - AT+NBAND=5//设置工作频段为850MHz，电信模组
  - AT+NBAND? //查询模块频段信息，确认设置成功
  - AT+CGATT?=1//小区附着
  - AT+CGATT? //查询是否模块附着成功,返回1代表成功
- **2. 通过AT command将20bytes的数据发送给服务器**
  - AT+NSOCR=DGRAM,17, 3000,1//创建UDP端口3000
  - AT+NSOST=0,120.162.113.26,3000,20,40, 6162632C3017A05C161948D317A02FD5C0000000//发送20bytes数据
  - AT+NUESTATS=ALL//返回模块状态信息，包括收发发功率、收发时间、收发数据大小、收发数据块数量等

案例：NB-IoT上行数据传输

- **3. 协议层包头转换**
  - ① UDP层添加8bytes的首部字段（包括端口号、校验和等）
  - ② IP层添加20bytes~40bytes的首部字段（包括源IP地址、目的IP地址等）
  - ③ 数据链路层首先将压缩数据包头（减小到1~2字节）并添加1~4字节的首部字段（包括数据类型等）
  - 假设: 20bytes的payload数据经过各协议层包头转换后，其长度变为25bytes
- **4. 信道编码**
  - 首先添加冗余字段（3bytes CRC），再使用1/3-turbo code编码
  - 问：编码后总长度变为多少?
- **5. 速率匹配**

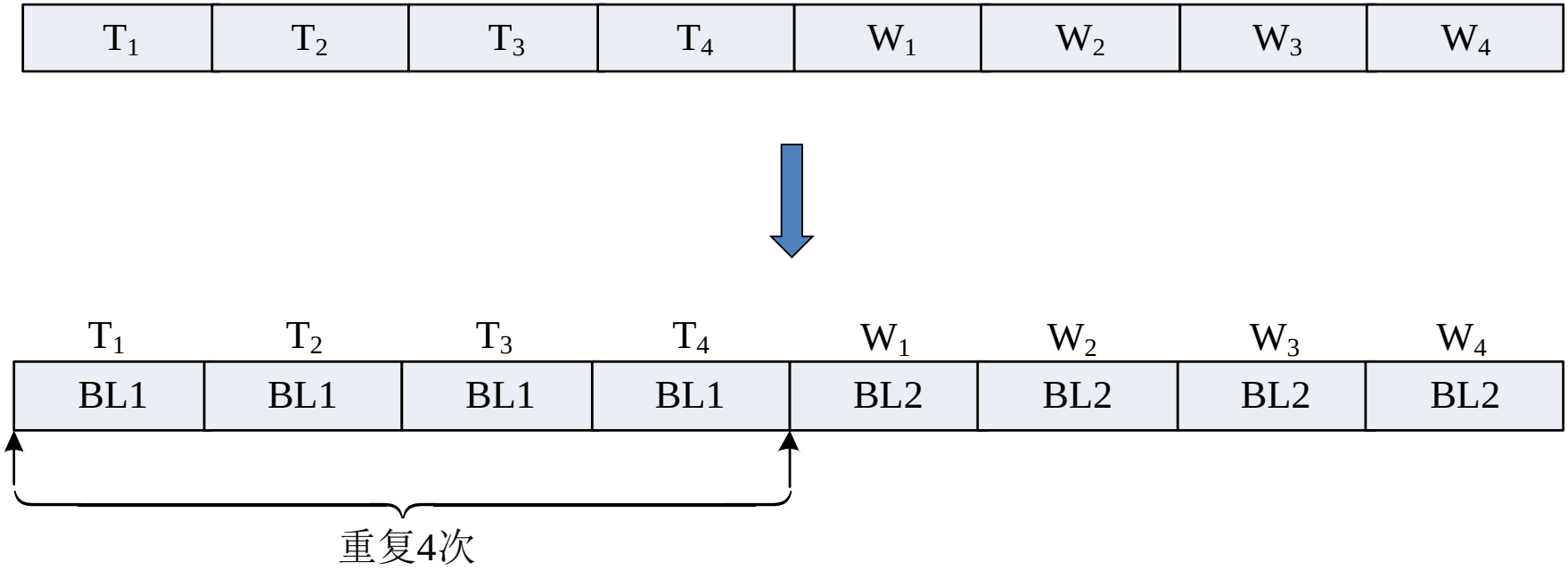
案例：NB-IoT上行数据传输

- **6. 分段**
  - 假设基站分配8个RUs（基站DCI指定），将编码块分成若2个相同大小的编码子块（BL1、BL2）
- **7. 调制**
  - 每个编码子块使用QPSK调制
- **8. 资源映射**
  - 将每个编码块调制后的符号映射到RU上传输
  - 假设UE分配到8个RUs (且每RU占3个subcarriers和8个slots)，同时重复传输次数 N_{REP}=4 (DCI指定)

案例：NB-IoT上行数据传输

8. （续）资源映射

- T_n, W_n: 代表分配的一个RU

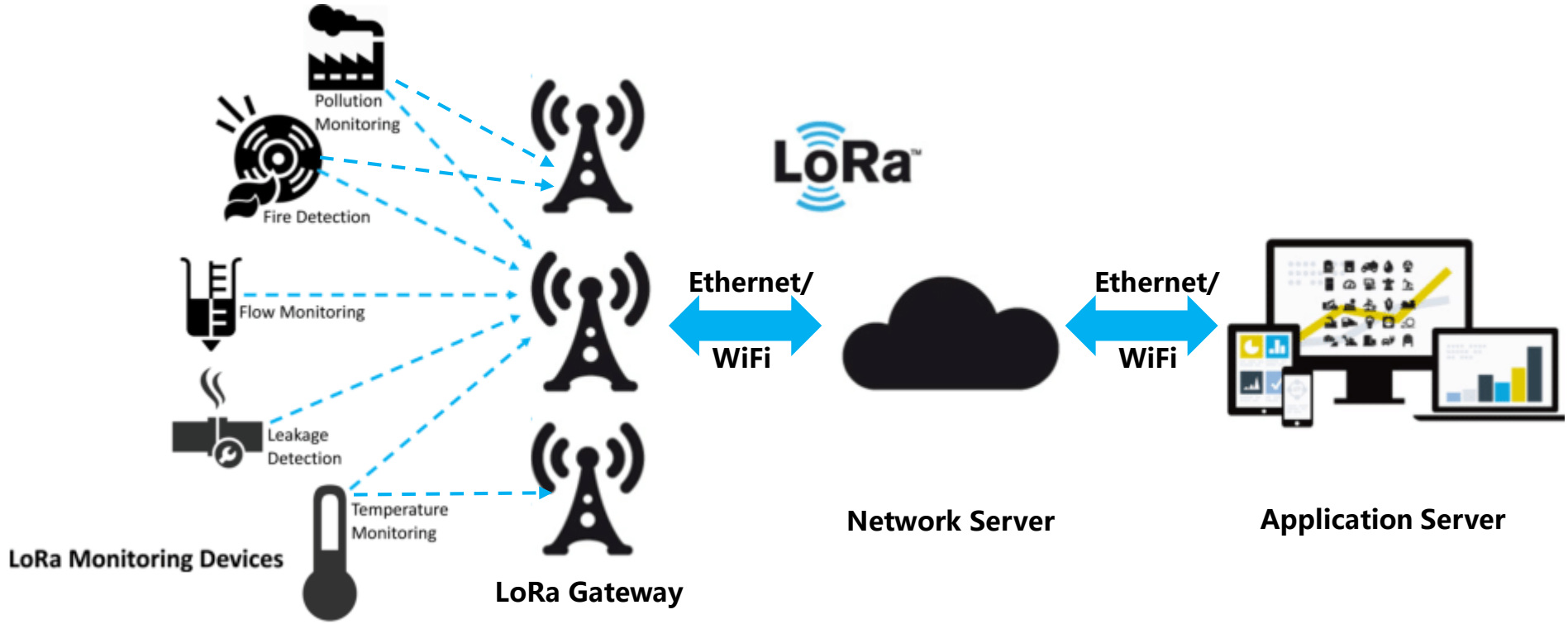


LoRaWAN

- **使用场景**
- **物理层**
  - 频段划分
  - 信道分配
  - 数据调制
- **MAC层**
- **案例**

LoRaWAN使用场景

- 设备通过LoRa与网关通信，网关通过Ethernet/WiFi等与后端网络通信
- 可被多网关接收，由网络服务器挑选合适的网关
- 应用场景
  - 工业园区、智慧校园



LoRaWAN

- 使用场景
- 物理层
  - 频段划分
  - 信道分配
  - 数据调制
- MAC层
- 案例

物理层-LoRa

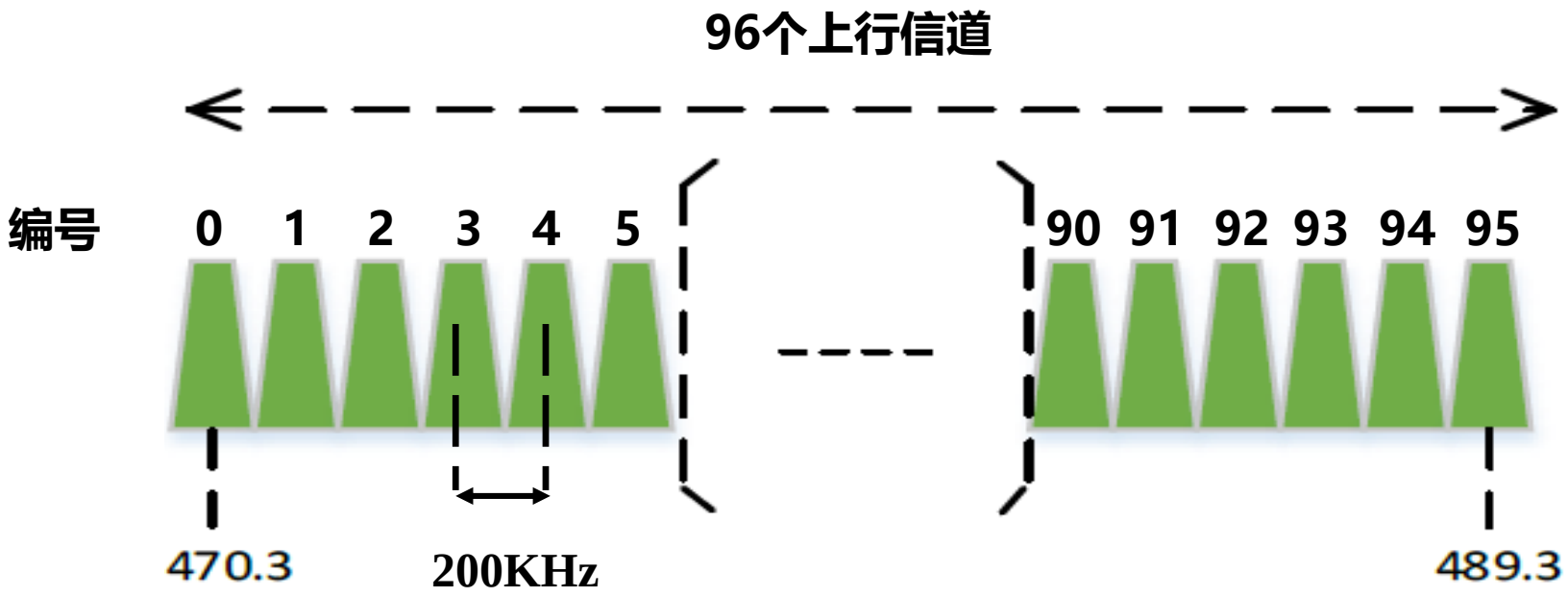
- 频段划分：非授权频段（小于1GHz)

地区	频段分配(MHz)
欧洲	863-870，433
美国	902-928
中国	470-510,779-787
亚洲	923
韩国	920-923
印度	865-867
俄罗斯	864-870
澳大利亚	915-928

物理层-LoRa

❑ 信道划分

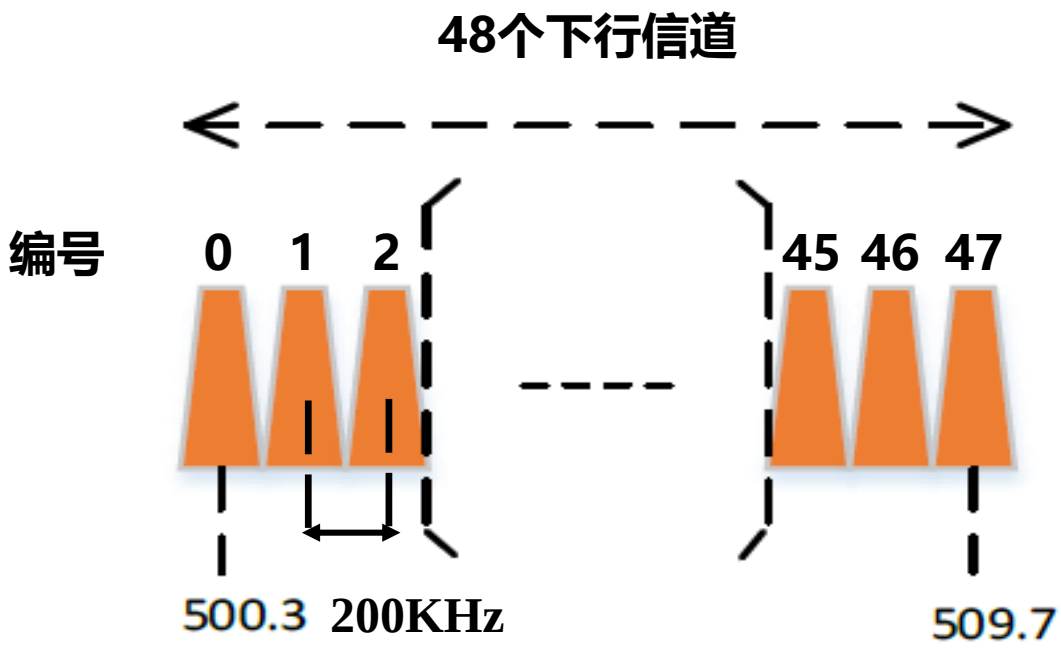
- 频段：470-510MHz（中国，40MHz），每个信道125KHz
- 上行
  - *6-38,45-77信道预留给中国电网，30个可用*



物理层-LoRa

❑ 信道划分

➢ 下行



物理层-LoRa

LoRaWAN中信道分配

- 上行
  - *网关发送命令，配置设备的通信信道*
    - LinkADRReq <ChMaskCntl>//网关配置设备可用信道
    - LinkADRAns <Channel mask ACK>//设备回复确认信息

ChMaskCntl	信道
0	0-15
1	16-31
2	32-47
3	48-63
4	64-79
5	80-95
6	所有信道开启
7	保留

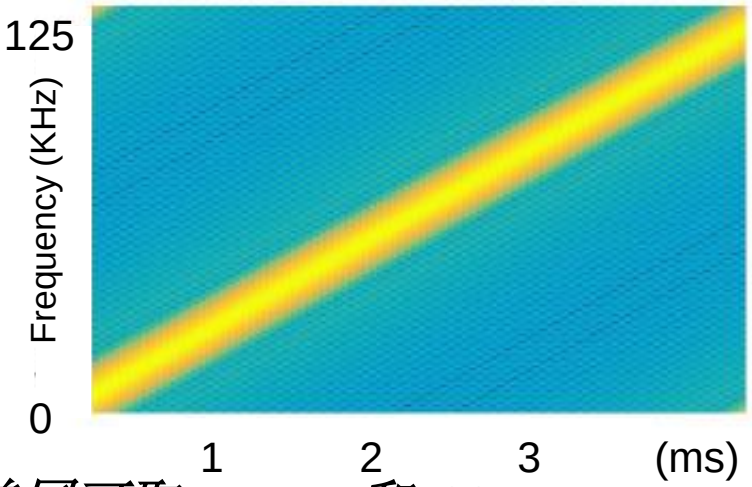
物理层-LoRa

- **LoRaWAN中信道分配**
  - 下行
    - *网关发送命令，配置设备的通信信道*
      - RXParamSetupReq <Frequency>//网关配置设备可用信道
      - RXParamSetupAns <Channel ACK>//设备回复确认信息

物理层-LoRa

❑ 扫码序列扩频（Chirp）调制

- Chirp信号：频率随时间线性变化的一种信号

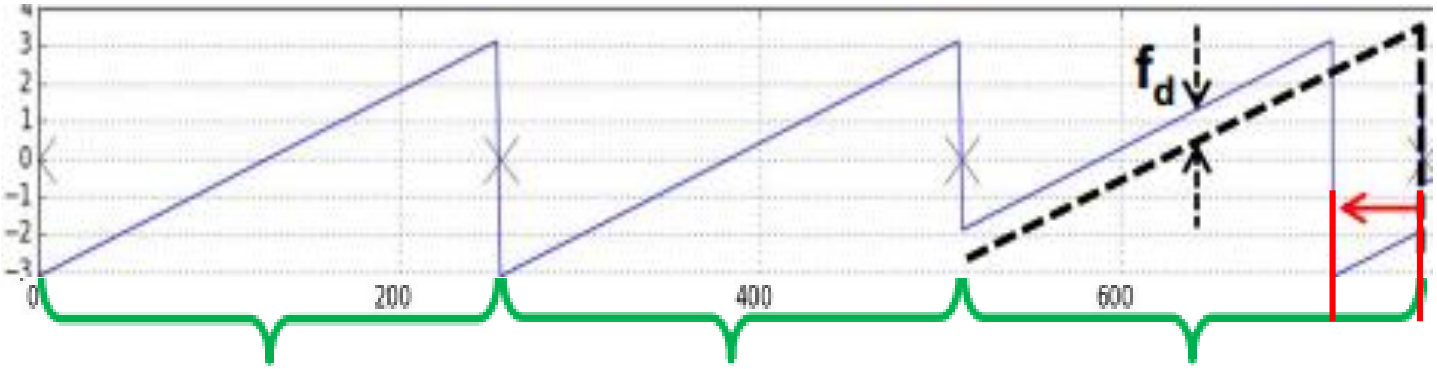


- Chirp调制关键参数
  - ***BW：带宽，中国取125KHz，美国可取125KHz和500KHz***
  - ***SF：spreading factor，一个chirp表示SF个bits，7-12***
  - T_s***：一个chirp发送时间***
    - $T_s = \frac{2^{SF}}{BW}$
- 通过起始频率所在纵轴的offset调制数据

物理层-LoRa

❑ 调制数据案例

- SF=8，BW=125KHz发送数据[0, 0, 50]
- Chirp频率随时间线性变化
  - *频率偏移等价于时间偏移*



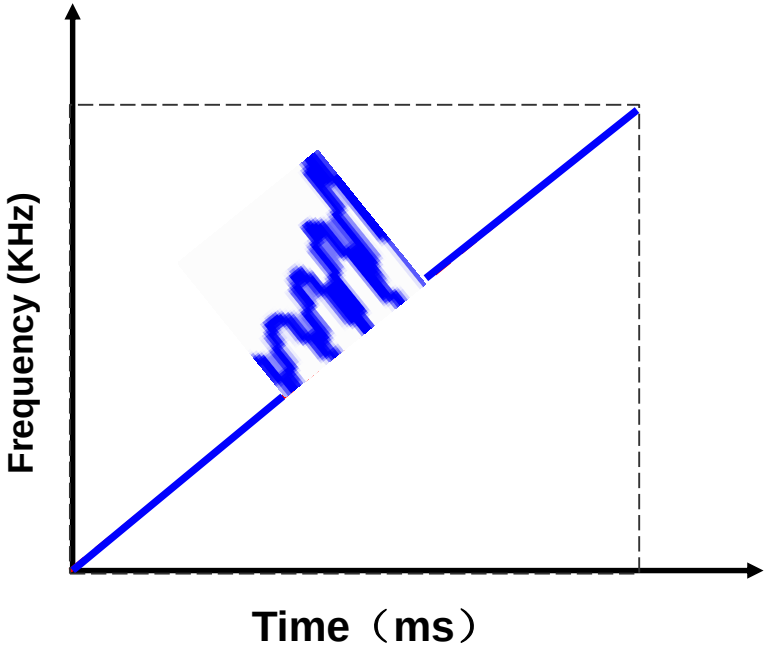
偏移0/256Ts **偏移0/256Ts** **偏移50/256Ts**

思考题

- **设备发送数据50**
  - 入网
    - ***OTAA或ABP***
  - 参数设置
    - ***SF=10，BW=125KHz，CR=4/5***
      - T_s？ R_b？
  - 调制
    - ***offset=？***

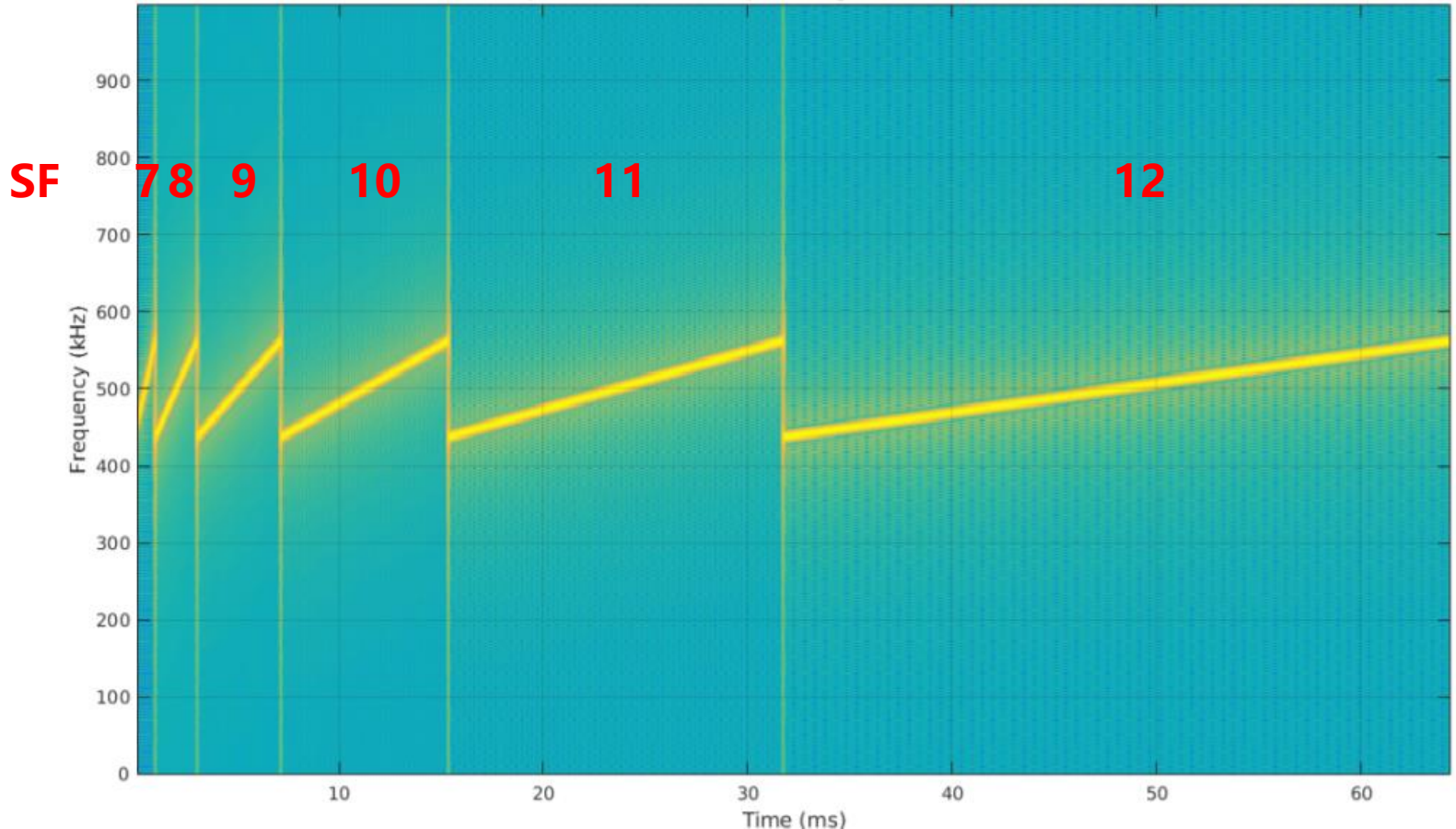
物理层-LoRa

- **Chirp扩频调制优势**
  - 鲁棒性：对抗多径和多普勒频移
    - *轻微的频率抖动可以恢复*



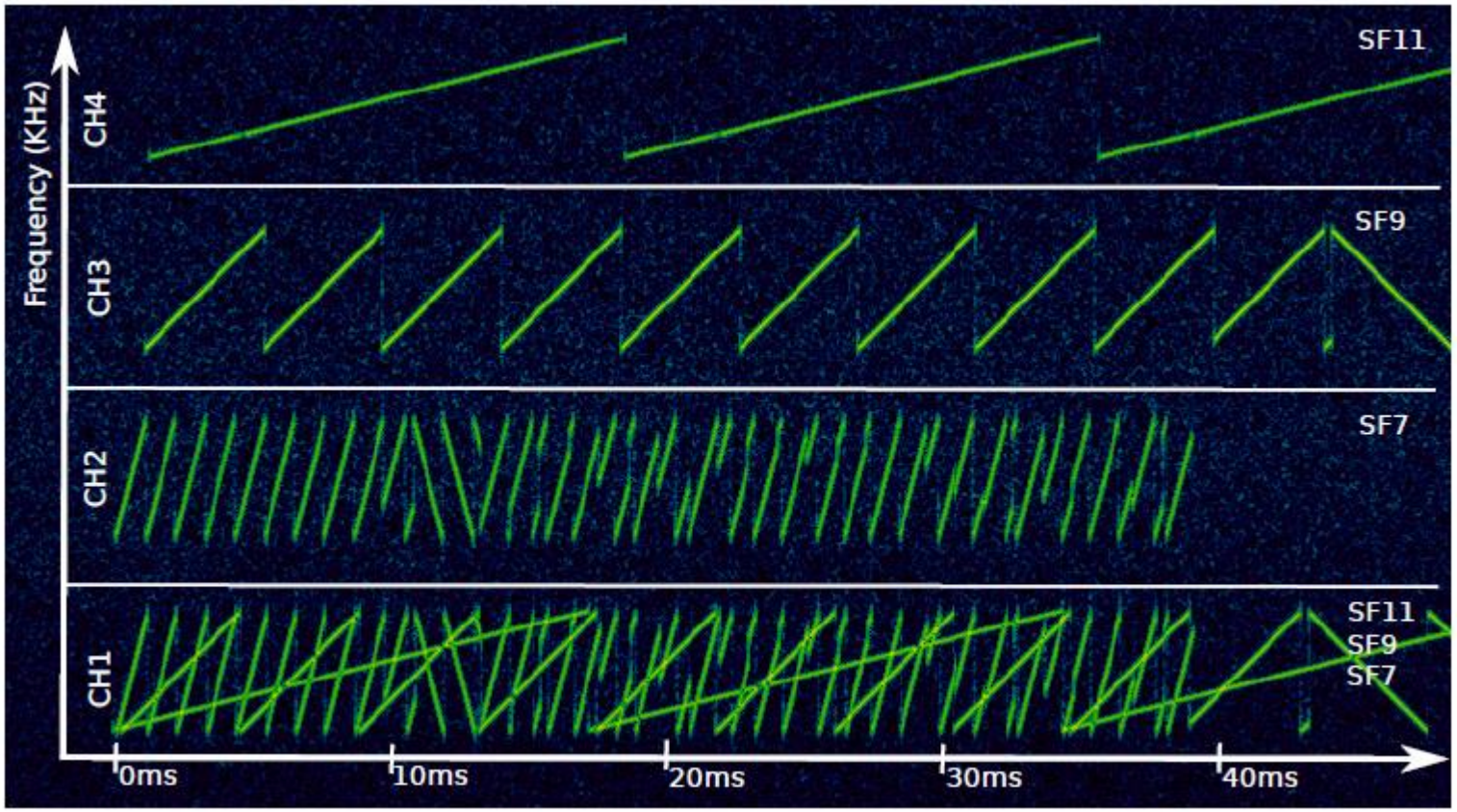
物理层-LoRa

- **Chirp扩频调制优势**
  - 提升传输并发性
    - *同一信道中，6个用户选择6个不同的SF，则他们相互正交，可并发传输*



物理层-LoRa

- **Chirp扩频调制优势**
  - 提升传输并发性
    - *CH1上同时收到3个用户发送的SF分别为7、9、11的数据，仍可区分*



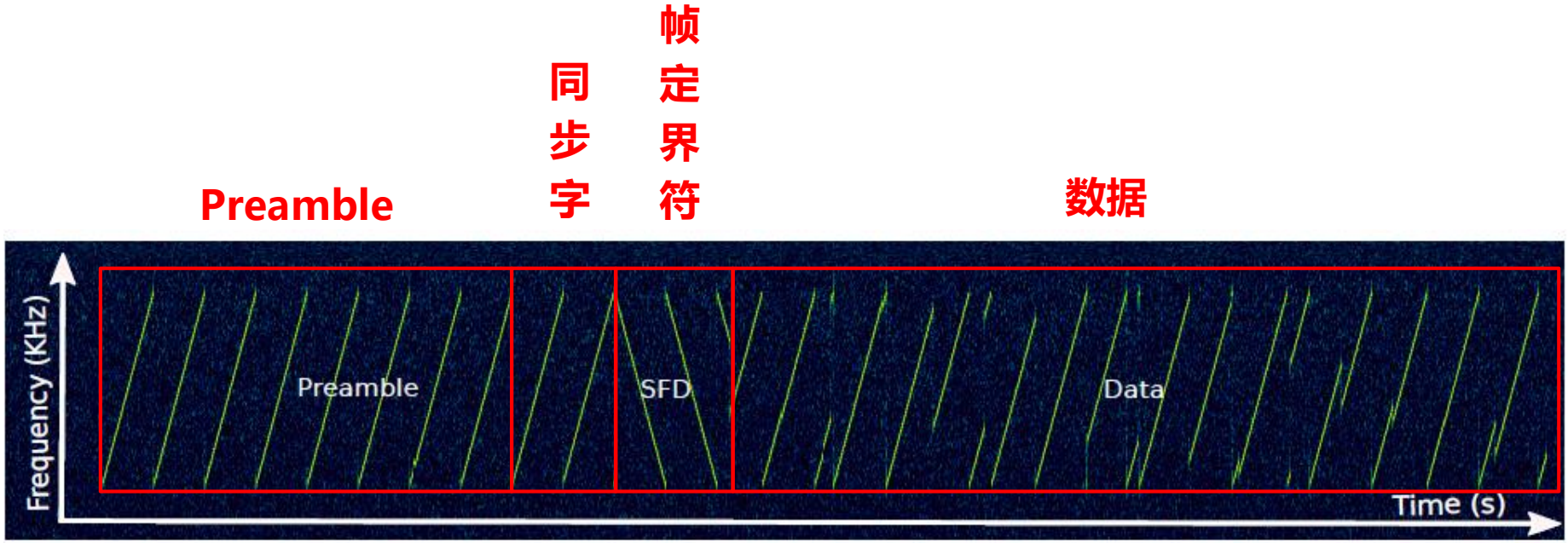
物理层-LoRa

- **Chirp扩频调制优势**
  - 扩大传输距离
    - *香农公式*$D = BW * \log_2(1 + SNR)$*表明，当传输速率**D**一定时，可通过增大带宽**BW**，降低对信道信噪比**SNR**的要求。*
    - ***接收端灵敏度=-174 + 10lg(BW) + NF + SNR=-137dBm***
      - BW: 125kHz
      - NF: 一般为6dB
      - SNR: LoRa解调最低为-20dB
    - ***链路预算=发送功率 – 接收端灵敏度=157dB***
      - LoRa最大发送功率为20dBm
    - ***相同条件下，GPRS/GSM的MCL为：144dB，因此有13dB增益***

物理层-LoRa

包结构

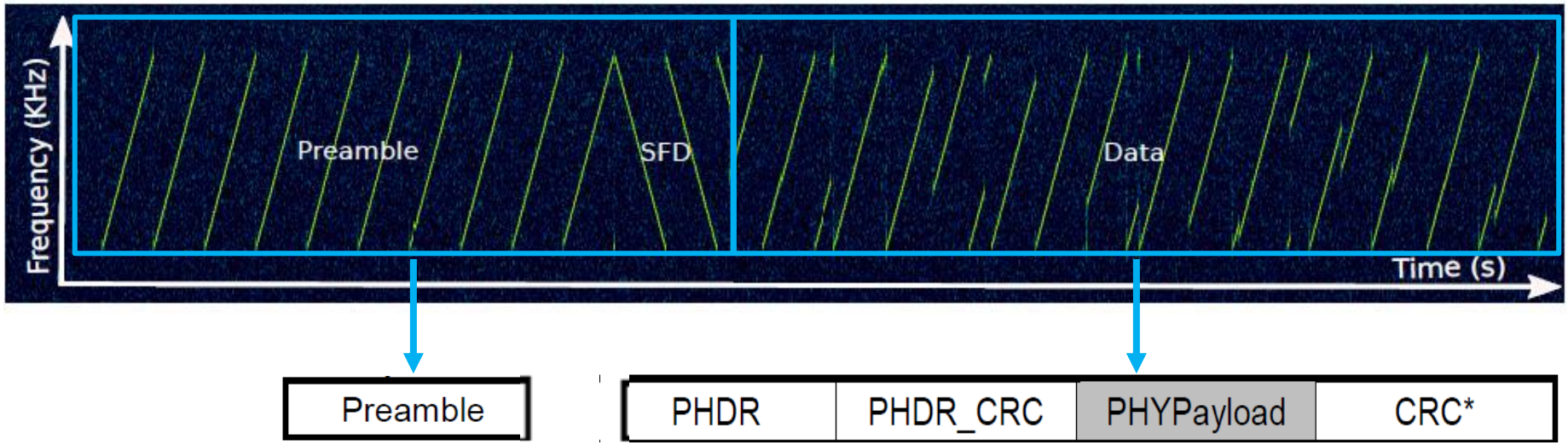
- Preamble：up-chirp信号
- 同步字： 2个up-chirp信号
- 帧定界符：2.25个dwon-chirp信号



物理层-LoRa

❑ 包结构

- Preamble：up-chirp信号
- 同步字： 2个up-chirp信号
- 帧定界符：2.25个dwon-chirp信号



LoRaWAN

- 使用场景
- 物理层
  - 频段划分
  - 信道分配
  - 数据调制
- MAC层
- 案例

入网激活

- **在发送数据之前，设备需要入网激活**
- **空中激活（OTAA）和自定义（ABP）两种方式**
  - OTAA需要入网流程，与服务器协商，获取设备地址和会话密钥
  - ABP则事先将必需的通信信息，烧录在设备上

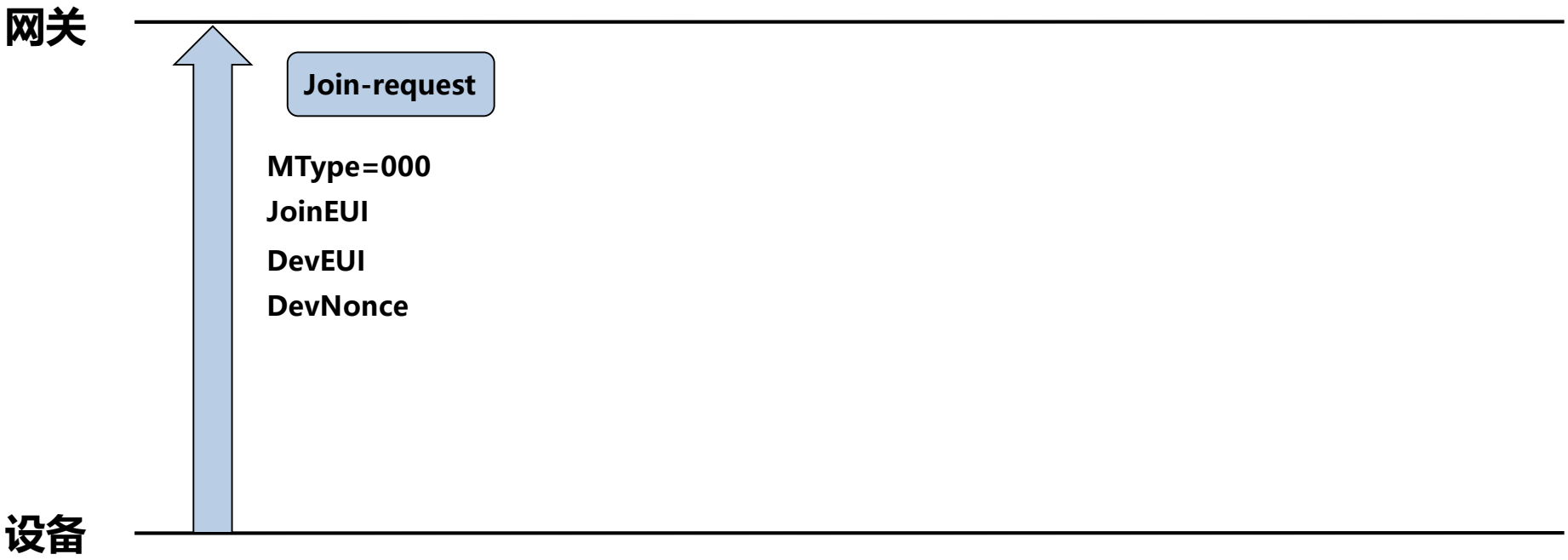
入网前

- **JoinEUI**
  - 8字节长地址，OTAA中唯一标识处理入网过程的入网服务器
- **DevEUI**
  - 8字节长地址，唯一标识设备
- **NwkKey**
  - AES-128密钥，制造时分配给设备
  - OTAA中用于生成网络会话密钥FNwkSIntKey、SNwkSIntKwy和NwkSEncKey
- **AppKey**
  - AES-128密钥，制造时分配给设备
  - OTAA中用于生成应用会话密钥AppSKey

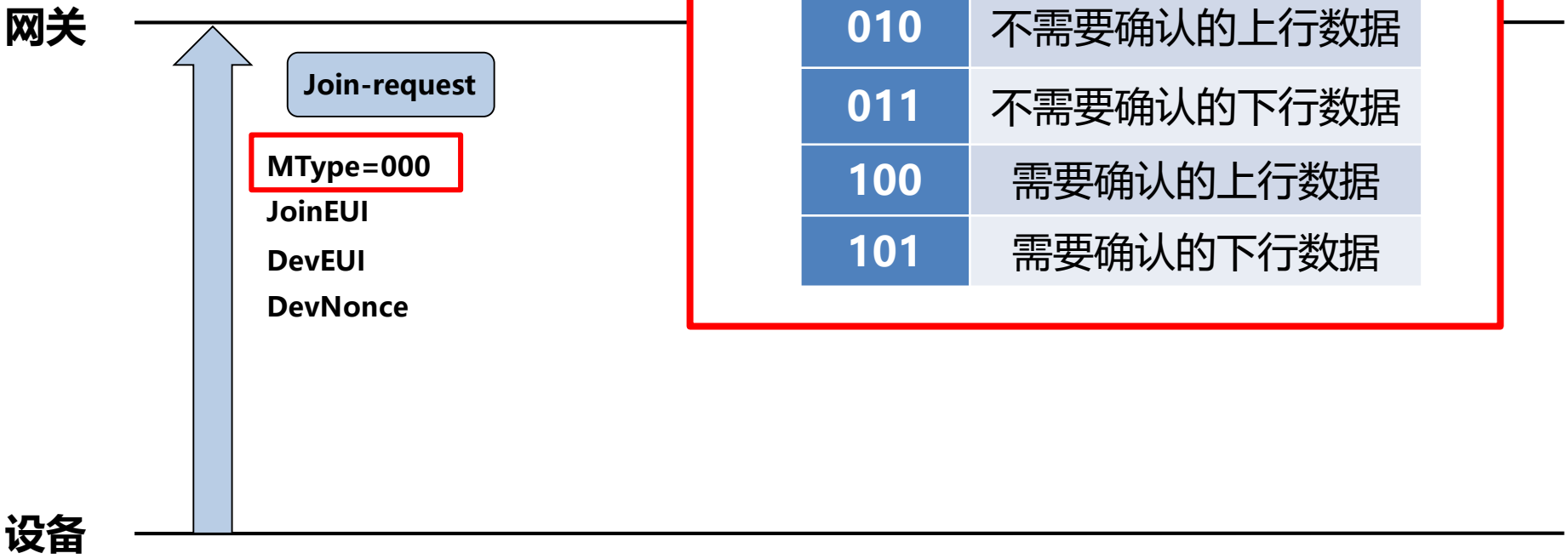
入网后

- **DevAddr**
  - 4字节设备地址，用于当前网络
- **FNwkSIntKey**
  - 计算上行数据的MIC
- **SNwkSIntKey**
  - 计算上行数据的MIC
  - 检查下行数据的MIC
- **NwkSEncKey**
  - 加密和解密上下行的MAC指令
- **AppSKey**
  - 加密和解密上下行的应用数据

入网流程

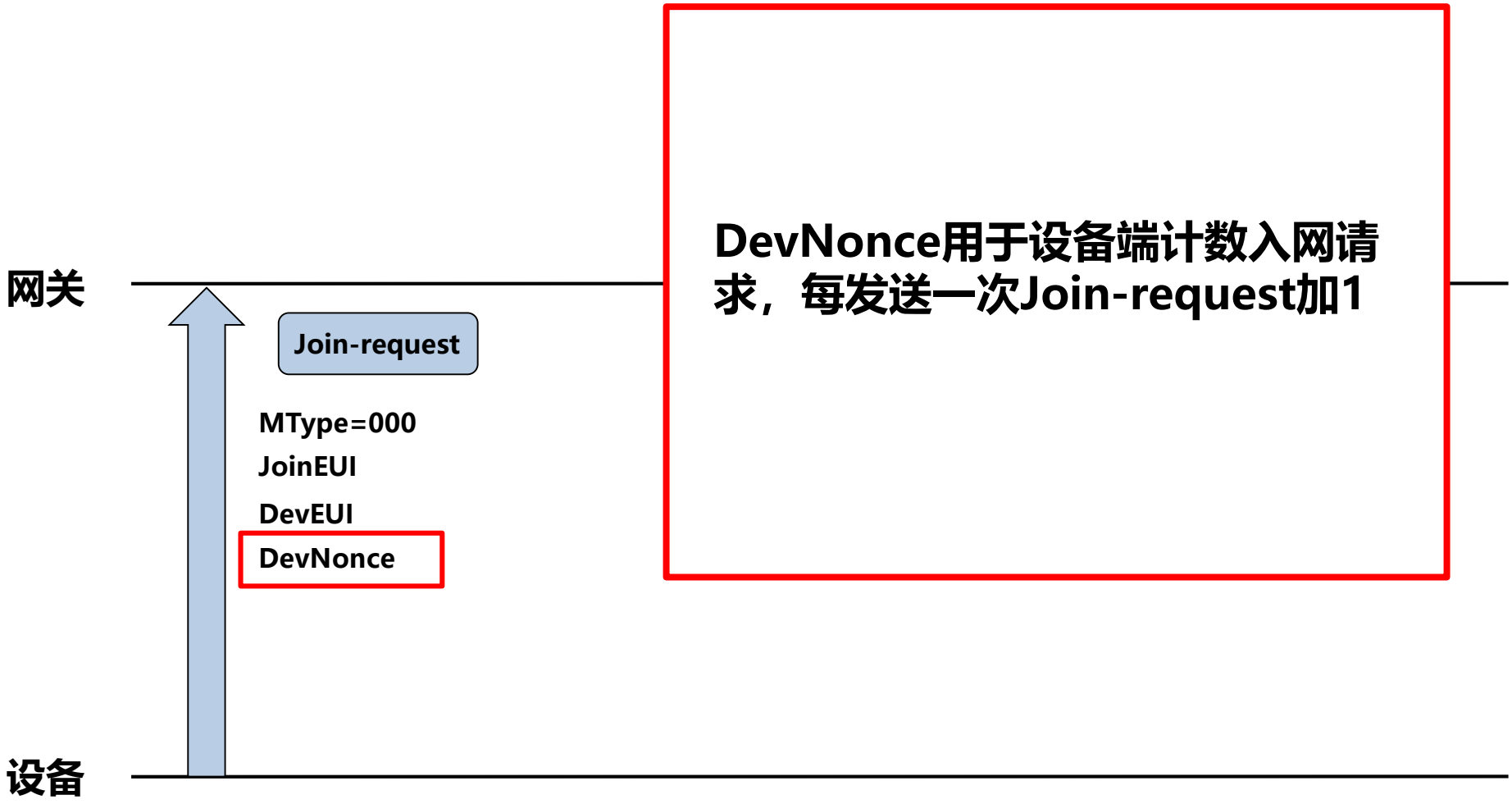


入网流程



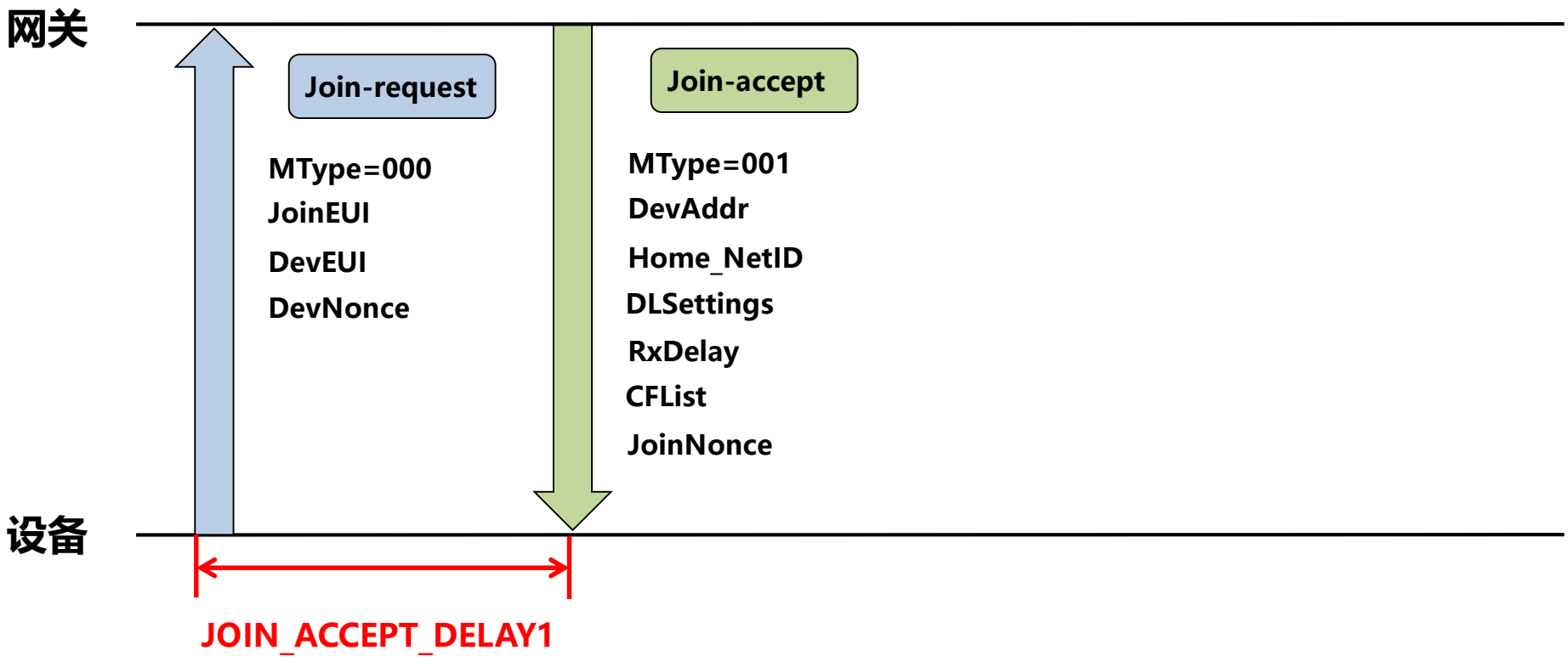
MType	消息类型
000	入网请求
001	入网许可
010	不需要确认的上行数据
011	不需要确认的下行数据
100	需要确认的上行数据
101	需要确认的下行数据

入网流程

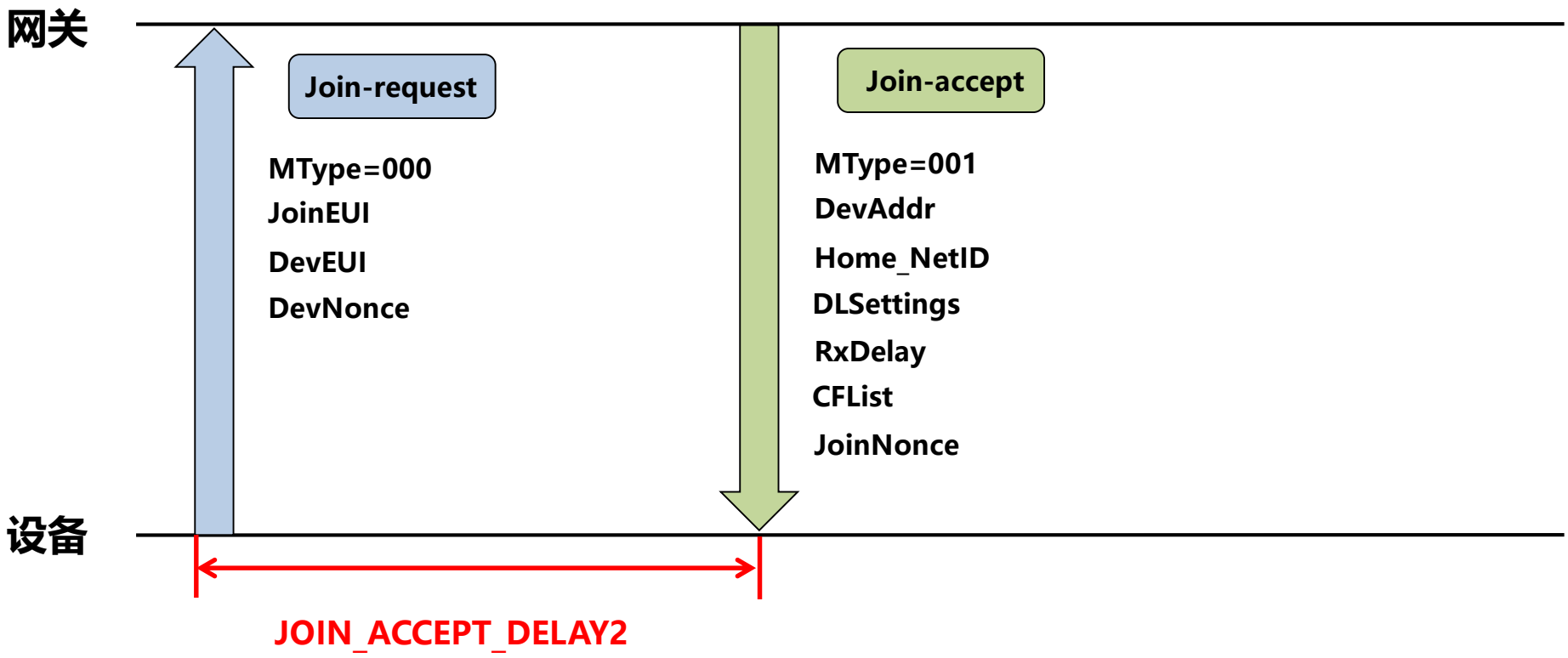


DevNonce用于设备端计数入网请求，每发送一次Join-request加1

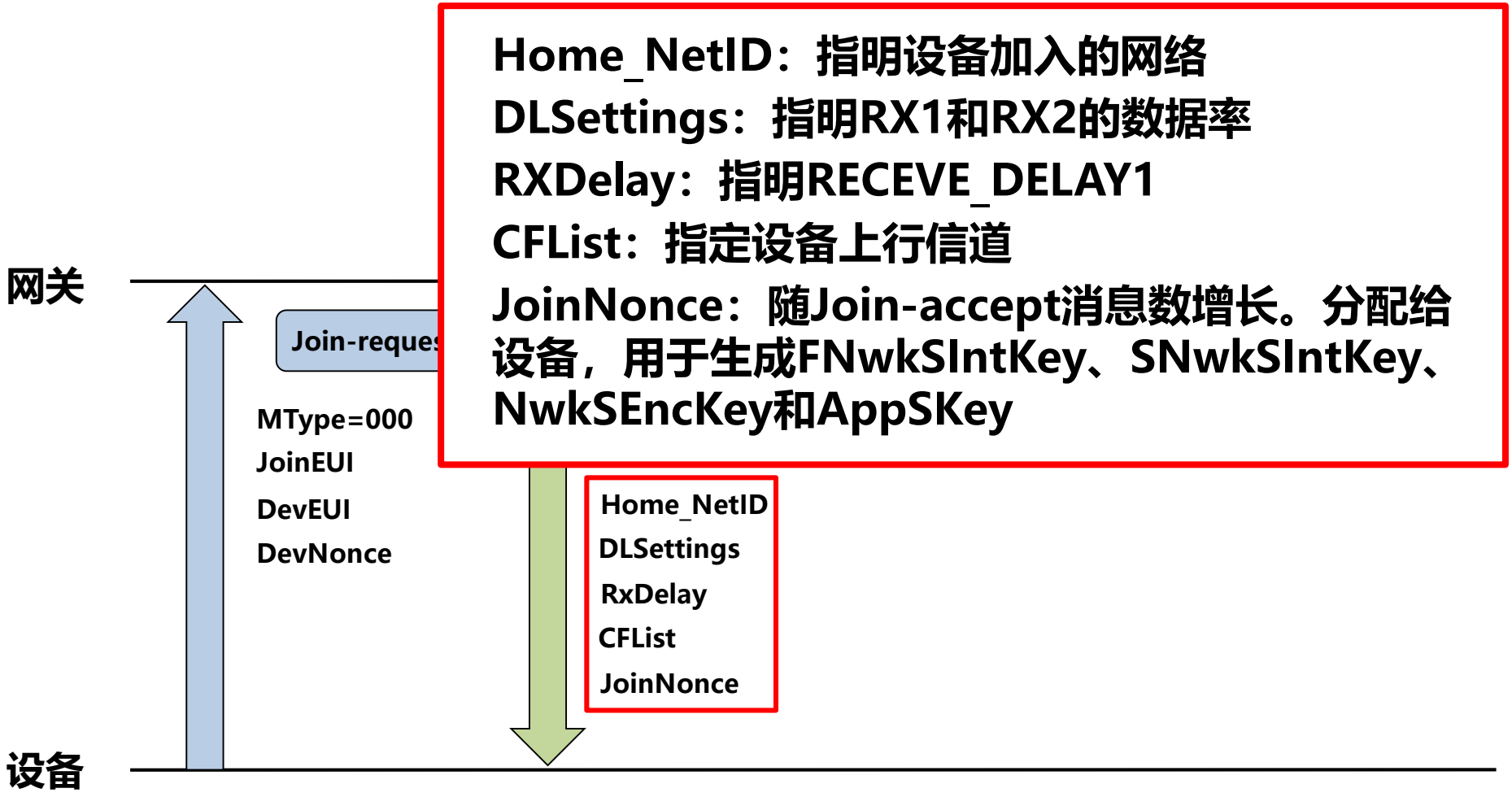
入网流程


网关
Join-request
MType=000
JoinEUI
DevEUI
DevNonce
Join-accept
MType=001
DevAddr
Home_NetID
DLSettings
RxDelay
CFList
JoinNonce
设备
JOIN_ACCEPT_DELAY1

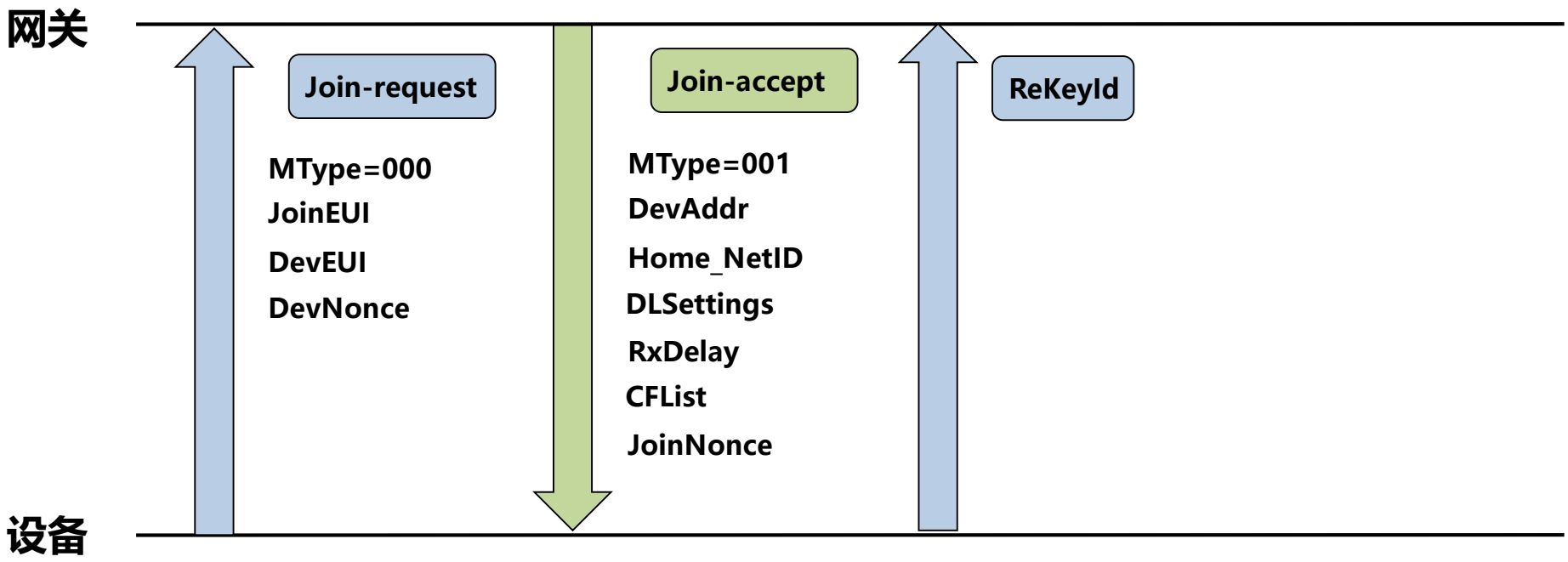
入网流程


网关
Join-request
MType=000
JoinEUI
DevEUI
DevNonce
Join-accept
MType=001
DevAddr
Home_NetID
DLSettings
RxDelay
CFList
JoinNonce
设备
JOIN_ACCEPT_DELAY2

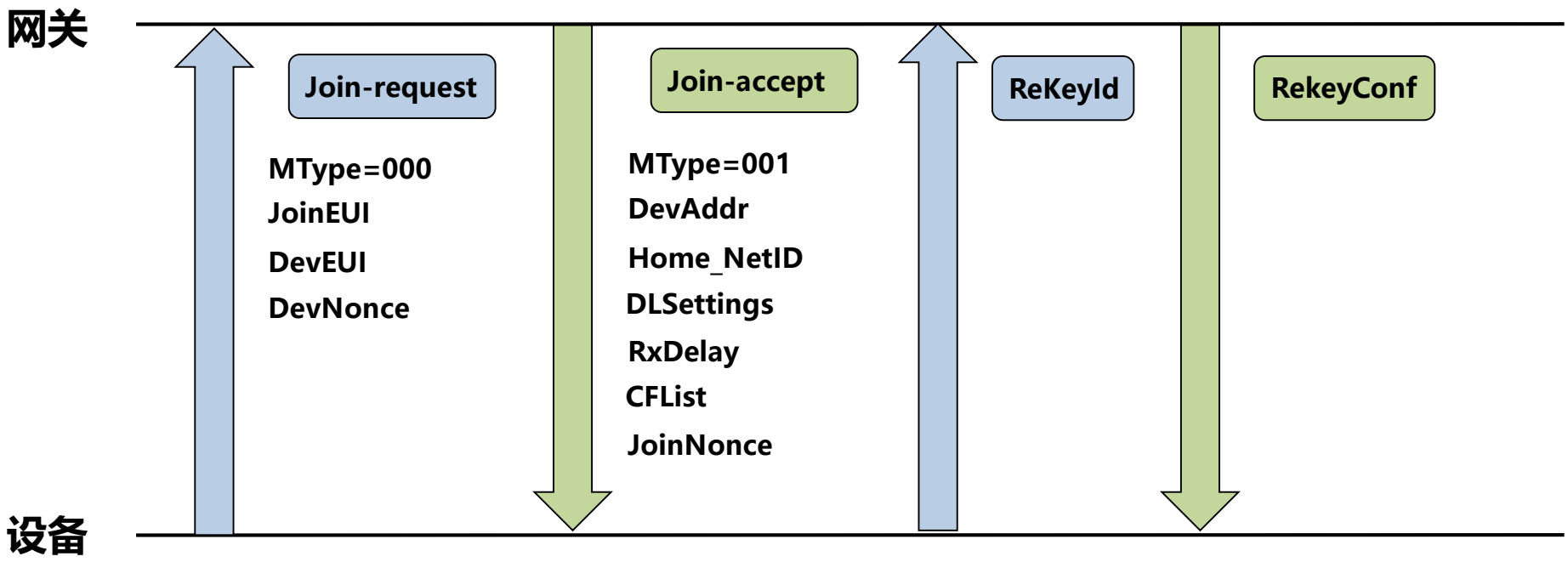
入网流程


Home_NetID：指明设备加入的网络
DLSettings：指明RX1和RX2的数据率
RXDelay：指明RECEVE_DELAY1
CFList：指定设备上行信道
JoinNonce：随Join-accept消息数增长。分配给设备，用于生成FNwkSIntKey、SNwkSIntKey、NwkSEncKey和AppSKey
网关
Join-reques
MType=000
JoinEUI
DevEUI
DevNonce
Home_NetID
DLSettings
RxDelay
CFList
JoinNonce
设备

入网流程

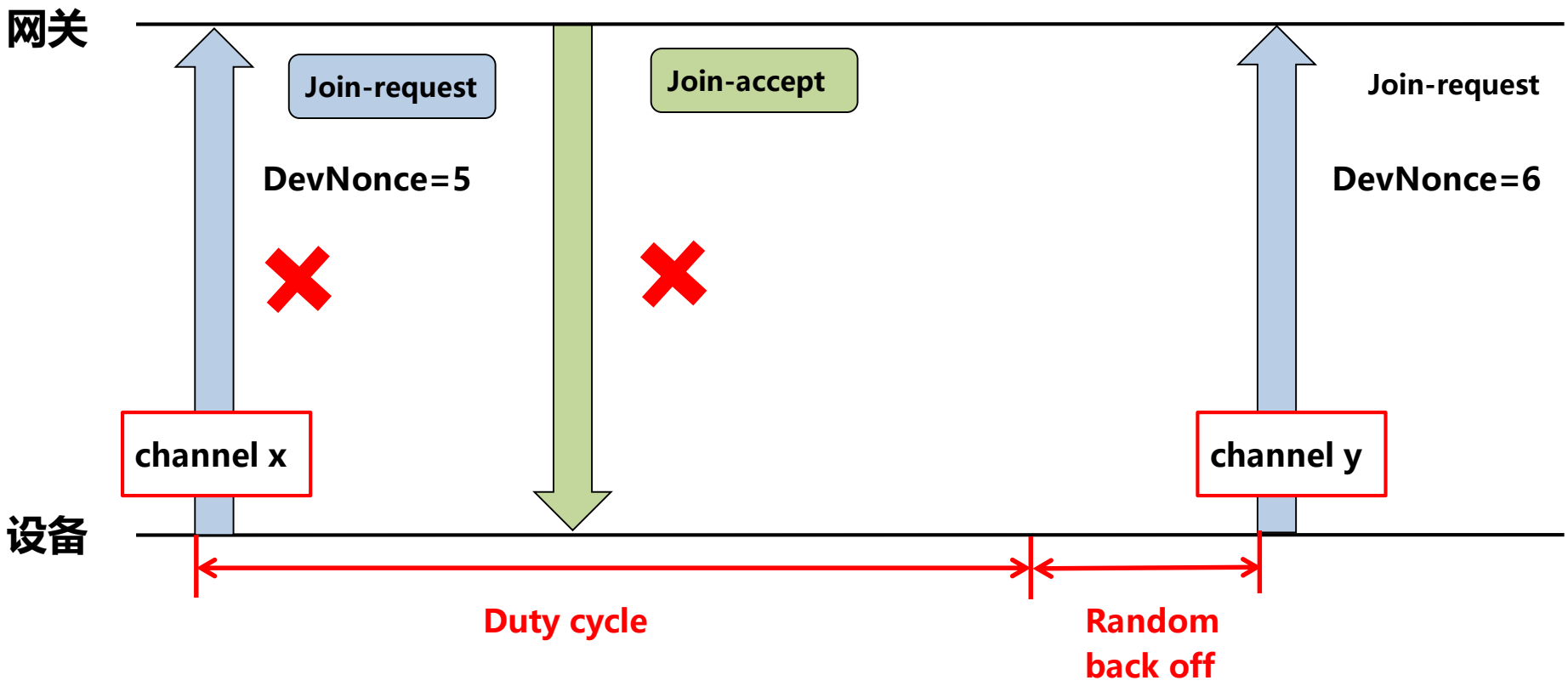

网关
Join-request
MType=000
JoinEUI
DevEUI
DevNonce
Join-accept
MType=001
DevAddr
Home_NetID
DLSettings
RxDelay
CFList
JoinNonce
ReKeyId
设备

入网流程


网关
设备
Join-request
MType=000
JoinEUI
DevEUI
DevNonce
Join-accept
MType=001
DevAddr
Home_NetID
DLSettings
RxDelay
CFList
JoinNonce
ReKeyId
RekeyConf

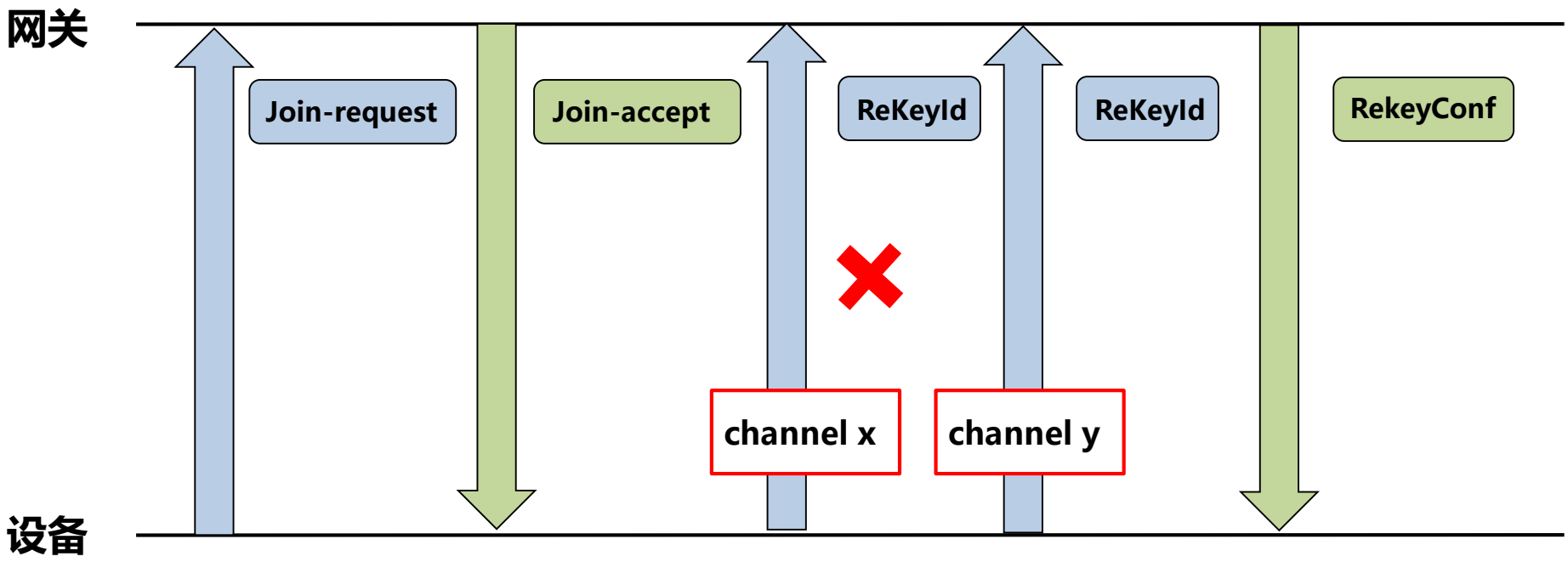
入网流程

- 未收到Join-accept



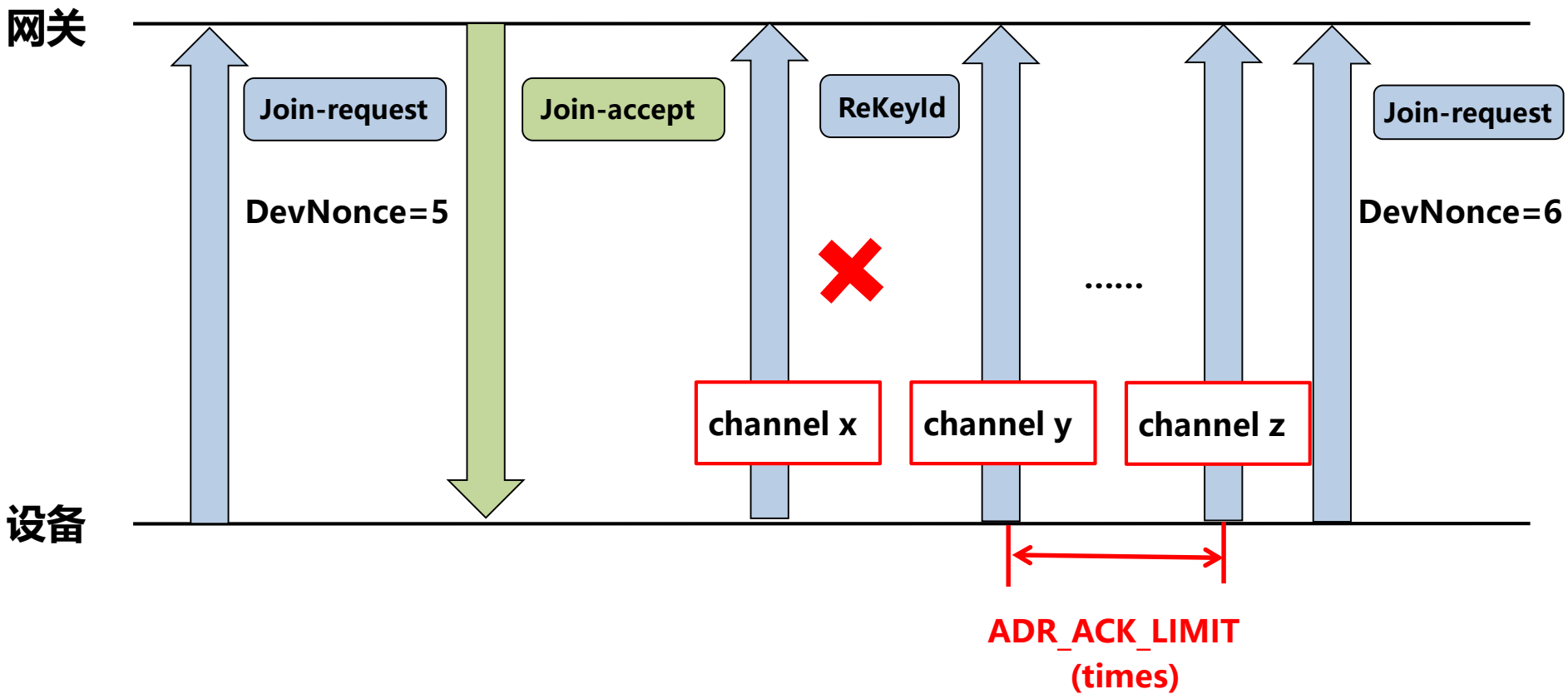
入网流程

❑ 未收到RekeyConf



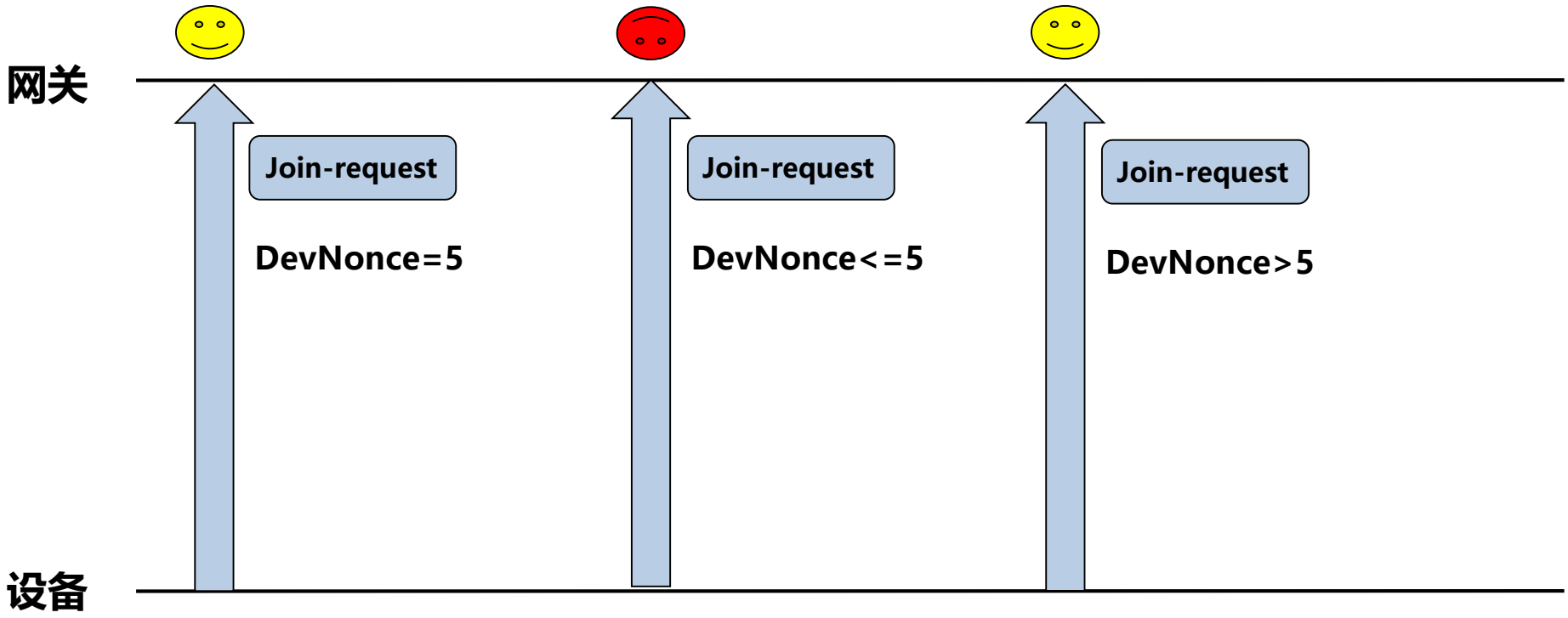
入网流程

❑ 未收到RekeyConf



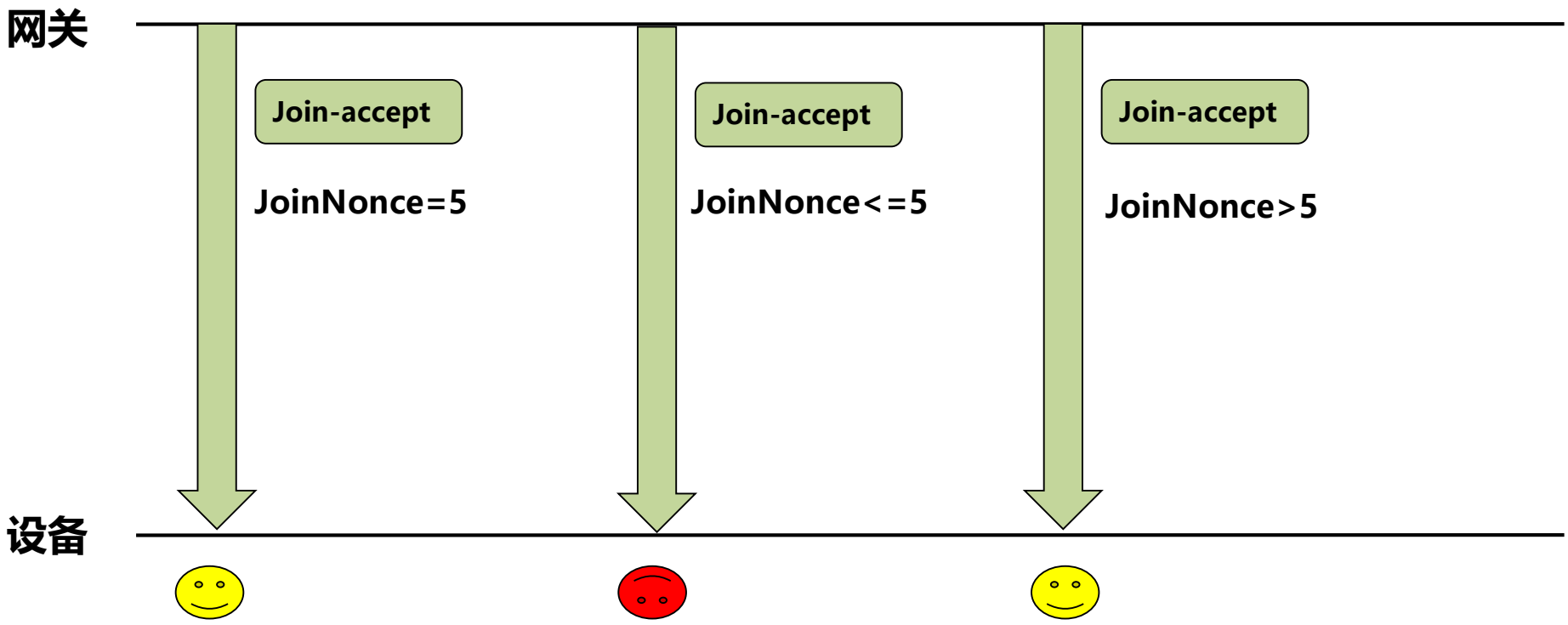
入网流程

- 重放攻击
  - 网关侧



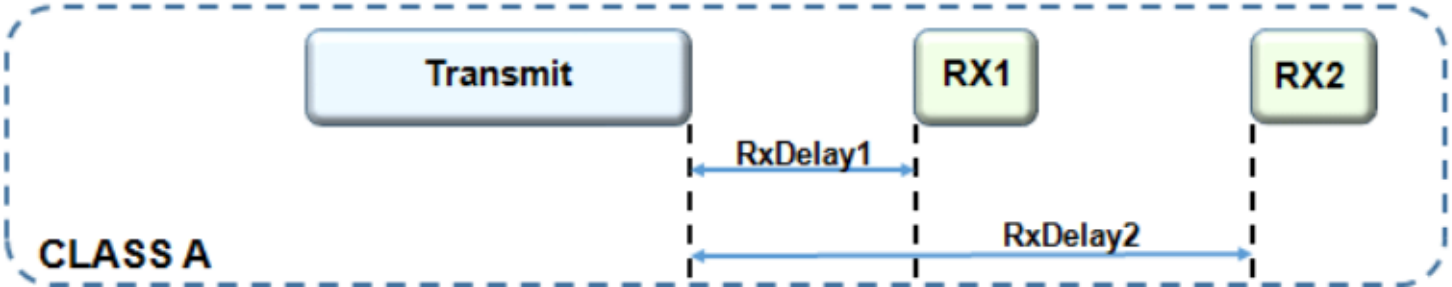
入网流程

- 重放攻击
  - 设备侧



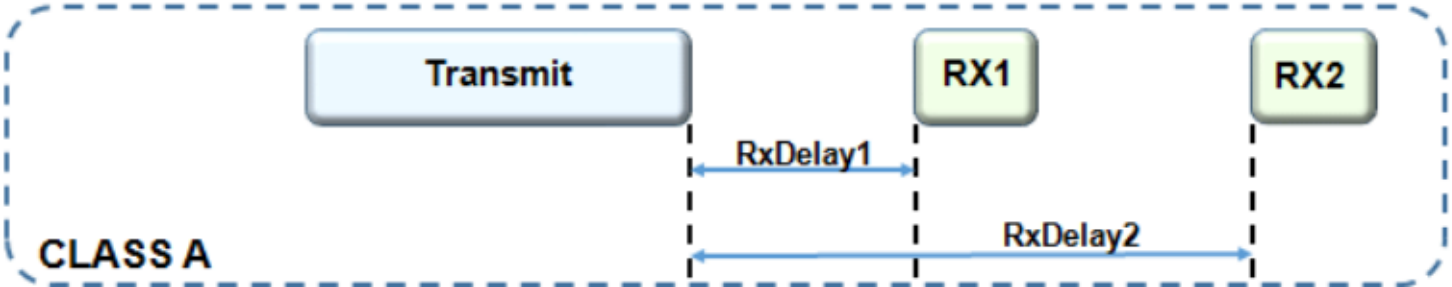
Class A

- 设备**想发即发**，发送的信道和数据率在入网过程中，由网关指定
- RXDelay1和RXDelay2在不同地区的频段各有不同。通常默认为上行传输完成后的1s和2s
- RX1的信道频率和数据率与上行传输相关
- RX2的信道频率和数据率由网关指定
- 网关在RX1和RX2的起始处发送数据
- 接收窗口长度至少保证设备能检测到5个symbol。没有则休眠，有则进行数据的接收、解调、地址匹配和一致性校验
- RX1成功接收数据，RX2不打开



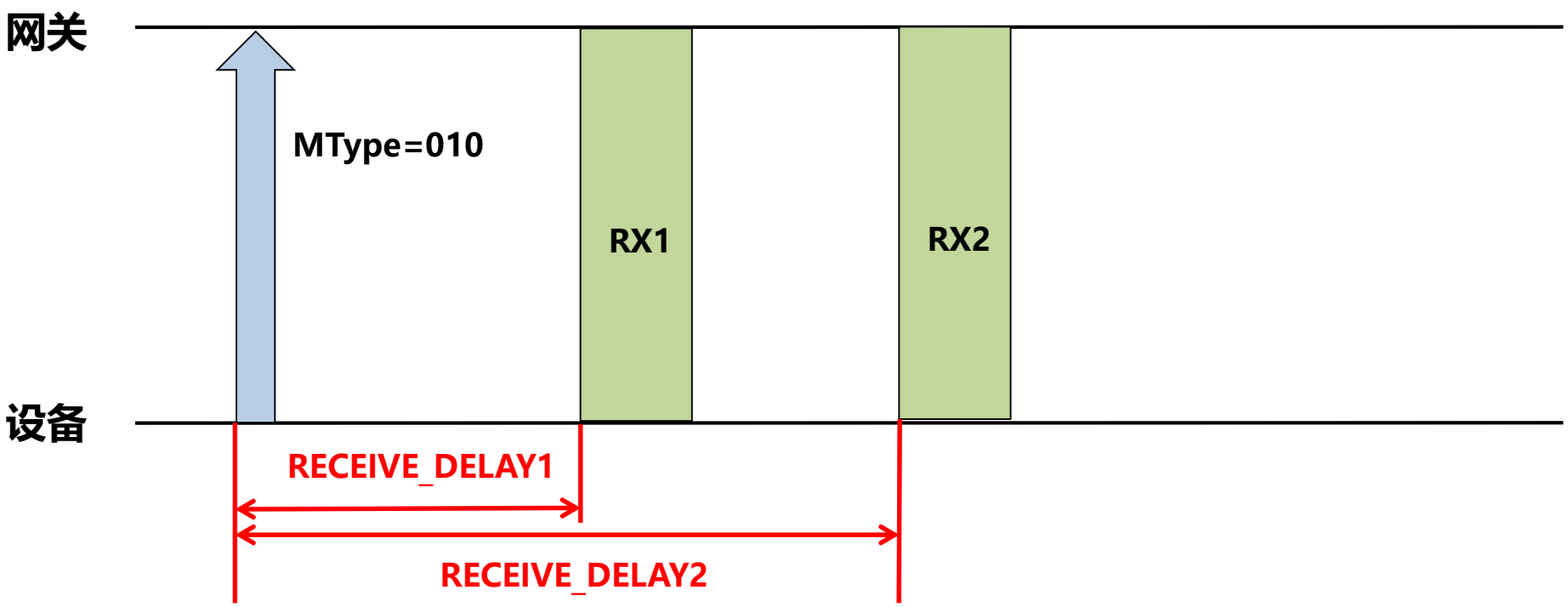
Class A

- 当网关需要发送下行数据时，只能等到设备发来上行数据之后，在随后的RX1、RX2中发送



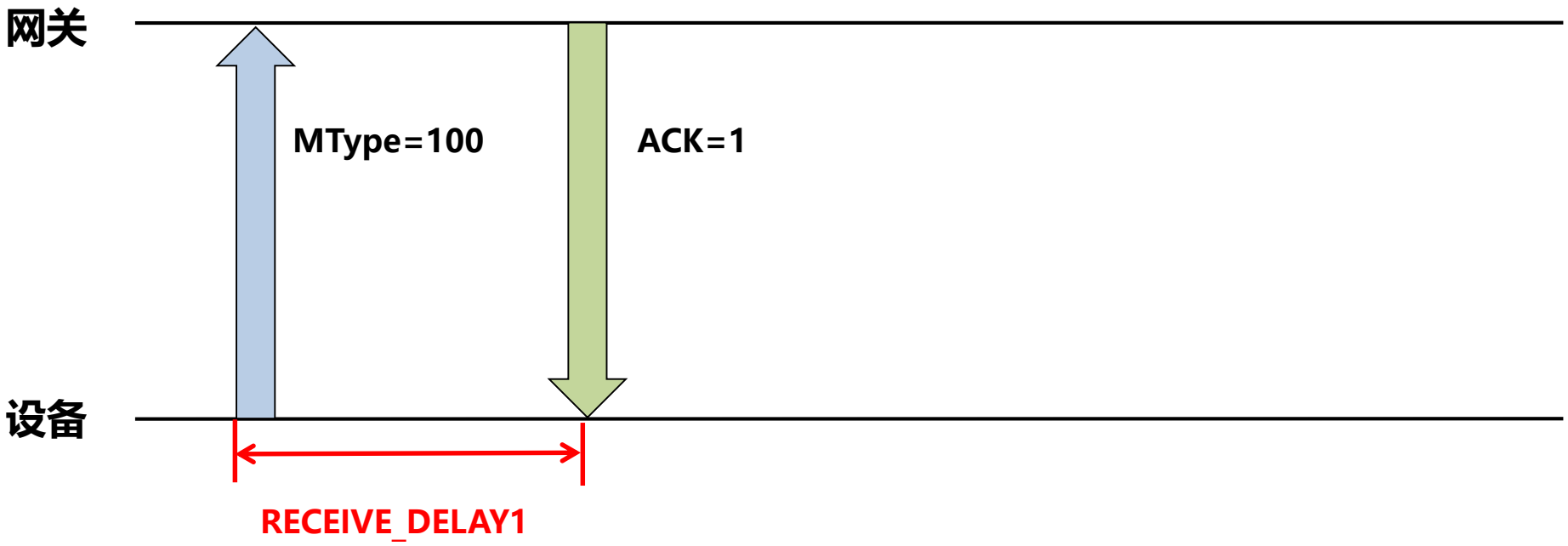
Class A数据传输

- 不需要确认的上行数据



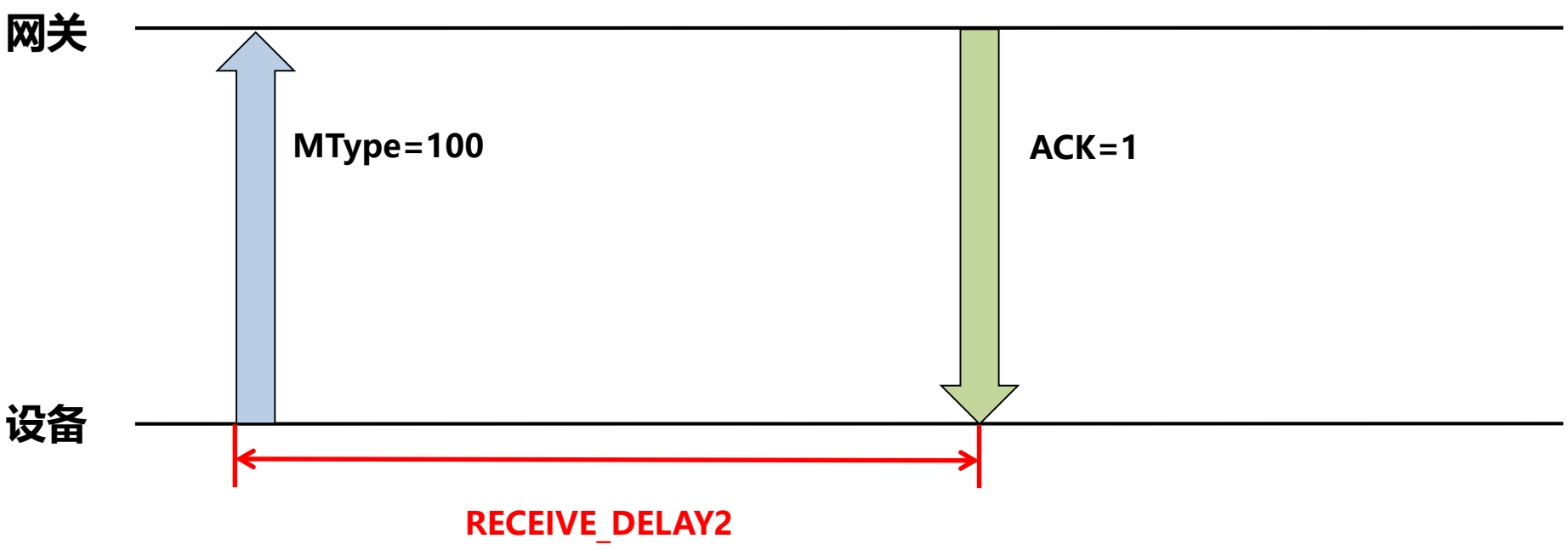
Class A数据传输

- 需要确认的上行数据



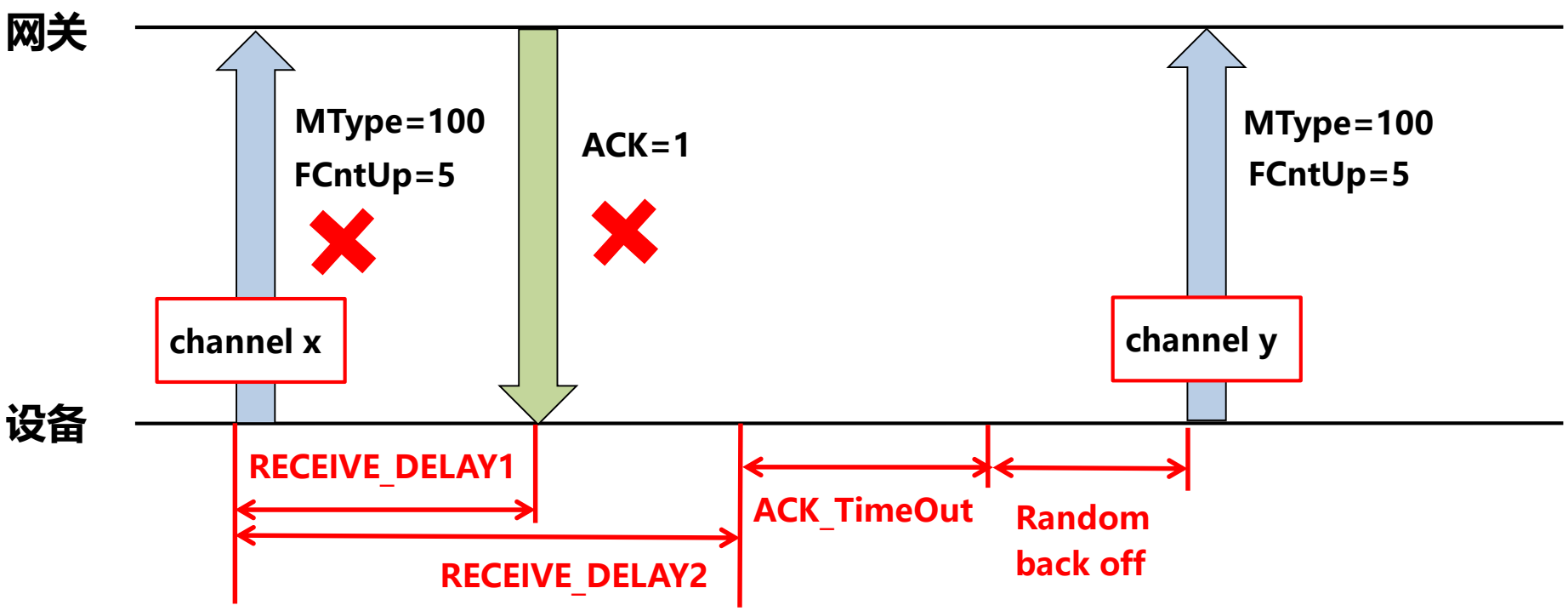
Class A数据传输

- 需要确认的上行数据



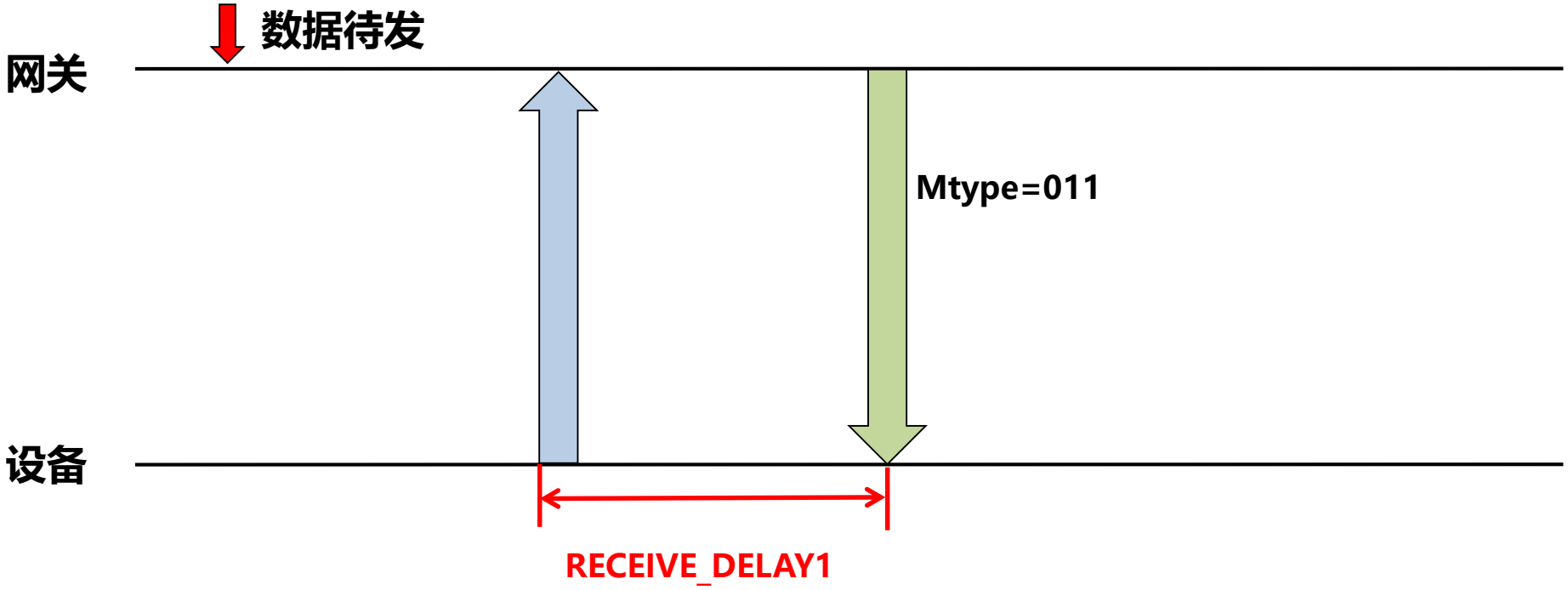
Class A数据传输

- 需要确认的上行数据
  - 未收到ACK
  - 传输次数[1,15]



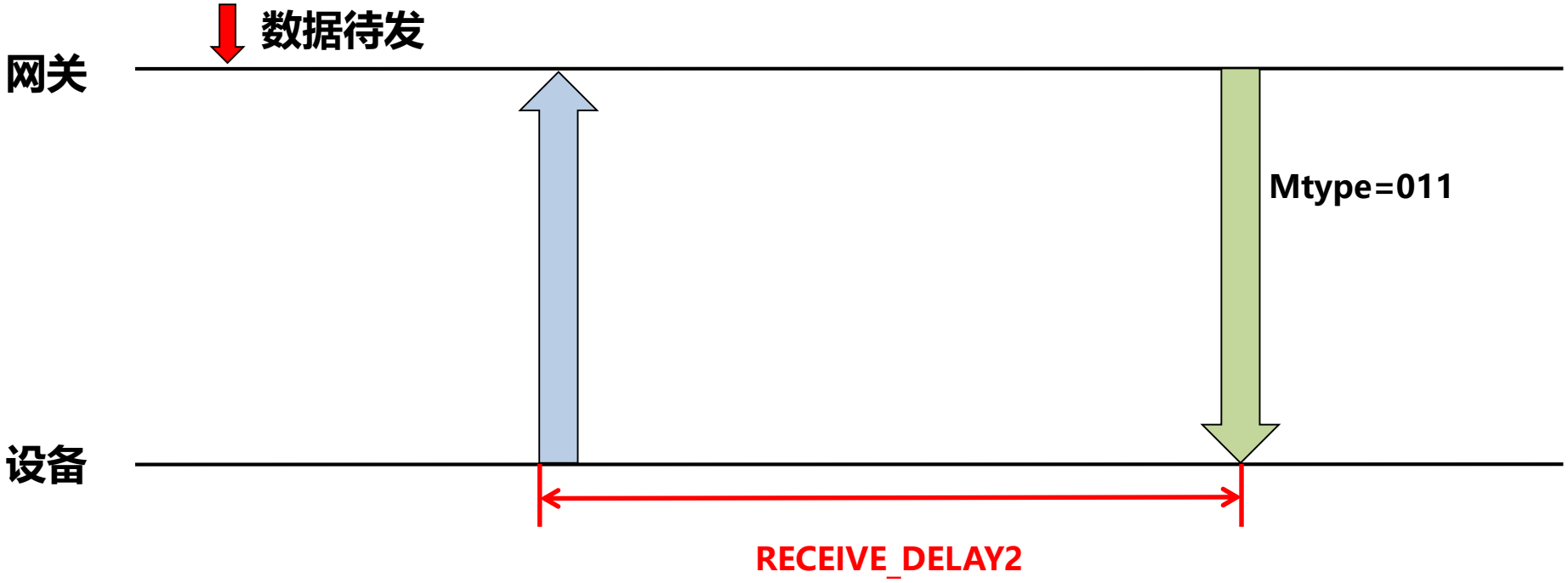
Class A数据传输

- 不需要确认的下行数据



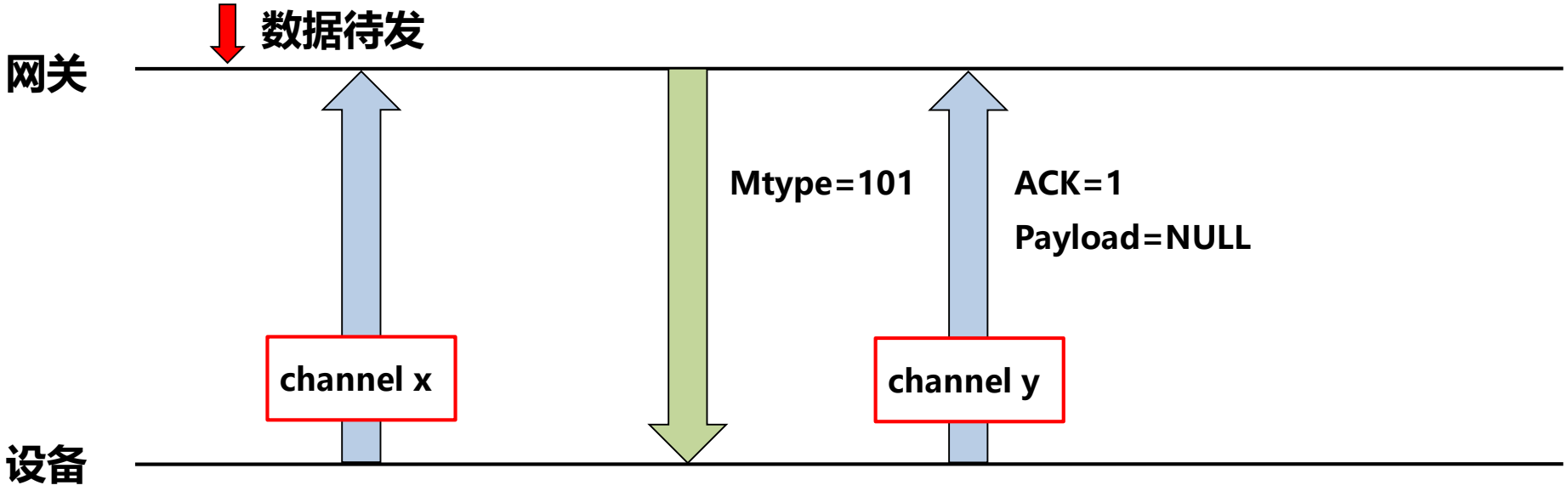
Class A数据传输

- 不需要确认的下行数据



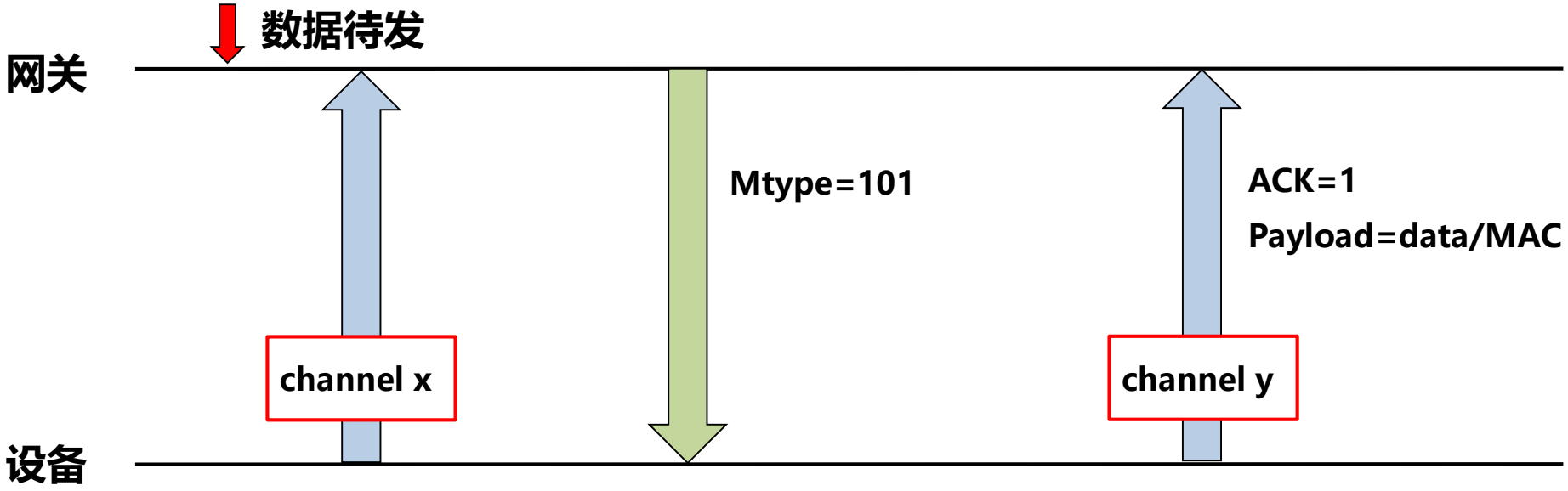
Class A数据传输

- 需要确认的下行数据
  - 设备立刻回复ACK，负载为空



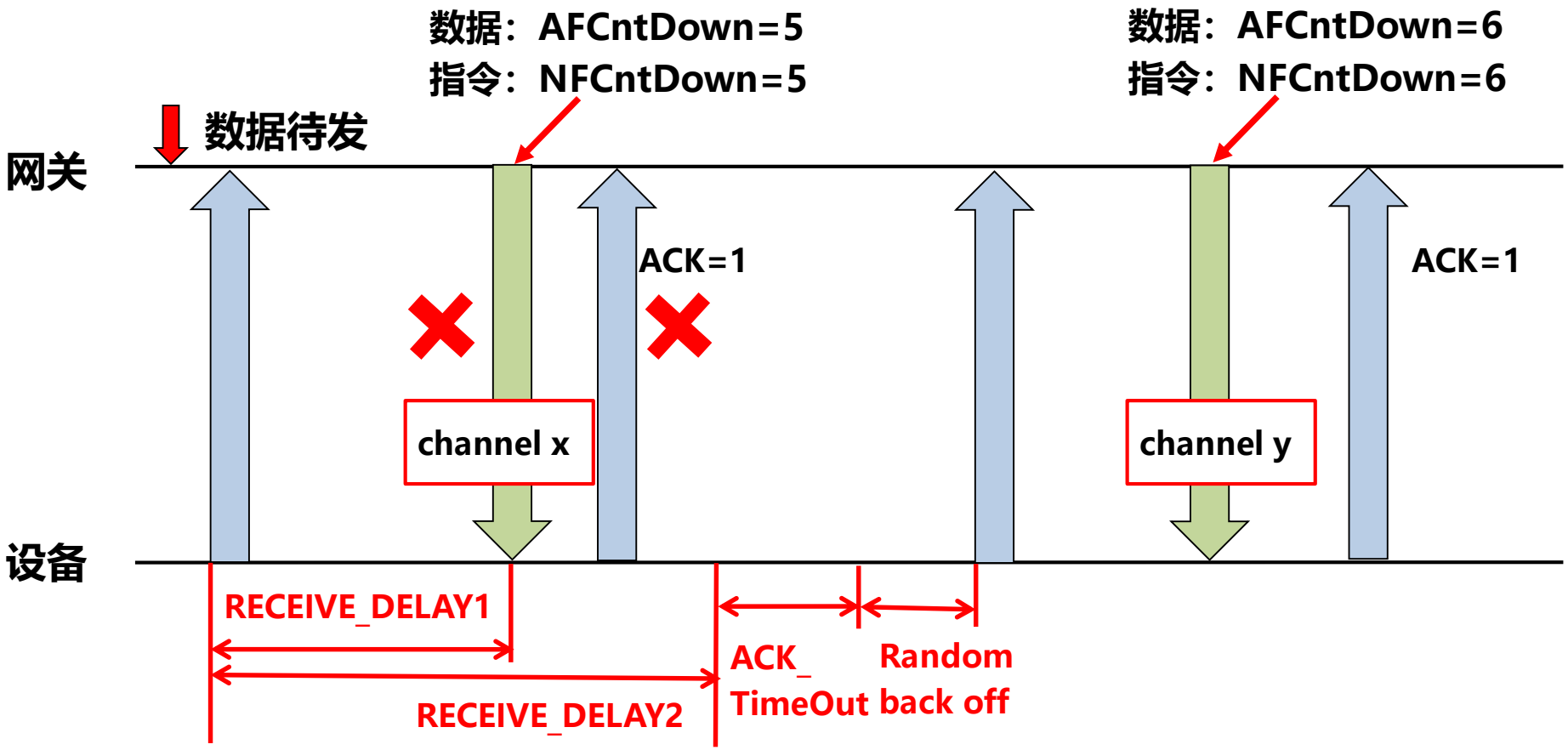
Class A数据传输

- 需要确认的下行数据
  - 设备发送上行数据时捎带ACK



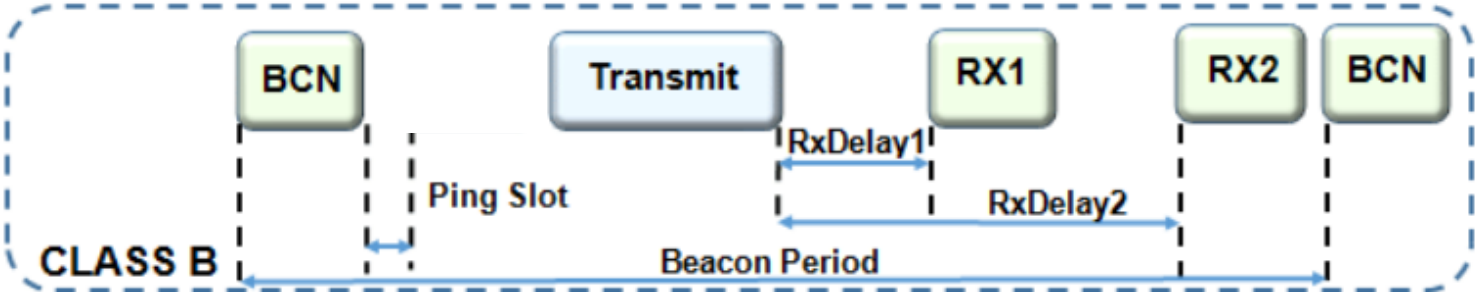
Class A数据传输

- 需要确认的下行数据
  - 未收到ACK



Class B

- 在Class A的RX1和RX2窗口的基础之上，给网关提供更多的下行窗口
- 设备通过接收网关周期性发送的beacon进行同步。在beacon周期内，设备定时开启ping slot进行数据的接收。设备以beacon中包含的时间信息，作为时间参考
- 设备首先处于class A状态，搜索到beacon之后，先通过在上行帧中将class B位置1通知网关，再切换到class B
- 设备设置其ping slot周期，网关设置设备ping slot的信道和数据率

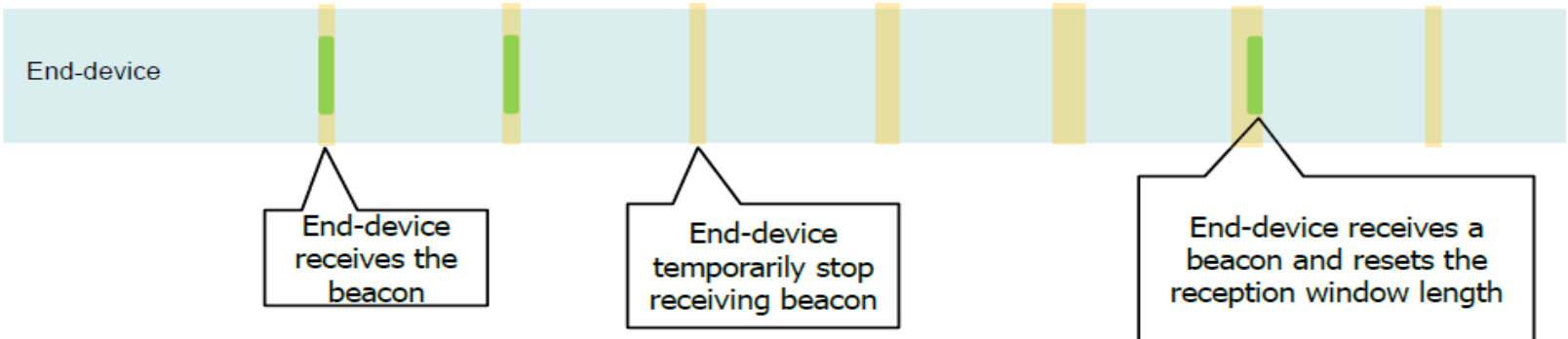


Class B

- **所有网关同时广播beacon，设备能接收最强信号的beacon。根据接收到的beacon，设备能从中获取时间参考，用于周期性开启接收窗口ping slot**
- **由于多个网关能同时收到一个设备发送的上行数据，网络服务器根据接收到的信号强度，选择发送下行数据ping帧的网关**
- **设备均以class A入网，此后按需切换成class B**
  - 设备搜寻beacon，可通过DeviceTimeReq指令加快搜寻
  - 一旦切换到class B，设备所有上行数据中class B位置1
  - 网关通过PingSlotChannelReq指令修改ping slot的频率和数据率
  - 设备先修改自己的ping slot周期。设备先切到class A，通过PingSlotInfoReq通知网关后，再切换到回class B
  - 2小时内没有再收到beacon，设备切回class A

Class B

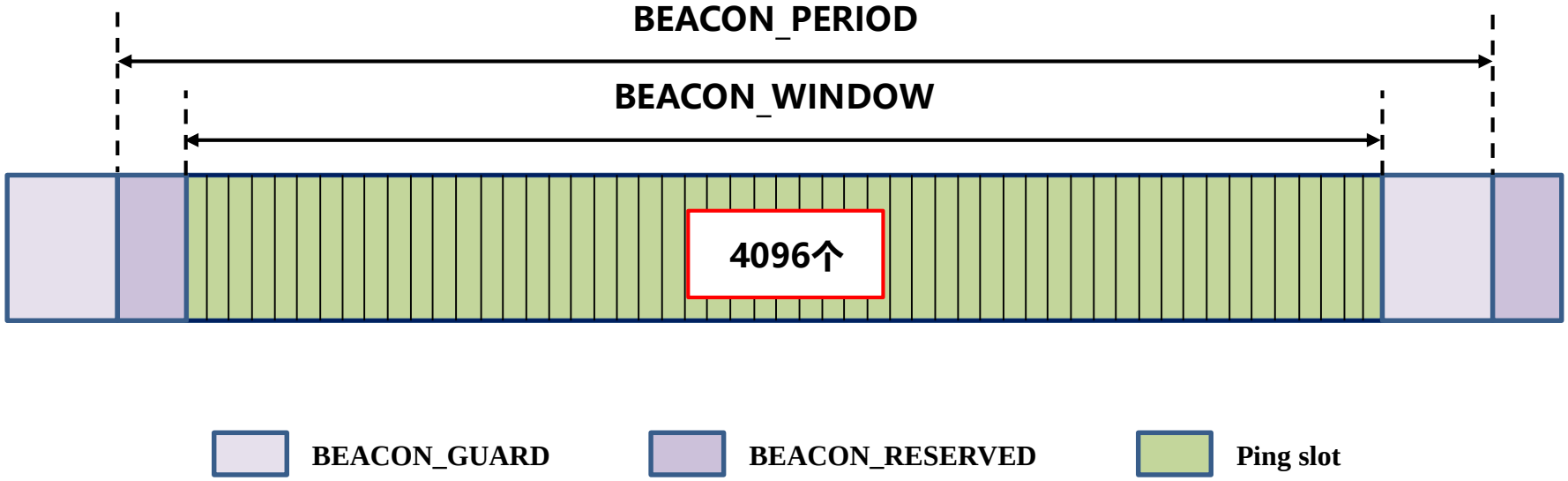
- **支持单播、多播。多播组的设备共享多播地址和会话密钥**
  - 单播和多播ping帧和class A下行帧格式大致一样，除了
    - *多播帧不携带MAC指令*
    - *不支持下行ACK*
  - 多播建立和密钥分发应用层确定
- **周期性搜索beacon，校准网络时间**
  - 暂时搜不到beacon时，设备逐渐增大beacon和ping接收窗口，以应对可能的时钟偏差
  - 最大能允许丢失beacon 2小时，在此期间，设备处于beacon-less状态，按本地时间开启窗口。



窗口时隙

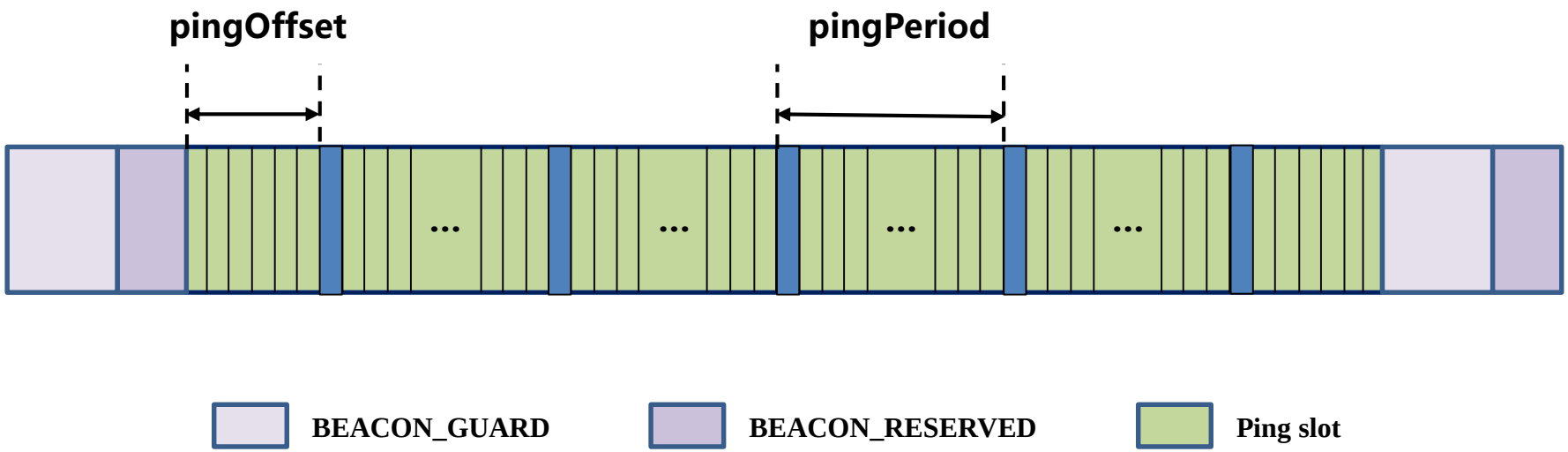
- beacon在BEACON_RESERVED起始时发送
- ping slots在BEACON_WINDOW里打开

BEACON_PERIOD	128s
BEACON_RESERVED	2.12s
BEACON_GUARD	3s
BEACON_WINDOW	122.88s
Slot length	30ms



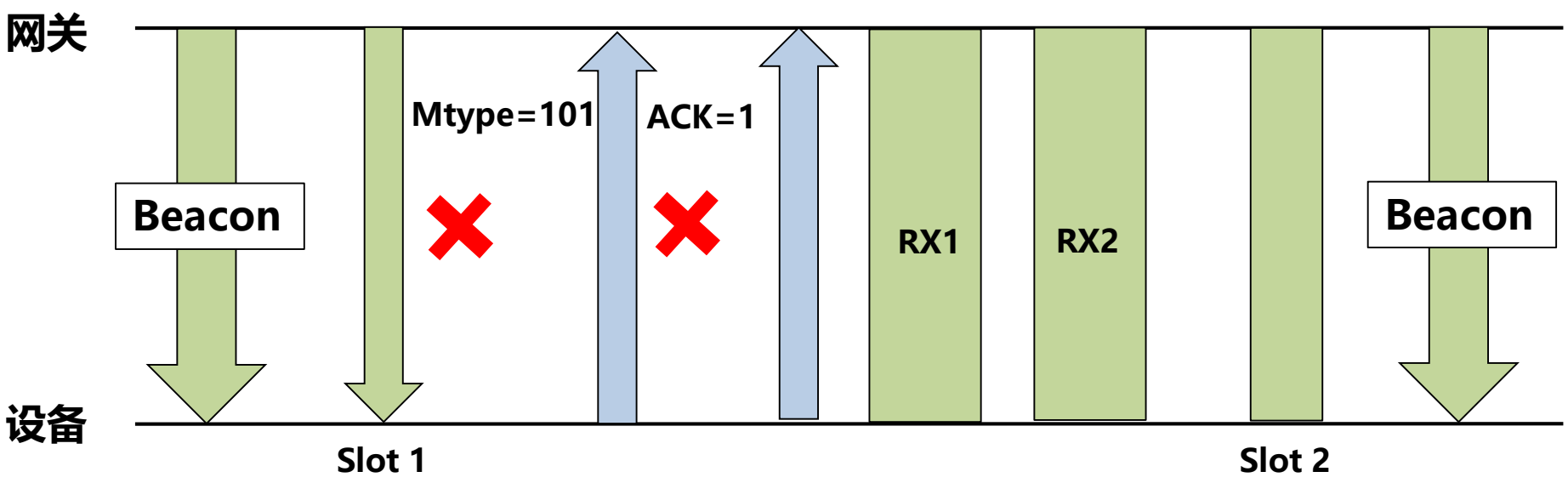
窗口时隙

pingNb	实际可用ping slot数量，为2^{k}，$0 \leq k \leq 7$
pingPeriod	可用ping slot间间隔，为$2^{12}/pingNb$
pingOffset	第一个可用ping slot的偏移，为$[0, pingPeriod - 1]$



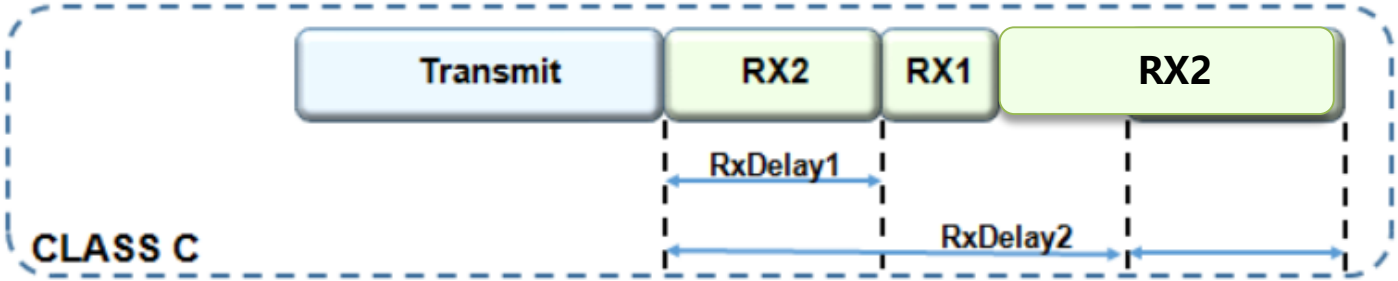
Class B数据传输

- Confirmed Downlink
  - 未收到ACK
    - *网关可利用RX1、RX2和slot重传*



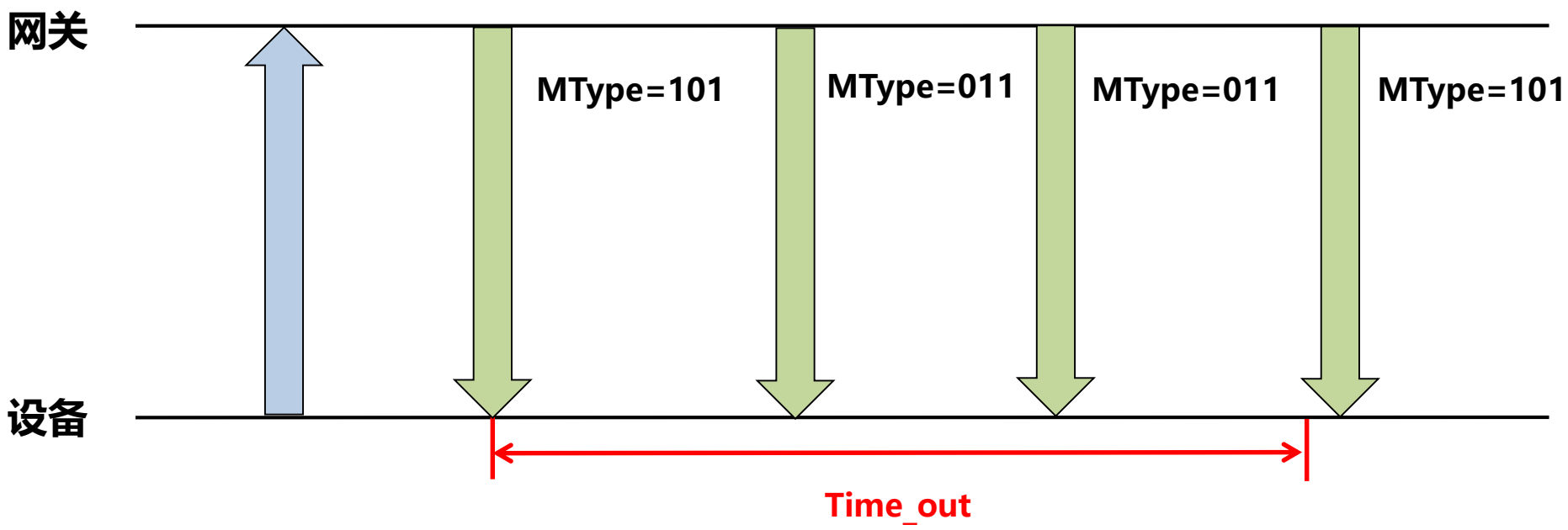
Class C

- 当设备处于供电状态，不必考虑能耗问题时，设备始终开启接收窗口
- 在Class A的RX1和RX2窗口的基础之上，当设备不在发送数据或者不在RX1状态时，均通过配置RX2参数，开启设备进行数据的接收
- RX1开启时，若网关仍在发送数据，设备停止当前的数据接收



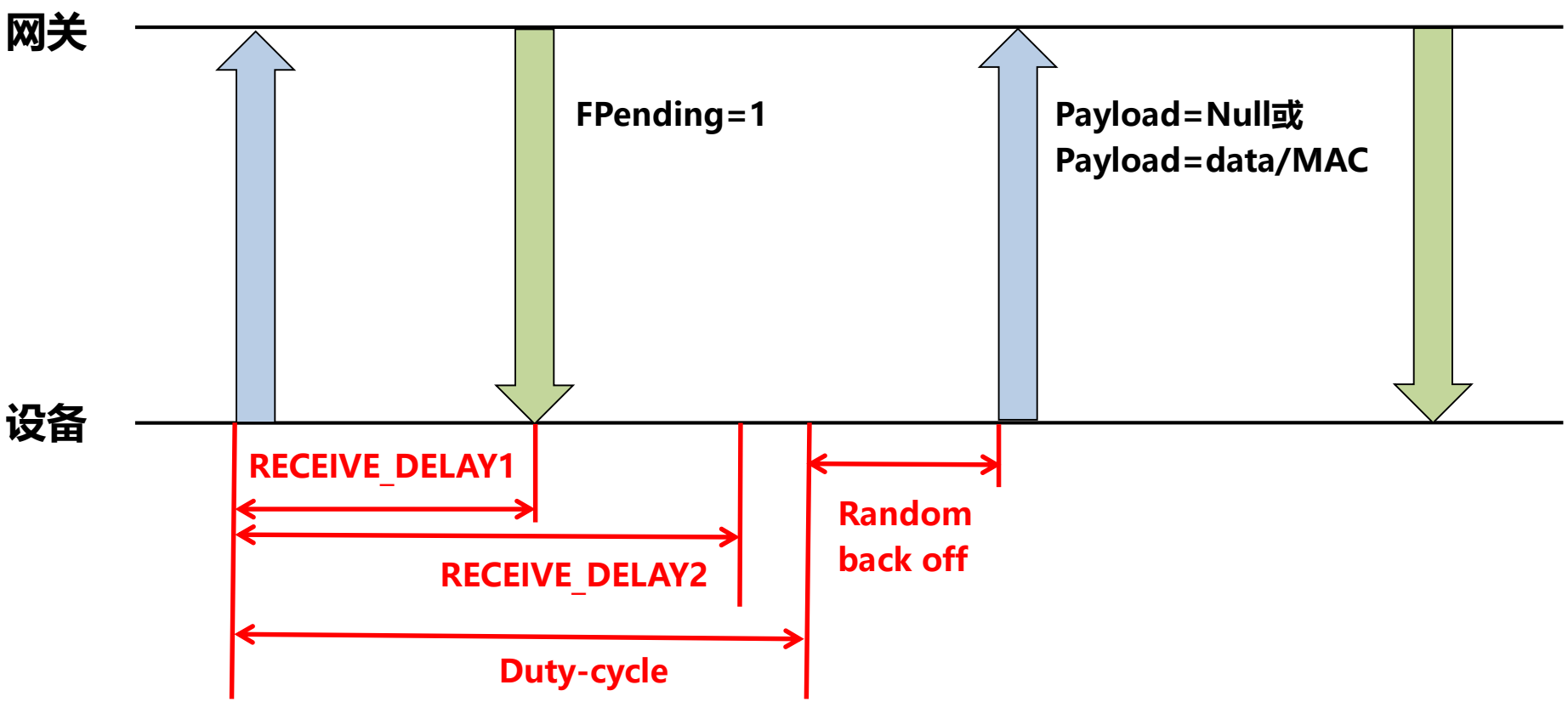
Class C数据传输

- 设备上行窗口有限，因此网关不能频繁发送请求确认的下行数据
  - Time_out之后或者收到上行确认数据，才能再次发送请求确认的下行数据。期间可以发送任意不需要请求确认的下行数据



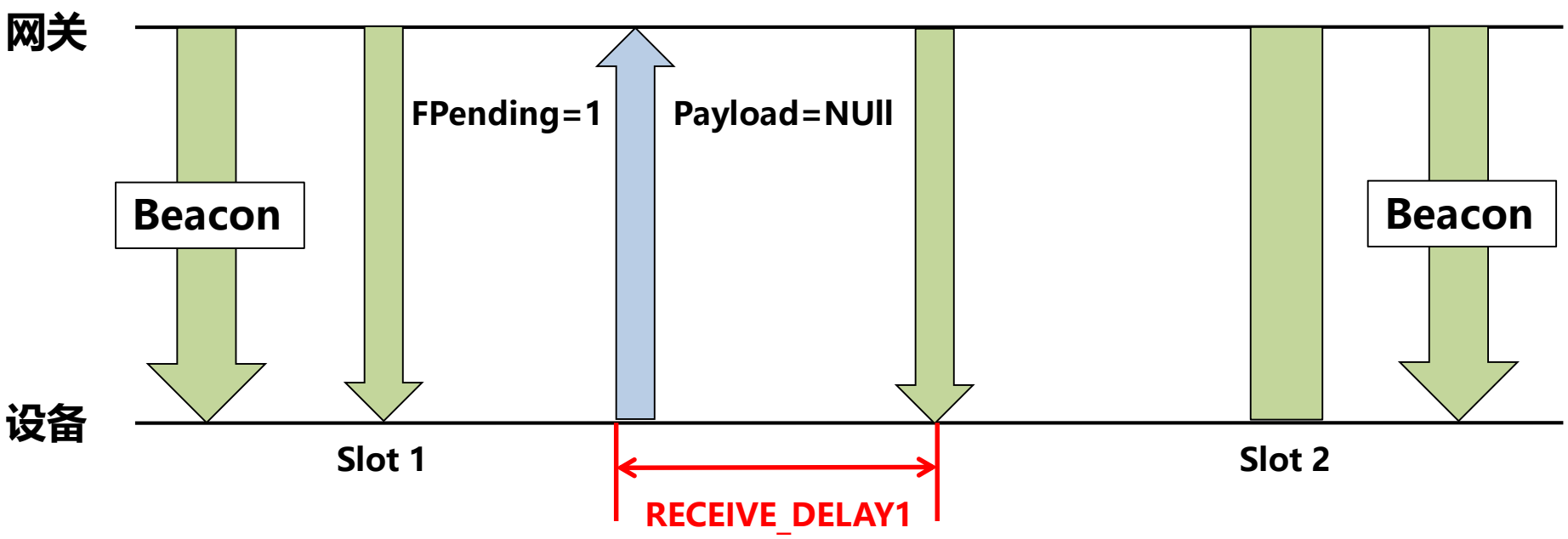
FPending

- FPending=1，表明网关有更多数据要传
  - Class A



FPending

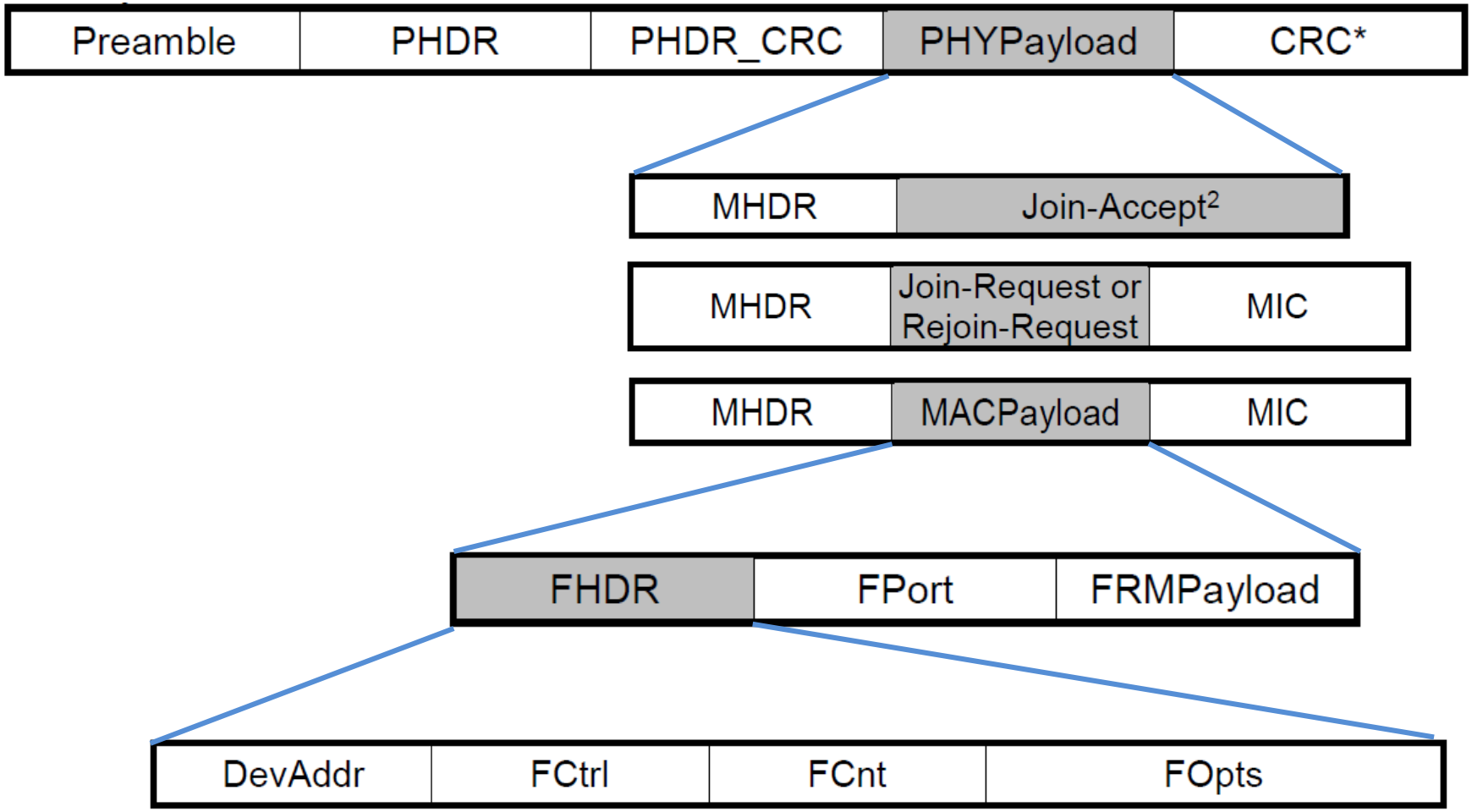
- FPending=1，表明网关有更多数据要传
  - Class B
    - *设备立即发送空白帧，以开启RX1、RX2*



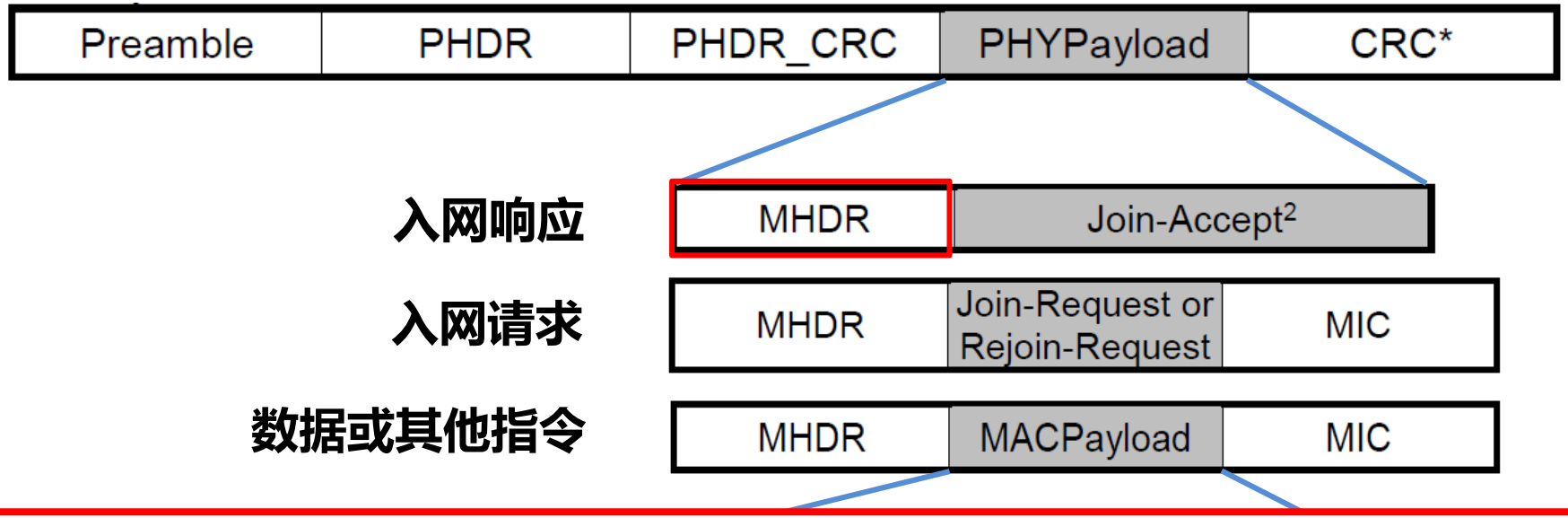
FPending

- **FPending=1，表明网关有更多数据要传**
  - **Class C**
    - *设备始终打开，除了设备发送数据时，网关想发即发，因此FPending=1不触发设备任何行为*

帧结构


Preamble
PHDR
PHDR_CRC
PHYPayload
CRC*
MHDR
Join-Accept2
MHDR
Join-Request or Rejoin-Request
MIC
MHDR
MACPayload
MIC
FHDR
FPort
FRMPayload
DevAddr
FCtrl
FCnt
FOpts

帧结构

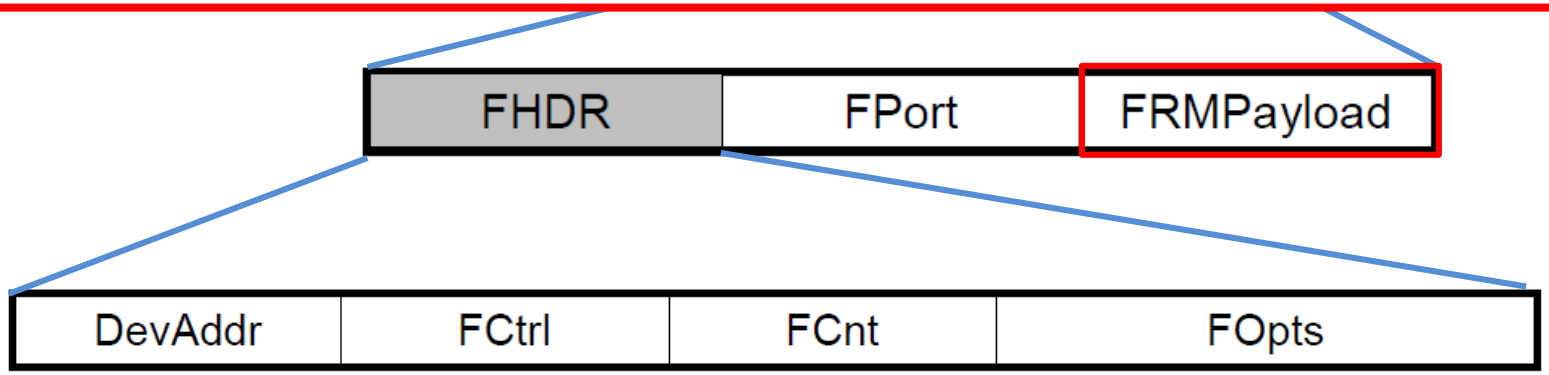


MHDR中Mtype字段指明消息类型

MType	Description
000	Join-request
001	Join-accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	Rejoin-request
111	Proprietary

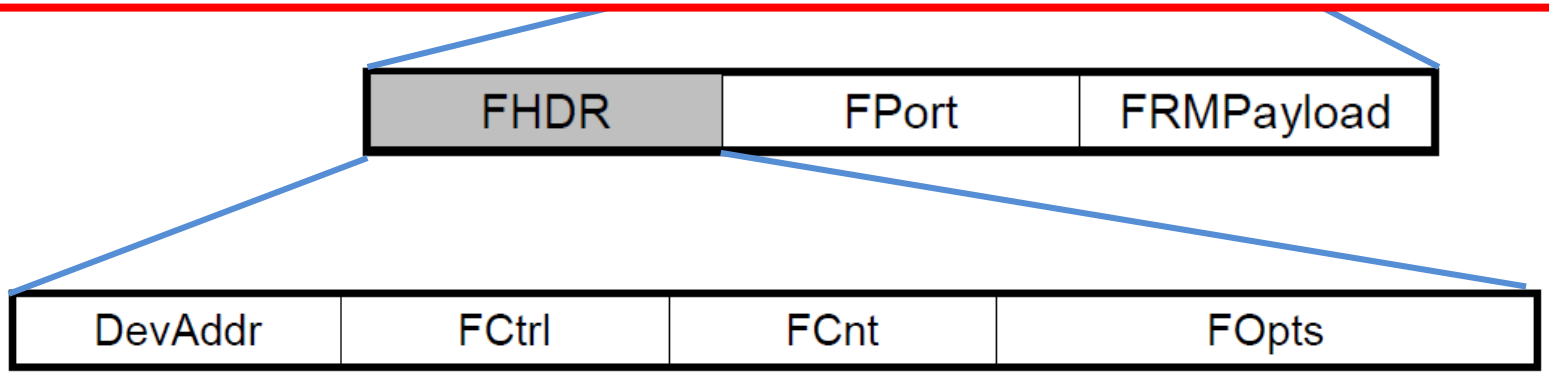
帧结构

FPort=0，FRMPayload存放MAC指令（如LinkADRReq）
FPort≠0， FRMPayload存放应用层数据
Fport空， FRMPayload为空



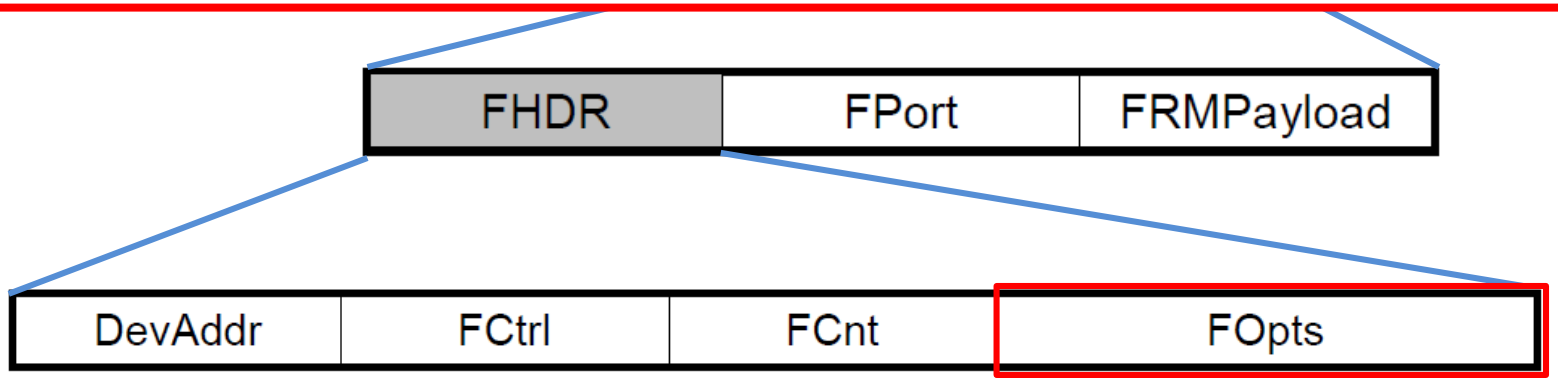
帧结构

一个帧中可发送多条指令，接收方在一个帧中按序组织回复
若回复超过FOpts最大长度，则在FRMPayload中回复
同时有数据和MAC指令回复时，优先MAC指令回复
发送MAC指令序列时，保证接收能在一个帧中答复完成



帧结构

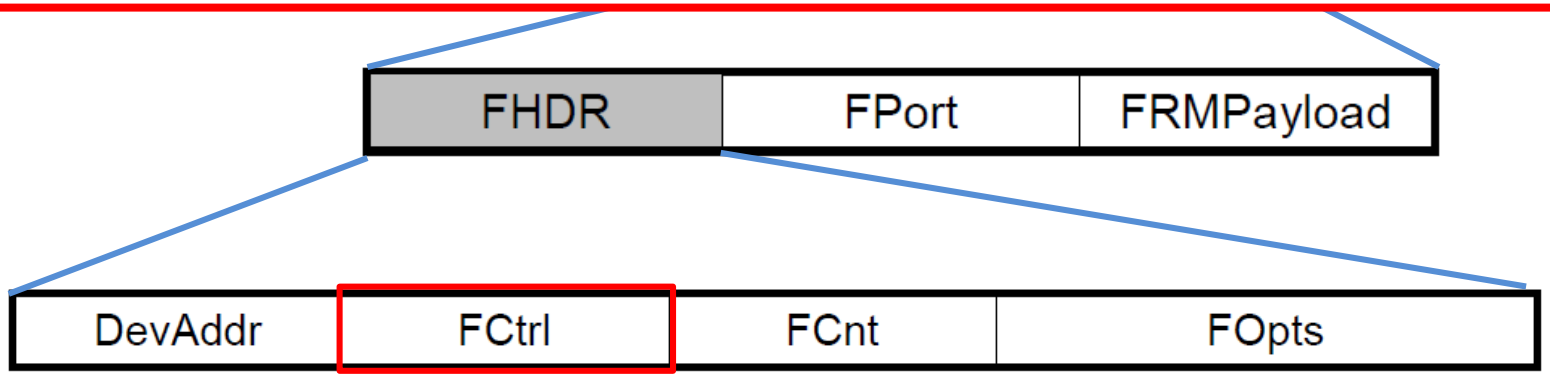
FOpts不为空时，存放MAC指令，此时FPort≠0，即FRMPayload和FOpts不能同时存放MAC指令



帧结构

下行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	RFU	ACK	FPending	FOptsLen

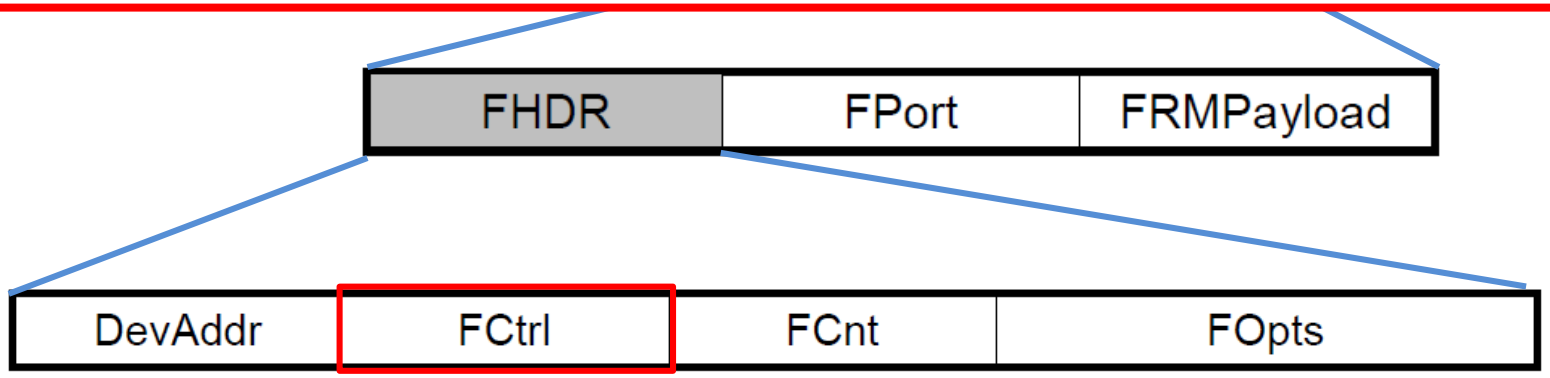


帧结构

下行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	RFU	ACK	FPending	FOptsLen

网关通知设备，使用更高的数据率和更小的发送功率

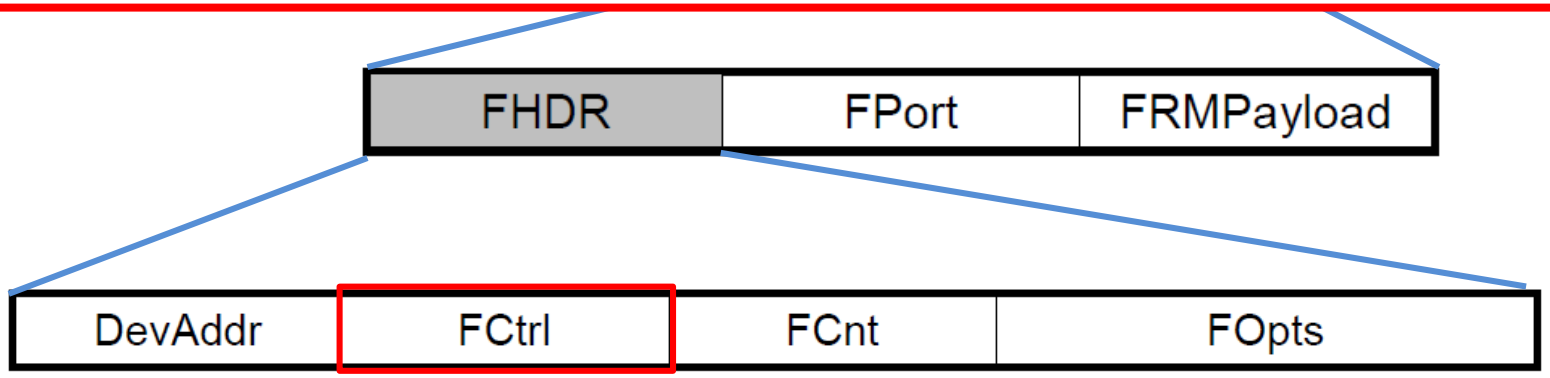


帧结构

下行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	RFU	ACK	FPending	FOptsLen

ACK=1，回复上行请求确认的数据

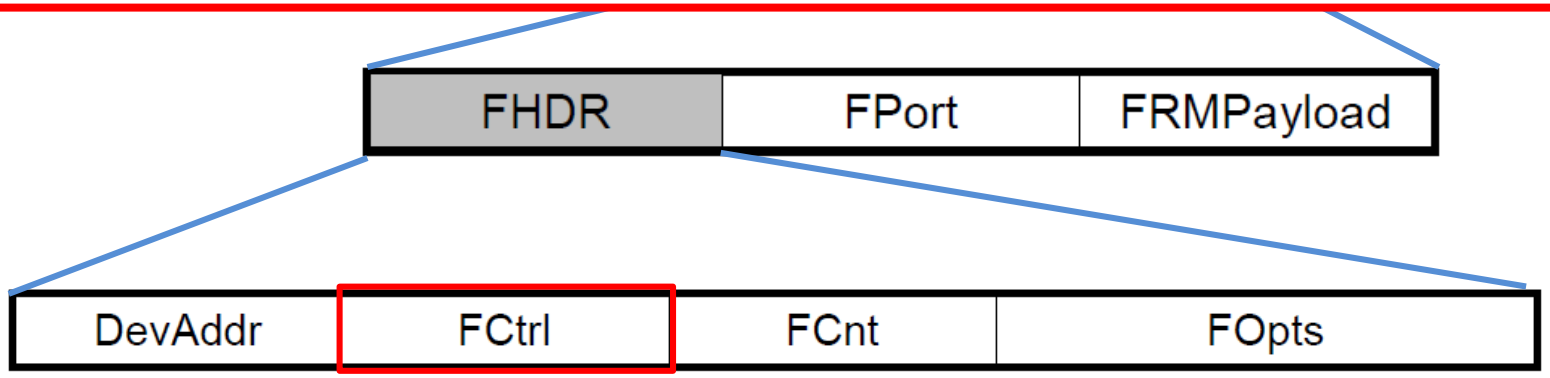


帧结构

下行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	RFU	ACK	FPending	FOptsLen

FPending=1，网关通知设备其还有下行数据要发送，请求为其安排接收窗口

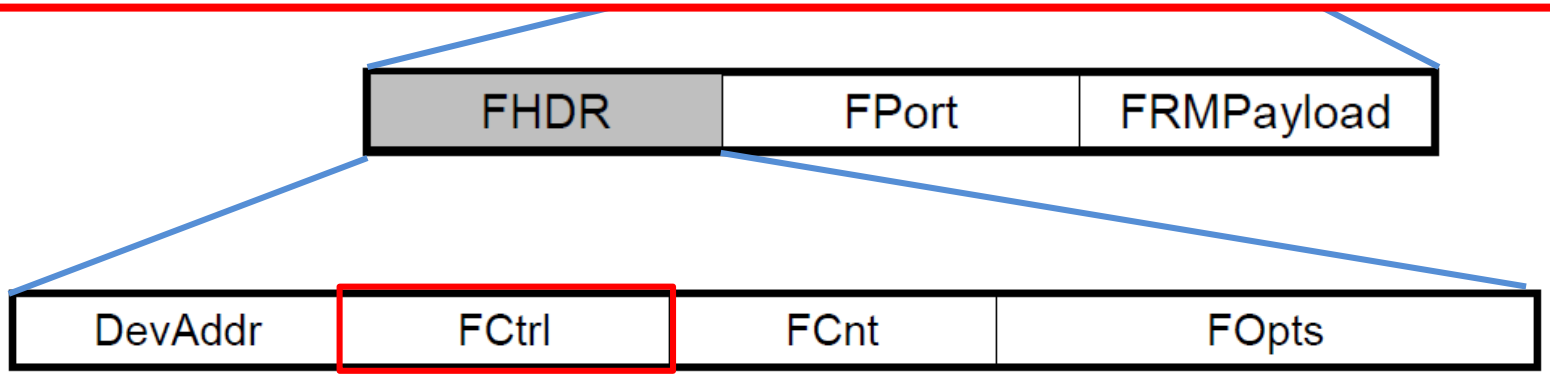


帧结构

下行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	RFU	ACK	FPending	FOptsLen

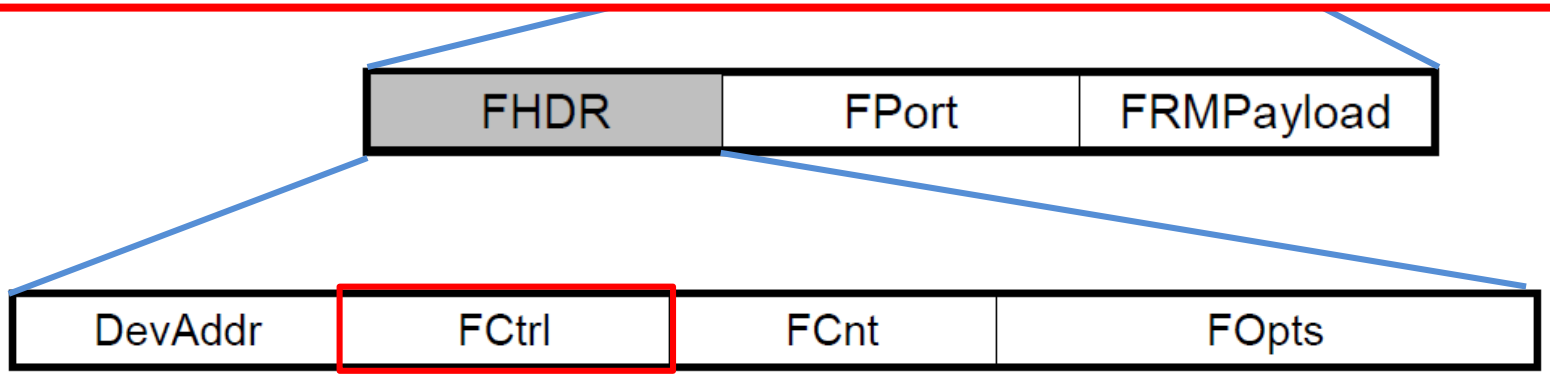
指明FOpts的长度。若FOptsLen=0，则FOpts不存在



帧结构

上行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	ADRACKReq	ACK	ClassB	FOptsLen

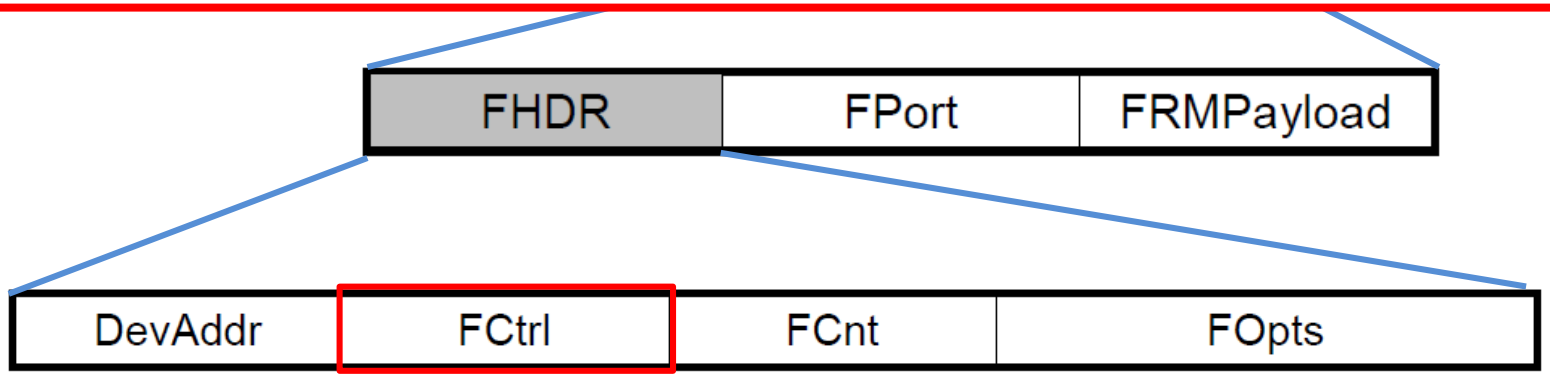


帧结构

上行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	ADRACKReq	ACK	ClassB	FOptsLen

设备通知网关，为其设置更高的数据率和更小的发送功率

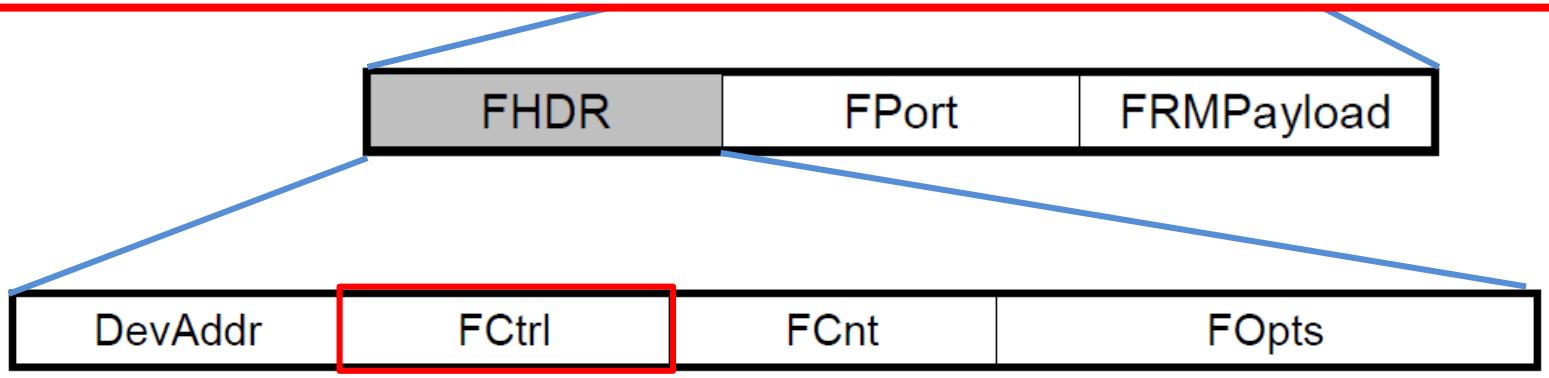


帧结构

上行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	ADRACKReq	ACK	ClassB	FOptsLen

设备向网关确认，通过ADR机制增大发送速率或减小发送功率后，网关仍然能收到设备的上行数据

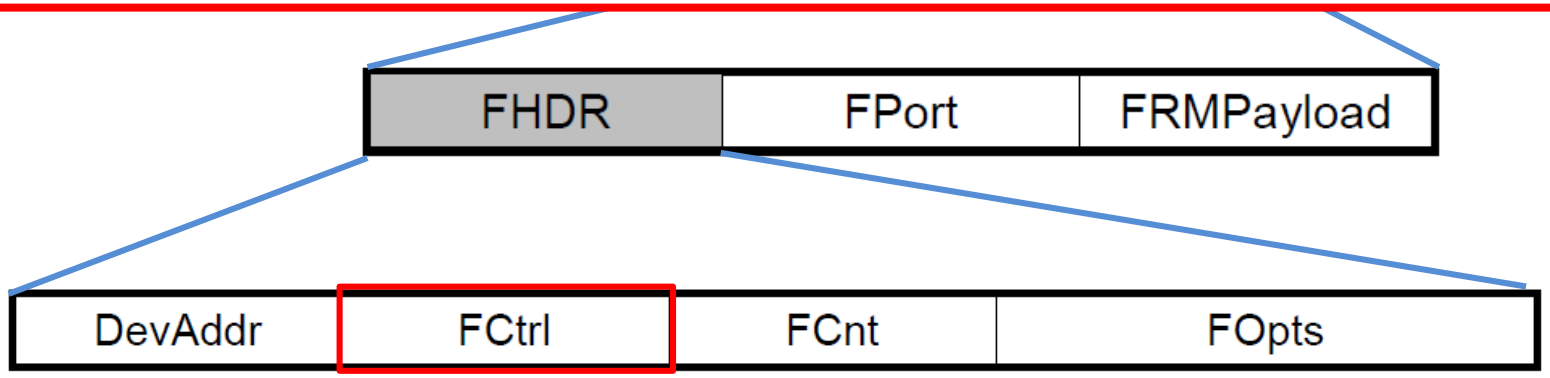


帧结构

上行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	ADRACKReq	ACK	ClassB	FOptsLen

ACK=1，回复下行请求确认的数据

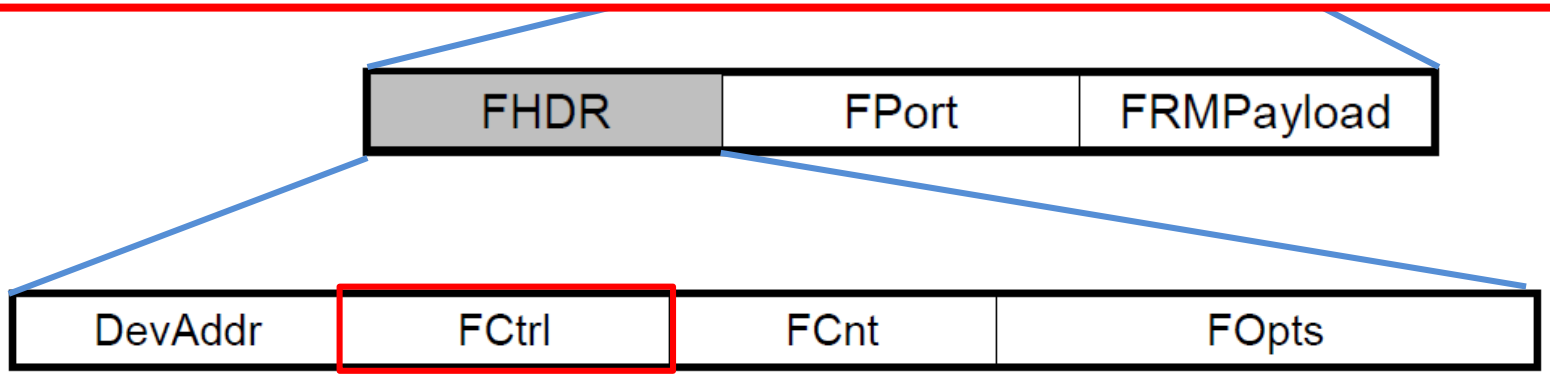


帧结构

上行

Bit#	7	6	5	4	[3..0]
FCtrl bits	ADR	ADRACKReq	ACK	ClassB	FOptsLen

ClassB=1，告诉网关设备处于classB状态



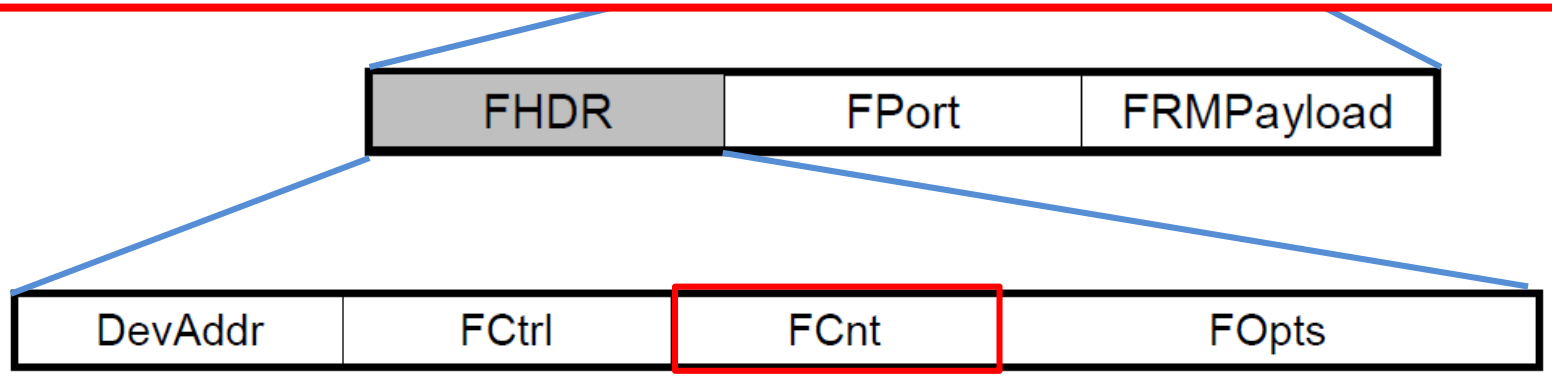
帧结构

上行：FCntUp

下行：NFCntDown统计收到的MAC指令帧，由网络服务器管理

AFCntDown统计收到的应用数据帧，由应用服务器管理

OTAA设备入网清0，ABP不重置



常见MAC指令

- LinkADRReq
  - 网关要求设备改变数据率、发射功率、发送信道
- RXParamSetupReq
  - 网关配置设备RX1数据率、RX2的数据率和信道
- DLChannelReq
  - 网关配置设备RX1信道
- RXTimingSetupReq
  - 网关配置设备RECEIVE_DELAY1
- RekeyInd
  - OTAA设备用于通知网关更新密钥
- RekeyConf
  - 网关通知OTAA设备确认密钥更新

常见MAC指令

- ADRParamSetupReq
  - 网关用于设置ADR_ACK_LIMIT
- PingSlotInfoReq
  - Class B设备告知网关其ping slot周期
- PingSlotChannelReq
  - 网关设置class B设备的ping slot信道
- BeaconFreReq
  - 网关修改beacon的发送信道
- DeviceModeInd
  - 设备告知网关其当前是class A还是class C

安全性

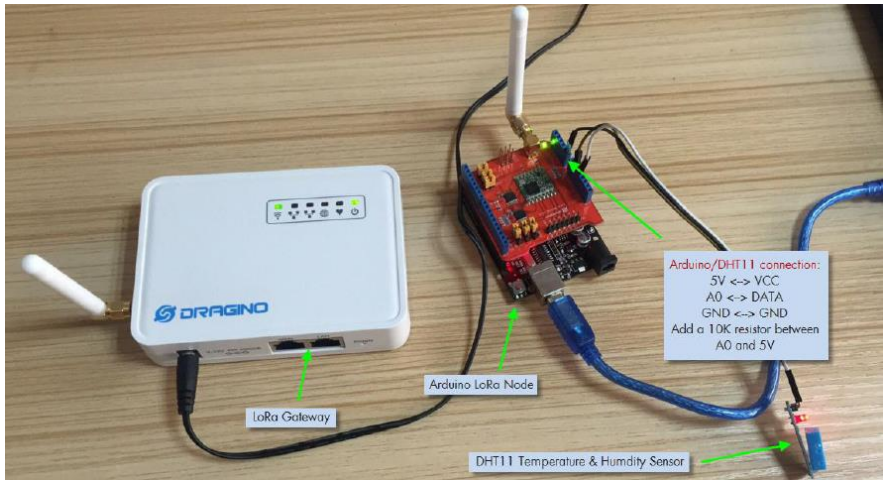
- 终端到网关通信采用128AES加密
- 网关到后端服务器再进行BASE64加密
- NwkSKey用于进行MAC命令的加密，AppSKey用于进行数据的加密
- 全球唯一标识的128位AES Key（AppKey）和唯一标识的64位DevEUI同时进行设备的识别
- LoRaWAN网络也使用LoRa联盟分配的24位唯一标识符标识

LoRaWAN

- **使用场景**
- **物理层**
  - 频段划分
  - 信道分配
  - 数据调制
- **MAC层**
- **案例**

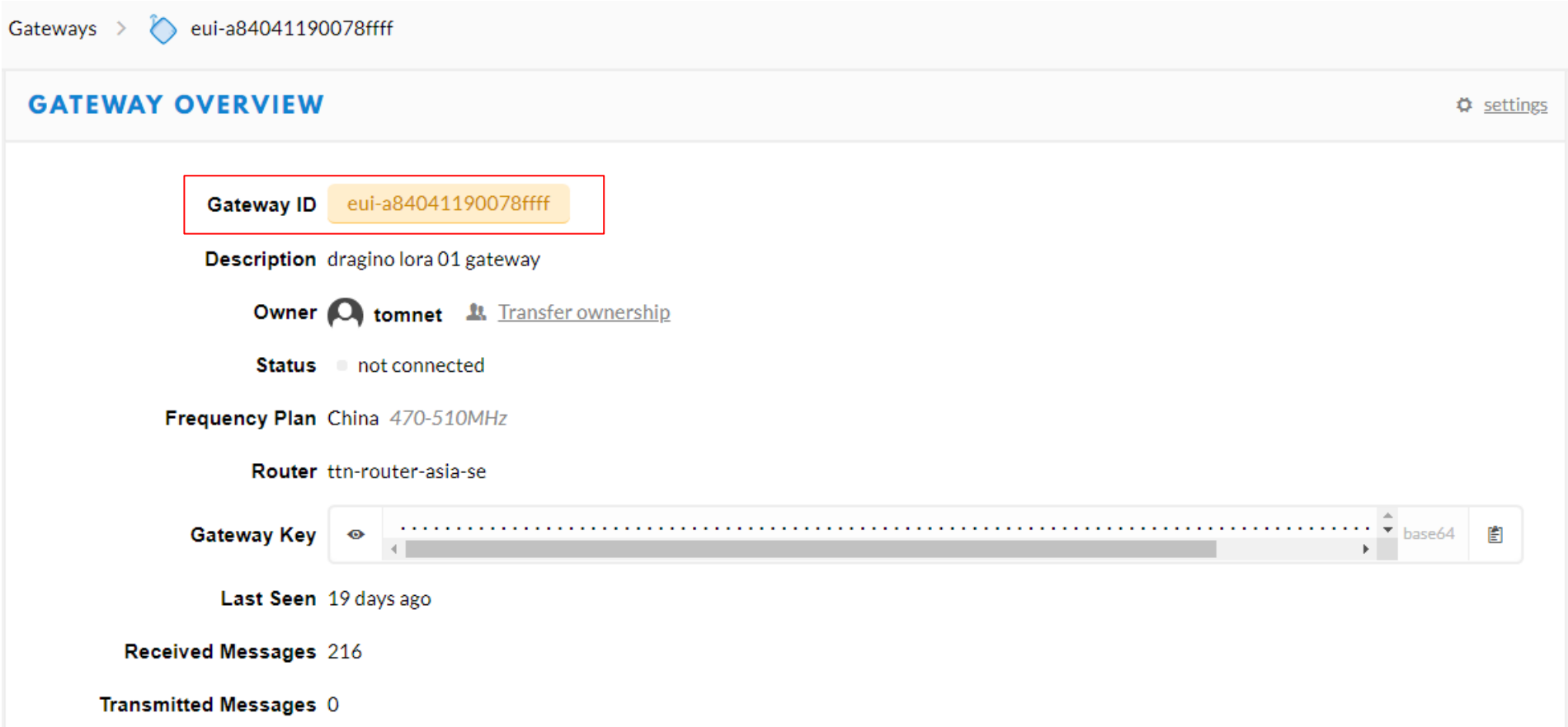
实际传输案例-基于LoRa的数据采集与云端上传

- **实验设置：LoRa节点将从传感器中获取监测数据，并将这些数据发送到LoRa网关。当网关收到数据后，它将把数据传递给物联网服务器**
- **实验搭建**
  - LoRa 终端节点：LoRa Shield + Arduino UNO + DHT11温湿度传感器
  - LoRa 网关：LG01
  - IoT server：TTN LoRaWAN 服务器



实际传输案例-基于LoRa的数据采集与云端上传

- 步骤一：创建TTN物联网服务器账户(网址：https://www.thethingsnetwork.org/)
- 步骤二：注册登录后，点击进入“Console”页面，在“Console”页面中创建一个gateway目录



实际传输案例-基于LoRa的数据采集与云端上传

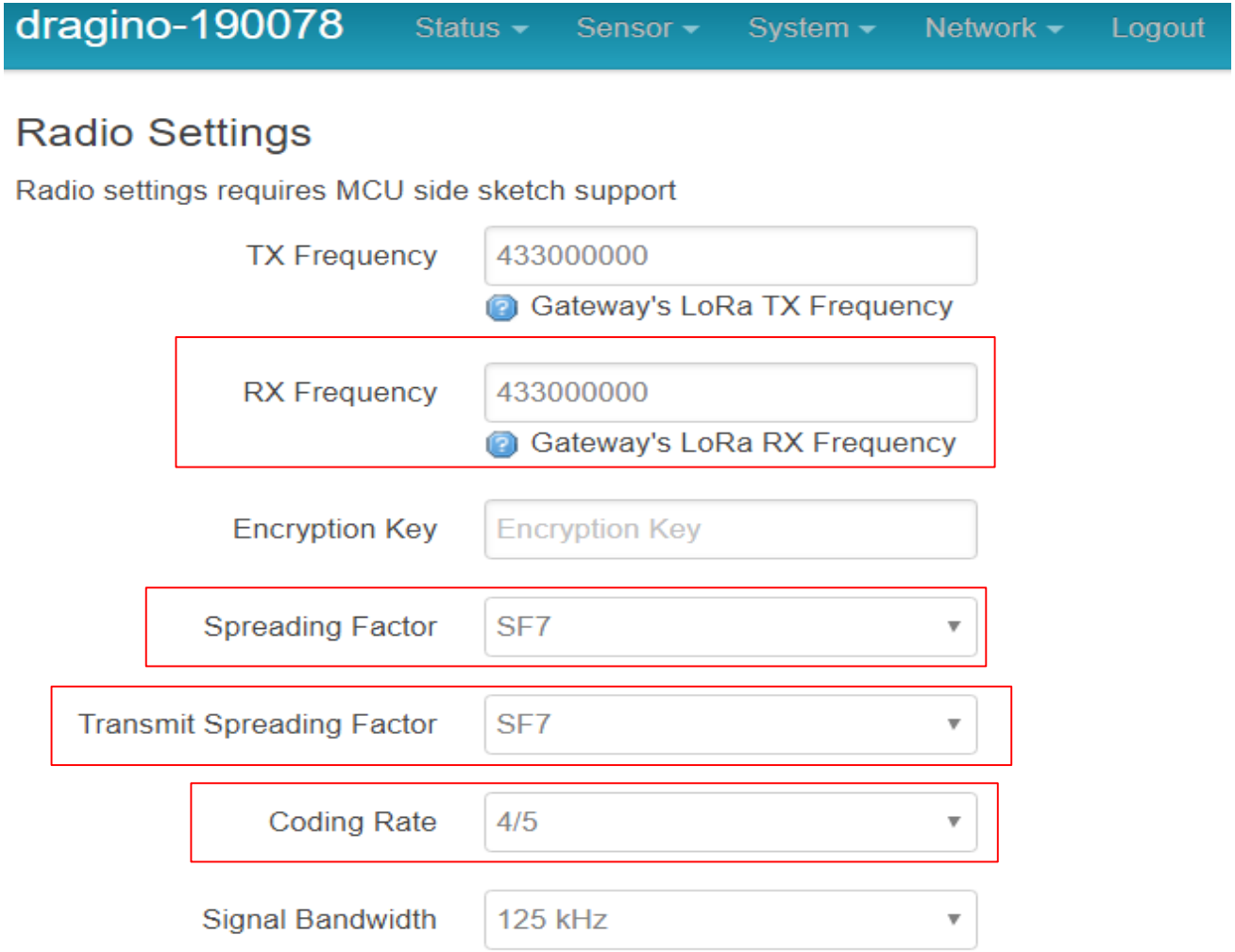
- 步骤三：在TTN服务器上创建一个终端节点目录。具体地，在“Console”页面上，添加一个application；接着在“Application”页面上，点击“register device”。同时将节点激活方式设置为ABP

Applications > dragino_test_aaplication1 > Devices > lorashield

Application ID	dragino_test_aaplication1
Device ID	lorashield
Activation Method	ABP
Device EUI	00 28 24 A3 EB ED 92 5B
Application EUI	70 B3 D5 7E D0 01 50 77
Device Address	26 01 14 67
Network Session Key	70 11 23 59 F7 E2 91 47 C9 07 F2 95 3B 8F 49 BB
App Session Key	17 48 32 98 F9 FC 25 3E 22 CE D0 2A 00 65 61 A1

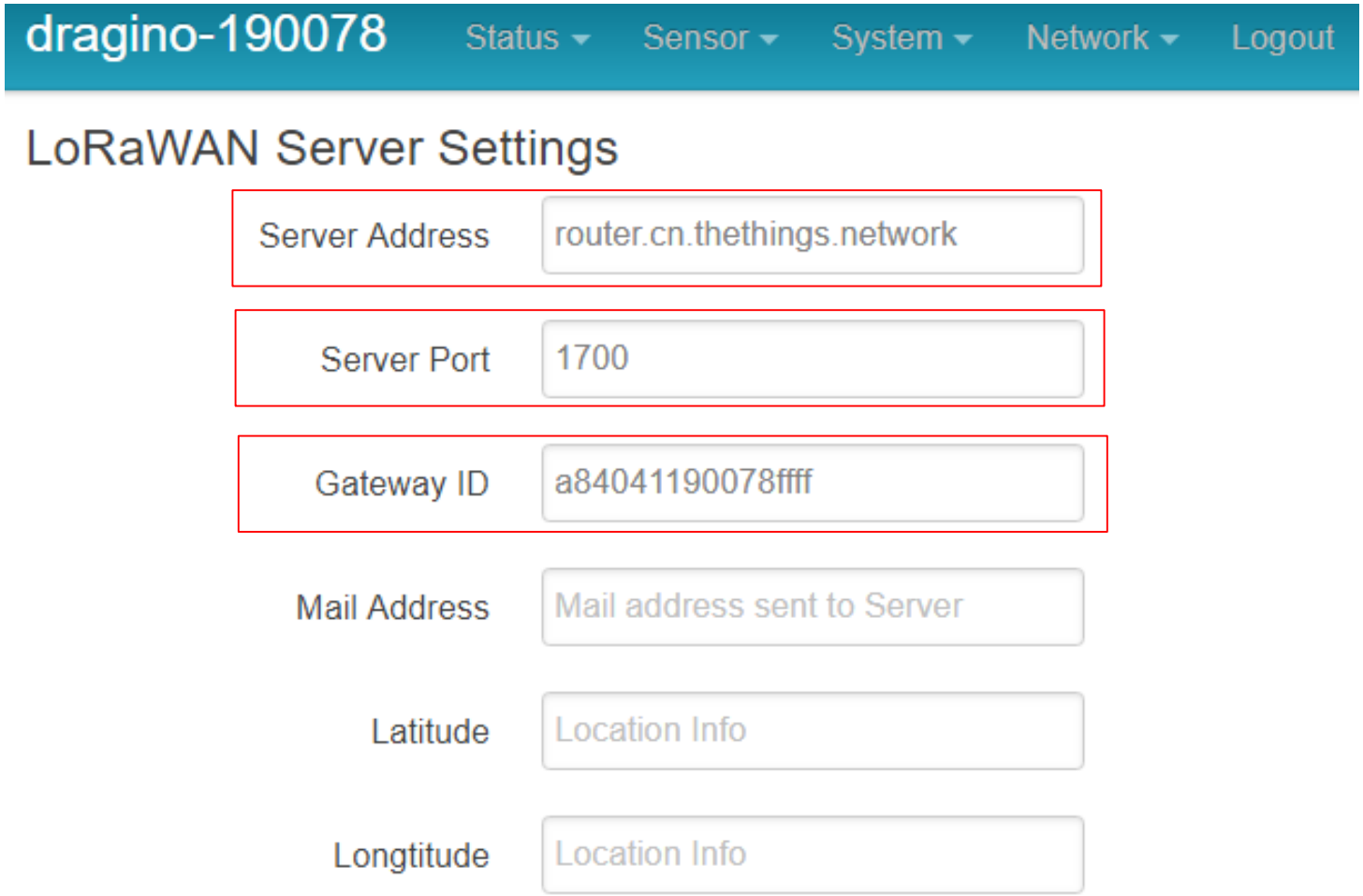
实际传输案例-基于LoRa的数据采集与云端上传

- 步骤四：将LG01网关用网线连接入网，并在LG01页面（网址：10.130.1.1）上配置LG01端的Lora radio参数



实际传输案例-基于LoRa的数据采集与云端上传

- 步骤五：在LG01页面（网址：10.130.1.1）上配置LoRaWAN服务器的地址及端口信息



实际传输案例-基于LoRa的数据采集与云端上传

- 步骤六：使用Arduino IDE烧录应用程序代码（包括数据接收和转发）到LG01网关

```
#include <Console.h>;
#include <LoRa.h>
static uint8_t packet[256];

void setup(){
        Bridge.begin(115200); // Setup Bridge
        if (!LoRa.begin(freq)){
            Console.println(F("init LoRa failed"));}
        setLoRaRadio();// Set LoRa Radio to Semtech Chip
}
void loop(){
        if (!send_mode) {
            receivepacket();  // receive LoRa packet and forward it
         } else if (send_mode == 1) {
            sendpacket(); //send packet to node
         } else {
            emitpacket();
 }
```

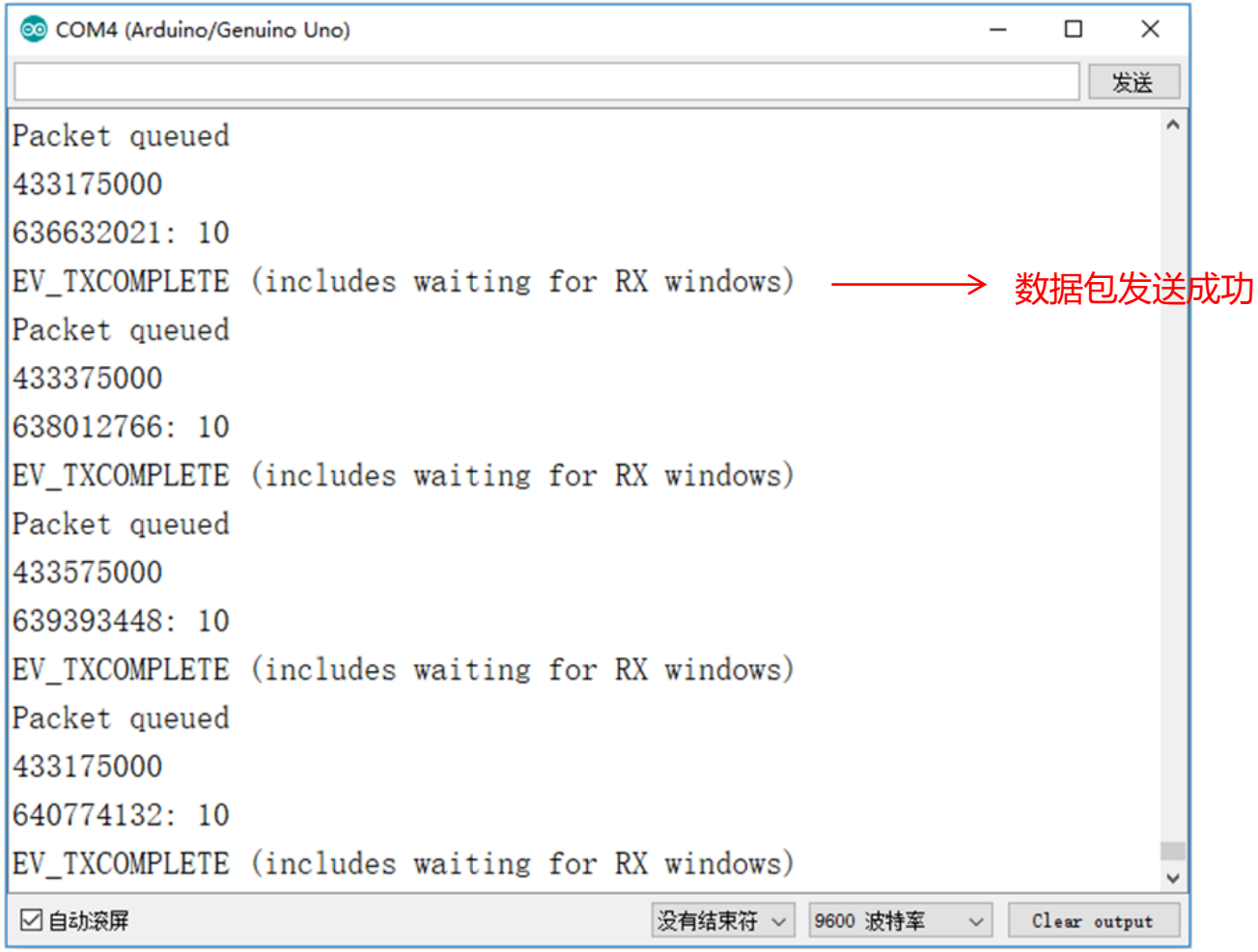
实际传输案例-基于LoRa的数据采集与云端上传

- 步骤七：连接DHT11和LoRa Shield节点，接着使用Arduino IDE烧录应用程序代码（包括传感器数据收集和转发）到LoRa Shield

```
#include <lmic.h>;
#include <hal.h>

static const PROGMEM u1_t NWKSKEY[16] = { 0xB3, 0xC2, 0x99, 0x8A, 0x2B, 0x04,
0x83, 0x5A, 0x34, 0xF6, 0xF1, 0x2E, 0x5A, 0x5F, 0xF8, 0x17 };  //LoRaWAN NwkSKey

static const u1_t PROGMEM APPSKEY[16] = { 0xE6, 0xA3, 0xC2, 0x13, 0x08, 0xFF,
0x89, 0xA0, 0x5B, 0x41, 0x39, 0x83, 0xC7, 0xFD, 0xB6, 0xAA }; //LoRaWAN AppSKey

static const u4_t DEVADDR = 0x26011880; //LoRaWAN end-device address

void setup(){
        Serial.begin(9600);
         LMIC.setFrequency(433000000); //set tx frequency
         LMIC_setDrTxpow(DR_SF7,14); //set data rate and transmission power
         do_send(&sendjob);// start job
}
```

实际传输案例-基于LoRa的数据采集与云端上传

- 步骤八：打开Arduino IDE串口监视器查看实时的LoRa Shield发送结果



数据包发送成功

实际传输案例-基于LoRa的数据采集与云端上传

❑ 步骤九：打开TTN网页查看实时的服务器接收数据

THE THINGS NETWORK CONSOLE COMMUNITY EDITION　Applications　Gateways　Support　tomnet

Applications > dragino_test_aaplication1 > Data

14:27:56	446	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:26:50	443	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:25:44	440	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:24:38	437	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:23:31	434	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:22:25	431	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:21:18	428	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:20:13	425	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:19:06	422	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21
14:18:00	419	1	dev id: lorashield	payload: 61 62 63 64 65 66 67 21

→ 收到节点发送的数据
(ASCII码表示)

研究进展

❑性能分析

- 在NB-IoT标准形成初期，有较多工作分析验证了NB-IoT系统在不同实际场景中（市区/郊区，独立/带内部署等）的性能，其中包括对NB-IoT基站覆盖范围、连接数目的测试，对NB-IoT终端设备数据通信、功耗、安全模式等的测试。
- 对LoRaWAN技术进行介绍讲解，研究其网络架构、协议设计、安全性分析等，大部分工作也将LoRaWAN与其他LPWAN技术进行比较，分析其电池寿命、网络容量、通信时延、通信距离等。

研究进展

❑性能优化及建模仿真

- 随着NB-IoT网络规模的扩大和结构的复杂化，其性能难以同时满足各类用户的实际需求。近两年，有一些工作关注于NB-IoT网络关键性能指标的优化，并提出了自适应的网络参数设置方法。例如：通过调整编解码策略提高数据传输的可靠性；通过调整节点duty cycle模式降低其使用能耗。
- 对LoRaWAN进行建模分析，通过对网络通信参数的修改，仿真大规模节点的性能，如研究不同终端设备数量和分布下，使用相同扩频因子时的网络性能。

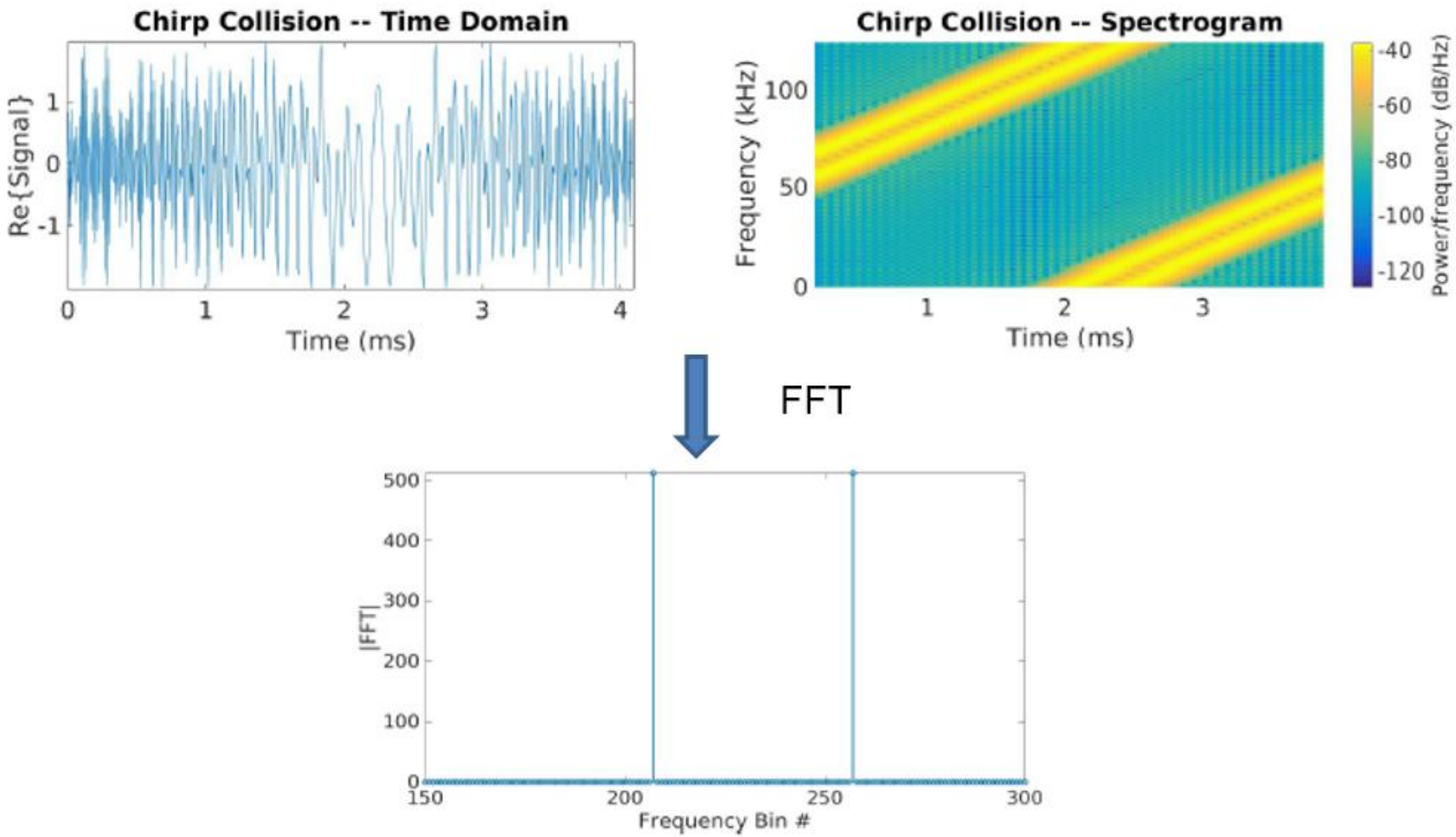
研究进展

❑实际应用

- NB-IoT作为低功耗广域网的一项重要技术，具备突出的性能优势，催生了越来越多的实际应用。比如，在智慧城市、智能家居、农业物联网、可穿戴智能设备等方面已经开始有一些实际的部署和使用。
- 设计了各种原型系统，用于收集数据用于研究,如环境相关数据的收集。或者基于特定应用场景，部署了具体的LoRaWAN系统，使用的硬件常为Semtech LoRa模组与Arduino或者Raspberry Pi。

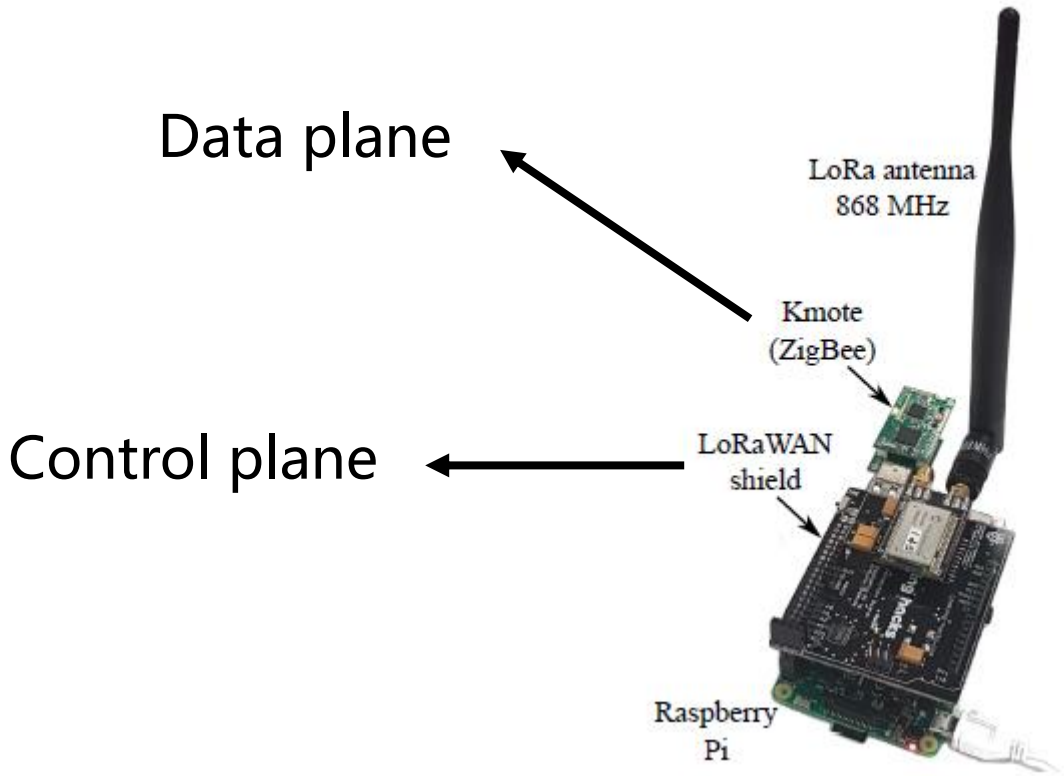
研究进展

[SIGCOMM'17] Empowering Low-Power Wide Area Networks in Urban Settings Devices



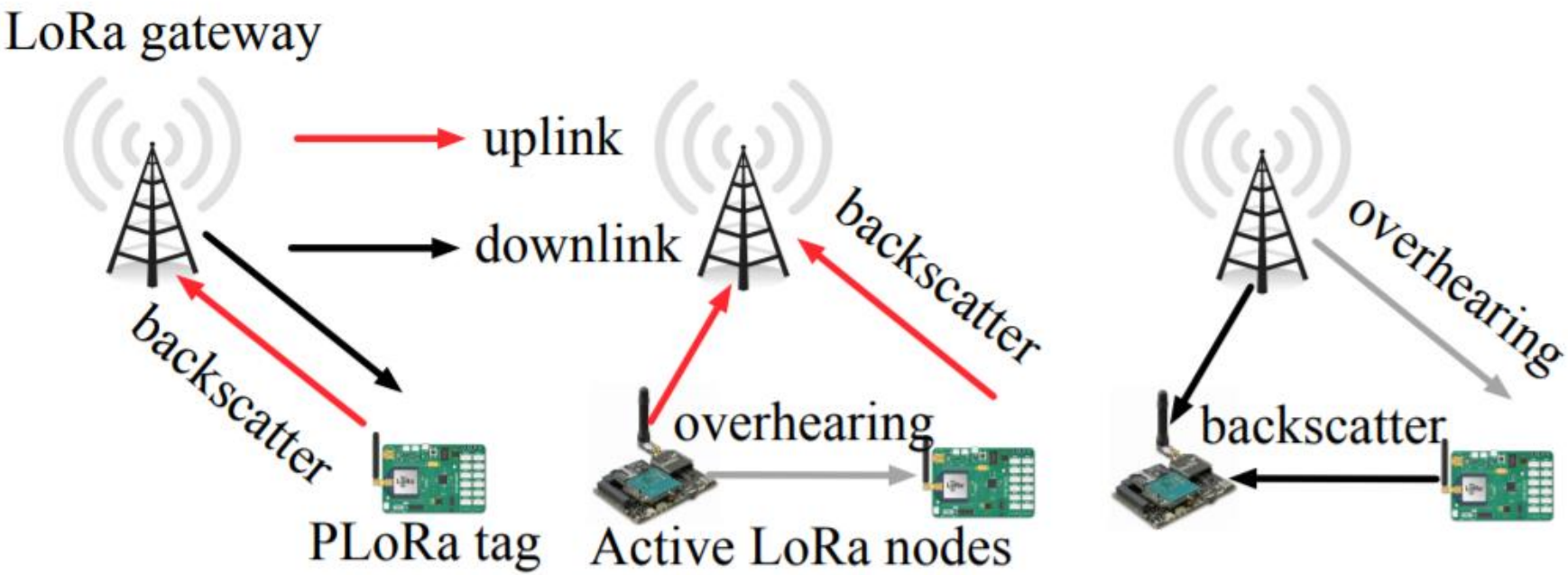
研究进展

[INFOCOM’18] One-Hop Out-of-Band Control Planes for Low-Power Multi-Hop Wireless Networks



研究进展

[SIGCOMM'18] PLoRa: A Passive Long-Range Data Network from Ambient LoRa Transmissions



浙江大学
Zhejiang University

谢谢！

Email: dongw@zju.edu.cn