

CHANGE REQUEST

LoRaWAN	Version	1.0.3	CR#	TC23-00017	rev	02
----------------	----------------	--------------	------------	-------------------	------------	-----------

CR Title ADR – harmonize state transitions

Source Senet, Inc (Dave Kjendal – dkjendal@senetco.com)
Ross Gilson (machineQ) to merge TC23-00003.000.CR-
Allowing.ADRACKReq.at.default.DR

Topic or Working Group	Technical workgroup/main	Date	2018-10-23
-------------------------------	--------------------------	-------------	------------

Category **EFCD**

<i>E</i> (errata)	<i>B</i> (feature addition)
<i>F</i> (correction)	<i>C</i> (functional modification of a feature)
<i>A</i> (corresponds to a correction in an earlier release)	<i>D</i> (editorial modification / clarification)

Reason for change The specification is incomplete in its treatment of several ADR controlled state variables.

Summary of change Treatment of nbTrans during ADR back-off is described.
The AdrAckReq bit is allowed to be set even in default data-rate and transmit power conditions.

Clauses affected 4.3.1.1 Adaptive data rate control in frame header (ADR, ADRACKReq in FCtrl)
6.2.5 Join-accept messages

Other deliverables affected The Regional Parameters document needs to be updated such that all regions clearly define the “Join-request channels”. Currently only the dynamic channel plan regions define this term.

1. Discussion

As LinkAdrReq commands also control the nbTrans (number of retransmissions) and this field is an integral component to how the network manages (in conjunction with data-rate and transmit-power) the effective performance of the end-device, the field must also be controlled during ADR back-off. This change will require the end-device to reset nbTrans to its default value of one during the last step of ADR back-off. This point of control was chosen as nbTrans typically improves performance (unlike typical data-rate and transmit-power controls) and the last step of the ADR back-off algorithm is to revert all ADR controlled variables to defaults.

The AdrAckReq bit is required be set by the end-device in all cases after ADR_ACK_LIMIT has been reached and is only cleared upon successful reception of a downlink message. Previously the AdrAckReq bit was cleared once default data-rate and transmit power conditions were reached. This change provides additional, valuable information to the network (specifically that the end-device is still unable to receive downlinks) even in these default conditions.

Treatment of ADR parameters in response to Join-Accept messages are described in each of the four cases where Join-Accept messages are successfully processed by the end-device.

2. Proposed changes (normative/informative)

***** Start of proposed change 1 *****

4.3.1.1 Adaptive data rate control in frame header (ADR, ADRACKReq in FCtrl)

LoRa network allows the end-devices to individually use any of the possible data rates [and Tx power](#). This feature is used by the LoRaWAN to adapt and optimize the [number of retransmissions, the data rate and the Tx power](#) of ~~static~~ end-devices. This is referred to as Adaptive Data Rate (ADR) and when this is enabled the network will be optimized to use the fastest data rate possible.

Adaptive Data Rate control may not be possible when the radio channel attenuation changes [fast and constantly rapidly and/or continuously](#). When the network is unable to control the data rate of a device, the device's application layer ~~should~~ **SHOULD** control it. It is recommended to use a variety of different data rates in this case. The application layer should always try to minimize the aggregated air time used given the network conditions.

If the [uplink ADR](#) bit is set, the network will control the [number of retransmissions, the data rate and the Tx power](#) of the end-device through the appropriate MAC commands. If the **ADR** bit is not set, the network will not attempt to control the [number of retransmissions, the data rate nor the transmit power](#) of the end-device regardless of the received signal quality. [The network MAY still send commands to inform the end-device of the recommended configuration](#)

of the parameters present in the LinkAdrReq command. An end-device SHOULD accept the channel mask controls present in LinkADRReq even when the ADR bit is not set.

When the downlink ADR bit is set, it informs the end-device that the Network Server is in a position to send ADR commands. The device MAY independently set/unset the uplink ADR bit.

When the downlink ADR bit is unset, it signals the end-device that due to rapid changes of the radio channel, the network temporarily cannot estimate the best data rate. In that case the device has the choice to either

- Unset the ADR uplink bit, and control its uplink data rate, transmit power and channel plan following its own strategy. This SHOULD be the typical strategy for a mobile end-device.
- Ignore it (keep the uplink ADR bit set) and apply the normal ADR backoff algorithm in the absence of downlinks. This SHOULD be the typical strategy for a stationary end-device^[DK1].

The **ADR** bit MAY be set and unset by the end-device or the Network on demand. However, whenever possible, the ADR scheme ~~should~~ **SHOULD** be enabled to increase the battery life of the end-device and maximize the network capacity.

Note: Even mobile end-devices are actually immobile most of the time. So depending on its state of mobility, an end-device can request the network to optimize its data rate using the uplink ADR bit.

Default Tx Power is the maximum transmission power allowed for the device considering device capabilities and regional regulatory constraints. The device shall use this power level, until the network ~~asks for less~~ attempts to change the power level, ~~throughby~~ using the LinkADRReq MAC command.

If an end-device ~~wishes to check for connectivity loss or if an end-device's whose~~ data rate is optimized by the network to use a data rate higher than its ~~lowest available~~ default data rate, ~~or a TXPower lower than its default TXPower, the end device~~ periodically needs to validate that the network still receives the uplink frames. Each time the uplink frame counter is incremented (for each new uplink ~~frame~~, repeated transmissions do not increase the ~~frame~~ counter), the device increments an ADR_ACK_CNT counter. After ADR_ACK_LIMIT uplinks (ADR_ACK_CNT >= ADR_ACK_LIMIT) without ~~receiving any a class A~~ downlink response, ~~the end device~~ sets the ADR acknowledgment request bit (**ADRACKReq**) ~~on uplink transmissions~~. ~~The network is required to respond with a downlink frame within the next ADR_ACK_DELAY frames, a~~ Any ~~received class A~~ downlink frame ~~received~~ following an uplink frame resets the ADR_ACK_CNT counter. ~~Upon receipt~~ ~~reception of any class A~~ downlink the end-device **MUST** clear the ADRACKReq bit. The downlink **ACK** bit does not need to be set as any ~~response class A downlink frame received by~~ during the receive slot of the end-device indicates that the ~~gateway network~~ has ~~still~~ received the uplinks from this device. If no ~~reply class A downlink frame~~ is received within the next ADR_ACK_DELAY uplinks (i.e., after a total of ADR_ACK_LIMIT + ADR_ACK_DELAY ~~transmitted frames~~), the end-device ~~MAY~~ **MUST** try to regain connectivity by ~~first stepping up setting the transmit power to the default power if possible then~~ switching to the next lower data rate that provides a longer radio range. The end-device ~~will~~ **MUST** further lower its data rate step by step every time ADR_ACK_DELAY ~~are transmitted is reached~~. ~~Once the device has completed the ADR backoff algorithm reached the lowest data rate without receiving a class A downlink frame, it~~ **MUST** re-enable all default uplink frequency channels and reset nbTrans to its default value of 1.

~~The ADRACKReq SHALL not be set if the device uses its lowest available data rate because in that case no action can be taken to improve the link range.~~^[GR2]

Note: For fixed channel plan regions (US, AU, CN, etc), the device must enable ALL channels. For dynamic channel plan regions (EU, IN, AS, etc.) the device must enable the region's default channels and will make no change to the configuration of the dynamically configured channels.^[GR3]

Note: Not ~~requesting~~ requiring an immediate response to an ADR acknowledgement request provides flexibility to the network to optimally schedule its downlinks.

Note: In ~~uplink transmissions the~~ **ADRACKReq** bit is set if ~~ADR_ACK_CNT >= ADR_ACK_LIMIT, and the current data rate is greater than the device defined minimum data rate, it is cleared in other conditions.~~^[GR4]

The following table provides an example of data rate back-off sequence in the EU region assuming ADR_ACK_LIMIT and ADR_ACK_DELAY constants are both equal to 32.

<u>ADR_ACK_CNT</u>	<u>ADRACKReq bit</u>	<u>Data Rate</u>	<u>TX power</u>	<u>nbTrans</u>	<u>Channel Mask</u>
<u>0 to 31</u>	<u>0</u>	<u>SF11</u>	<u>Max – 9dBm</u>	<u>3</u>	Single channel enabled <u>Normal Operations Channel Mask</u>
<u>32 to 63</u>	<u>1</u>	<u>Keep</u>	<u>Keep</u>	<u>Keep</u>	<u>Keep</u>
<u>64 to 95</u>	<u>1</u>	<u>Keep</u>	Max <u>Default</u>	<u>Keep</u>	<u>Keep</u>
<u>96 to 127</u>	<u>1</u>	<u>SF12</u>	Default <u>Max</u>	<u>Keep</u>	<u>Keep</u>
<u>>= 128</u>	<u>1</u>	<u>SF12</u>	Default <u>MAX</u> ^[GR5]	<u>1</u>	<u>All channels enabled</u>

Figure 1 : data rate back-off sequence example

***** End of proposed change 1 *****

***** Start of proposed change 2 *****

6.2.5 Join-accept message

The network ~~server~~ will respond to the **join-request** message with a **join-accept** message if the end-device is permitted to join a network. The join-accept message is sent like a normal downlink but uses delays JOIN_ACCEPT_DELAY1 or JOIN_ACCEPT_DELAY2 (instead of RECEIVE_DELAY1 and RECEIVE_DELAY2, respectively). The channel frequency and data rate used for these two receive windows are identical to the one used for the RX1 and RX2 receive windows described in the “receive windows” section of the “LoRaWAN regional physical layer specification” document

No response is given to the end-device if the ~~join-join~~ request is not accepted.

The join-accept message contains an application nonce (**AppNonce**) of 3 octets, a network identifier (**NetID**), an end-device address (**DevAddr**), a delay between TX and RX (**RxDelay**) and an optional list of channel frequencies (**CFList**) for the network the end-device is joining. The CFList option is region specific and is defined in the “LoRaWAN regional physical layer specification” document.

Size (bytes)	3	3	4	1	1	(16) Optional
Join Accept	AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList

The **AppNonce** is a random value or some form of unique ID provided by the network **server** and used by the end-device to derive the two session keys **NwkSKey** and **AppSKey** as follows:¹

$NwkSKey = \text{aes128_encrypt}(\text{AppKey}, 0x01 \mid \text{AppNonce} \mid \text{NetID} \mid \text{DevNonce} \mid \text{pad}_{16})$

$\text{AppSKey} = \text{aes128_encrypt}(\text{AppKey}, 0x02 \mid \text{AppNonce} \mid \text{NetID} \mid \text{DevNonce} \mid \text{pad}_{16})$

The MIC value for a join-accept message is calculated as follows:²

$\text{cmac} = \text{aes128_cmac}(\text{AppKey},$
 $\text{MHDR} \mid \text{AppNonce} \mid \text{NetID} \mid \text{DevAddr} \mid \text{DLSettings} \mid \text{RxDelay} \mid \text{CFList})$

$\text{MIC} = \text{cmac}[0..3]$

The join-accept message itself is encrypted with the **AppKey** as follows:

$\text{aes128_decrypt}(\text{AppKey}, \text{AppNonce} \mid \text{NetID} \mid \text{DevAddr} \mid \text{DLSettings} \mid \text{RxDelay} \mid \text{CFList} \mid \text{MIC})$

Note: ~~The network server uses an AES decrypt operation in ECB mode is used~~ to encrypt the join-accept message so that the end-device can use an AES encrypt operation to decrypt the message. This way an end-device only has to implement AES encrypt but not AES decrypt.

Note: Establishing these two session keys allows for a federated network **server** infrastructure in which network operators are not able to eavesdrop on application data. ~~In such a setting, the application provider must support the network operator in the process of an end-device actually joining the network and establishing the NwkSKey for the end-device. At the same time t~~The application provider commits to the network operator that it will take the charges for any traffic incurred by the end-device and retains full control over the AppSKey used for protecting its application data.

The format of the **NetID** is as follows: The seven LSB of the **NetID** are called **NwkID** and match the seven MSB of the short address of an end-device as described before. Neighboring or overlapping networks must have different **NwkIDs**. The remaining 17 MSB can be freely chosen by the network operator.^[DK6]

The DLsettings field contains the downlink configuration:

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

The RX1DRoffset field sets the offset between the uplink data rate and the downlink data rate used to communicate with the end-device on the first reception slot (RX1). By default this offset is 0. The offset is used to take into account maximum power density constraints for base stations in some regions and to balance the uplink and downlink radio link margins.

The actual relationship between the uplink and downlink data rate is region specific and detailed in the “Physical Layer” section

The delay **RxDelay** follows the same convention as the **Delay** field in the **RXTimingSetupReq** command.

¹ The pad₁₆ function appends zero octets so that the length of the data is a multiple of 16.

² [RFC4493]

If the Join-accept message is received following the transmission of:

- A Join-Request the device SHALL revert to its default channel and RF parameters definitions. If the CFlist is present, it is used to update the default channels and RF parameters. The MAC layer parameters (except RXdelay1, RX2 data rate, and RX1 DR Offset that are transported by the join-accept message) SHALL all be reset to their default values.

***** *End of proposed change 2* *****