Thinking LoRaWAN Metrics — WP1

A document to initiate discussions

Renzo E. Navas (Post-doc)
renzo.navas@inria.fr

Outline

Inria

Objective

Objective of this presentation:

- Initiate discussions on LoRaWAN metrics.

01

Background/Baseline Info

Inria

Objectives of this section:

- Validate the primary LoRa-PHY and LoRaWAN information collected at a Network Server (NS).
- Validate and discuss the supported LoRaWAN MAC commands.

Objectives of this section:

- Validate the primary LoRa-PHY and LoRaWAN information collected at a Network Server (NS).
- Validate and discuss the supported LoRaWAN MAC commands.

NOTE: I pose some questions to have a common understanding on the primary information available at NS-level. (Detailed answers expected to continue in offline communication.)

[LoRa-PHY] Is this is the raw Info the GWs forward to the NS?

A RF packet and associated metadata with the following fields:

Name	Type	Function
time	string	UTC time of pkt RX, us precision, ISO 8601 'compact' format
tmms	number	GPS time of pkt RX, number of milliseconds since 06.Jan.1980
tmst	number	Internal timestamp of "RX finished" event (32b unsigned)
freq	number	RX central frequency in MHz (unsigned float, Hz precision)
chan	number	Concentrator "IF" channel used for RX (unsigned integer)
rfch	number	Concentrator "RF chain" used for RX (unsigned integer)
stat	number	CRC status: 1 = OK, −1 = fail, 0 = no CRC
modu	string	Modulation identifier "LORA" or "FSK"
datr	string	LoRa datarate identifier (eg. SF12BW500)
datr	number	FSK datarate (unsigned, in bits per second)
codr	string	LoRa ECC coding rate identifier
rssi	number	RSSI in dBm (signed integer, 1 dB precision)
lsnr	number	Lora SNR ratio in dB (signed float, 0.1 dB precision)
size	number	RF packet payload size in bytes (unsigned integer)
data	string	Base64 encoded RF packet payload, padded

Source (packet_forwarder):
`https://github.com/Lora-net/packet_forwarder/blob/master/PROTOCOL.TXT`

Inria

Information collected at NS-level – II

What LoRA-PHY- and LoRaWAN- Info is stored/aggregated at NS-level?:

- Can we have an example? E.g.,
  - (PHY-level) #CRC failed RF-packets,
  - (MAC-level) #frames w/failed Message Integrity Codes (MIC),
  - Number of GWs that received the same "frame".
- ... database description?
- How (de)coupled are NSs with Application Servers?

LoRaWAN MAC commands – I

... just for reference (LoRaWAN 1.0.4):

CID	Command	Transmitted by		Brief Description
		End-device	Network Server	
0x02	***LinkCheckReq***	x		Used by an end-device to validate its connectivity to a network.
0x02	***LinkCheckAns***		x	Answers ***LinkCheckReq***. Contains the received signal power estimation, which indicates the quality of reception (link margin) to the end-device.
0x03	***LinkADRReq***		x	Requests the end-device to change data rate, TX power, redundancy, or channel mask.
0x03	***LinkADRAns***	x		Acknowledges ***LinkADRReq***.
0x04	***DutyCycleReq***		x	Sets the maximum aggregated transmit duty cycle of an end-device.
0x04	***DutyCycleAns***	x		Acknowledges ***DutyCycleReq***.
0x05	***RXParamSetupReq***[7]		x	Sets the reception slot parameters.
0x05	***RXParamSetupAns***	x		Acknowledges ***RXParamSetupReq***.
0x06	***DevStatusReq***		x	Requests the status of the end-device.
0x06	***DevStatusAns***	x		Returns the status of the end-device, namely its battery level and its radio status.
0x07	***NewChannelReq***		x	Creates or modifies the definition of a radio channel.
0x07	***NewChannelAns***	x		Acknowledges ***NewChannelReq***.
0x08	***RXTimingSetupReq***[7]		x	Sets the timing of the reception slots.
0x08	***RXTimingSetupAns***	x		Acknowledges ***RXTimingSetupReq***.
0x09	***TXParamSetupReq***[7]		x	Used by a Network Server to set the maximum allowed dwell time and MaxEIRP of end-device, based on local regulations.
0x09	***TXParamSetupAns***	x		Acknowledges ***TXParamSetupReq***.
0x0A	***DlChannelReq***[7]		x	Modifies the definition of a downlink RX1 radio channel by shifting the downlink frequency from the uplink frequencies (i.e. creating an asymmetric channel).
0x0A	***DlChannelAns***	x		Acknowledges ***DlChannelReq***.
0x0B to 0x0C	RFU			
0x0D	***DeviceTimeReq***	x		Used by an end-device to request the current GPS time.
0x0D	***DeviceTimeAns***		x	Answers ***DeviceTimeReq***.
0x0E to 0x0F	RFU			
0x10 to 0x1F	Class B commands (cf. Sections 12).			
0x20 to 0x2F	Reserved for Class C commands.			
0x30 to 0x7F	RFU			
0x80 to 0xFF	Proprietary	x	x	Reserved for proprietary network command extensions.

Table 14: MAC commands

CID	Command	Transmitted by		Brief Description
		End-device	Network Server	
0x10	***PingSlotInfoReq***	x		Used by the end-device to communicate the unicast ping-slot periodicity to the Network Server
0x10	***PingSlotInfoAns***		x	Used by the Network to acknowledge a ***PingInfoSlotReq*** command
0x11	***PingSlotChannelReq***[13]		x	Used by the Network Server to set the unicast ping channel frequency and data rate of an end-device
0x11	***PingSlotChannelAns***	x		Used by the end-device to acknowledge a ***PingSlotChannelReq*** command
0x12	***BeaconTimingReq***	x		Deprecated
0x12	***BeaconTimingAns***		x	Deprecated
0x13	***BeaconFreqReq***		x	Command used by the Network Server to modify the frequency at which the end-device expects to receive a beacon broadcast
0x13	***BeaconFreqAns***	x		Used by the end-device to acknowledge a ***BeaconFreqReq*** command

Table 61: Class B MAC command table

Question to Acklio/Aguila:

- **Which LoRaWAN version (1.0.X) and which MAC commands the NS implementations [not] support?**
- *Heads-up*: We may need to define Proprietary MAC commands (CID: 0x80 to 0xFF).
  - > This is the proper –clean– way to communicate Information gathered at the NS[a] to End-Devices.

[a] or Network-related Info that the NS relays on behalf of other Entities that are NOT Application Servers. E.g., the beautiful metrics we will have :)

Inria

LoRaWAN MAC commands – III

From an End-Device perspective –at least– we need:

- **"Link Check" commands** (*LinkCheckReq/Ans*)
- **"Link ADR" commands** (*LinkADRReq/Ans*)
- ... all Class A MAC if possible? (... except "DeviceTime"?)
- If we plan on doing things with **multicast** (Class B):
  - > "Class B end-device SHALL support at least one multicast group"
  - > MAC Class B: "BeaconFreq", "PingSlotInfo", "PingSlotChannel".

Inria

02

Leveraging on ADR Biblio

Inria

Objectives of this section:

- Leverage ideas from "metrics" used in LoRaWAN ADR bibliography.
  - > Synthesis of the 22 articles surveyed in [KHA20].

Table: Metrics from ADR Bibliography [KHA20]

Metric Name	Definition	NS wide	Per SF	Per Frq	Per GW	Per Node
Data Extraction Rate (DER) [Bor+16]	The ratio of received messages to transmitted messages over a period of time.*	✓	✗**	✗**	✗**	✓
Network Energy Consumption (NEC)[Bor+16]	Energy spent by the network to successfully extract a message†	✓	✓	✓	✓	✓‡
Packet Delivery Ratio (PDRo)	$\frac{\#msg\_sent\_by\_nodes}{\#msg\_rcv\_NS}$ (See§)	✓	✓¶	✓¶	✓¶	✓
Packet Reception Rate (PRR)	*—equivalent to DER metric—*					

* *NB1:* Apparently, only for Uplink traffic. *NB2:* Received TX-Msg can be calculated using the frame counters per node.

** Not applicable in a real deploy, because a lost packet can not be –easily– attributed to a 'PHY-link'. But applicable on simulation.

† NB1: "NEC depends on the number of nodes, frequency of transmissions and transmitter communication parameters". NB2: Does not count energy spent on not received packets (?!...).

‡ Even if a network-centric metric, we can discriminate per node.

§ As many metrics, makes sense only over a period of time (we could calculate over a fixed period or sliding window).

¶ Theory/Sim: Very useful! Real Deploy: Makes only sense if attached to a Per-Node-Calculation.

Inria

Synthesis – II

Table: Metrics from ADR Bibliography [KHA20] — Cont.

Metric Name	Definition	NS wide	Per SF	Per Frq	Per GW	Per Node
Packet Error Rate (PER)	$\frac{\#\text{packets\_crc\_error}}{\#\text{sent\_packets}}$, over a period of time.*	✓	✓	✓	✓	✓
Packet Loss Rate (PLR)	$\frac{\#\text{lost\_packets}}{\#\text{sent\_packets}}$, over a period of time†	✓	✗‡	✗‡	✗‡	✓
Packet Loss Ratio (PLRo)	$\frac{\#\text{lost\_packets}}{\#\text{sent\_packets}}$ (See§)	✓	✗‡	✗‡	✗‡	✓
Jain's Fairness index [DBP19]	$\frac{(\sum_{i=1}^{n} x_i)^2}{n\sum_{i=1}^{n} x_i^2}$ (See¶)	✓	✓	✓	✓	✗

* In the biblio, PER and PLR sometimes is used indistinctly because it is mostly simulations. In a Real-like deploy, we should define PER as a packet that was attempted to be demodulated at the GW but got a CRC error.

† NB1: Uplink? #sent_packets can be calculated with Frame Counter of Received Packets. **NB2: In a real deploy, this is actually an indirect metric = 1−(P. Error Rate + P. Reception Rate).**

‡ Not applicable in a real deploy, because a lost packet can not be –easily– attributed to a 'PHY-link'. But applicable on simulation.

§ In the biblio, PLRo is used as 1-PDRo (they do not discriminate CRC errors)

¶ NB1: x_i denotes the *normalized throughput* of each device and n the total number of active devices in each "slice". NB2: Index varies between 0 and 1, with 1 being perfectly fair.

Inria

Summary–Thoughts

- **DER** and **NEC** [Bor+16], used explicitly by several articles
  - > ... **Data Extration Rate**, is simply a **PRR**.
- **Jain's Fairness Index** [DBP19], very interesting for slicing.

Inria

Summary–Thoughts

- **DER** and **NEC** [Bor+16], used explicitly by several articles
  - > ... **Data Extration Rate**, is simply a **PRR**.
- **Jain's Fairness Index** [DBP19], very interesting for slicing.
- The rest are the "usual suspects" (packet-based): very useful.
  - > *Fact*: a packet is the unity of information.
  - > I avoided raw "throughput" considerations (bits/second).
  - > yes.. packets come in different bit-sizes. If needed, we can transform packet metrics into raw-bits metrics.
  - > I propose to focus on packets of **59 Bytes** of MACPayload –51B of application payload–, the maximum on DR0–SF12.

Inria

Summary–Thoughts

- **DER** and **NEC** [Bor+16], used explicitly by several articles
  - > ... **Data Extration Rate**, is simply a **PRR**.
- **Jain's Fairness Index** [DBP19], very interesting for slicing.
- The rest are the "usual suspects" (packet-based): very useful.
  - > *Fact*: a packet is the unity of information.
  - > I avoided raw "throughput" considerations (bits/second).
  - > yes.. packets come in different bit-sizes. If needed, we can transform packet metrics into raw-bits metrics.
  - > I propose to focus on packets of **59 Bytes** of MACPayload –51B of application payload–, the maximum on DR0–SF12.
- From a node-perspective:
  - > **Packet Delivery Ratio (or Rate)** and **PER**/**PLR**, are the most useful to take local-decisions (non collaborative, greedy).
    - – (PDR, etc ... aggregated by LoRa-PHY metadata.)
    - – E.g., if a node is too far away from GWs, SF7 will never be exploitable, independently of GW/Network global load/factors.

Inria

LoRaWAN ADR Bibliography

[Bor+16] Martin C Bor et al. "Do LoRa low-power wide-area networks scale?" In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2016, pp. 59–67. DOI: 10.1145/2988287.2989163.

[DBP19] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. "Adaptive dynamic network slicing in LoRa networks". In: *Future generation computer systems* 98 (2019), pp. 697–707. DOI: 10.1016/j.future.2019.01.042.

[KHA20] Rachel Kufakunesu, Gerhard P Hancke, and Adnan M Abu-Mahfouz. "A survey on Adaptive Data Rate optimization in LoRaWAN: Recent solutions and major challenges". In: *Sensors* 20.18 (2020), p. 5044. DOI: 10.3390/s20185044.

03

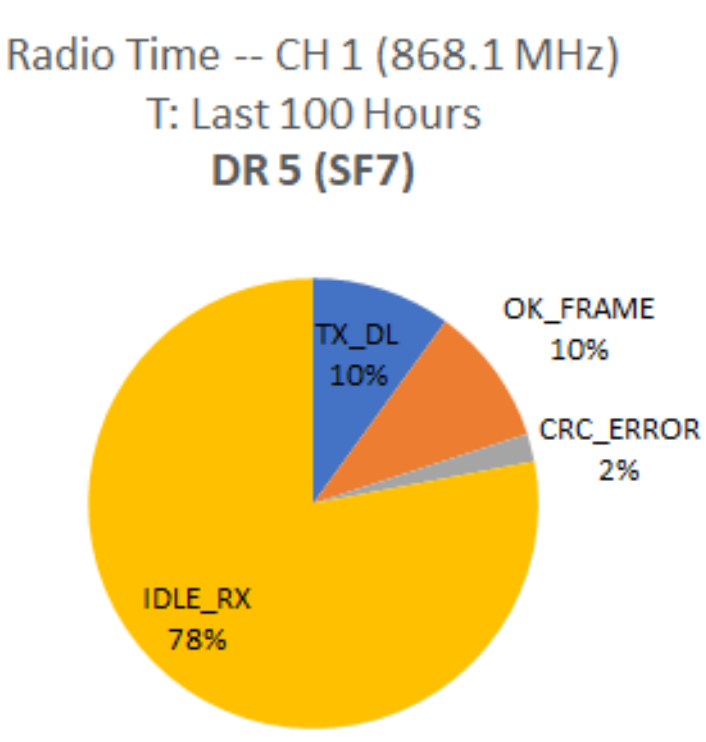
Possible Metrics

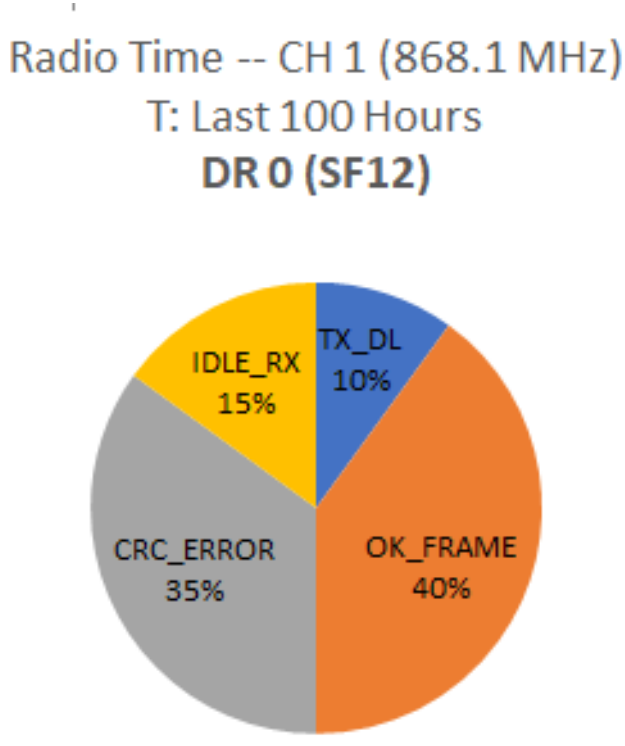
Inria

Ideas – Network-centric

- Metrics, aggregation of Info (See slides 6-7):
  - > Can be aggregated in many meaningful ways (dimensions).
  - > E.g., makes sense to aggregate per:
    - – Orthogonal channels –Frequency–,
    - – Semi-orthogonal Channels –DR-,
    - – Other: Nodes, frames.
  - > ΔTime: sliding window or fixed-period?
- Think about Down Link (DL) traffic information/metrics
  - > ... we really want to minimize DL, penalize abusive nodes, etc.

From GW perspective, relevant to know how radio-time is used.





- Aggregation per multiple DR/SF is not easy
  - > DRs for a given Frequency/CH are **not** time-independent.
  - > But is very relevant to have. TODO: think.

Node-centric Info collected at NS-level (see slides 6-7):

I) Store info for the *N* most recent uplink packets.

- E.g., N=20 (See: The Things Network ADR)
- **Signal-to-Noise Ratio (SNR)**.
  - > **Question: is (I)snr independent of the Spreading Factor?**
- PHY-info, from lora_packet_forwarder:
  - > DR, Freq, **Timestamp**, RSSI, CRC status...
- MAC-info:
  - > Frame counter, bad MIC, #GWs that received same packet, ...

II) Calculate, upon node-request (or scheduled DL uni/multicasts):

- Signal-to-Noise Ratio $\rightarrow$ (average , std_dev)
- (TODO)
  - > Time-related information (E.g., packets per unity of Time)

Inria

Node-centric Metric:

"Energy spent to successfully deliver a 59 Bytes frame"

- ... Let's name it EPDR.
- Will be very useful aggregated per Data Rate.
- If we assume constant TX-power, we can think in the domain of Time, instead of Energy; e.g., time-radio-on (seconds).

Ideas – Node-centric

- The “EPDR” metric can only be calculated in-node[a].
- But, the node needs info available only at NS-level (PDR).

Ideas – Node-centric

- The "EPDR" metric can only be calculated in-node[a].
- But, the node needs info available only at NS-level (PDR).
- We need DL traffic. A MAC-level protocol, draft:
  1. (UL) $\rightarrow$: Stats Request
  2. (DL) $\leftarrow$: Stats Response [**max 15 Bytes/120 bits**[bc]]
     - Contains Packet-Delivery Ratio info aggregated per DR
     - **Note: DL traffic is catastrophic for overall network performance (GWs are half-duplex!)$\rightarrow$ minimize!**

---

[a] Only the node knows the PHY params (= Energy spent) of lost frames.

[b] "Piggybacked MAC commands SHALL always be sent without encryption and SHALL NOT exceed 15 octets" (Page 26 of LoRaWAN TS001-1.0.4 spec)

[c] "MAC commands sent as FRMPayload SHALL always be encrypted and SHALL NOT exceed the maximum FRMPayload length" (56 Bytes)

Inria

04

Summary–Discussion

Inria

Summary – Thinking LoRaWAN Metrics

- Node-perspective:
  - > **Energy**-centric metrics (aggregated per SF).
    - – **Packet**-metrics can be transformed to Energy using the LoRA-PHY params.
  - > ... or Radio-ON **time** (if power constant).
- Network-perspective:
  - > **Spectrum**-use-centric metrics (?)
    - – Use of **radio time**
    - – Time-Information: "throughput"
    - – PDR of "distant" nodes: **"fairness"**
  - > ...

Inria

Thank you!
(... and more discussion)