UMR IRISA

Institut de Recherche en Informatique et Systèmes Aléatoires

J-LoRaNeS: A Julia based LoRa Network Simulator

Jules COURJAULT, PhD student - jules.courjault@irisa.fr

Supervised by: Olivier BERDER and Baptiste VRIGNEAU

Industrial collaboration with CG-wireless

CG-WIRELESS
Innovative Radio Frequency Solutions

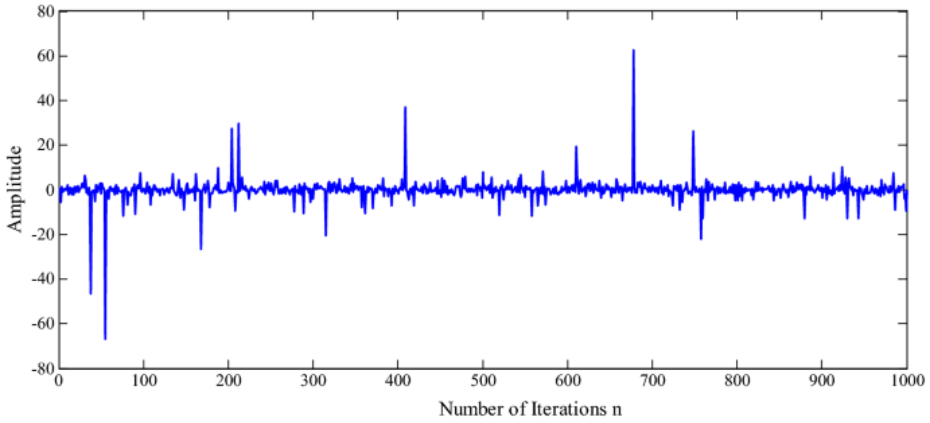
7/07/2022

eit Digital

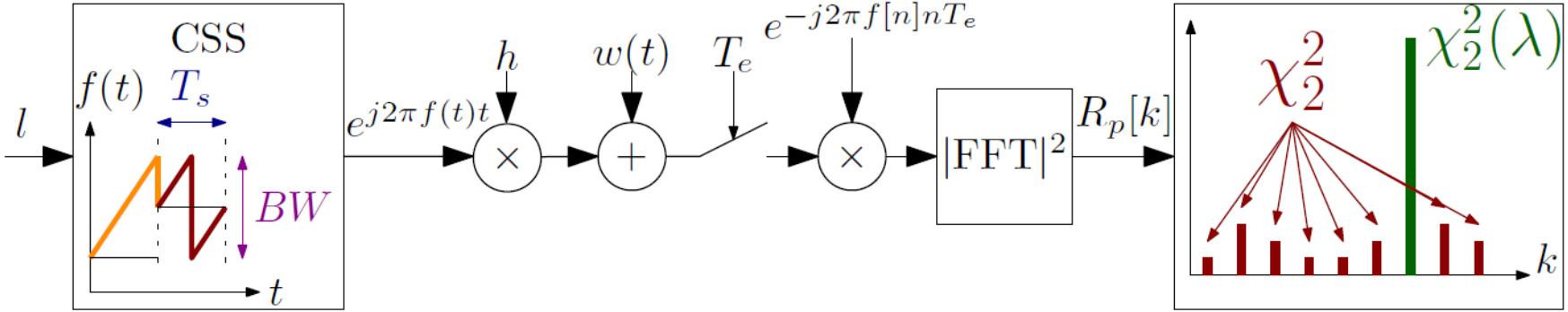
I. Context: Industrial environment

- Industry 4.0 => Wireless sensor networks (Indutrial IoT)
- Indoor transmission
- High attenuation and multiple reflections due to metallic structure
- Impulsive noise due to high-power machinery

➔ Specific channel model + Non-Gaussian noise => Is LoRa a good candidate for IIoT ?



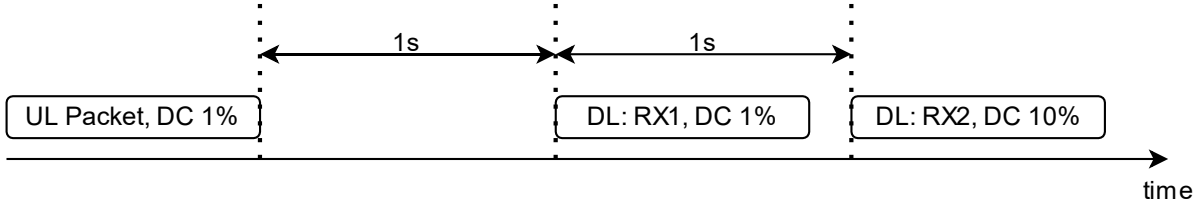
II. LoRa principle: PHY-layer description



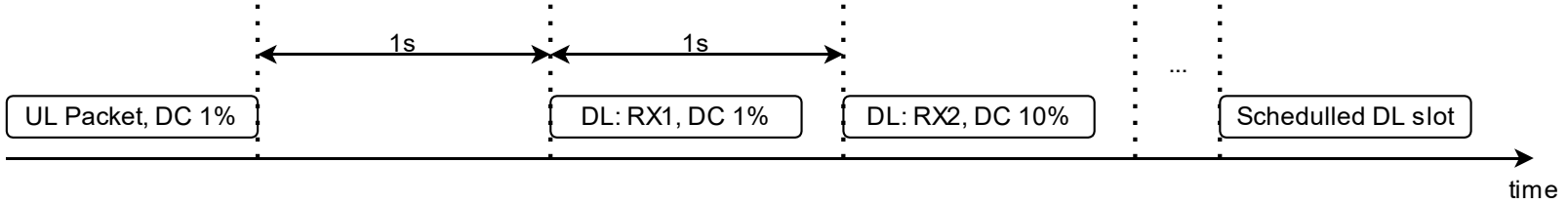
- $BW \in \{125, 250, 500\}$ *kHz,* $CR \in \{4/5\,, 4/6\,, 4/7\,, 4/8\,\}$, $SF \in [\![7, 12]\!]$
- Symbol l is coded using CSS modulation with a frequency offset depending on the value of the symbol
- At reception the symbol is decoded thanks to de-chirping and finding the index with the highest square modulus of the FFT
- Increasing SF increases the communication robustness to impulsive noise [1]
- SF and TP management => ToA and energy consumption => trade-off between QoS and energy consumption

II. LoRa principle: devices classes

- Class A: After sending an UL, the ED listen for DL response on RX1 and RX2 windows



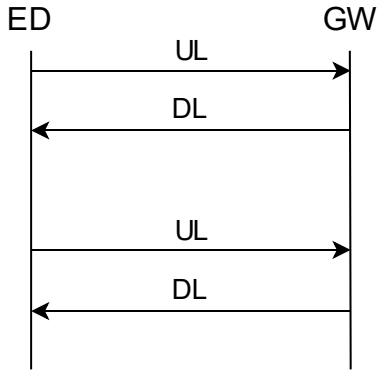
- Class B: Class A + Listen for DL on scheduled timeslots



- Class C: Always listen for DL packets

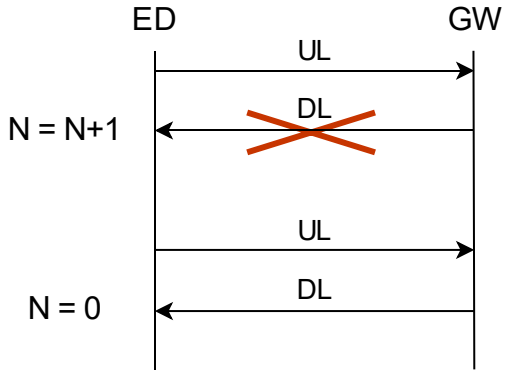
II. LoRa principle: LoRaWAN ADR on node side [2]

Case 1



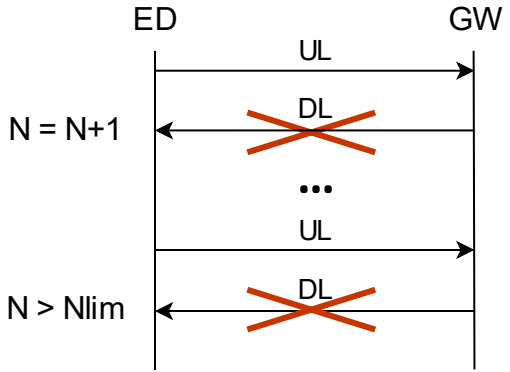
DL is received after each UL.
Nothing happens

Case 2



Increases counter N when DL is not received on RX1 or RX2.
Counter resets if DL is received

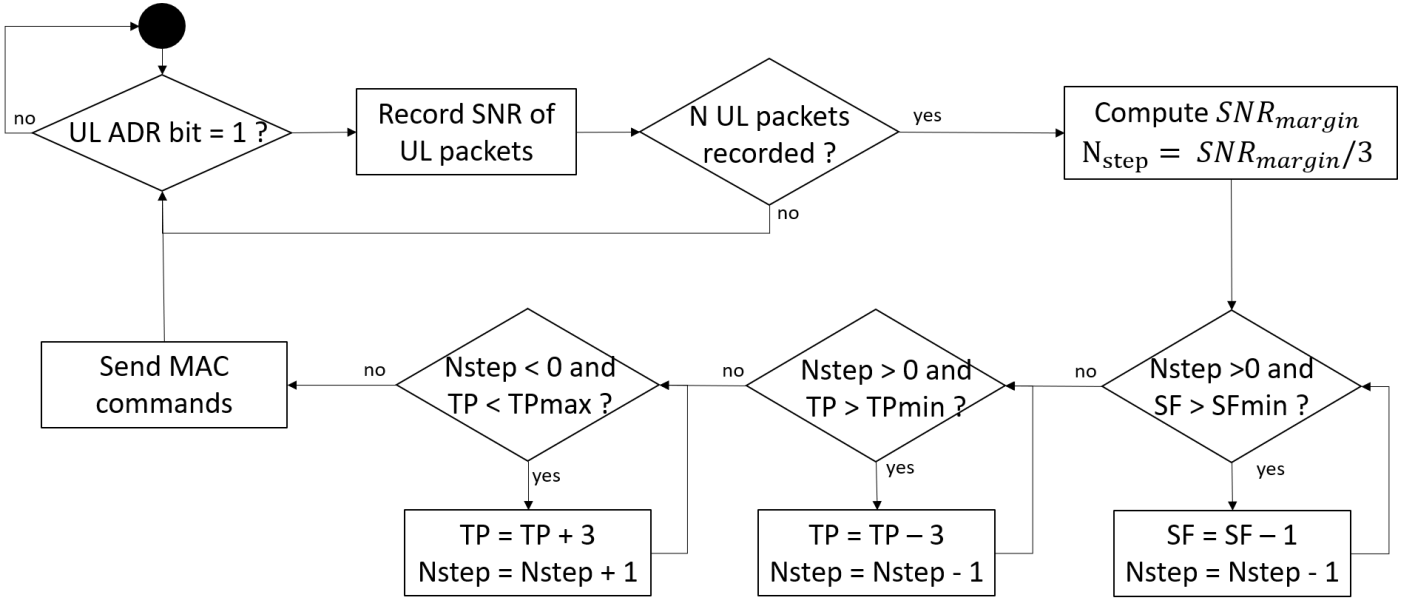
Case 3



If the counter goes above a certain limit, node will increase SF or TP for next ULs

➔ **Increase the probability to reach a GW,** whatever the energy cost is.

II. LoRa principle: LoRaWAN ADR on server side [2]



➔ **Increase energy efficiency** of the communication.

III. LoRa Network Simulator

Need of a simulator

Simulator objectives:

- Simulation of physical phenomenon as accurate as possible
- Class A device simulation
- Network performance monitoring:
  - Packet Delivery Ratio
  - Power consumption
  - Convergence and stability of the ADR algorithm
- Test new algorithms based on RL to replace the current ADR

II. LoRa Network Simulator

Existing simulators

Simulator	Perf. metrics	Language	Channel model	DC	DT	ISFO	ADR
FLoRa [3]	EC per successful transmission Packet delivery ratio	Omnet++	Log-distance	Yes	Yes	No	LoRaWAN ADR
LoRaWANSim [4]	Network/node EC Packet delivery ratio	Matlab	Okumura-Hata	Yes	Yes	Yes	Pre-computed LoRaWAN ADR
LoRaSim [5]	Network EC Data extrac. Rate	Python	Log-distance	No	No	No	No
LoRa-MAB [6]	Average EC Packet delivery ratio	Python	Log-distance	Yes	Yes	Yes	MAB-based

DC: Duty cycle, DT: Downlink Traffic, ISFO: Imperfect SF Orthogonality

LoRaWANSim seems suitable but has several defaults:

- ADR is pre-computed and not dynamic during the simulation
- All UL then all DL are processed => Performance overestimation for dynamic ADR

III. LoRa Network Simulator

~~Who~~ What is Julia ? [7]

Julia is a recent language launched in 2012, and the version 1.0 introduced in 2018, with following features:

- Compiled language, performance should be close to C/C++ performance
- Python-like syntax => allows short development time
- Multiple-dispatch => A generic function is dynamically dispatched based on the runtime type of its argument

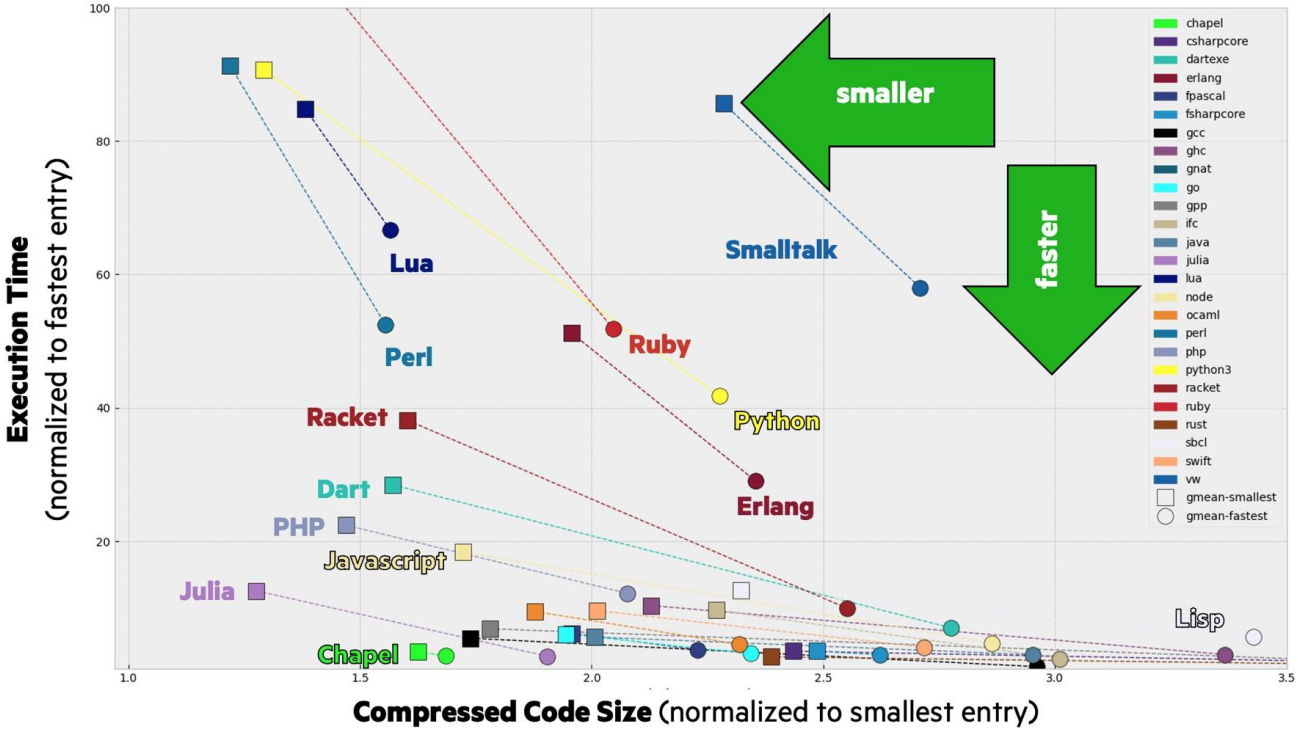
```
julia> f(a::Int64, b::Int64) = a + b
f (generic function with 1 method)

julia> f(a::Int64, b::Float64) = a * b
f (generic function with 2 methods)

julia> f(12,8)
20

julia> f(12,8.0)
96.0
```

III. LoRa Network Simulator: Julia Benchmark



Source from Chapel language: https://salsa.debian.org/benchmarksgame-team/benchmarksgame

III. LoRa Network Simulator: Input and Output

Input

Network topology:
- Number of GWs
- Number of EDs
- EDs/GWs positions

Channel configuration:
- Attenuation model
- Shadowing

Application requirement:
- Payload size
- Preamble size
- Packet generation rate

Node/GW config:
- Receiver sensitivity
- Capture-effect threshold
- ADR algorithm
- Power consumption model

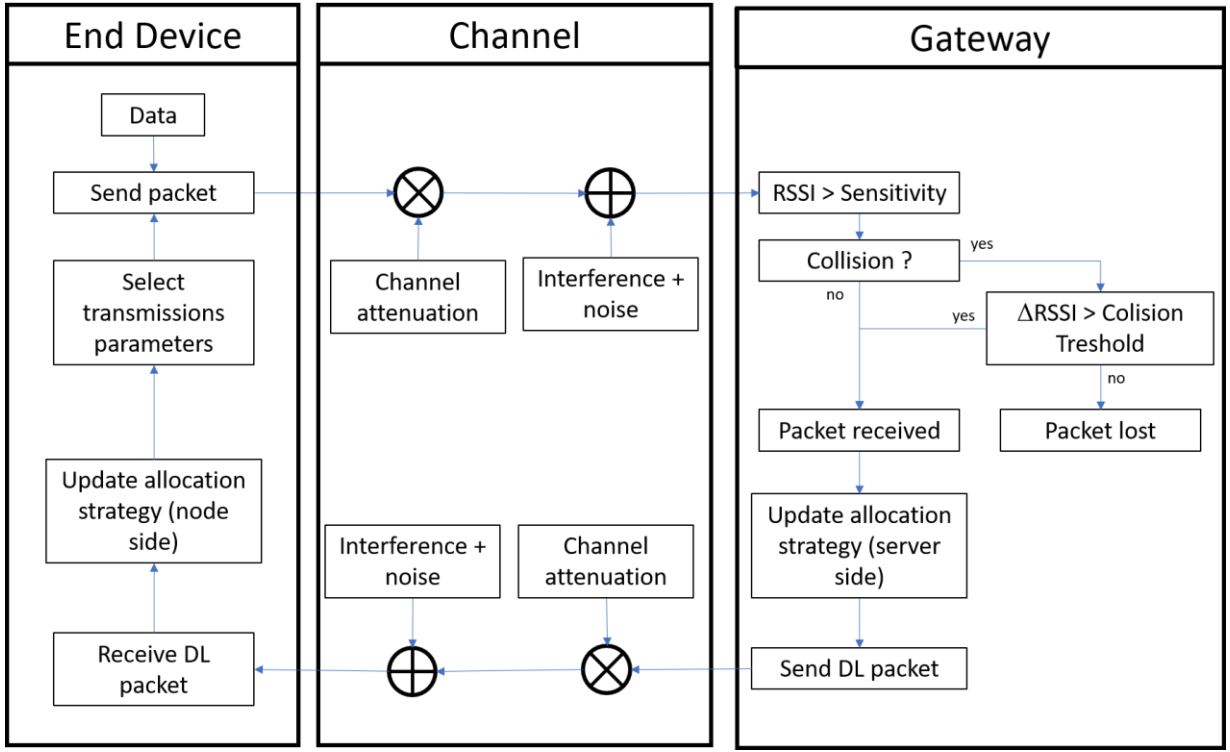
Output

Packet delivery ratio
- UL PDR
- PDR on RW1 and RW2
- Network response rate

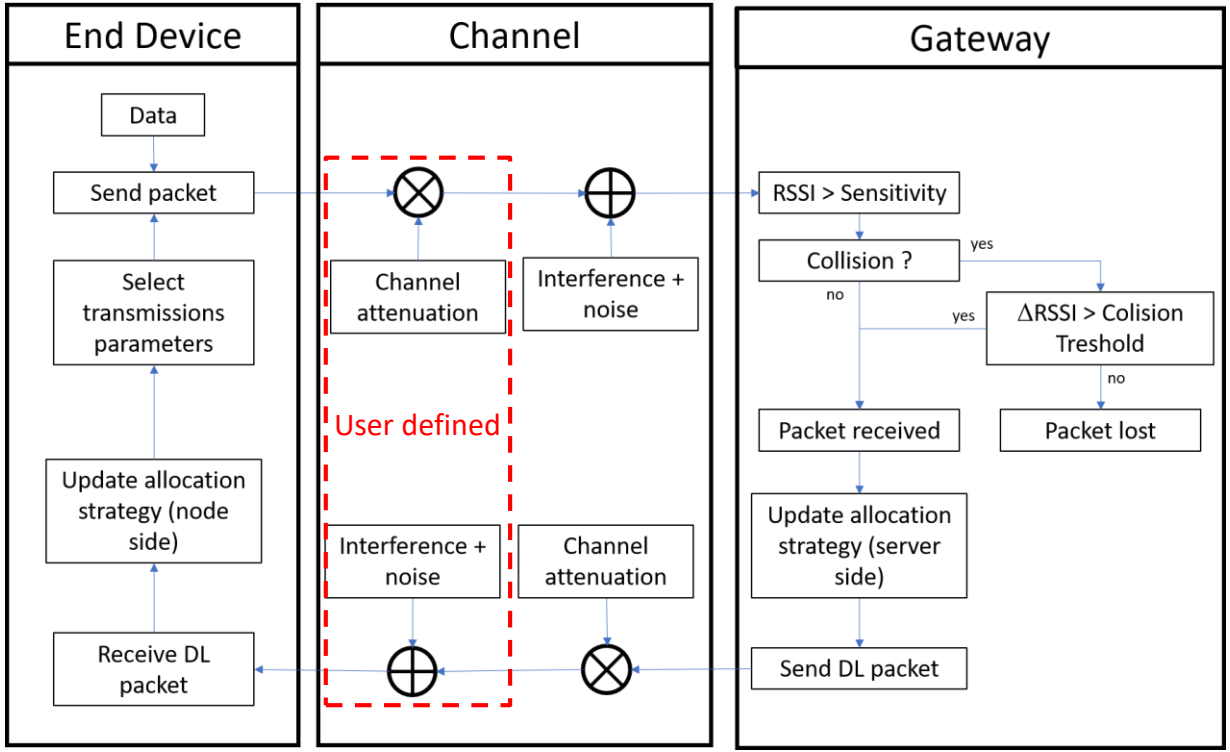
Energy consumption
- per nodes
- In average
- Max/min

As is the configuration is made through Julia initialization function. The goals is to read the configuration in a JSON file

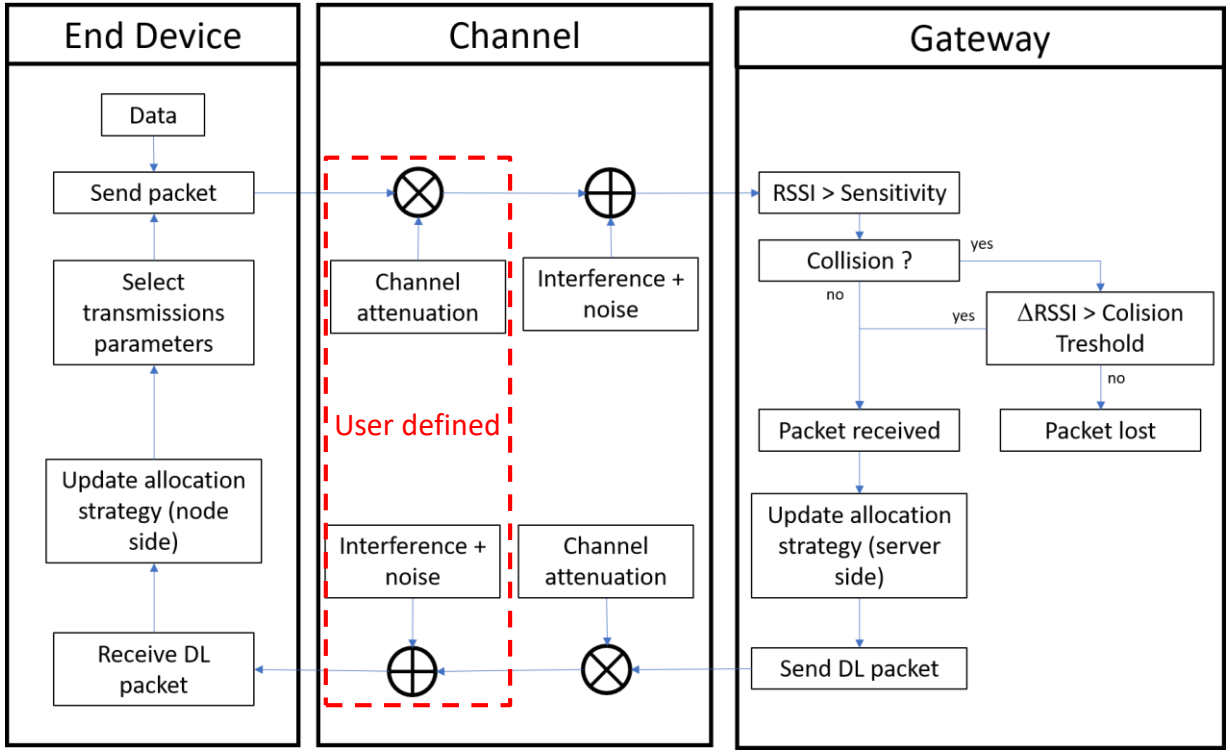
III. LoRa Network Simulator: GW and ED behavior



III. LoRa Network Simulator: GW and ED behavior



III. LoRa Network Simulator: GW and ED behavior

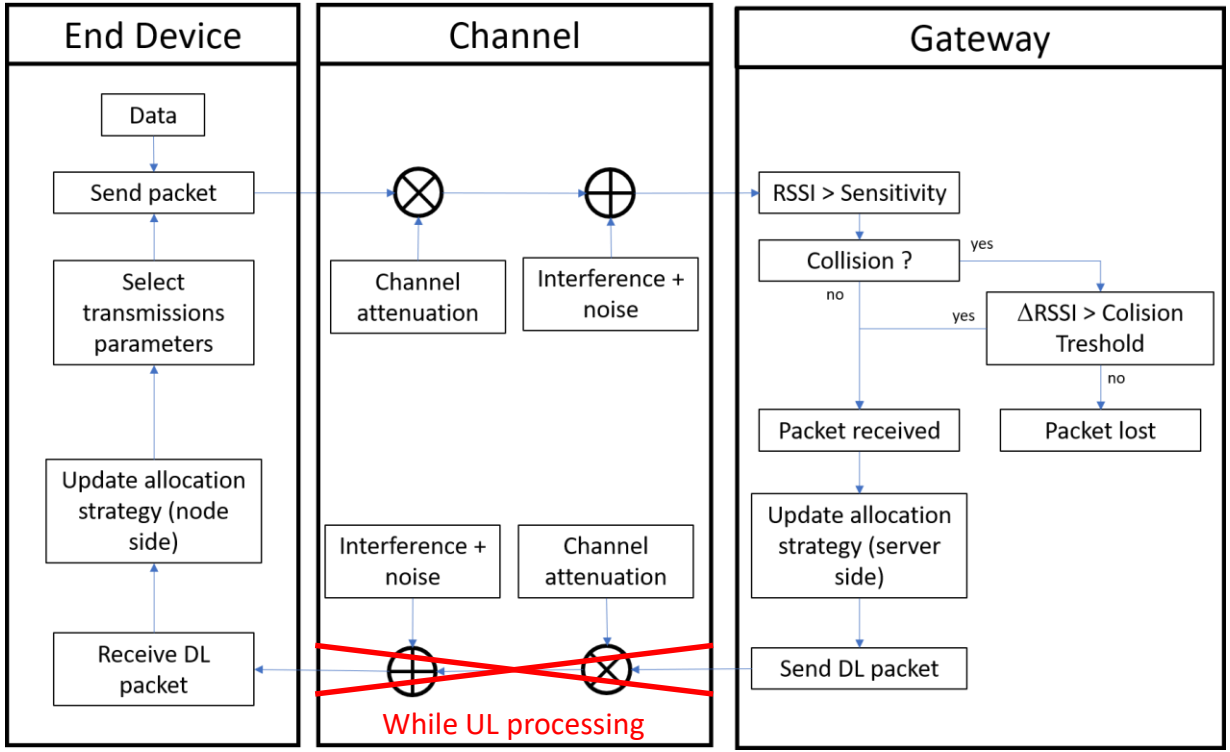


$$RSSI = TP - Att. - shadowing + G_{ant.}$$
$$S = -174 + 10\log_{10} BW + NF + SNR_{min}$$

Collision threshold (dB)[4]:

	SF7	SF8	SF9	SF10	SF11	SF12
SF7	1	−8	−9	−9	−9	−9
SF8	−11	1	−11	−12	−13	−13
SF8	−15	−13	1	−13	−14	−15
SF10	−19	−18	−17	1	−17	−18
SF11	−22	−22	−21	−20	1	−20
SF12	−25	−25	−25	−24	−23	1

III. LoRa Network Simulator: GW and ED behavior

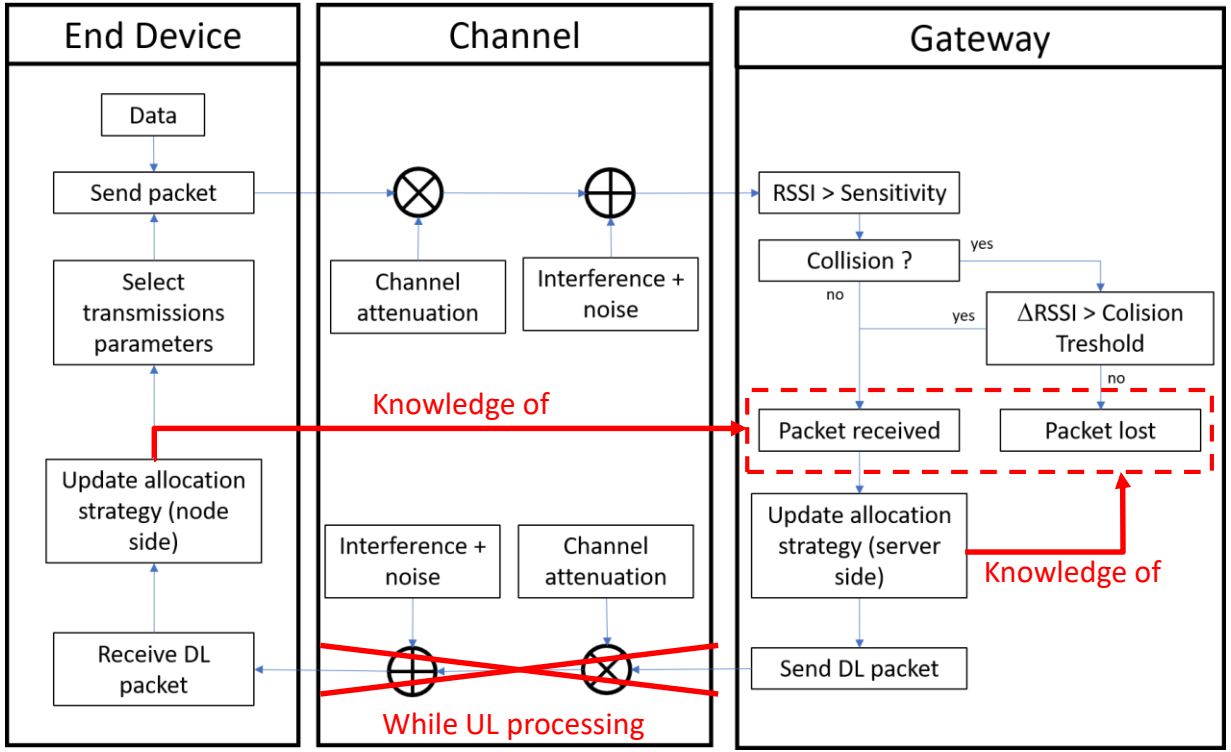


$$RSSI = TP - Att. - shadowing + G_{ant.}$$
$$S = -174 + 10\log_{10} BW + NF + SNR_{min}$$

Collision threshold (dB)[4]:

	SF7	SF8	SF9	SF10	SF11	SF12
SF7	1	−8	−9	−9	−9	−9
SF8	−11	1	−11	−12	−13	−13
SF8	−15	−13	1	−13	−14	−15
SF10	−19	−18	−17	1	−17	−18
SF11	−22	−22	−21	−20	1	−20
SF12	−25	−25	−25	−24	−23	1

III. LoRa Network Simulator: GW and ED behavior



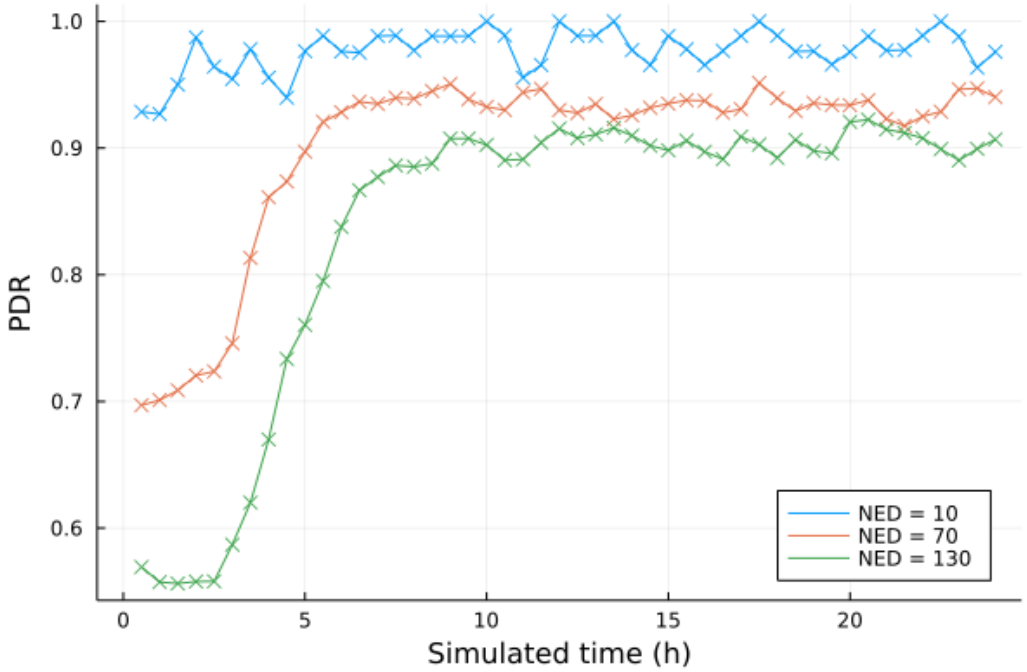
$$RSSI = TP - Att. - shadowing + G_{ant.}$$
$$S = -174 + 10\log_{10} BW + NF + SNR_{min}$$

Collision threshold (dB)[4]:

	SF7	SF8	SF9	SF10	SF11	SF12	
	1	−8	−9	−9	−9	−9	SF7
	−11	1	−11	−12	−13	−13	SF8
	−15	−13	1	−13	−14	−15	SF8
	−19	−18	−17	1	−17	−18	SF10
	−22	−22	−21	−20	1	−20	SF11
	−25	−25	−25	−24	−23	1	SF12

III. LoRa Network Simulator

First results: PDR vs. time



Simulation setup:

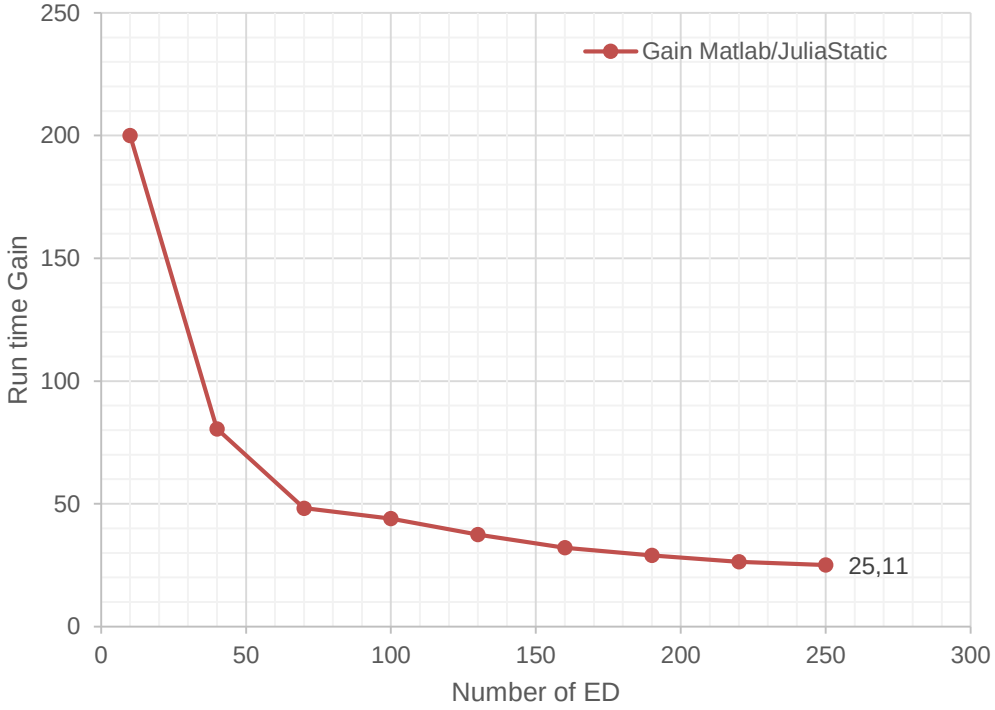
- Number of GW = 1
- Number of ED = 10, 70 or 130
- Simulated time = 24h
- Network radius = 10km
- Packet rate: 1 every 10 min
- Channel: Okumura-Hata with 3dB shadowing
- Payload: 51 B in UL and 20 B in DL

$$PDR = \frac{\#\ Packet\ received}{\#\ Packet\ sent}$$

- Convergence time
- DC saturation ?

III. LoRa Network Simulator

Run time comparison with LoRaWANSim



Simulation setup:

- Number of GW = 1
- Number of ED = 1:30:250
- Simulated time = 24h
- Network radius = 10km
- Packet rate: 1 every 10 min
- Channel: Okumura-Hata with 3dB shadowing
- Payload: 51 B in UL and 20 B in DL
- ADR is static and pre-computed in this comparison

$$\text{Gain} = \frac{\text{Matlab runtime}}{\text{Julia runtime}}$$

- For 100 ED :
  - Julia runtime is 5.8 s
  - Matlab runtime is 255 s

IV. Conclusions and future works

1) Conclusion

- J-LoRaNeS is faster than LoRaWANSim
- J-LoRaNeS can manage ISFO, capture effect, duty-cycle and DL traffic.
- J-LoRaNeS is able to emulate a Gaussian noise environment with a customized channel attenuation model
- We can run dynamic ADR → RL methods can be easily implemented
- PDR delivery and ADR convergence can be monitored

2) Future works

- Add power consumption metrics
- Merge UL and DL processing
- Add Impulsive noise in the simulator
- Explore RL algorithms

References

[1]: J. Courjault & al., *How robust is a LoRa communication against impulsive noise ?*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2020

[2]: LoRa Alliance, *LoRaWAN specification v1.1*, 2017 https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/

[3]: M. Slabicki & al., *Adaptive Configuration of LoRa Networks for Dense IoT Deployments.* IEEE/IFIP Network Operations and Management Symposium (NOMS), 2018.

[4]: R. Marini & al., *LoRaWANSim: A flexible simulator for LoRaWAN Networks.* Sensors, 2021

[5]: M. Bor & al., *Do LoRa Low-Power Wide-Area Networks Scale?.* ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2016.

[6]: D.-T. Ta & al., *LoRa-MAB: Toward an Intelligent Resource Allocation Approach for LoRaWAN.* IEEE Global Communications Conference (GLOBECOM), 2019

[7]: https://julialang.org/

Thanks for listening

CG-WIRELESS
Innovative Radio Frequency Solutions

eit Digital

www.irisa.fr @irisa_lab

Institut de Recherche en Informatique et Systèmes Aléatoires

I. Thesis objectives

2. Adaptive transmission scheme based on Reinforcement Learning

- Learns environment characteristics (noise nature, fading,…)
- Tune communication parameters to achieve required Quality of Service while minimizing power consumption:
  - Existing example: Adaptive Data Rate algorithm for LoRaWAN but converge slowly
  - Theoretical performance knowledge might help for parameters tuning

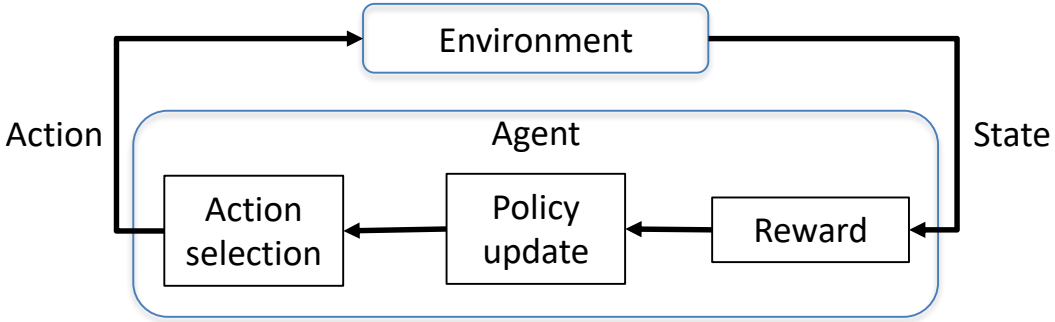


Figure 3: Reinforcement learning principles.

II. LoRa principle: LoRaWAN ADR on node side