Table of Contents

LoRaWAN-Specification_ZH_CN

项目介绍

这是《LoRaWAN-Specification》的中文译本。

《LoRaWAN-Specification》是 LoRa 联盟规范的核心协议，由于国内LoRa从业者数量众多，难免有不少伙伴需要中文译本，所以诞生了这个小项目。

项目采用 gitbook 进行编写，地址在 https://www.gitbook.com/book/twowinter/lorawan-specification_zh_cn。

由于这是民间自发的翻译，一些地方翻译可能不够恰当。如果觉得协议的某处比较晦涩，请不要怀疑自己，大概率是翻译的问题。

非常欢迎朋友们反馈翻译问题，争取给行业伙伴们提供一份相对可靠的译本。

项目进展

LoRaWAN_V1.0.2 的中文版本经过1年半时间断断续续的调整，现已优先发布，可点此下载。

其他版本待新建分支来跟进。大家可持续关注github 仓库地址。

贡献者介绍

- IoT小能手 twowinter
- 厦门四信的小伙伴

> 厦门四信的几个小伙伴在业余时间对协议的部分内容做了校对，尤其是 kevin 同学贡献了 CLASS B 等主要章节的翻译，在此表示感谢。
>
> 四信是 LoRa 联盟成员，阿里云金牌合作伙伴，CLAA 钻石合作伙伴，一直致力于为行业伙伴提供稳定可靠的 LoRa 模组、终端、网关系列产品，同时还提供消防、电力、水利等多个垂直行业解决方案。

- 其他积极反馈问题或者提交修改的伙伴，如果你愿意，你的信息将会展示在这。

第1章 介绍

本文档描述了LoRaWAN网络协议，是针对电池供电的终端设备(不管移动还是固定位置)进行优化的一套网络协议。

LoRaWAN网络通常采用星型拓扑结构，由拓扑中的网关来转发终端与后台网络服务器间的消息。网关通过标准IP连接来接入网络服务器，而终端则通过单跳的 LoRa 或者 FSK 来和一个或多个网关通讯。虽然主要传输方式是终端上行传输给网络服务器，但所有的传输通常都是双向的。

终端和网关间的通讯被分散到不同的信道频点和数据速率上。数据速率的选择需要权衡通信距离和消息时长两个因素，使用不同数据速率的设备互不影响。LoRa的数据速率范围可以从 0.3kbps 到 50kbps。为了最大程度地延长终端的电池寿命和扩大网络容量，LoRa网络使用速率自适应(ADR)机制来独立管理每个终端的速率和RF输出。

每个设备可以在任意可用的信道，任意时间，使用任意数据速率发送数据，只要遵守如下规定：

- 终端的每次传输都使用伪随机方式来改变信道。频率的多变使得系统具有更强的抗干扰能力。
- 终端要遵守相应频段和本地区的无线电规定中的最大发射占空比要求。
- 终端要遵守相应频段和本地区的无线电规定中的最大发射时长要求。

> twowinter注：
>
> 发射占空比定义：发射时长占总时长的比例。按照无线电规定，每个设备不能持续占用信道，通过最大发射占空比来限制终端占用信道的时间。 例如某终端发送某数据时的发射时长为1s，当地无线电规定中的最大发射占空比为1%，则该终端需要等候99s才能进行下一次的发射。

1.1 LoRaWAN Classes

所有的LoRaWAN设备都必须至少实现本文档描述的 Class A 功能。另外也可以实现本文档中描述的 Class B 和 Class C 及后续将定义的可选功能。但是在任何情况下，设备都必须兼容 Class A。

1.2 文档约定

MAC命令的格式为 ***LinkCheckReq***(粗斜体)，位和位域的格式为 **FRMPayload** (粗体)，常量的格式为 RECEIVE_DELAY1(非粗非斜体)，变量的格式为 *N*(斜体)。

在本文档中，

- 所有多字节字段的字节序均采用小端模式
- EUI 是8字节字段，采用小端模式传输
- 默认所有RFU保留位都设为0

第2章 LoRaWAN Classes 类型介绍

LoRa 是由Semtech面向长距离、低功耗、低速率应用而开发的无线调制技术。本文档中，将 Class A 基础上实现了更多功能的设备称为“更高 class 终端”。

2.1 LoRaWAN Classes

LoRa网络包含基础LoRaWAN（称之为Class A）和可选功能（Class B，Class C）：

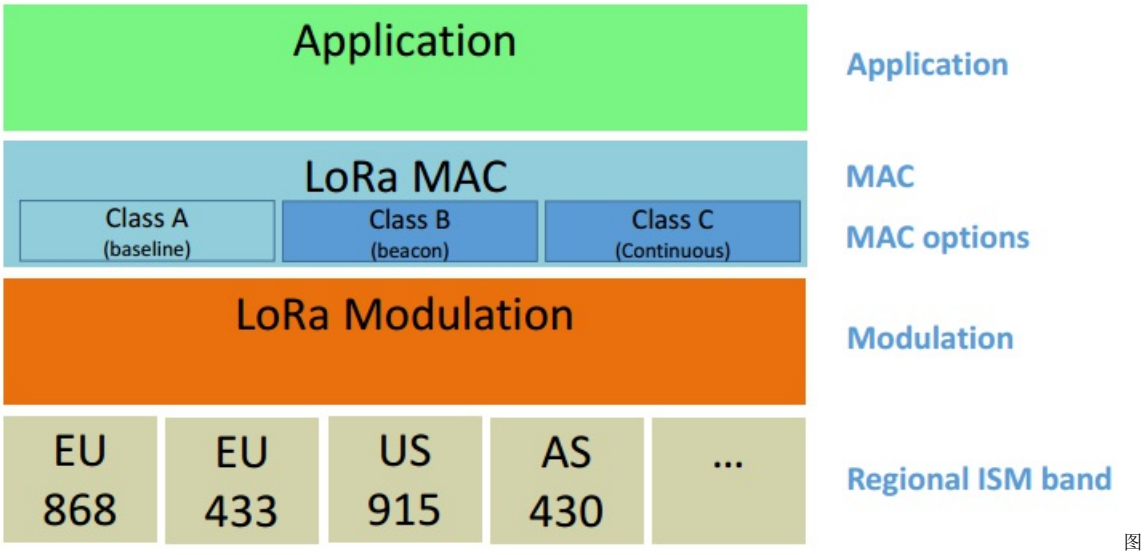


图1.LoRaWAN Classes

- 双向传输终端**(Class A)**： Class A 的终端在每次上行后都会紧跟两个短暂的下行接收窗口，以此实现双向传输。终端基于自身通信需求来安排传输时隙，在随机时间的基础上具有较小的变化(即ALOHA协议)。这种Class A 操作为应用提供了最低功耗的终端系统，只要求应用在终端上行传输后的很短时间内进行服务器的下行传输。服务器在其他任何时间进行的下行传输都得等终端的下一次上行。
- 划定接收时隙的双向传输终端**(Class B)**： Class B 的终端会有更多的接收时隙。除了Class A 的随机接收窗口，Class B 设备还会在指定时间打开别的接收窗口。为了让终端可以在指定时间打开接收窗口，终端需要从网关接收时间同步的信标**(Beacon)**。这使得服务器可以知道终端何时处于监听状态。
- 最大化接收时隙的双向传输终端**(Class C)**： Class C 的终端基本是一直打开着接收窗口，只在发送时短暂关闭。Class C 的终端会比 Class A 和 Class B 更加耗电，但同时从服务器下发给终端的时延也是最短的。

2.2 文档范围

这份LoRaWAN协议还描述了与 Class A 不同的其他 Class 的额外功能。更高 Class 的终端必须满足 Class A 定义的所有功能。

> 注意：物理层帧格式，MAC帧格式，以及协议中更高 class 和 Class A 相同的内容都写在了 Class A 部分，避免内容重复。

CLASS A - ALL END-DEVICE

所有的LoRaWAN终端都必须满足Class A的规定。

第3章 PHY 帧格式

LoRa 有上行消息和下行消息。

3.1 上行消息

上行消息是由终端发出，经过一个或多个网关转发给网络服务器。

上行消息使用 LoRa 射频帧的严格模式，消息中含有 PHDR 和 PHDR_CRC 。载荷有CRC校验来保证完整性。

PHDR，PHDR_CRC 及载荷 CRC 域都通过射频收发器加入。

上行 PHY:

Preamble	PHDR	PHDR_CRC	PHYPayload	CRC

图2.上行PHY帧格式

3.2 下行消息

下行消息是由网络服务器发出，经过单个网关转发给单个终端。

下行消息使用射频帧的严格模式，消息中包含 PHDR 和 PHDR_CRC。

下行 PHY:

Preamble	PHDR	PHDR_CRC	PHYPayload

图3.下行PHY帧格式

3.3 接收窗口

每个上行传输后终端都要开两个短的接收窗口。接收窗口开始时间的规定，是以传输结束时间为参考。

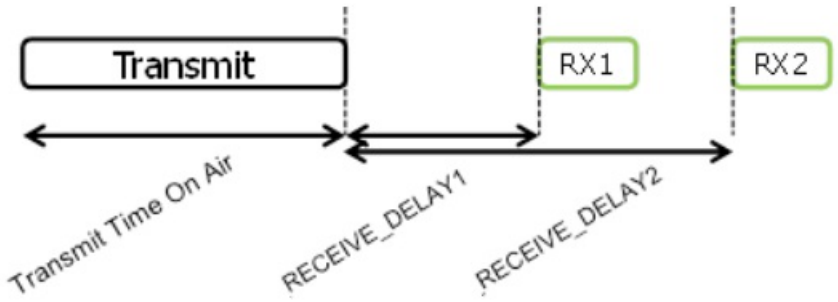


图4.终端接收时隙的时序图

3.3.1 第一接收窗口的信道，数据速率和启动

第一接收窗口 RX1 使用的频率和上行频率有关，使用的速率和上行速率有关。RX1 是在上行调制结束后的 RECEIVE_DELAY1秒（+/- 20微秒）打开。上行和 RX1 时隙下行速率的关系是按区域规定，详细描述在[LoRaWAN地区参数]文件中。默认第一窗口的速率是和最后一次上行的速率相同。

3.3.2 第二接收窗口的信道，数据速率和启动

第二接收窗口 RX2 使用一个固定可配置的频率和数据速率，在上行调制结束后的 RECEIVE_DELAY2秒（+/- 20微秒）打开。频率和数据速率可以通过 MAC 命令(见 第5章)。默认的频率和速率是按区域规定，详细描述在[LoRaWAN地区参数]文件中。

3.3.3 接收窗口的持续时间

接收窗口的长度至少要让终端射频收发器有足够的时间来检测到下行的前导码。

3.3.4 接收方在接收窗口期间的处理

如果在任何一个接收窗口中检测到前导码，射频收发器需要继续激活，直到整个下行帧都解调完毕。如果在第一接收窗口检测到数据帧，且这个数据帧的地址和MIC校验通过确认是给这个终端，那终端就不必开启第二个接收窗口。

3.3.5 网络发送消息给终端

如果网络想要发一个下行消息给终端，它会精确地在两个接收窗口的起始点发起传输。

3.3.6 接收窗口的重要事项

终端在第一或第二接收窗口中收完下行消息后，或者在第二接收窗口失效之后(第一或第二窗口均未收到下行消息)，才能再发起另一个上行消息。

3.3.7 其他协议的收发处理

节点在LoRaWAN收发窗口阶段可以收发其他协议，只要终端能满足当地要求以及兼容LoRaWAN协议。

第4章 MAC帧格式

LoRa所有上下行链路消息都会携带PHY载荷，PHY载荷以1字节MAC头(MHDR)开始，紧接着MAC载荷(MACPayload)，最后是4字节的MAC校验码(MIC)。

射频PHY层：

Preamble	PHDR	PHDR_CRC	PHYPayload	CRC

图5.射频PHY结构(注意 CRC只有上行链路消息中存在) PHY载荷：

MHDR	MACPayload	MIC

或者

MHDR	Join-Request	MIC

或者

MHDR	Join-Response	MIC

图6.PHY载荷结构 MAC载荷：

FHDR	FPort	FRMPayload

图7.MAC载荷结构 FHDR：

DevAddr	FCtrl	FCnt	FOpts

图8.帧头结构

图9.LoRa帧格式元素(即图5~8)

4.1 MAC层(PHYPayload)

Size (bytes)	1	1..M	4
PHYPayload	MHDR	MACPayload	MIC

MACPayload字段的最大长度M，在第6章有详细说明。

4.2 MAC头(MHDR字段)

Bit#	7..5	4..2	1..0
MHDR bits	MType	RFU	Major

MAC头中指定了消息类型(MType)和帧编码所遵循的LoRaWAN规范的主版本号(Major)。

4.2.1 消息类型(MType位字段)

LoRaWAN定义了六个不同的MAC消息类型：join request, join accept, unconfirmed data up/down, 以及 confirmed data up/down 。

MType	描述
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	RFU
111	Proprietary

表1.MAC消息类型

- 4.2.1.1 Join-request and join-accept 消息

join-request和join-accept都是用在空中激活流程中，具体见章节6.2

- 4.2.1.2 Data messages

Data messages 用来传输MAC命令和应用数据，这两种命令也可以放在单个消息中发送。 Confirmed-data message 接收者需要应答。 Unconfirmed-data message 接收者则不需要应答。 Proprietary messages 用来处理非标准的消息格式，不能和标准消息互通，只能用来和具有相同拓展格式的消息进行通信。

不同消息类型用不同的方法保证消息一致性，下面会介绍每种消息类型的具体情况。

4.2.2 数据消息的主版本(Major位字段)

Major位字段	描述
00	LoRaWAN R1
01..11	RFU

表2.Major列表

> 注意：Major定义了激活过程中(join procedure)使用的消息格式（见章节6.2）和MAC Payload的前4字节（见第4章）。终端要根据不同的主版本号实现不同最小版本的消息格式。终端使用的最小版本应当提前通知网络服务器。

4.3 MAC载荷(MACPayload)

MAC载荷，也就是所谓的“数据帧”，包含：帧头（FHDR）、端口（FPort）以及帧载荷(FRMPayload），其中端口和帧载荷是可选的。

4.3.1 帧头(FHDR)

FHDR是由终端短地址(DevAddr)、1字节帧控制字节(FCtrl)、2字节帧计数器(FCnt)和用来传输MAC命令的帧选项(FOpts，最多15个字节)组成。

Size(bytes)	4	1	2	0..15
FHDR	DevAddr	FCtrl	FCnt	FOpts

FCtrl在上下行消息中有所不同，下行消息如下：

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

上行消息如下：

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

- 4.3.1.1 帧头中 自适应数据速率 的控制(ADR, ADRACKReq in FCtrl)

LoRa网络允许终端采用任何可能的数据速率。LoRaWAN协议利用该特性来优化固定终端的数据速率。这就是自适应数据速率(Adaptive Data Rate (ADR))。当这个使能时，网络会优化使得尽可能使用最快的数据速率。

移动的终端由于射频环境的快速变化，数据速率管理就不再适用了，应当使用固定的数据速率。

如果**ADR**的位字段有置位，网络就会通过相应的MAC命令来控制终端设备的数据速率。如果**ADR**位没设置，网络则无视终端的接收信号强度，不再控制终端设备的数据速率。**ADR**位可以根据需要通过终端及网络来设置或取消。不管怎样，ADR机制都应该尽可能使能，帮助终端延长电池寿命和扩大网络容量。

> 注意：即使是移动的终端，可能在大部分时间也是处于非移动状态。因此根据它的移动状态，终端也可以请求网络使用ADR来帮助优化数据速率。

如果终端被网络优化过的数据速率高于自己默认的数据速率，它需要定期检查下网络仍能收到上行的数据。每次上行帧计数都会累加(是针对于每个新的上行包，重传包就不再增加计数)，终端增加 ADR_ACK_CNT 计数。如果直到ADR_ACK_LIMIT次上行(ADR_ACK_CNT >= ADR_ACK_LIMIT)都没有收到下行回复，它就得置高ADR应答请求位(**ADRACKReq**)。 网络必须在规定时间内回复一个下行帧，这个时间是通过ADR_ACK_DELAY来设置，上行之后收到任何下行帧就要把ADR_ACK_CNT的计数重置。当终端在接收时隙中的任何回复下行帧的ACK位字段不需要设置，表示网关仍在接收这个设备的上行帧。如果在下一个ADR_ACK_DELAY上行时间内都没收到回复(例如，在总时间

ADR_ACK_LIMIT+ADR_ACK_DELAY之后)，终端必须切换到下一个更低速率，使得能够获得更远传输距离来重连网络。终端如果在每次ADR_ACK_DELAY到了之后依旧连接不上，就需要每次逐步降低数据速率。如果终端用它的默认数据速率，那就不需要置位**ADRACKReq**，因为无法帮助提高链路距离。

> 注意：不要ADRACKReq立刻回复，这样给网络预留一些余量，让它做出最好的下行调度处理。
>
> 注意：上行传输时，如果 ADR_ACK_CNT >= ADR_ACK_LIMIT 并且当前数据速率比设备的最小数据速率高，就要设置 ADRACKReq，其它情况下不需要。

- 4.3.1.2 消息应答位及应答流程(ACK in FCtrl)

收到confirmed类型的消息时，接收端要回复一条应答消息(应答位ACK要进行置位)。如果发送者是终端，网络就利用终端发送操作后打开的两个接收窗口之一进行回复。如果发送者是网关，终端就自行决定是否发送应答。 应答消息只会在收到消息后回复发送，并且不重发。

> 注意：为了让终端尽可能简单，尽可能减少状态，在收到confirmation类型需要确认的数据帧，需要立即发送一个严格的应答数据帧。或者，终端会延迟发送应答，在它下一个数据帧中再携带。

- 4.3.1.3 重传流程

当需要应答却没收到应答时就会进行重发，重发的个数由终端自己定，可能每个终端都不一样，这个参数也可以由网络服务器来设置调整。

> 注意：一些应答机制的示例时序图在第18章中有提供。
>
> 注意：如果终端设备重发次数到达了最大值，它可以降低数据速率来重连。至于后面是否再重发还是说丢弃不管，都取决于终端自己。
>
> 注意：如果网络服务器重发次数到达了最大值，它就认为该终端掉线了，直到它再收到终端的消息。一旦和终端设备的连接出现问题时，要不要重发都取决于网络服务器自己。
>
> 注意：在重传期间的数据速率回退的建议策略在章节18.4中有描述。

- 4.3.1.4 帧挂起位(FPending in FCtrl 只在下行有效)

帧挂起位(FPending)只在下行交互中使用，表示网关还有挂起数据等待下发，需要终端尽快发送上行消息来再打开一个接收窗口。

FPending的详细用法在章节18.3。

- 4.3.1.5 帧计数器(FCnt)

每个终端有两个计数器跟踪数据帧的个数，一个是上行链路计数器（FCntUp），由终端在每次上行数据给网络服务器时累加；另一个是下行链路计数器（FCntDown），由服务器在每次下行数据给终端时累计。 网络服务器为每个终端跟踪上行帧计数及产生下行帧计数。 终端入网成功后，终端和服务端的上下行帧计数同时置0。 每次发送消息后，发送端与之对应的 FCntUp 或 FCntDown 就会加1。 接收方会同步保存接收数据的帧计数，对比收到的计数值和当前保存的值，如果两者相差小于 MAX_FCNT_GAP （要考虑计数器滚动），接收方就按接收的帧计数更新对应值。如果两者相差大于 MAX_FCNY_GAP 就说明中间丢失了很多数据，这条以及后面的数据就被丢掉。

LoRaWAN的帧计数器可以用16位和32位两种，节点上具体执行哪种计数，需要在带外通知网络侧，告知计数器的位数。 如果采用16位帧计数，**FCnt**字段的值可以使用帧计数器的值，此时有需要的话通过在前面填充0（值为0）字节来补足；如果采用32位帧计数， **FCnt**就对应计数器32位的16个低有效位(上行数据使用上行FCnt，下行数据使用下行FCnt)。

终端在相同应用和网络密钥下，不能重复用相同的FCntUp数值，除非是重传。

- 4.3.1.6 帧可选项(FOptsLen in FCtrl, FOpts) FCtrl 字节中的FOptsLen位字段描述了整个帧可选项(FOpts)的字段长度。

FOpts字段存放MAC命令，最长15字节，详细的MAC命令见章节4.4。

如果FOptsLen为0，则FOpts为空。在FOptsLen非0时，则反之。如果MAC命令在FOpts字段中体现，port0不能用(FPort要么不体现，要么非0)。

MAC命令不能同时出现在FRMPayload和FOpts中，如果出现了，设备丢掉该组数据。

4.3.2 端口字段(FPort)

如果帧载荷字段不为空，端口字段必须体现出来。端口字段有体现时，若FPort的值为0表示FRMPayload只包含了MAC命令；具体见章节4.4中的MAC命令。 FPort的数值从1到223(0x01..0xDF)都是由应用层使用。 FPort的值从224到255(0xE0..0xFF)是保留用做未来的标准应用拓展。

Size(bytes)	7..23	0..1	0..N
MACPayload	FHDR	FPort	FRMPayload

N是应用程序载荷的字节个数。N的有效范围具体在第7章有定义。

N应该小于等于： N <= M - 1 - (FHDR长度) M是MAC载荷的最大长度。

4.3.3 MAC帧载荷加密(FRMPayload)

如果数据帧携带了载荷，FRMPayload必须要在MIC计算前进行加密。 加密机制是采用IEEE802.15.4/2006的AES128算法。

默认的，加密和解密由LoRaWAN层来给所有的FPort来执行。如果加密/解密由应用层来做更方便的话，也可以在LoRaWAN层之上给特定FPorts来执行，除了端口0。具体哪个节点的哪个FPort在LoRaWAN层之外要做加解密，必须要和服务器通过out-of-band信道来交互(见第19章)。

- 4.3.3.1 LoRaWAN的加密

密钥K根据不同的FPort来使用：

FPort	K
0	NwkSKey
1..255	AppSKey

表3: FPort列表 具体加密是这样： pld = FRMPayload 对于每个数据帧，算法定义了一个块序列Ai，i从1到k，k = ceil(len(pld) / 16)：

Size(bytes)	1	4	1	4	4	1	1
Ai	0x01	4 x 0x00	Dir	DevAddr	FCntUp or FCntDown	0x00	i

方向字段(**Dir**)在上行帧时为0，在下行帧时为1. 块Ai通过加密，得到一个由块Si组成的序列S。

> Si = aes128_encrypt(K, Ai) for i = 1..k S = S1 | S2 | .. | Sk

通过异或计算对payload进行加解密：

- 4.3.3.2 LoRaWAN层之上的加密

如果LoRaWAN之上的层级在已选的端口上(但不能是端口0，这是给MAC命令保留的)提供了预加密的FRMPayload给LoRaWAN，LoRaWAN则不再对FRMPayload进行修改，直接将FRMPayload从MACPayload传到应用层，以及从应用层传到MACPayload。

4.4 消息校验码(MIC)

消息检验码要计算消息中所有字段。 msg = MHDR | FHDR | FPort | FRMPayload

MIC是按照[RFC4493]来计算：

> cmac = aes128_cmac(NwkSKey, B0 | msg) MIC = cmac[0..3]

块B0的定义如下：

Size(bytes)	1	4	1	4	4	1	1
B0	0x49	4 x 0x00	Dir	DevAddr	FCntUp or FCntDown	0x00	len(msg)

方向字段(**Dir**)在上行帧时为0，在下行帧时为1。

第5章 MAC命令

对网络管理者而言，有一套专门的MAC命令用来在服务器和终端MAC层之间交互。这套MAC命令对应用程序或者应用服务器或者运行在终端设备上的应用程序是不可见的。

单个数据帧中可以包含MAC命令序列，要么在FOpts字段中捎带，要么作为独立帧将**FPort**设成0后放在FRMPayload里。如果采用FOpts捎带的方式，MAC命令不进行加密并且长度不能超过15字节。如果采用独立帧放在**FRMPayload**的方式，那就必须采用加密方式，并且不能超过**FRMPayload**的最大长度。

> 注意：如果MAC命令不想被窃听，那就必须以独立帧形式放在FRMPayload中进行发送。

每个MAC命令是由 1字节命令码 (CID) 跟着一段可能为空的特定命令字节序列组成的。

CID	Command	由谁发送		描述
		终端	网关	
0x02	LinkCheckReq	x		终端利用这个命令来判断网络连接质量
0x02	LinkCheckAns		x	LinkCheckReq的回复。包含接收信号强度，告知终端接收质量
0x03	LinkADRReq		x	向终端请求改变数据速率，发射功率，重传率以及信道
0x03	LinkADRAns	x		LinkADRReq的回复。
0x04	DutyCycleReq		x	向终端设置发送的最大占空比。
0x04	DutyCycleAns	x		DutyCycleReq的回复。
0x05	RXParamSetupReq		x	向终端设置接收时隙参数。
0x05	RXParamSetupAns	x		RXParamSetupReq的回复。
0x06	DevStatusReq		x	向终端查询其状态。
0x06	DevStatusAns	x		返回终端设备的状态，即电池余量和链路解调预算。
0x07	NewChannelReq		x	创建或修改 1个射频信道 定义。
0x07	NewChannelAns	x		NewChannelReq的回复。
0x08	RXTimingSetupReq		x	设置接收时隙的时间。
0x08	RXTimingSetupAns	x		RXTimingSetupReq的回复。
0x09	TxParamSetupReq		x	网络服务器用于设置基于当地规定的终端的最大允许驻留时间和最大EIRP
0x09	TxParamSetupAns	x		TxParamSetupReq的回复。
0x0A	DlChannelReq		x	通过从上行链路频率移位下行链路频率（即创建非对称信道）来修改下行链路RX1无线电信道的定义
0x0A	DlChannelAns	x		DlChannelReq的回复。
0x80~0xFF	私有	x	x	给私有网络命令拓展做预留。

表4：MAC命令表

> 注意：MAC命令的长度虽然没有明确给出，但是MAC执行层必须要知道。因此未知的MAC命令无法被忽略，且前面未知的MAC命令会终止MAC命令的处理队列。所以建议按照LoRaWAN协议介绍的MAC命令来处理MAC命令。这样所有基于LoRaWAN协议的MAC命令都可以被处理，即使是更高版本的命令。

> 注意：由网络服务器调整的任何值（例如，RX2、新的或已调整的信道定义）仅在终端设备的下一次加网之前有效。因此，在每个成功加网之后，终端设备将再次使用默认参数，并由网络服务器根据需要重新调整值。

5.1 Link Check 命令 (LinkCheckReq, LinkCheckAns)

通过LinkCheckReq命令，终端可以知道是否已连接上服务器。该命令没有载荷。

当网络服务器通过一个或者多个网关接收到LinkCheckReq命令时，它会以LinkCheckAns命令进行回复。

Size (bytes)	1	1

LinkCheckAns Payload	Margin	GwCnt

解调预算(**Margin**)是一个范围为0~254的8位无符号整数，表示成功接收最新的**LinkCheckReq**命令的链路预算(单位为dB)。若 Margin 值为"0"则意味着数据帧是在解调水平上进行接收(0 dB或者没有预算)，当 Margin 值为"20"时则意味着数据帧到达在解调水平之上20dB的网关。

网关计数(**GwCnt**)是成功接收最新的**LinkCheckReq**命令的网关个数。

5.2 Link ADR 命令(LinkADRReq, LinkADRAns)

通过 LinkADRReq 命令，NS(网络服务器)可以调整终端的数据速率。

Size (bytes)	1	2	1
LinkADRReq Payload	DataRate_TXPower	ChMask	Redundancy

Bits	[7:4]	[3:0]
DataRate_TXPower	DataRate	TXPower

所请求的数据速率(**DataRate**)和发射功率(**TXPower**)是根据区域规定，体现在LoRaWAN协议中文版_配套文件 地区参数(物理层) 中。命令中的发射功率字段指的是设备可操作的最大发射功率。如果命令中的发射功率高于终端实际发射功率的最大值，终端也要应答成功，这种情况下，将终端的发射功率尽可能提高到最大值。 信道掩码(ChMask)字段指示了上行链路的可用信道，从最低位bit0表示开始。

Bit#	Usable channels
0	Channel 1
1	Channel 2
..	..
15	Channel 16

表5：信道状态表

ChMask 字段的对应位如果设置为1，则表示对应的信道可以进行上行传输，只要该信道允许终端使用该数据速率。如果对应位设置为0，则表示相应信道不可用。

Bits	7	[6:4]	[3:0]
Redundancy bits	RFU	ChMaskCntl	NbTrans

Redundancy 字段中的 **NbTrans** 位域，指的是每个上行消息的发送次数，这仅对 "unconfirmed" 消息有作用。这个字段的默认值为1，相对应的是每个数据帧只进行单次传输，有效范围是[1:15]。如果收到 **NbTrans** == 0，终端需要使用默认值。这个位域可以被NS(网络服务器)用来控制节点上行的 Redundancy 从而获得QOS(服务质量)。在重传帧时节点通常会跳频，每次重传需要等到接收窗口超时。只要在RX1期间收到下行消息，该上行消息则不再进行任何重传。对于 Class A 设备，RX2时隙的接收也是一样处理。

ChMaskCntl 位域和之前定义的 ChMask 字段有关，它控制了ChMask所指定的16个信道块。也可以对所用信道进行全局的打开或关闭。这个位域的使用是根据区域规定，体现在LoRaWAN协议中文版_配套文件 地区参数(物理层) 中。

NS(网络服务器)可能会在单个下行帧中包含多个 LinkAdrReq 命令。终端为了配置 channel mask ，将会按照下行消息中的命令块的顺序，逐一地处理所有的 LinkAdrReq 消息。 终端可能会接收或者拒绝命令块中所有 channel mask 的控制，在逐个 LinkAdrAns 命令块中体现连续的 Channel Mask ACK 状态，来指示相应的 channel mask 接受与否。 终端在连续命令块时只处理最后一个消息中的 DataRate, TXPower 和 NbTrans 字段，因为这些参数将会决定终端的全局状态。终端需要在每一个 LinkAdrAns 命令中体现 ACK 状态，来指示对这些最终设置的接受与否。

信道频点信息是按地区规定，在第6章中有定义。终端使用 LinkADRAns 命令来应答 LinkADRReq 命令。

Size (bytes)	1
LinkADRAns Payload	Status

Bits	[7:3]	2	1	0
Status bits	RFU	Power ACK	Data rate ACK	Channel mask ACK

LinkADRAns 的 Status 位域按照如下定义：

	Bit = 0	Bit = 1
Channel mask ACK	所发的 channel mask 使能了未定义的信道或者禁用了所有信道。命令被丢弃，终端状态不变。	所发的 channel mask 已成功解析，已按照 mask 设置了当前的信道状态。

Data rate ACK	所请求的数据速率，终端无法识别，或者无法应用在当前信道中（不支持任何使能的信道)。命令被丢弃，终端状态不变。	数据速率成功设置。
Power ACK	所请求的发射功率不能在终端上执行。命令被丢弃，终端状态不变。	功率等级成功设置。

如果这三个位中有任何一位等于0，则**LinkADRReq**命令没有成功，节点保持之前的状态。

5.3 终端发射占空比(DutyCycleReq, DutyCycleAns)

DutyCycleReq命令被网络协调者用来限制终端的最大总计发射占空比。最大总计发射占空比覆盖所有子频段上的发射占空比。

Size (bytes)	1
DutyCycleReq Payload	DutyCyclePL

Bits	7:4	3:0
DutyCyclePL	RFU	MaxDCycle

终端所允许的最大发射占空比为： aggregated duty cycle = 1/(2^MaxDcycle)

MaxDutyCycle的有效范围为[0:15].MaxDutyCycle的值若为0则表示“无发射占空比限制”，除非各地区有对发射占空比进行限制。

5.4 接收窗口参数(RXParamSetupReq,RXParamSetupAns)

RXParamSetupReq命令可以对每个上行消息之后的第二接收窗口(RX2)的频率以及数据速率进行改变。该命令还可以对上行数据速率和RX1下行数据速率的偏移量进行改变。

Size (bytes)	1	3
RXParamSetupReq Payload	DLsettings	Frequency

Bits	7	6:4	3:0
DLsettins	RFU	RX1DRoffset	RX2DataRate

RX1DRoffset位域设置上行数据速率和RX1下行数据速率的偏移量。默认情况下偏移量为0（意思就是上行数据速率与下行数据速率相等)。偏移量用于考虑一些地区的基站最大功率密度限制和平衡上下行射频链路预算。

RX2DataRate位域定义了第二接收窗口的下行链路数据速率，遵循与**LinkADRReq**命令相同的规则（例如，0表示DR0/125kHz)。**Frequency**位域所设置的是第二接收窗口所使用信道的频率，该频率按照与**NewChannelReq**命令相同的规则进行定义。

终端使用**RXParamSetupAns**命令对**RXParamSetupReq**命令进行应答。**RXParamSetupAns**命令应该添加在所有的上行链路数据帧的**Fopt**字段中直到终端接收到一个Class A类型的下行链路数据帧。这样就可以保证即使在上行链路帧丢失的情况之下，网络服务器总是可以知道终端所使用的下行链路参数。

RXParamSetupAns命令的载荷为一个字节的状态信息。

Size (bytes)	1
RXParamSetupAns Payload	Status

Status各位的含义如下：

Bits	7:3	2	1	0
Status bits	RFU	RX1DRoffset ACK	RX2 Data Rate ACK	Channel ACK

	Bit = 0	**Bit = 1**
Channel ACK	终端无法使用请求的频率	RX2时隙信道频率设置成功
RX2 Data rate ACK	终端无法识别请求的数据速率	RX2时隙数据速率设置成功
RX1DRoffset ACK	上行数据速率与RX1下行数据速率的偏移量不在允许的范围之内	RX1DRoffset设置成功

如果3个位的任何一位为0，则**RXParamSetupReq**命令不成功，节点保持之前的状态。

5.5 终端状态(DevStatusReq, DevStatusAns)

通过 **DevStatusReq** 命令，NS(网络服务器)可以获取终端的状态信息。该命令无载荷。一旦终端收到 DevStatusReq 命令，则会回复 **DevStatusAns** 命令。

Size (bytes)	1	1
DevStatusAns Payload	Battery	Margin

报告电池电量（**Battery**)的编码如下:

Battery	**Description**
0	终端连接到外部电源
1..254	数值表示电池电量，1表示最低，254表示最高
255	终端无法测量电池电量

图8：电池电量码表

Margin是最近一次成功接收**DevStatusReq**命令的解调信噪比(该值必须是四舍五入到最近的整数值，单位为dB）。它是6位的有符号整数（最小值为 -32，最大值为31）。

Bits	7:6	5:0
Status bits	RFU	Margin

5.6 信道的创建和修改(NewChannelReq, NewChannelAns, DlChannelReq, DlChannelAns)

NewChannelReq命令可以用于修改现有的双向信道或者创建一个新的信道。这个命令设置了新信道的中心频率还有上行数据速率的可用范围：

Size (bytes)	1	3	1
NewChannelReq Payload	ChIndex	Freq	DrRange

信道索引**ChIndex**是正在创建或者正在修改的信道的索引。根据所使用的区域和频带，LoRaWAN规范强加了所有设备通用的默认信道，该信道不能被**NewChannelReq**命令修改(具体见章节6)。如果默认信道的个数为N，则默认信道的编号从0~(N-1)，并且**ChIndex**的可接受范围为N~15。一个设备必须至少能处理16个 不同的信道定义。在某些特定的区域，设备可能必须存储超过16个信道定义。

Freq位域是一个24位无符号整数。实际信道频率为（100×**Freq**），单位为HZ，其中表示低于100MHz的频率数值将会保留供将来使用。**Freq**可以设置从100MHz~1.67GHz之间的信道频率，但必须以100Hz为单位(因为实际信道频率为（100×**Freq**））。终端必须检测该频率是否能被射频硬件所使用，若不行则需返回错误。

DrRange位域规定了这个信道所允许的上行数据速率范围。该位域被分为两个4位字段：

Bits	7:4	3:0
DrRange	MaxDR	MinDR

按照章节5.2所定义的规则，最小数据速率**MinDR**字段规定了这个信道所允许的最低上行数据速率。例如0表示DR0/125kHz。类似的，最大数据速率**MaxDR**规定了最高上行数据速率。例如，若**DrRange**=0x77则表示一个信道只允许50kbps的GFSK；若**DrRange**=0x50表示一个信道支持DR0/125kHz到DR5/125kHZ的频率范围。

最近定义以及修改的信道被使能之后可以立刻用于通信。RX1的下行频率与上行频率相等。

终端以**NewChannelAns**命令对**NewChannelReq**进行应答。**NewChannelAns**命令的载荷包含了以下信息：

Size (bytes)	1
NewChannelAns Payload	Status

Status位有以下含义：

Bits	7:2	1	0
Status	RFU	Data rate range ok	Channel frequency ok

	Bit = 0	Bit = 1
Data rate range ok	指定的数据速率超出了终端当前定义的范围	该数据速率与终端能够兼容
Channel frequency ok	终端无法使用该频率	终端能够使用该频率

如果以上两位其中之一为0，则**NewChannelReq**命令不成功，新的信道将不会产生。

DlChannelReq命令允许服务器在RX1时隙使用不同的下行链路频率。这个命令可以适用于所有支持**NewChannelReq**命令的地理区域(例如欧盟和中国，但是不适用于美国和澳大利亚，具体详见《LoRaWAN Regional Parameters document [PARAMS]》。

该命令用于设置RX1时隙的下行消息中心频率，如下：

Size (bytes)	1	3
DlChannelReq Payload	ChIndex	Freq

ChIndex是要修改下行频率的信道的索引。

Freq位域是24位的无符号整数。实际信道频率为（100×**Freq**），单位为HZ，其中表示低于100MHz的频率数值将会保留供将来使用。终端必须检测该频率是否能被射频硬件所使用，若不行则需返回错误。

终端以**DlChannelAns**命令对**DlChannelReq**命令进行应答。**DlChannelAns**命令在终端没有接收到一个下行数据之前必须添加在所有的上行数据**FOpt**位域中。这样才能保证在上行数据包丢失的情况之下，网络服务器总是能够知道终端所使用的下行频率。

该命令的载荷有如下信息：

Size (bytes)	1
DlChannelAns Payload	Status

Status位有以下的含义：

Bits	7:2	1	0
Status	RFU	上行频率可用	信道频率可用

	Bit = 0	Bit = 1
Channel frequency ok	终端无法使用该频率	终端无法使用该频率
Channel frequency ok	该信道无法使用此上行频率，只能为已经具有一个有效上行频率的信道设置下行频率	信道的上行频率有效

5.7 TX 和 RX 之间的延时设置(RXTimingSetupReq, RXTimingSetupAns)

RXTimingSetupReq命令允许配置TX上行链路发送完毕之后与第一个接收窗口打开之间的延时。第二接收窗口在第一接收窗口打开之后的1秒打开。

Size (bytes)	1
RXTimingSetupReq Payload	Settings

Delay位域指定了延时时间。这个位域被分为两个4位字段：

Bits	7:4	3:0
Settings	RFU	Del

延时时间的单位为秒。Del的值为0时对应的延时时间为1s。

Del	**Delay[s]**
0	1
1	1
2	2
3	3
..	..
15	15

图11：延时时间映射表

终端用于应答**RXTimingSetupReq**命令的**RXTimingSetupAns**命令没有载荷。

RXTimingSetupAns命令在终端没有接收到一个下行数据之前必须添加在所有的上行数据**FOpt**位域中。这样才能保证在上行数据包丢失的情况之下，网络服务器总是能够知道终端所使用的下行频率。

5.8 终端发送参数(TxParamSetupReq, TxParamSetupAns)

该命令只需要在特定的可调节地区进行使用。具体请参考《LoRaWAN Regional Parameters document [PARAMS]》。

TxParamSetupReq可以用于通知终端的最大允许驻留时间，换言之，一包数据在空中的最大持续传输时间,以及终端所允许的最大等效全向辐射功率**(EIRP)**。

> KevinCao注： **EIRP**解释：为无线电发射机供给天线的功率与在给定方向上天线绝对增益的乘积。各方向具有相同单位增益的理想全向天线，通常作为无线通信系统的参考天线。EIRP定义为：EIRP=Pt*Gt，它表示同全向天线相比，可由发射机获得的在最大天线增益方向上的 发射功率。Pt表示发射机的发射功率，Gt表示发射天线的天线增益。在无线通信工程中，通常用来衡量干扰的强度，以及发射机发射强信号的能力。

Size (bytes)	1
TxParamSetup payload	EIRP_DwellTime

EIRP_DwellTime位域的结构如下所述：

Bits	7：6	5	4	3：0
MaxDwellTime	RFU	DownlinkDwellTime	UplinkDwellTime	MaxEIRP

TxParamSetupReq命令的[0..3]位是用于表示**EIRP**的最大值，每个编码值对应的最大**EIRP**值的映射表如下。这张表中的最大**EIRP**的值的变化范围由各个地区来进行自行规定。

Coded Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Max EIRP(dBm)	8	10	12	13	14	16	18	20	21	24	26	27	29	30	33	36

最大**EIRP**指的是设备无线电发射功率的上限。设备不需要使用该功率进行传输，但是绝不会辐射超过指定的EIRP。

第4位和第5位分别定义了上行链路和下行链路的驻留时间，驻留时间的映射编码表如下所示：

Coded Value	**Dwell Time**
0	No Limit
1>	400 ms

当该Mac命令生效时，(因区域而定）终端会以**TxParamSetupAns**命令对**TxParamSetupReq**命令进行回复。**TxParamSetupAns**命令不包含任何载荷。

当在某个区域内是不需要**TxParamSetupReq**命令时，则终端不会对该命令进行任何处理并且不会进行回复。

第6章 终端激活

为了加入LoRaWAN网络，每个终端需要初始化及激活。

终端的激活有两种方式，一种是空中激活 Over-The-Air Activation (OTAA)，当设备部署和重置时使用; 另一种是独立激活 Activation By Personalization (ABP)，此时初始化和激活这两步就在一个步骤内完成。

> twowinter 备注： ABP 这个词不太好翻译，通常会翻成个性化激活，也就是通过独立配置参数的方式激活。但总感觉少点味道，与空中激活摆在一起，感觉独立激活这个词在语义上更有并列感。当然这是我的主观感觉，建议大家和同行交流时，还是说 ABP激活 吧。

6.1 终端激活后的数据存储

激活后，终端会存储如下信息：设备地址(DevAddr)，应用ID(AppEUI)，网络会话密钥(NwkSKey)，应用会话密钥(AppSKey)。

- 6.1.1 终端地址(DevAddr)

终端地址(DevAddr)由可标识当前网络设备的32位ID所组成，具体格式如下：

Bit#	[31..25]	[24..0]
DevAddr bits	NwkID	NwkAddr

它的高7位是NwkId，用来区别同一区域内的不同网络，另外也保证防止节点窜到别的网络去。它的低25位是NwkAddr，是终端的网络地址，可以由网络管理者来分配。

- 6.1.2 应用ID(AppEUI)

AppEUI是一个类似IEEE EUI64的全球唯一ID，标识终端的应用提供者。 APPEUI在激活流程开始前就存储在终端中。

- 6.1.3 网络会话密钥(NwkSKey)

NwkSKey被终端和网络服务器用来计算和校验所有消息的MIC，以保证数据完整性。也用来对单独MAC的数据消息载荷进行加解密。

- 6.1.4 应用会话密钥(AppSKey)

AppSKey被终端和网络服务器用来对应用层消息进行加解密。当应用层消息载荷有MIC时，也可以用来计算和校验该应用层MIC。

6.2 空中激活 OTAA

针对空中激活，终端必须按照加网流程来和网络服务器进行数据交互。如果终端丢失会话消息，则每次必须重新进行一次加网流程。 加网流程需要终端准备好如下这三个参数：DevEUI，AppEUI，AppKey。

APPEUI在上面的6.1.2已经做了描述。

> 注意：对于空中激活，终端不会初始化任何网络密钥。只有当终端加入网络后，才会被分配一个网络会话密钥，用来加密和校验网络层的传输。通过这样，使得终端在不同网络间的漫游处理变得方便。同时使用网络和应用会话密钥，使得网络服务器中的应用数据，不会被网络提供者读取或者篡改。

- 6.2.1 终端 ID (DevEUI)
  DevEUI 是一个类似IEEE EUI64的全球唯一ID，标识唯一的终端设备。
- 6.2.2 应用密钥(AppKey)

AppKey 是由应用程序拥有者分配给终端，很可能是由应用程序指定的根密钥来衍生的，并且受提供者控制。当终端通过空中激活方式加入网络，AppKey用来产生会话密钥NwkSKey和AppSKey，会话密钥分别用来加密和校验网络层和应用层数据。

- 6.2.3 加网流程

从终端角度看，加网流程是由和服务器的两个MAC命令交互组成的，分别是 join request 和 join accept。

- 6.2.4 Join-request 消息

加网流程总是由终端发送 join-request 来发起。

Size (bytes)	8	8	2
Join Request	AppEUI	DevEUI	DevNonce

join-request 消息包含了AppEUI 和 DevEUI ，后面还跟了2个字节的声明 DevNonce。

DevNonce 是一个随机值。网络服务器为每个终端记录过去的 DevNonce 数值，如果相同设备发出相同的 DevNonce 的join request就会忽略。

join-request 消息的MIC数值(见第4章 MAC帧格式)按照如下公式计算：

> cmac = aes128_cmac(AppKey, MHDR | AppEUI | DevEUI | DevNonce) MIC = cmac[0..3]

join-request 消息不用加密。

- 6.2.5 Join-accept 消息

如果网络服务器准许终端加入网络，就会用 **join-accept** 对 **join-request** 进行应答。**join-accept** 是作为一个普通下行帧进行下发的，唯一的区别是它使用的是 JOIN_ACCEPT_DELAY1 或者 JOIN_ACCEPT_DELAY2 (分别代替 RECEIVE_DELAY1 和 RECEIVE_DELAY2)但是它所使用的两个接收窗口的信道频率和数据率和 LoRaWAN 地区参数文件[PARAMS]"接收窗口"部分所描述的 RX1 和 RX2 接收窗口相同。

如果 **join-request** 不被接受，则终端不会收到回应。

join-accept 消息的帧格式包括3字节的应用随机数(**AppNonce**)，网络标识符(**NetID**)，终端地址(**DevAddr**)，TX和RX之间的延时(**RxDelay**)，用于终端所加入的网络的可选信道频率列表(**CFList**)。**CFList** 的选择是由区域指定的，在 LoRaWAN 地区参数文件[PARAMS]中进行定义。

Size (bytes)	3	3	4	1	1	(16)Optional
Join Accpet	AppNonce	NetID	DevAddr	DLSettings	RXDelay	CFList

AppNonce是由网络服务器所提供的一个随机值或者某些形式的唯一ID，用于终端得到两个会话密钥**NwkSKey**和**AppSKey**，如下:

```
NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad 16 )
AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad 16 )
```

join-accept的 MIC 值由如下计算得到:

```
cmac = aes128_cmac(AppKey,MHDR | AppNonce | NetID | DevAddr | DLSettings | RxDelay | CFList)
MIC = cmac[0..3]
```

join-accept消息是使用**AppKey**进行加密的，如下:

```
aes128_decrypt(AppKey, AppNonce | NetID | DevAddr | DLSettings | RxDelay | CFList | MIC)
```

> 注意:网络服务器在 ECB 模式下使用一个 AES 解密操作去对 **join-accept** 消息进行加密，因此终端就可以使用一个 AES 加密操作去对消息进行解密。这样终端只需要去实现 AES 加密而不是 AES 解密。
>
> 注意:建立这两个会话密钥使得 网络服务器 中的网络运营商无法窃听应用层数据。在这样的设置中，应用提供商必须支持网络运营商处理终端的加网以及为终端生成 NwkSkey。同时应用提供商向网络运营商承诺，它将承担终端所产生的任何流量费用并且保持用于保护应用数据的AppSKey的完全控制权。

NetID的格式如下所述:**NetID**的7个最低有效位称为**NwkID**并且和之前所描述的终端的短地址的7个最高有效位相对应。保留的17个最高有效位可以由网络运营商进行自由选择。

DLsettings字段包含了下行配置:

Bits	7	6:4	3:0
DLsettings	RFU	RX1DRoffset	RX2 Data rate

RX1DRoffset位域设置上行数据速率和RX1下行数据速率的偏移量。默认情况下偏移量为0（意思就是上行数据速率与下行数据速率相等)。偏移量用于考虑一些地区的基站最大功率密度限制和平衡上下行射频链路预算。

上行和下行的数据率之间的实际关系是由区域指定的，在LoRaWAN地区参数文件[PARAM]中进行定义。

延时**RxDelay**和**RXTimingSetupReq**里的**Delay**字段有着相同的约定。

6.3 独立激活 ABP

在某些情况下，终端可以独立激活。独立激活是让终端绕过 join request - join accept的加网流程，直接加入到指定网络中。

独立激活终端，意味着 DevAddr 和两个会话密钥 NwkSKey 和 AppSKey 直接存储在终端中，而不是DevEUI，AppEUI，AppKey。终端在一开始就配置好了入网必要的信息。

每个终端必须要有唯一的 NwkSKey 和 AppSKey。这样，一个设备的密钥被破解也不会造成其他设备的安全性危险。创建那些密钥的过程中，密钥不允许通过公开可用信息获得(例如节点地址)。

第7章 重传退避

上行帧如下面这样：

- 需要网络或者应用服务器进行确认或者应答时，如果超时没有接收到，终端需要进行重传。

同时，

- 某些外部事件（断电，无线电干扰，网络断电，地震），将会导致大量的（>100）的设备出现同一时间上行的行为。

这样可能会引发灾难性的、持久的、射频网络过载的情况。

> 注意：这种上行帧的一个典型例子是 JoinRequest，当发生网络中断时，一组终端将会复位 MAC 层。
>
> 这一组的终端将会开始广播 JoinRequest 上行帧，只有当从网络中接收到 JoinResponse 命令时才会停止。

对于这些数据帧的重传，RX2时隙的末端和下一个上行帧的重传间隔应该是随机的，并且每个终端遵循不同的顺序（例如使用设备地址作为伪随机生成器的种子）。这些消息的发射占空比，要根据当地的参数要求，以及如下的限定，取二者中更严格的一个限定。

在上电或者复位后的第1个小时	T0 < t < T0+1h	发射时间 < 36秒
在接下来的10小时	T0+1 < t < T0+11h	发射时间 < 36秒
前11小时后的24小时	T0+11+N < t < T0+35+N，N>=0	发射时间 < 8.7秒/24小时

CLASS B – BEACON

Class B在当前协议版本中还仅作实验性参考。

第8章 Class B 介绍

这章描述了LoRaWAN Class B层，这是为电池节点优化设计的，不管节点是移动的还是固定在某个位置。

Class B 的终端必须执行如下操作，为了获得服务端发起的下行消息，终端必须按要求开启一个固定时间间隔的接收窗口。

LoRaWAN Class B 就是在终端上增加一个经过同步的接收窗口。

LoRaWAN Class A 的限制之一就是终端发送数据使用的Aloha算法；这使得客户应用程序或者服务端不能在确定时间内联系上终端。Class B 的目的就是在Class A 终端随机上行后的接收窗口之外，让终端也能在可预见的时间内开启接收。Class B 是让网关周期发送信标来同步网络中的所有终端，以便终端能够在周期时隙的确定时间点打开一个短的接收窗口(叫做“ping slot”)。

> 注意：是否要从Class A 切换到 Class B，这个要在终端的应用层进行处理。如果打算从网络端将Class A 切换到 Class B，客户程序只能利用终端 Class A的上行包来反馈一个下行包给节点，需要应用层上处理来识别这个请求 - 这个处理不在LoRaWAN层面。

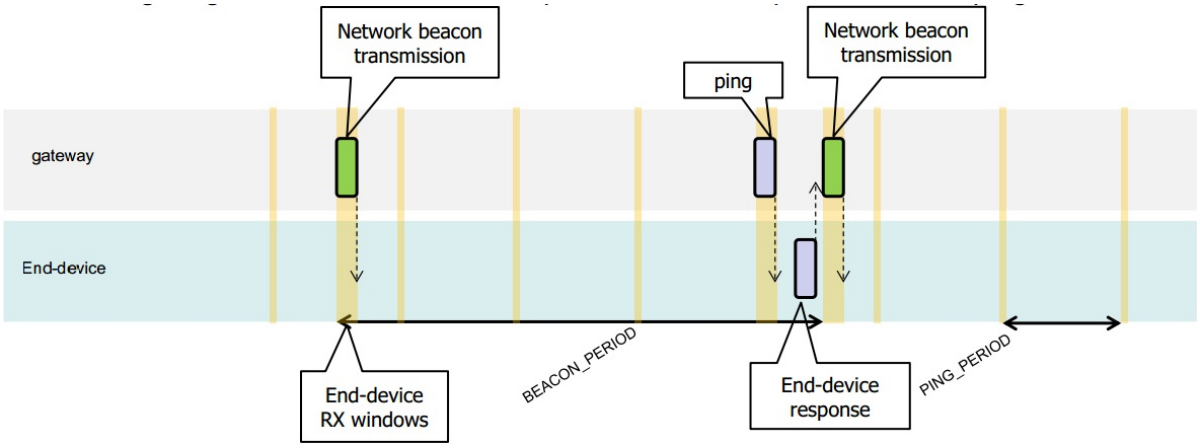
第9章 下行同步网络的原理

对于一个支持ClassB的网络，所有网关必须同步广播一个信标，以给所有终端提供一个参考时间。基于这个时间参考，终端可以周期性地打开接收窗口，下文称之为“ping slot”，这个“ping slot”被网络建设者用于发起下行通信。网络使用ping slots其中之一来发起下行通讯的行为，称之为“ping”。用来发起下行通讯的网关，是network server根据终端最近一次上行包的信号传输质量来选择的。基于此，如果终端根据广播的信标帧发现网络发生了切换(通信的网关发生了变化)，它必须发出上行帧给network server，以使server端更新下行路径的数据库。

所有终端启动后，以Class A来加入网络。之后终端应用层可以切换到Class B。通过以下步骤来实现：

- 终端应用层请求LoRaWAN层切换到Class B模式。终端的LoRaWAN层搜索信标帧，如果搜索到并且锁定了信标帧，那么就向应用层返回BEACON_LOCKED的服务原语，反之则返回BEACON_NOT_FOUND的服务原语。为了促进信标帧的搜索，LoRaWAN层可以使用稍后介绍的 “BeaconTimingReq” 消息。
- 基于信标的强度和电池寿命，终端的应用层选择ping slot所需的数据速率和周期，这可以从LoRaWAN层获取到。
- 一旦处于Class B模式，MAC层需要在所有上行帧的FCTRL字段中，将Class B的位域置为1。这个位用来通知server，设备已经切换到Class B模式。MAC层会给每个beacon和ping时隙安排接收时隙。当成功接收信标，终端的LoRaWAN层将会转发beacon内容给应用层，同时携带测量的射频信号强度。终端的LoRaWAN层在安排beacon和ping时隙时，需要考虑可能的最大时钟偏移。当在ping时隙成功解调出下行帧，它的处理和Class A 的方式一样。
- 移动的终端，必须周期性地通知network server其位置信息，以便确定下行路径。这是通过发送普通的(可能是空包)“unconfirmed”或者“confirmed”上行包来实现。终端的LoRaWAN层需要将Class B的位域置为1。如果应用程序通过解析beacon内容来判断节点移动，那将会使得这个事情变得更高效。这种情况下终端需要在beacon接收后随机延时一段时间(具体见章节15.5)再上行，避免上行帧冲突。
- 如果在指定周期内没有接收到beacon(具体见章节12.2)，则意味着网络同步丢失。MAC层必须通知应用层切换回Class A。随后终端在上行帧的LoRaWAN层中将不再设置Class B的位域，用以通知network server终端不再处于 Class B 模式。终端的应用程序可以周期性地尝试切换回 Class B。在做这个处理时要先探寻下beacon。

下面这张图展示了beacon接收时隙和ping时隙。



在这个示例中，指定beacon周期是128秒，ping接收时隙的周期是32秒。大部分时候server并没有使用ping时隙，因此终端可以在接入信道时监听下是否有前导码，如果没有则立即关闭接收窗口。如果监测到前导码，则射频会持续接收，直到下行帧解调完毕。MAC层随后处理数据帧，检查确认地址域匹配和MIC校验有效之后再转发给应用层。

第10章 Class B 模式的上行帧

Class B 模式的上行帧和 Class A 的基本一样，除了帧头Fctrl字段的RFU位域有所不同。在 Class A 上行帧中这个位没有使用(RFU)，而在 Class B 中有使用。

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

上行帧中的 Class B 位域置为1，用于通知network server设备已切换到 Class B 模式，准备好接收下行ping包。

下行帧的FPending位域的定义是不变的，仍然和Class A的定义一样，表示server有多个下行帧要下发，设备应当继续接收。

第11章 Class B 模式的下行帧(Class B选项)

11.1 物理层帧格式

下行 Ping 帧使用和 Class A 下行帧相同的格式，但可能会采用不同的信道频率规划。

11.2 单播和多播 MAC消息

信息的传播方式可以是“单播”或者“多播”。单播是指将信息传递给一个指定的终端，多播是指将信息传递给多个终端。多播组内的所有终端都必须共享一个相同的多播地址和相关的加密密钥。LoRaWAN Class B 协议中并没有明确规定如何去建立这样的多播组，以及如何安全地分配多播密钥。这必须通过 节点个性化设置 或者 应用层 来实现。

11.2.1 单播 MAC 消息格式

单播下行 Ping 帧的 MAC 载荷格式和 Class A 的定义一样。终端的处理也采用相同的方式。同时也采用相同的帧计数，在收到 Class B ping 时隙或者 Class A 应答时隙时都进行递增处理。

11.2.2 多播 MAC 消息格式

多播帧和单播帧大部分都一样，仅有一些区别：

- 不允许携带 MAC 命令，既不能在 FOpt 字段里，也不能 port 0 时的载荷里携带，因为多播下行不像单播帧那样具备认证鲁棒性。
- **ACK** 和 **ADRACKReq** 位必须为 0。**MType** 字段必须为 “Unconfirmed Data Down”。
- FPending 位表示还有多播数据要传输。如果设置了这个位，将会在下个多播接收时隙里传输数据帧。如果没设置这个位，则不确定下个多播接收时隙是否会传输数据。这个位可以让终端来评估正在冲突的接收时隙的优先级。

第12章 信标的获得和追踪

在从 Class A 切换到 Class B 之前，终端必须首先接收一个网络的信标来将它自身的时间基准与网络时间进行校准。

一旦处于 Class B 模式，终端必须定期地去搜索并且接收一个网络信标，以消除自身内部基准时间相对于网络时间的任何漂移。

Class B 模式的设备也许会短暂性地无法接收信标(超出与网关的通信范围，存在干扰，...）在这种情况之下，终端必须考虑它内部时钟可能产生的漂移，逐步地加大信标和ping时隙的接收窗口时间。

> 例如，一个设备精度为 10ppm 的内部时钟每个信标周期(128s)就会有+/-1.3ms的漂移。

12.1 最小 beacon-less 操作时间

在信标丢失的情况之下，一个终端需要在接收到最后一个信标节点时间开始算起保持2小时的 Class B 操作。这种短暂的没有信标的 Class B 操作就称之为 “beacon-less” 操作。这种情况之下就需要依赖终端自己的时钟来保持时间。

在 “beacon-less” 的情况下，单播、多播还有信标接收时隙都必须逐步地扩大接收窗口时间以容纳终端可能的时钟漂移。

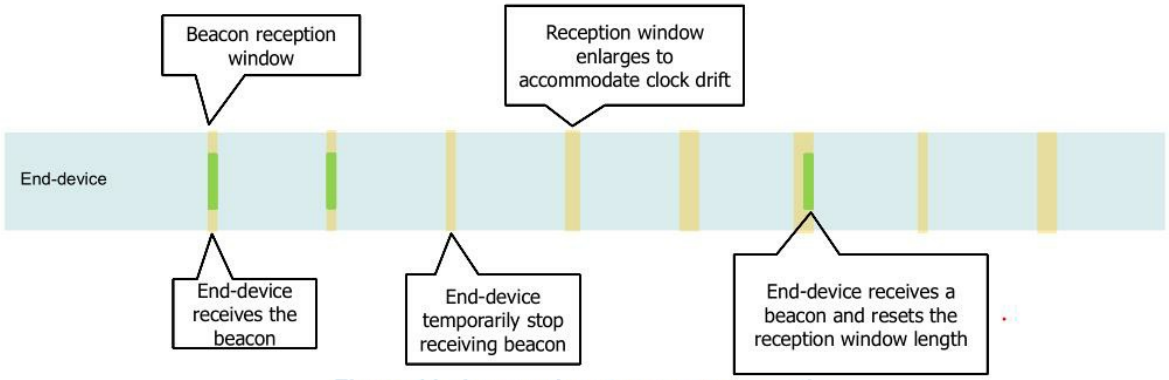


Figure 11 : beacon-less temporary operation

12.2 beacon-less 接收操作的延长

在120分钟的时间间隔之内，一旦终端接收到任何信标，就需要将 Class B 的 “beacon-less” 操作进一步延长120分钟，使得终端可以校正时序漂移以及重置接收时隙持续时间。

12.3 减少时序漂移

终端可以使用信标的准确周期(当信标可用时)去校准他们的初始化时钟，这样可以减少初始化时钟频率的不准确性。由于定时振荡器会表示出可预知的温度漂移，因此使用温度传感器可以尽可能地减小时序漂移。

第13章 Class B下行时隙时序

13.1 定义

为了使 Class B模式能够正常运行，终端必须以信标规定的精准时刻打开接收时隙窗口。这章节定义了所需的时序操作。

两个连续的信标起始点之间的间隔称为信标周期。信标帧的传输以 **BEACON_RESERVED** 时间间隔的起始端对齐。每个信标都有一个保护时间间隔，在该时间间隔之内是没有 ping 时隙的。保护间隔的长度对应于允许帧在空中的最长时间。这样就能保证在保护时间之前的一个 ping 时隙内发起的下行数据帧总是有时间去完成传输而不与信标的传输发生冲突。因此用于ping时隙的时间间隔是从 **BEACON_RESERVED** 时间间隔的末尾节点到下一 **BEACON_GUARD** 时间间隔的起始节点。

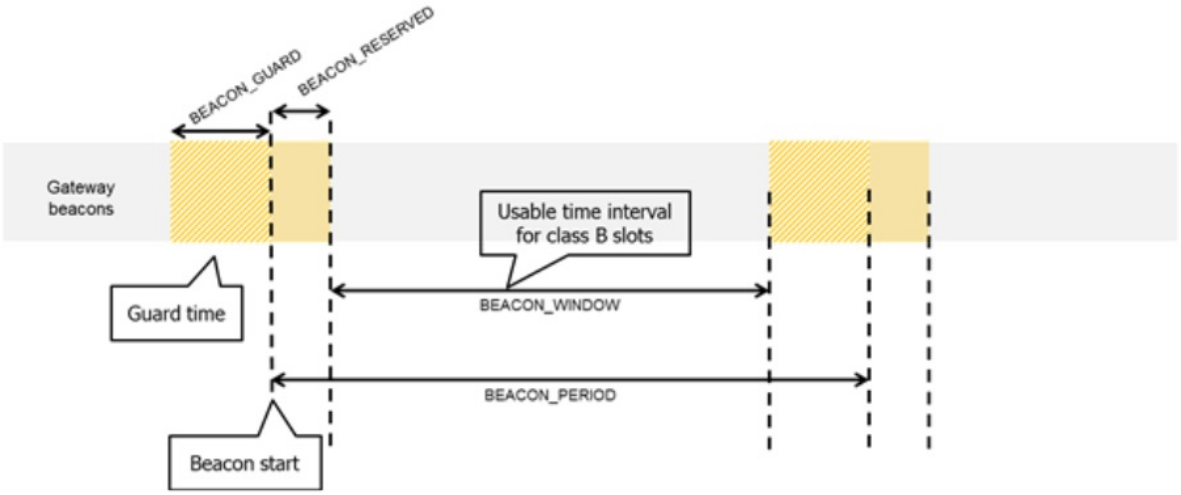


图 12 ：信标时序

Beacon_period	128 s
Beacon_reserved	2.120 s
Beacon_guard	3.000 s
Beacon-window	122.880 s

表 12 ：信标时序

信标帧在空中的时间实际上是远小于 **BEACON_RESERVED** 时间间隔的，目的是将来用于添加网络管理广播帧。

BEACON_WINDOW 时间间隔被划分为2^12=4096个小时段，每一段时长为 30ms，所有时段的编号从0~4095。

每个使用时隙号**N**的终端必须在 **Beacon start** 开始之后的 **Ton** 秒打开它的接收窗口，**Ton**的计算公式如下:

```
Ton = beacon_reserved + N * 30ms
```

N 称为时隙编号。

最后一个 ping 时隙时段(编号为4095)的开始时间是在 beacon start 后的 beacon_reserved + 4095 * 30 ms = 124970ms 或者下一个信标开始前的 3030ms。

13.2 时隙随机化

为了避免系统冲突或者过载问题，所以时隙编号是随机的并且在每个信标周期都会改变。

使用以下参数:

DevAddr	设备32位网络单播或者多播地址
pingNb	每个信标周期的ping时隙数量。必须为2的整数幂：pingNb = 2^k,1<=k<=7 <="" td="">
pingPeriod	设备唤醒接收所间隔的时隙周期，其单位是时隙数量：pingPeriod = 2^12 / pingNb
pingOffset	在每个信标周期开始计算的随机偏移。值的范围为0到(pingPeriod-1)
beaconTime	这个时间将会在**BCNPayload**中携带。前一个信标帧的时间

slotLen	一个单元ping时隙长度=30ms(就是前面所说的将ping时隙划分的时间段长度，4096段)

在每个信标周期终端和服务器都会计算出一个新的伪随机偏移将接收时隙对齐。使用全零的固定密钥的AES加密去进行随机化：

```
Key = 16 x 0x00
Rand = aes128_encrypt(Key,beaconTime | DevAddr | pad16)
pingOffset = (Rand[0] + Rand[1]x256) modulo pingPeriod
```

信标周期所使用的时隙是:

```
pingOffset + N x pingPeriod with N = [0:pingNb-1]
```

因此节点打开接收时隙的时间是:

First slot	Beacon_reserved + pingOffset x slotLen
Slot 2	Beacon_reserved + (pingOffset + pingPeriod) x slotLen
Slot 3	Beacon_reserved + (pingOffset + 2 x pingPeriod) x slotLen
...	...

如果一个终端同时服务于单播和一个或多个多播时隙，则该计算将会在一个新的信标周期开始时执行多次。一次用于单播地址(节点网络地址)，一次用于每个多播组地址。

当一个多播ping时隙和一个单播ping时隙发生了冲突并且终端接收窗口无法进行处理的情况之下，终端应该优先监听多播时隙的数据。如果多播接收时隙之间发生了冲突，则前一个多播帧的**FPending**位就可以用于设置优先级处理。

随机机制可以避免单播和多播时隙的系统冲突。如果在一个信标周期内发生了冲突，则下一个信标周期就不大可能发生。

第14章 Class B Mac命令

所有在 Class A 协议中描述的命令都应该在 Class B 中实现。Class B 协议还额外添加了如下的 **MAC** 命令。

CID	Command	由谁传输		描述
		终端	网关	
0x10	**PingSlotInfoReq**	x		终端设备用于将 ping 单播时隙数据速率和周期性传送给网络服务器
0x10	**PingSlotInfoAns**		x	用于网络应答**PingInfoSlotReq**命令
0x11	**PingSlotChannelReq**		x	用于网络服务器设置一个终端的单播 ping 通道
0x11	**PingSlotFreqAns**	x		终端用于应答**PingSlotChannelReq**命令
0x12	**BeaconTimingReq**	x		用于终端向网络请求下一个信标时间和信道
0x12	**BeaconTimingAns**		x	用于网络应答**BeaconTimingReq**命令
0x13	**BeaconFreqReq**		x	用于网络服务器修改终端希望接收信标广播的频率
0x13	**BeaconFreqAns**	x		用于终端应答**BeaconFreqReq**命令

14.1 PingSlotInfoReq

终端可以使用**PingSlotInfoReq**命令来告知服务器它的单播 ping 时隙周期以及期望的数据速率。这个命令只能用于告知服务器单播 ping 时隙的参数。多播 ping 时隙完全由应用程序进行定义而不应该使用此命令设置。

Size(bytes)	1
PingSlotInfoReq Payload	Periodicity & data rate

Bit#	7	[6:4]	[3:0]
Periodicity & data rate	RFU	Periodicity	Data rate

Periodicity字段是一个无符号3位整数，用于终端当前使用的 ping 时隙周期的编码，编码公式如下:

```
pingSlotPeriod = 2^Periodicity(单位是s)
```

- Periodicity = 0 表示终端每1s打开一个 ping 时隙。
- Periodicity = 7 表示终端每128s打开一个 ping 时隙，这是 LoRaWAN Class B 协议中所支持的最大 ping 时隙周期。

Data rate 字段表示终端期望收到 ping 时隙的数据率。它使用的编码方式与 Class A 协议中所描述的**LinkAdrReq**命令相同。

服务器需要知道终端的 ping 时隙周期或者期望的数据率，否则 Class B 的下行将不会成功。因此终端在 **PingSlotInfoReq** 命令发出之后必须收到 **PingSlotInfoAns** 命令的回复才能从 Class A 切换到 Class B。当终端需要改变 ping 时隙周期以及数据率时，需要先恢复到 Class A 模式，在发送 **PingSlotInfoReq** 命令并且收到服务器端的 **PinSlotInfoAns** 命令回复之后，就可以使用新的参数切换回 Class B模式。**PingSlotInfoReq** 命令可以和 **FHDRFOpt** 字段里的任何 MAC 命令进行连接，如 Class A 协议中的帧格式所述。

14.2 BeaconFreqReq

该命令由服务器发往终端，用于修改终端期望信标的频率。

Octets	3
PingSlotChannelReq Payload	Frequency

Frequency字段和Class A协议中定义的 **NewChannelReq** MAC命令有着相同的编码方式。

Frequency是24位的无符号整数。实际的信标信道频率是100 x Frequency,单位Hz。信标的信道以 100Hz 为基本单位，变化范围在 100MHz 到 1.67GHz 之间。终端必须检查该频率是不是射频硬件所允许的范围，不是则需要返回错误。

一个有效的非零频率将会强制终端以一个固定的频率信道去监听信标，即使默认是指定跳频信标(即美国ISM频段)。

若频率为0则表示终端使用“信标物理层”部分所定义的默认信标频率计划，在适用的情况之下恢复成跳频信标搜索。

14.3 PingSlotChannnelReq

该命令由服务器发往终端，用于修改终端期望下行 ping 的频率。

Octets	3	1
PingSlotChannelReq Payload	Frequency	DrRange

Frequency 字段和Class A协议中定义的 **NewChannelReq** MAC 命令有着相同的编码方式。

Frequency是24位的无符号整数。实际的信标信道频率是100 x Frequency,单位Hz。信标的信道以100Hz为基本单位，变化范围在100MHz到1.67GHz之间。终端必须检查该频率是不是射频硬件所允许的范围，不是则需要返回错误。

若频率为0则表示终端使用默认信标频率计划。

DrRange是信道允许的数据率范围。这个字节被等分为两个4位。

Bits	[7:4]	[3:0
DrRange	Max data rate	Min data rate

按照LoRaWAN地区参数文件[PARAMS]的定义，“Min data rate”字段指定了信道允许的最低数据率。例如0在欧盟物理层中指定DR0/125 kHz。类似的，“Max data rate”指定了最高数据率。例如在欧盟规范中，DrRange = 0x77 意味着在信道上只允许50 kpbs GFSK，DrRange = 0x50意味着支持 DR0/125kHz 到 DR5/125 kHz的数据率。

将终端接收到以上的命令之后，需要以**PingSlotFreqAns**命令进行回复。这个MAC命令的载荷包含了以下的信息:

Size(bytes)	1
pingSlotChannelAns Payload	Status

Status的位段有以下含义:

Bits	[7:2]	1	0
Statusd	RFU	Data rate range ok	Channel frequency ok

	Bit = 0	**Bit = 1**
Data rae range ok	设置的数据率超过了该终端当前定义的范围，保持之前的数据率范围	数据率范围与终端兼容
Channel frequency ok	终端无法使用该频率，保持之前的频率	终端可以使用这个频率

14.4 BeaconTimingReq

终端使用该命令来请求下一个信标的时间以及信道，该MAC命令没有载荷。**BeaconTimingReq & BeaconTimingAns**机制仅仅用于加快刚开始的信标搜索以降低终端的能量需求。

网络在给定的时间周期内可能只应答有限数量的请求。终端不能期望在发出**BeaconTimingReq**命令之后立刻收到**BeaconTimingAns**命令的应答。想要切换到Class B模式的处于Class A模式的终端一小时之内不应该发送超过一个**BeaconTimingReq**命令。

需要快速锁定信标的终端必须实现自主信标查找算法。

14.5 BeaconTimingAns

网络用该命令来应答**BeaconTimingReq**命令的请求。

Size(bytes)	2	1
BeaconInfoReqPayload	Delay	Channel

"Delay"字段是一个16位的无符号整数。如果当前下行帧的结束与下一个信标帧的开始之间的剩余时间记为 RTime:

```
30ms x (Delay+1) > RTime >= 30ms x Delay
```

在信标交替使用多个信道的网络中，**“Channel”**字段是下一个信标广播所使用的信道编号。对于信标广播频率固定的网络来说，这个字段值为0。

第15章 信标(Class B选项)

15.1 信标物理层

所有网关除了可以为终端和网络服务器转发消息，还可以通过在可配置的固定时间间隔上发送信标(**BEACON_INTERVAL**)来参与提供一个时间同步机制。所有信标都以无线分组隐式模式进行发送，即没有 LoRa 物理帧头和 CRC 校验。

PHY	Preamble	BCNPayload

信标的 **Preamble** 开始于(长于默认)10个未调制符号。这允许终端实现低功耗占空比信标搜索。

信标的帧长度与无线电物理层紧密耦合。因此实际的帧长度可能从一个区域实现变为另一个区域实现。更改字段在下面的部分以粗体显示。

15.1.1 欧盟 863-870MHz ISM 频段

信标使用下面的设置进行传送:

DR	3	对应于125kHz带宽的SF9扩频因子
CR	1	编码率=4/5
frequency	869.525MHz	这是推荐的允许+27 dBm EIRP的频率。

只要符合ETSI的要求，网络运营商也可以使用一个不同的频率。

信标帧的内容如下:

Size(bytes)	3	4	1	7	2
BCNPayload	NetID	Time	**CRC**	GwSpecific	**CRC**

15.1.2 美国 902-928 MHz ISM 频段

信标使用下面的设置进行传送:

DR	10	对应于500kHz带宽的SF10扩频因子
CR	1	编码率=4/5
frequency	923.3到927.5MHz(以600kHz为单位)	信标的传送与Class A规范中定义的下行链路所使用的信道相同。

用于给定的信标所使用的下行链路信道是:

```
Channel = [floor(beacon_time/beacon_preiod)] modulo 8
```

- beacon_time是信标帧“Time”4字节字段的整数值。
- beacon_period是信标帧的周期，128s
- floor(x)意思是四舍五入到临近x的整数。

> 例子:第一个信标在923.3MHz上进行传送，第二次在932.9MHz，第九次再一次在923.3MHz进行传送。

Beacon channel nb	Frequency[MHz]
0	923.3
1	923.9
2	924.5
3	925.1
4	925.7
5	926.3
6	926.9
7	927.5

信标帧的内容如下:

Size(bytes)	3	4	2	7	1	2
BCNPayload	NetID	Time	**CRC**	GwSpecific	**RFU**	**CRC**

15.2 信标帧内容

信标帧的**BCNPayload**载荷由一个网络的公共部分和一个网关的特定部分组成。

Size(bytes)	3	4	1/2	7	0/1	2
BCNPayload	NetID	Time	CRC	GwSpecific	RFU	CRC

网络的公共部分包含了一个网络的标识符**NetID**，用于唯一标识发送信标的网络还有一个时间戳**Time**(单位为s)，这个时间戳是从1970年1月1日的Coordinated Universal Time(UTC) 00:00:00开始计时的。信标网络的公共部分的完整性由8位或者16位的 CRC 校验码来进行保护，是8位还是16位取决于PHY层参数。CRC-16是在IEEE 802.15.4-2003 7.2.1.8部分所定义的NetID+Time字段上进行计算。当需要8位CRC时计算CRC-16的低8位就会被使用。

例如：这是一个有效的 EU868 信标帧:

```
AA BB CC| 00 00 02 CC | 7E | 00 | 01 20 00 | 00 81 03 | DE 55
```

字节是从左向右进行传送。相对应字段的值是：

Field	NetID	Time	**CRC**	InfoDesc	lat	long	**CRC**
Value Hex	CCBBAA	CC020000	7E	0	002001	038100	55DE

NetID+Time字段的CRC-16校验码是0xC87E，但是在这种情况之下只使用低8位。

NetID的7个最低有效位被称之为**NwkID** ，与终端短地址的7位最高有效位相匹配。相邻的或者重叠的网络必须有不同的**NwkID**。

网络的特定部分提供网关发送一个信标的额外信息，因此对于每个网关可能不同。当RFU字段适用时(区域特定)应该等于0。可选择的部分由GwSpecific+RFU字段计算出的CRC-16校验码进行保护。CRC-16的定义与强制部分相同。

例如：这是一个有效的 美国900 信标

Field	NetID	Time	**CRC**	InfoDesc	lat	long	RFU	**CRC**
Value Hex	CCBBAA	CC020000	C87E	0	002001	038100	00	D450

在空中，字节以以下顺序进行发送：

```
AA BB CC| 00 00 02 CC | 7E C8 | 00 | 01 20 00 | 00 81 03 | 00 | 50 D4
```

监听和同步网络的公共部分足以在Class B模式去操作一个固定的终端。一个移动的终端也应该解调出信标的网关特定部分，以便信标在从一个网络移动到另一个网络时可以通知网络服务器。

> 注意：如前所述，所有的网关在同一个时间点(即时间同步)发送他们的信标，因此对于网络公共部分来说，即使一个终端同时从多个网关接收信标，监听的终端也不存在明显的空中冲突。至于网关的特定部分，当冲突发生时，位于多个网关附近的一个终端仍然有能力以高概率去解码最强的信标。

15.3 信标GwSpecific字段格式

GwSpecific字段的内容如下所述:

Size(bytes)	1	6
GwSpecific	InfoDesc	Info

InfoDesc描述符描述了如何解释**Info**字段信息。

InfoDesc	**Meaning**
0	网关第一天线的GPS坐标
1	网关第二天线的GPS坐标
2	网关第三天线的GPS坐标
3:127	RFU

128:255	为自定义网络特定广播预留

对于一个单一的全向天线网关，当广播GPS坐标时**InfoDesc**的值为0。例如，对于一个具有3扇区电线的站点，第一天线广播信标时**InfoDesc**的值为0，第二天线广播信标时**InfoDesc**的值为1，等等...

15.3.1 网关**GPS**坐标:**InfoDesc = 0**，**1**或者**2**

对于**InfoDesc**=0，1或者2，**Info**字段所包含的内容编码了天线广播信标的GPS坐标

Size(bytes)	3	3
Info	Lat	Lng

纬度和经度字段(分别对应于**Lat**和**Lng**)编码了网关的地理位置，如下:

- 南北纬度使用24位字来进行编码，-2^23对应于南90°(南极点)，2^23对应于北90°(北极点)。赤道对应于0。
- 东西经度使用24位字来进行编码，-2^23对应于西180°，2^23对应于东180°。格林尼治子午线对应于0。

15.4 信标精确定时

信标从Coordinated Universal Time(UTC)，1970年1月1日00:00:00 加上 NwkID 加 TBeaconDelay 开始，每128秒发送一次。因此信标是在 Coordinated Universal Time(UTC) 1970年1月1日00:00:00 之后 Bt = k*128 + NwkID + TBeaconDelay 的时间点进行发送。

其中 k 是最小的整数: k*128 + NwkID > T

其中 T = 从1970年1月1日的 Coordinated Universal Time(UTC) 00:00:00 以后的秒数。

> 注意: T 不是 Unix 时间。类似于 GPS 时间，不像 Unix 时间，T 是严格单调递增的并且不受闰秒的影响。
>
> **KevinCao**注:闰秒，是指为保持协调世界时接近于世界时时刻，由国际计量局统一规定在年底或年中（也可能在季末）对协调世界时增加或减少1秒的调整。由于地球自转的不均匀性和长期变慢性（主要由潮汐摩擦引起的），会使世界时（民用时）和原子时之间相差超过到±0.9秒时，就把协调世界时向前拨1秒（负闰秒，最后一分钟为59秒）或向后拨1秒（正闰秒，最后一分钟为61秒）； 闰秒一般加在公历年末或公历六月末。

其中**TBeaconDelay**是网络的特定延时，范围在0到50ms之间。**TBeaconDelay**在不同的网络之间可能不同，并且它意味着通信时允许网关有轻微的延时。**TBeaconDelay**对于给定的一个网络中的所有网关必须相同。**TBeaconDelay**必须小于50ms。所有终端的ping时隙使用信标传输时间作为定时基准。因此网络在调度Class B下行时需要将TBeaconDelay时间考虑在内。

15.5 网络下行路由更新要求

当网络使用Class B下行时隙去与终端进行通信时，当网络接收到最后一个上行数据帧之后，它会从最接近终端的一个网关进行下行数据发送。因此网络服务器需要追踪Class B终端的粗略位置。

只要一个Class B终端移动并且改变网络，它需要告知服务器以更新下行路由。可以通过发送“confirmed”类型或者“unconfirmed”类型的上行数据帧来完成更新，可能没有应用载荷。

终端可以在2个基础策略之间做出选择:

- 系统周期上行:最简单的方式，不需要对信标的“gateway specific”字段解调。只适用于缓慢移动的或者固定的终端。对于这些周期性上行链路没有要求。
- 网络改变的上行:终端对信标的“gateway specific”字段进行解调，检测到广播其解调的信标的网关的ID已经改变并且发送上行数据帧。在这种情况之下终端应当遵守信标解调和上行数据帧发送之间0~120s的伪随机延时。这用于确保当信标广播之后，同一个信标周期内进入或者离开网络的多个Class B设备的上行数据帧不会立即系统性地同时发生。

无法告知网络改变将会导致 Class B 的下行暂时性地无法运行。网络服务器可能必须等到下一个终端上行才能传输下行。

第16章 Class B单播/多播下行信道频率

16.1 欧盟 863-870MHz ISM 频段

所有的 Class B 的下行单播和多播都使用由 “**PingSLotChannelReq**” MAC 命令所定义的单频信道。默认的频率是 869.525MHz。

16.2 美国 902-928MHz ISM 频段

默认的，Class B的下行使用最后一个信标(鉴信标帧格式内容)的 **Time** 字段的信道函数和 **DevAddr**。

```
Class B downlink channel = [DevAddr + floor(Beacon_Time/Beacon_period)] modulo 8
```

- 其中 Beacon_Time 是当前信标周期的 32 位 **Time** 字段。
- Beacon_period 是信标周期的长度(协议中定义的是128s)
- Floor 指的是四舍五入到临近的较低整数值。
- DevAddr 是终端的32位网络地址。

因此 Class B 的下行在 ISM 频段的 8 个信道进行跳跃并且所有的 Class B 终端平等地使用 8 个下行信道进行传输。

如果带有一个有效的非零参数的 “**PingSlotChannelReq**” 命令被用于设置 Class B 下行频率，则随后所有的 ping 时隙都应该只使用这个频率而不是上一个信标频率。

如果发送参数为零的 “**PingSlotChannelReq**” 命令，则终端应该恢复成默认的频率计划，同上所述，Class B ping 时隙在8个信道之间进行跳跃。

其基本思想是允许网络运营商在可行的情况之下配置终端使用一个专用的频段用于 Class B 下行，并且当 ISM 频段可用时尽可能地保持频率多样性。

CLASS C – CONTINUOUSLY LISTENING

第17章 持续接收的终端

具备Class C 能力的终端，通常应用于供电充足的场景，因此不必精简接收时间。

Class C 的终端不能执行 Class B 。

Class C 终端会尽可能地使用 RX2 窗口来监听。按照 Class A 的规定，终端是在 RX1 无数据收发才进行 RX2 接收。为了满足这个规定，终端会在上行发送结束和 RX1 接收窗口开启之间，打开一个短暂的 RX2 窗口，一旦 RX1 接收窗口关闭，终端会立即切换到 RX2 接收状态； RX2 接收窗口会持续打开，除非终端需要发送其他消息。

> 注意：没有规定节点必须要告诉服务端它是 Class C 节点。这完全取决于服务端的应用程序，它们可以在 join 流程通过协议交互来获知是否是 Class C 节点。

17.1 Class C 的第二接收窗口持续时间

Class C 设备执行和 Class A 一样的两个接收窗口，但它们没有关闭 RX2 ，除非他们需要再次发送数据。因此它们几乎可以在任意时间用 RX2 来接收下行消息，包括MAC命令和ACK传输的下行消息。另外在发送结束和 RX1 开启之间还打开了一个短暂的RX2窗口。

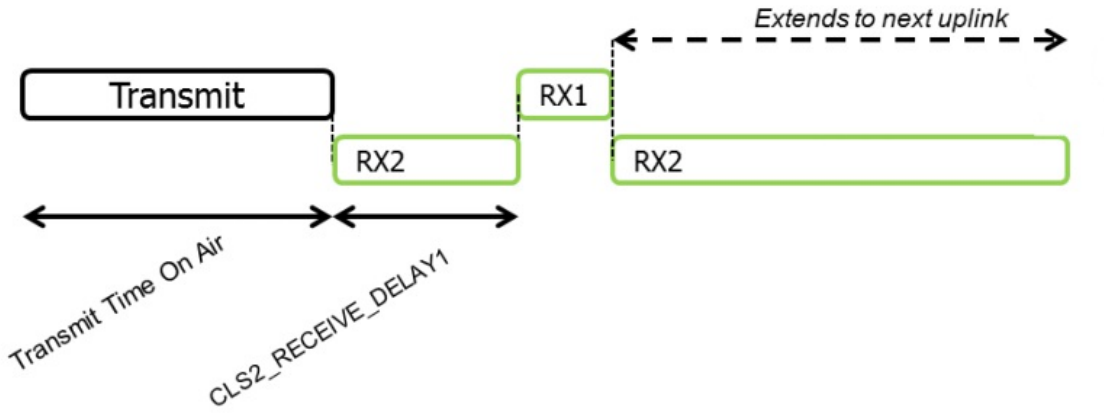


图13.Class C 终端的接收时隙时序图

17.2 Class C 对多播下行的处理

和 Class B 类似，Class C 设备也可以接收多播下行帧。多播地址和相关的 NWKSKEY 及 APPSKEY 都需要从应用层获取。Class C 多播下行帧也有相同的限制：

- 不允许携带MAC命令，既不能放在FOpts域中，也不能放在 port 0 的 payload 中，因为多播下行无法像单播帧那样具备相同的鲁棒性。
- ACK 和 ADRACKReq 位必须要为0。MType 域需要为 Unconfirmed Data Down 类型的数值。
- FPending 位表明有更多的多播数据要发送。考虑到 Classs C 设备在大部分时间处于接收状态，FPending位不触发终端的任何特殊行为。