

Physical Message Format

UPLINK

Preamble	PHDR	PHDR-CRC	PHYPAYLOAD	CRC
----------	------	----------	------------	-----

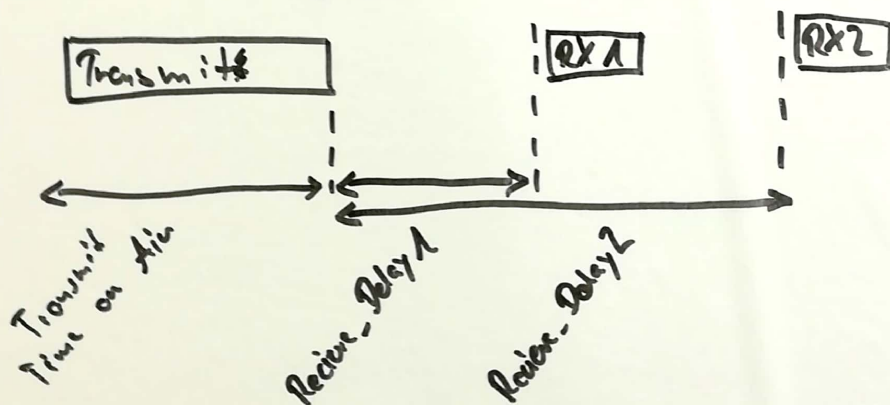
- sent by end-devices to server
- relayed by many gateways
- radio packet explicit mode (PHDR + PHDR-CRC included)

DOWNLINK

Preamble	PHDR	PHDR-CRC	PHYPAYLOAD
----------	------	----------	------------

- sent by network server to only one end-device
- relayed by a single gateway
- radio packet explicit mode

Receive Windows: after uplink transmission end-devices open two short receive windows



First receive Window (RX1):

- has a frequency that is a function of the frequency in uplink
- ~~for a data rate it has a function same for data rate~~
- by default, first receive window data rate = data rate of last uplink

Second receive Window (RX2):

- configurable data-rate & frequency (through MAC-Commands)

Receive Window duration: must be at least the time required by the radio transceiver

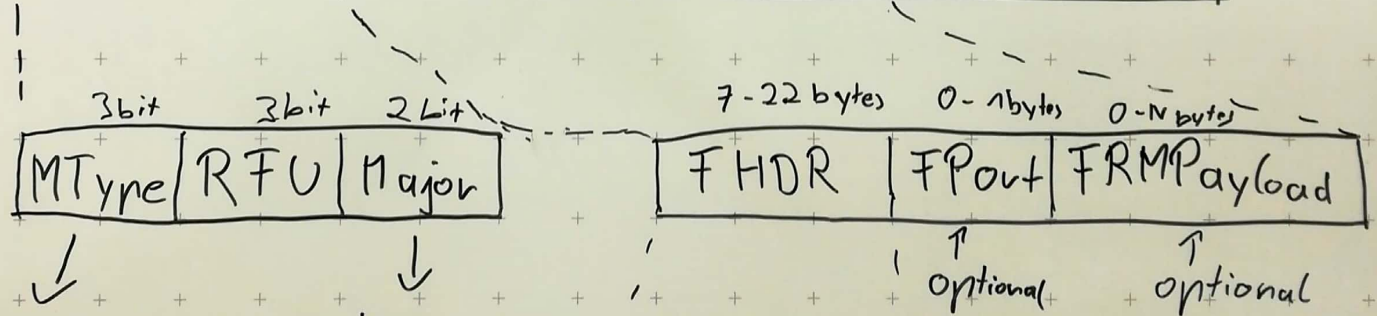
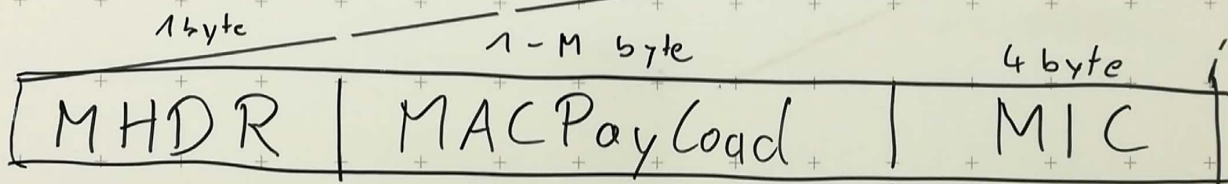
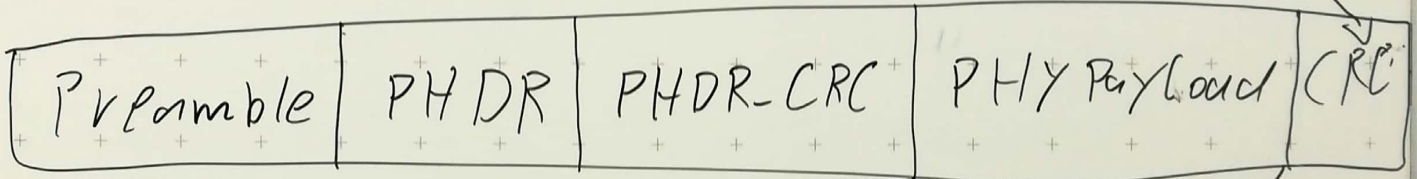
Receiver activity: If Preamble detected \rightarrow Receiver stays active until downlink frame demodulated

(Important notice: end-devices SHALL NOT transmit another uplink message before it either has received a downlink message in RX1 or RX2 of the previous transmission, or second receive window of previous transmission is expired)

MAC Message Formats

Radio Phy layer

new uplink



Join request

Join accept

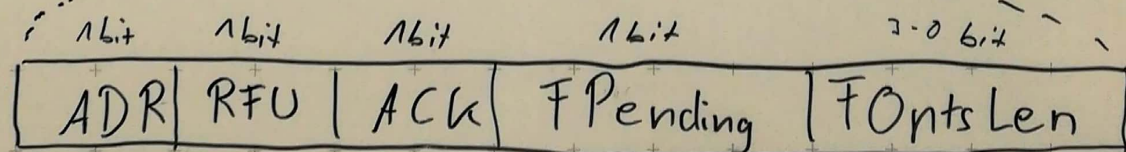
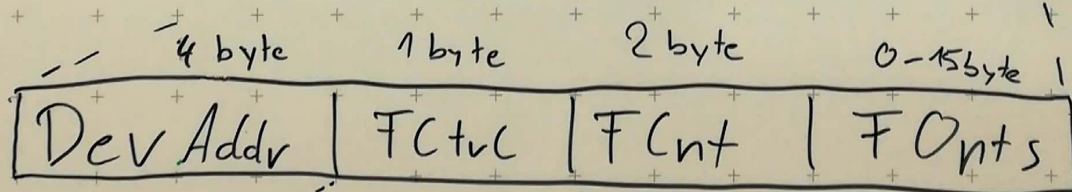
Unconfirmed Data Up/Down

Confirmed Data Up/Down

Version
(aktuell novRA)

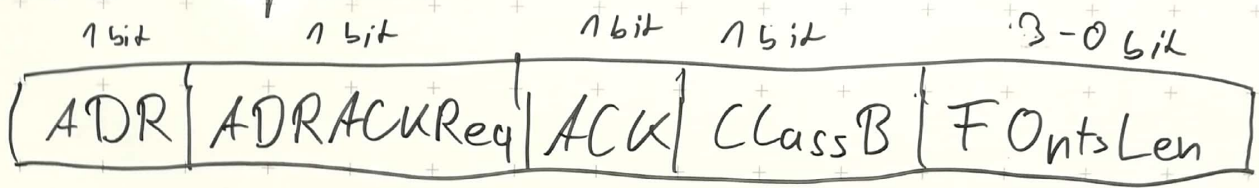
↑ optional

↑ optional



DOWNLINK

FCtrl - uplink



if set:

Netzwerk kontrolliert DataRate

if not set

Netzwerk versucht nicht
die DataRate zu beeinflussen
unabhängig der Signalqualität

Reset der Counter

Adress zur letzter
Message kein
Retransmission

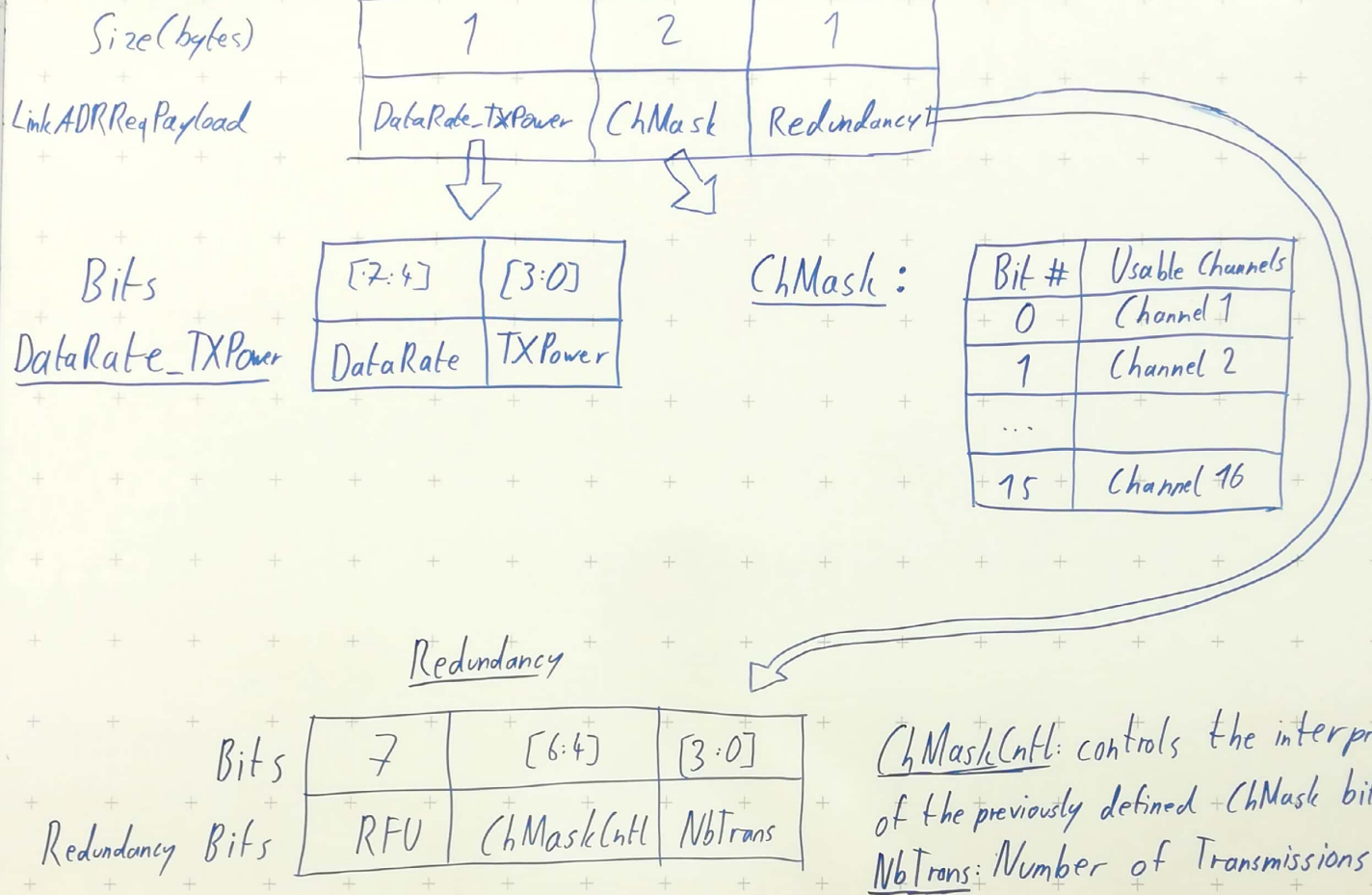
Retransmission:

abhängig vom Endgerät

MAC Commands

(ID)	Command	Transmitted by		Short description
		End device	Gateway	
0x02	LinkCheck Req	X		End device (ED) via (i) data network connectivity
0x02	LinkCheck Ans		X	Answer with reception quality
0x03	LinkADRReq		X	Change data rate, transmit power, repetition rate or power (ED)
0x03	LinkADR Ans	X		Ack
0x04	DutyCycleReq		X	Sets max aggregated Duty cycle
0x04	DutyCycle Ans	X		Ack
0x05	RXParamSetupReq		X	Sets reception slots parameters
0x05	RXParamSetupAns	X		Ack
0x06	DevStatusReq		X	Requests ED status
0x06	DevStatus Ans	X		ED status, battery & demodulation margin
0x07	NewChannelReq		X	Creates/modifies definition of Radio channel
0x07	NewChannel Ans	X		Ack

Network Server



End-Device

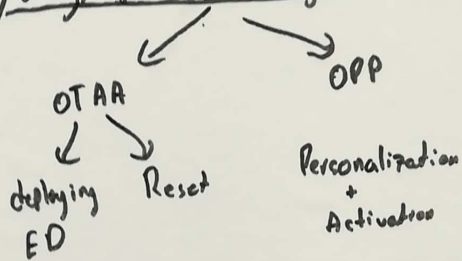
Bits	2	7	0
Status Bits	Power ACK	Data rate ACK	Channel mask ACK

Bit = 0 \Rightarrow Failure

Bit = 1 \Rightarrow Success

End device activation

1) Ways for Activating End device



1.1) Data stored in End Device

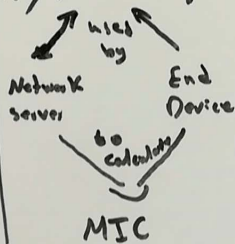
DevAddr, AppEUI, NWKKey, AppSKey

Bit	32... 15	24... 0
DevAddr	NWKID	NWKAddr

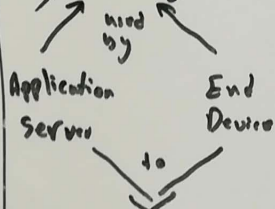
1.2) Application Identifier

Identifies the Entity able to Process of Join Req frame

1.3) NWKKey



1.4) AppSKey



Encrypt and Decrypt the payload for App Specific Data msgs

3) Activation by Personalization

End Device can be activated by personalization. It ties an End device to specific Network by passing the Join request and join accept procedure



DevAddr + NWKKey + AppSKey are stored into End Device

Instead of

DevEUI + AppEUI + AppKey

Each Device has unique set of NWKKey + AppSKey

compromising the keys in one Device should not affect the security of the other Device

Over-the Air Activation

- end device must follow joint procedure prior to participating in data exchange with network server.
- Joint procedure requires to be personalized with
 - Unique end-device Identifier (DevEUI)
 - Application Identifier (AppEUI) → network ID, ~~an AES 128 root key~~
 - Application Key (AppKey) → is an AES 128 root key specific to end device and is used to derive session keys: NwKskey & AppSkey.

→ join procedure → Join request → AppEUI & DevEUI

↳ followed by DevNonce → random value

Join-accept contains AppNonce, NetID, DevAddr, Delay
btwn Tx & Rx (Rx Delay)

AppNonce → is a random value

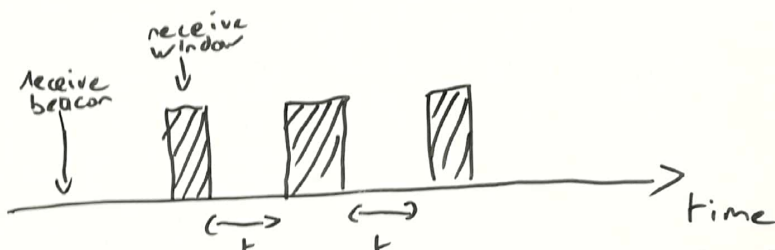
→ derives 2 Session Keys

Class B

Gateway provide a beacon to all end device.
It gives time reference in order to open "receive window".



□ gateway
○ end device



Network server: choose which ~~server~~ gateway will initiate the communication. It also send an uplink to give the routing path.

For each device, the network server must have information about:

- default periodicity between receive window
- receive window data rate
- receive window channel

All device join the class A.

Process to follow to be switched to class B:

1 - End device application must send a request to be switched to class B

2 - Change the FCTRL field to set the class B bit to 1

3 - Inform the network server to update routing path.

* - Periodicity of receive window can be changed. The end device must send a packet (Ping Slot Info Req) to the network server

** - If no beacon has been received for a given period, it means that the synchronisation with the network is lost.

The end device should be switched back to class A.