LoRaWAN 规范 1.0 （2~4 章）　　云帆物联

2 Introduction on LoRaWAN options

2.1 LoRaWAN Classes

- 终端双向通讯（A 类）

  低功耗，先发送后接收，发送和接收交替进行。终端只有再发送数据后才能接收处理服务器发送来的数据，发送数据不受接收数据的限制。收发比

- 具有接收时隙的终端双向通讯（B 类）

  同样是先发送后接收，不同的是每次发送后按照一定时间间隔启动接收窗口，接收多条数据。时间间隔从网关获取，以便服务器知晓终端接收消息

- 最大接收时隙的终端双向通讯（C 类）

  打开接收窗口的时间间隔很小，几乎不间断的接收消息。比 A 和 B 更耗能，但和服务器交互的延迟低。

2.2 规范

高级类的附加功能向下兼容低级类。所有 LoRaWAN 终端必须实现 A 类的功能。

注意：

本规范手册中：物理消息格式、MAC 消息格式以及 A 类和其它高级类都具备的东西，只在本手册的 A 类部分介绍。

3 Physical Message Formats

LoRa 中区分上行和下行的术语。

3.1 上行链路消息

上行链路消息由终端发送经过一个或多个网关中转到达网络服务器。

下行链路消息：

Preamble	PHDR	PHDR_CRC	PHYPayload

3.3 接收窗口

设备终端每次发送数据（上行传输）后打开两个短接收窗口（short receive windows）。接收窗口的启动时间是配置好的时间周期，该时间在最近一条

4 MAC Message Formats

LoRa 所有的上下行链路消息都会包含 PHY 负载，该负载以单字节 MAC 头为开始，MAC 头后面是 MAC 负载，结尾是 4 字节的消息一致码（MIC）。

Radio PHY layer:

Figure 5: Radio PHY structure (CRC* is only available on uplink messages)

PHYPayload:

Figure 6: PHY payload structure

MACPayload:

Figure 7: MAC payload structure

FHDR:

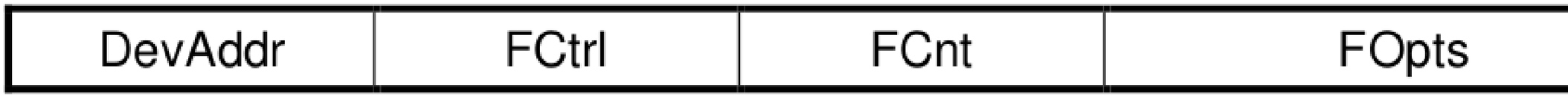

Figure 8: Frame header structure

4.2 MAC Header (MHDR field)

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

MAC 头中包含消息类型（MType）和帧编码所遵循的 LoRaWAN 规范的主版本号（Major）。RFU 是保留位。

- 4.2.1 Message type (MType bit field)

LoRaWAN 自定义了六个独特的 MAC 消息类型：join request, join accept, unconfirmed data up/down, 以及 confirmed data up/down

MType	Description	备注
000	Join Request	无线激活过程使用，具体见章节 6.2
001	Join Accept	无线激活过程使用，具体见章节 6.2
010	Unconfirmed Data Up	接受者不必回应
011	Unconfirmed Data Down	接受者不必回应
100	Confirmed Data Up	接受者必须回应
101	Confirmed Data Down	接受者必须回应
110	RFU	保留
111	Proprietary	用来实现自定义格式的消息，交互的设备之间必须有相同的处理逻辑，不能和标准消息互通

- 4.2.1.1 Join-request and join-accept messages

  已经添加上表的到备注

- 4.2.1.2 Data messages

  消息数据既能传输 MAC 命令又能传输应用数据，甚至可以一起发送。不同消息类型用不同的方法保证一致性，下面会介绍这一点。

- 4.2.2 Major version of data message (Major bit field)

- 4.3.1 Frame header (FHDR)

FHDR 由：终端短址（**DevAddr**）、一个帧控制字节（FCtrl）、2 字节的帧计数器（即帧大小，FCnt） 和 最多 15 个字节的配置（FOpts，用来传输 MAC 命

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

For downlink frames the FCtrl content of the frame header is:

Bit#	7	6	5	4
FCtrl bits	ADR	ADRACKReq	ACK	FPending

For uplink frames the FCtrl content of the frame header is:

Bit#	7	6	5	4
FCtrl bits	ADR	ADRACKReq	ACK	RFU

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

  LoRa 网络对终端的数据速率没有任何限制。LoRaWAN 协议通过该特性调整优化静态终端（相对移动终端来讲）的数据速率，即自适应数据速率（A
  来使用尽可能快的数据速率。
  移动终端在移动过程中会快速切换无线广播环境，该过程中进行数据速率管理没什么实际意义，此时移动终端应使用已经修正过的默认数据速率。
  设置 ADR 之后，网络通过 MAC 命令控制终端的数据速率。如果没有设置 ADR，网络会无视收到的信号的质量，不对终端的数据速率做任何调整。终
  就应该开启 ADR，这样可以延长终端的电池寿命并充分利用网络带宽。

  注意：

  哪怕移动终端在大多数时间下都是不移动的。因此终端可以根据它自己移动状态请求网络通过 ADR 进行数据速率优化。

- 4.3.1.2 消息确认位和确认**流程** (ACK in FCtrl)

  收到 confirmed 类型的消息时，接收者要回复一条确认消息（ACK，通过设置确认位实现）。如果发送者是终端，网络就把消息发送到该终端打开
  息的传输方式。
  确认消息只会在收到消息以后作为响应发送，并且不重发。

  注意：

  为了尽量简化终端处理、减少状态，一旦收到需要确认的消息要立刻发送确认消息，确认消息要简单直接（最好发空消息）。

- 4.3.1.3 重传机制（Retransmission procedure）

  如果终端设备发送一条需要确认的消息后没有收到响应，终端就会重新发送这条消息。不同设备间的消息重传的次数和每次的时间可能不同，当然
  注意：

  18 章给出了一些确认机制的时间

  注意：

  如果设备重传次数到限制后还没收到确认消息，就会降低自身的数据传输速率来增加连接距离再次尝试连接。这种条消息的重传或者放弃是

  注意：

  如果网络服务器重传次数达到限制后还没有收到确消息，在没有收到设备的消息之前会认为无法与终端建立连接（终端不可达）。这种消息

  注意：

  上面提到的重传期间的数据速率回归机制在 18.4 有详细介绍

- 4.3.1.4 帧挂起位 (FPending in FCtrl, downlink only)

  帧挂起位（FPending）只在下行交互中使用，表示网关还有数据挂起等待发送。此时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 18.3。

- 4.3.1.5 计数器 (FCnt)

注意（下面这段话译者一直没明白）：

> 由于 FCnt 字段只携带 32bits 的帧数中最不显著的 16bits，服务器必须通过观察流量（observation of the traffic）推断出帧号的 16 个
> 当计数器使用 32 位时，FCnt 字段只发送 32bits 中的 16 个最低有效位。此时服务器需要通过观察传输的数据来自己维护 16 个最高有效位。

- 4.3.1.6 帧配置 (FOptsLen in FCtrl, FOpts)

帧配置长度（FOptsLen）位于帧的 FCtrl 部分，表示 FOpts 的总长度。FOpts 搭载到数据帧中发送的 MAC 命令最长 15 字节，详细的 MAC 命令见 4
如果帧配置长度 FOptsLen=0，FOpts 为空；~~如果 FOptsLen 不是 0，端口必须有，也不能是 0（Fport=0 表示 MAC 命令放在 payload 中）。~~ 如果 F
，端口号要么省略，要么是一个非零值（具体看下面）。

MAC 命令不能同时出现在 payload（负载）和帧配置项中。

- 4.3.2 端口字段 Port field（FPort）

负载（payload）不为空的时候端口号（port）也不能是空。此时 port=0 表示 FRMPayload 中只有 MAC 命令（详情见章节 4.4）。FPort 值的可用
留值，方便以后扩展。

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

N 是应用负载的字节数，N 的取值范围见第 7 节

N ≤ M - 1 - (FHDR 的字节长度)

其中 M 是 MAC 的最大有效荷载长度。

- 4.3.3 MAC 帧负载据加密（FRMPayload）

如果帧数据中包含 payload，要先对 FRMPayload 进行加密，再计算消息的一致码（MIC）。

加密方案使用基于 IEEE 802.15.4/2006 Annex B [IEEE802154] 的 AES 加密，秘钥长度 128 位。默认情况下，所有 FPort 的加/解密都在 LoRaWAN
上完成。

字节数	1	4		1	4		4			1	1
Ai	0x01	4	×	0x00	Dir	DevAddr	FCntUp	或	FCntDown	0x00	i

Dir 字段：上行帧为 0，下行帧为 1
对 Ai 加密，得到 Si：

$$S_i = \text{aes128\_encrypt}(K, A_i) \text{ for } i = 1..k$$
$$S = S_1 \mid S_2 \mid .. \mid S_k$$

通过分割对 payload 进行加解密：

- 4.3.3.2 Encryption above the LoRaWAN layer

  对于选定的端口（FPort 不能是 0，因为 0 表示 MAC 命令），如果 LoRaWAN 前面的一些层给 LoRaWAN 的 FRMPayload 是加密过的，LoRaWAN 就把 FRM
  何改动。

- 4.4 Message Integrity Code (MIC)

  对整个消息进行 MIC 计算（AES 签名算法 CMAC），需要加密的消息包括以下字段：

  msg = MHDR | FHDR | FPort | FRMPayload

  len(msg) 表示消息的字节长度。
  MIC 算法参考 RFC4493：

$$cmac = \text{aes128\_cmac}(\mathbf{NwkSKey}, B_0 \mid msg)$$
$$\mathbf{MIC} = cmac[0..3]$$

  B0 定义如下：

放在 FOpts 的命令不加密（原因：加密 Payload，对整个数据签名），也不能超过 15 个字节（2^4 - 1）。
放在 FRMPayload 的 MAC 命令长度不能超过 FRMPayload 的最大值。

注意：

不想被别人截获的命令要放到 FRMPayload，并单独发送该数据帧

一条 mac 命令由一个命令 ID（CID，一个字节），和特定的命令序列组成，命令序列可以是空。

命令 ID	命令	终端发送	网关发送	简介
0x02	LinkCheckReq	×		用于终端验证网络连接
0x02	LinkCheckAns		×	回应验证请求， 同时包含终端~~接收质量相关的估算的信号功率~~ 的功率估算，估算值与终端链路余量
0x03	LinkADRReq		×	请求终端改变数据率、传输功率、接收率或者信道
0x03	LinkADRAns	×		LinkRateReq 的应答
0x04	DutyCycleReq		×	设置设备的最大总发射占空比
0x04	DutyCycleAns	×		DutyCycleReq 的应答
0x05	RXParamSetupReq		×	设置接收时隙相关参数
0x05	RXParamSetupAns	×		RXSetupReq 的应答
0x06	DevStatusReq		×	请求终端状态
0x06	DevStatusAns	×		返回终端装填，即电量和解调情况
0x07	NewChannelReq		×	创建或修改无线电信道
0x07	NewChannelAns	×		NewChannelReq 的应答
0x08	RXTimingSetupReq		×	设置接收时隙的时间
0x08	RXTimingSetupAns	×		RXTimingSetupReq 的应答
0x80 ~ 0xFF	Proprietary	×	×	保留命令

PS：下面给张 LoRaWAN 的协议图解

1. 正常通信

最近有不少朋友对 PHYPayload 还是搞不清，再分享两张图片，把整个流程都分享给大家

1. 入网激活

Join Request (MType = 0)

Preamble	PHDR	PHDR_CRC	PHYPayload							CRC*
			MHDR 1B			MACPayload			MIC	
			Mtype	RFU	Major	AppEUI	DevEUI	DevNonce	4B	
			3b	3b	2b	8B	8B	2B		

1. 接收入网

Join Accept (MType = 1)

Preamble	PHDR	PHDR_CRC	PHYPayload									CRC*
			MHDR 1B			MACPayload						MIC
			Mtype	RFU	Major	AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList	4B
			3b	3b	2b	3B	3B	4B	1B	1B	16B(变长)	

- LoRaWAN 规范 1.0 （章节 2~5）
- LoRaWAN 规范 1.0 （章节 6）

6 终端激活（End-Device Activation）

所有终端设备在正式加入 LoRaWAN 网络之前必须先进行初始化并激活。有两种激活方式：
无线激活（Over-The-Air Activation (OTAA)），设备部署和重置时使用;
手动激活（Activation By Personalization (ABP)），此时初始化和激活一步完成。

6.1 激活成功后存储在终端设备的数据

以下信息在激活成功后回存储在终端设备：设备地址（DevAddr）、应用 ID（AppEUI）、网络会话密钥（NwkSKey）和应用会话密钥（AppKey）。

- 6.1.1 终端设备地址（DevAddr）

  DevAddr 是终端在当前网络中的识别码，大小 32bits。结构如下：

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

  最高 7 位是网络 ID（**NwkID**），用以区分有地域重叠的不同网络运营商和弥补有路由问题的网络。接下来的 25bits，终端设备网络地址（**NwkAddr**

- 6.1.2 应用唯一识别 ID（AppEUI）

  AppEUI 是 IEEE EUI64 的全球唯一应用 ID，用以识别终端设备的应用服务提供商（等等）。**AppEUI** 在进行激活操作之前就存储在终端设备中了。

- 6.1.3 网络会话密钥（NwkSKey）

  NwkSKey 是分配给终端设备的网络会话密钥。**网络服务器**和**设备用**它来计算和校验所有消息的 MIC（消息一致码），来保证收发的数据一致。**也可加/解密**。

- 6.1.4 应用会话密钥（**AppSKey**）

  AppSKey 是分配给终端设备的应用会话密钥。网络服务器和设备用来对 应用指定的 Payload 字段进行加解密。也可以用来计算和校验应用层 MIC

- 6.2.2 应用密钥（AppKey）

  AppKey 是 AES-128 的应用密钥，由应用拥有者通过 应用指定的 根密钥衍生并分配给终端设备，根密钥只有应用供应商知晓和掌握。终端设备通过无
  并分发相应的终端设备，用来加密和校验网络通讯和应用数据。

- 6.2.3 入网流程

  从终端的角度看，和服务器交互的入网流程包含两个 MAC 消息：join request 和 join accept.

- 6.2.4 入网请求（Join-request message）

  入网流程由终端发起，终端入网时发送入网请求，消息格式（MACPayoad）如下：

字节数	8	8	2
Join Request	AppEUI	DevEUI	DevNonce

  入网请求消息包含：**AppEUI**、**DevEUI** 和终端设备产生的 2 字节的**随机数（DevNonce）**
  DevNonce 是个随机值，终端设备最近使用的一些(数量自定义)**DevNonce** 会保存在网络服务器（NS）。如果终端发送的入网请求中的 **DevNonce** 在

  注意：该机制的目的是防止重放攻击(replay attacks)，避免其它人通过发送之前的入网请求来断开终端设备和网络的连接。

  入网请求的消息一致性校验码（MIC）（见第 4 章 MAC 消息部分）通过下述算法计算：

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

  - 1
  - 2
  - 1
  - 2

  入网请求以明文发送。

- 6.2.5 接受入网消息（Join-accept）

  服务器同意终端入网后网络服务器（NS）会回复 **接受入网** 消息。**接受入网**使用 JOIN_ACCEPT_DELAY1 或 JOIN_ACCEPT_DELAY2（而不是 RECEIVE_

- 1
- 2
- 1
- 2

接受入网的 MIC 计算方式如下：

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

- 1
- 2
- 1
- 2

消息本身采用 AppKey 加密，加密方式如下：

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

- 1
- 1

注意：网络服务器加密接收入网消息使用的是 AEC ECB 模式的**解密**算法。这样终端就不必再实现 AES 解密算法，只用 AES 加密即可。

PS ：之前关于 NetID 的翻译有误，特此更正。

NetID 包含：7 个最低有效位的 NetID，称作 NwkID；17 个最高有效位，存放前面提到的终端短址。区域相邻或重叠的网络的 NwkID 不能相同。网络
NetID 包含：NetID 的 7 个最低有效位称作 NwkID，即 DevAddr（终端短址）的 7 个最高有效位。区域相邻或重叠的网络的 NwkID 不能相同。余下的

DLsetting 字段含有下行配置：

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

RX1DRoffset 设置上下行数据速率之间的偏移（偏差），和终端设备交互的首个接收时隙（RX1）。*（感冒好几天了，头懵，不知道该怎么翻译）*
些地区基站的最大功率密度限制，offset 用来平衡上行和下行的无线链路余量。