T3.1: Automating IoT device configuration. 6th July 2021

Renzo E. NAVAS (renzo.navas@inria.fr)

WP3: Virtualization and automation of IoT access networks.

Goals of Meeting

- **Highlight work in T3.1: we have (preliminary) results!**
- Inform of my depart :’/ : Transition

Recap T3.1: Goal

Description:

- Having a large number of IoT nodes to learn what parameters to use (time, power, spreading factor, etc.) for uploading data, and optimizing their global performance.
- Focus on mechanisms where IoT nodes make their own decisions (decentralized)
- But, end devices may have their strategy optimized globally by the orchestrator.

Goal:

- Propose and analyze the performance of machine learning algorithms that need few resources, like multi-armed bandit methods.
- **D3.1: Lightweight learning algorithms for massive IoT and analysis of their performance. (T15)**

Recap T3.1: Roadmap

Baseline: Do "better" than LoRaWAN's ADR –and SoA–, in massive IoT scenarios.

1. Related Work (Positioning, Survey. E.g. [1]) [Non-priority for D3.1]
2. **Proposal**: Reinforcement-Learning-based, particularly **Bandit Algorithms [2]**
3. **Evaluation**: We will use a realistic setting/evaluation scenario, NS-3 based [URL].
4. **Contribution:** A differentiating factor of our Bandit-based Algorithm(s) will be this applicability/evaluation in realistic LoRaWAN scenarios.
   - In the literature, proposals use strong hypotheses or simplified models.
   - However, evaluation/comparison against non-bandit proposals will be a challenge (i.e., implementation)

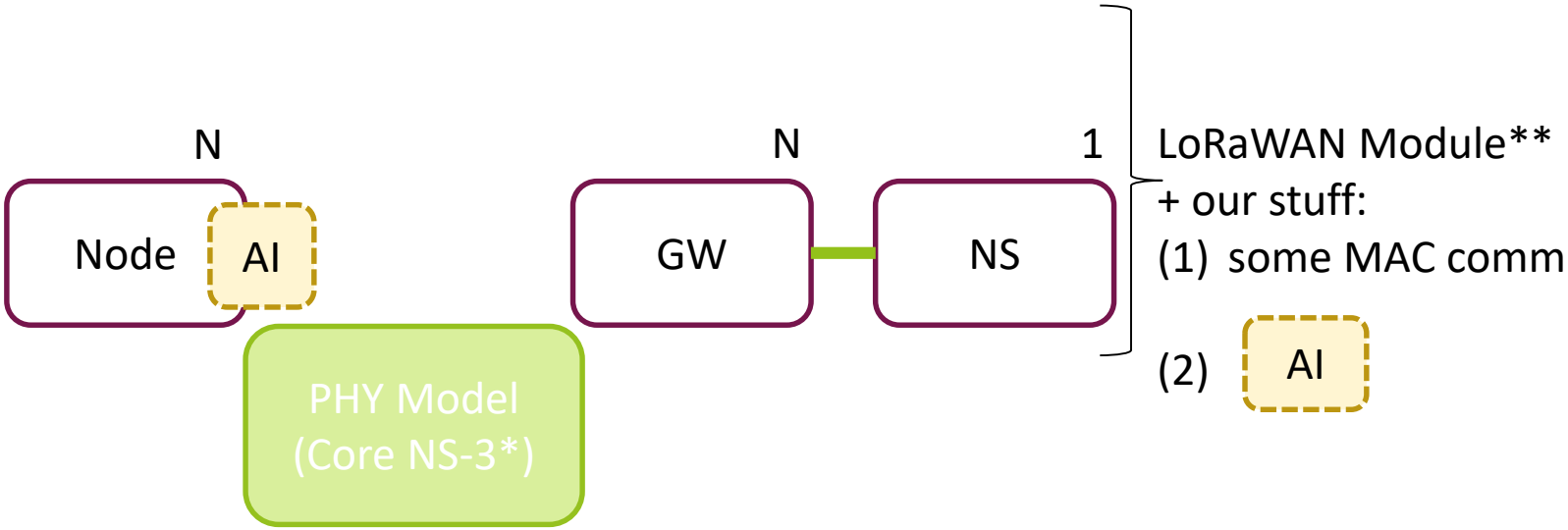
Outline

01. Software Components
02. LoRAWAN Bandit Rewards
03. First Experimental Results
04. Wrap Up and Future

01

Software Components

Software Components



*Core NS-3: A Discrete Event Simulator (in C++)

**LoRaWAN NS3 Module: https://github.com/signetlabdei/lorawan

AI-kernel: https://github.com/Svalorzen/AI-Toolbox

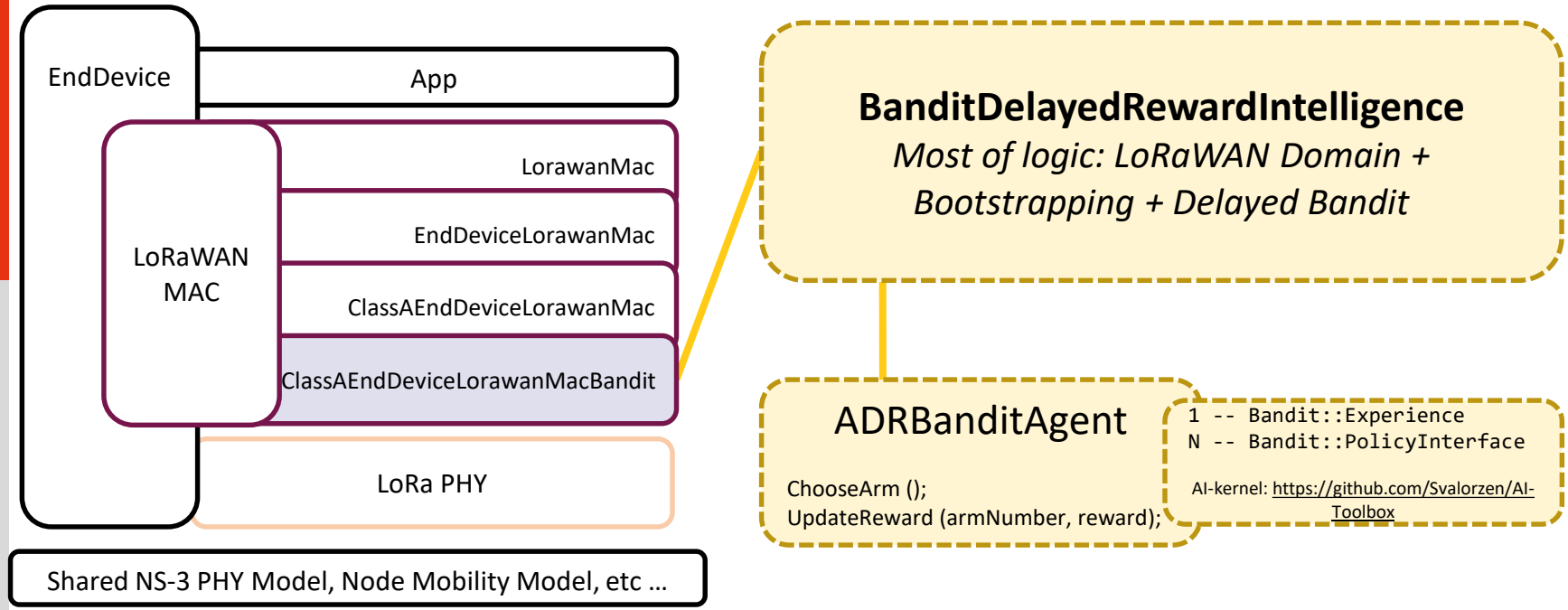
Software Components

- Code shared on Inria GitLab:
  - https://gitlab.inria.fr/intelligentsia/LoRaWAN-Bandits
  - We can give you access!
  - TODO: Clean, Document, etc… Code is preliminary/rough but “hardest” part is done

- Custom LoRaWAN MAC Command (0xBB) for Bandit Rewards:
  - BanditRewardReq / BanditRewardAns
  - … More on this later

Software Components: Node

AI

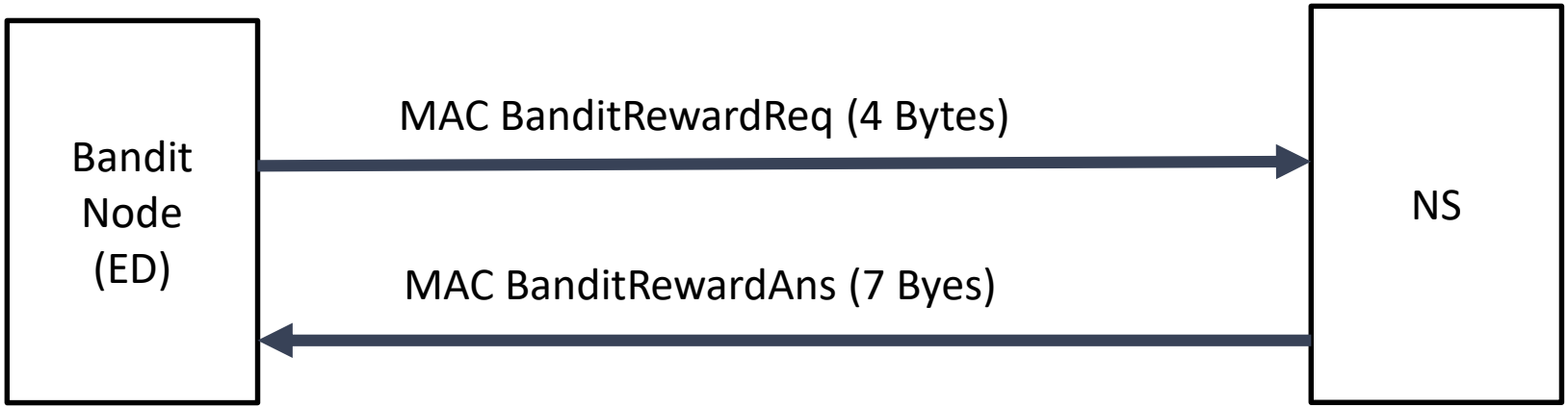


02

LoRAWAN Bandit Rewards

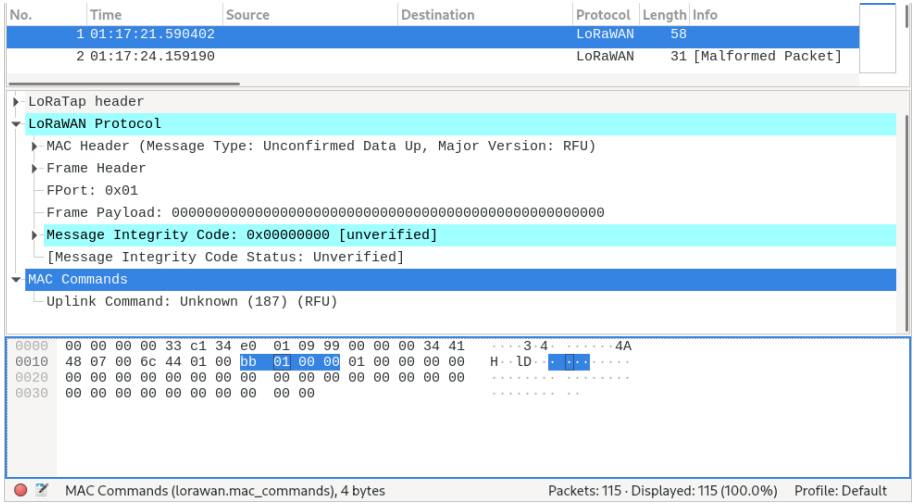
Delayed Reward (or Feedback)

(Remember : 12 Bytes is the maximum to be able to piggyback a MAC command)



[ED→NS] MAC BanditRewardReq

Size (Bytes)	1	2	1
Payload	CID (0xBB)	Max FCnt	Delta



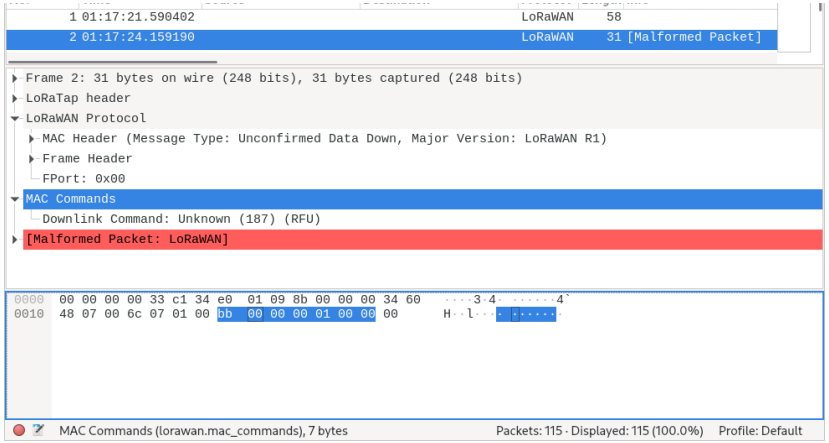
Eg: [0xBB , 0x01 0x00, 0x00]

FCnt=1, Delta = 0

“Send me stats for frame #1 to frame #1”

[ED←NS] MAC BanditRewardAns

Size (Bytes)	1	1	1	1	1	1	1
Payload	CID (0xBB)	#Pkt_RCV DR0 (SF12)	#Pkt_RCV DR1 (SF11)	#Pkt_RCV DR2 (SF10)	#Pkt_RCV DR3 (SF9)	#Pkt_RCV DR4 (SF8)	#Pkt_RCV DR5 (SF7)



Eg: [0xBB,00,00,00,01,00,00]
"1 packet received with SF9,
0 for other SFs"

(Frame# and Delta are not present)

Feedback Frequency

1) Ask for Reward at every Frame until Frame #10

2) Then, we ask for reward every N frames (deterministic!)

- There is room for improvement/ strategy implementations here ☺ (E.g., Probabilistic, Adaptive)
- Notion of **Convergence** is interesting to discuss/have in account (E.g., value and speed)
- Warning!
  - Everybody asking for reward at the same time affects the medium
  - Asymmetric UL/DL links:
    - RX sensitivity is different for ED (SX1276) and GW (SX1301):
    - UL may be ok, but we can not get the DL message using the same SF (Bandit will Converge slow)

MAC Command: High-level comments

- Rewards Request and Answer: “Hardcoded” … for now ?
  - Feedback only for Packets Received per SF (e.g., interesting to discriminate per Frequency)
  - If we want more: we need a more flexible/complex MAC command(s)
  - **I foresee research material on this topic (Compression, Frequency of Request, Stats Flexibility)**

- NS is agnostic to the notion of Bandit Rewards!
  - Good property!
    - Each Node optimizes whatever they want with the raw-data stats.
  - But… we can envisage a solution with Reward-Awareness @ NS
    - More NS complexity, but can be used for Slicing purposes! (NS has global visibility) [Beware of Ethical considerations!]

LoRAWAN Rewards (Optimizing@ED)

Arm Reward = (per packet sent)

- Table value* *if packet received*
- 0 *otherwise*

Objective [What] \ Arm [How]	SF12	SF11	SF10	SF9	SF8	SF7
Raw PDR	1	1	1	1	1	1
Energy-aware	1	2	4	8	16	32
?						

Final Remaks

Three fundamental Design Decisions:

1. Definition of Reward
   - > **Key** factor, and not as trivial as it seems (… reward ≠ 1-cost ≠ 1/cost !!)
2. Frequency of Reward MAC Command
   - > Related to Convergence speed (and primordial for mobility Use Case in future)
3. Bandit Arm's Rewards "Internal Bootstrapping"
   - > a.k.a. heuristics for starting "cold" arms (Currently: two fake samples per arm, reward 0 and 1)
   - > … may affects Convergence/Exploration!

…. Experimental results will reflect these decisions :) (Iterative "tunning")

03

First Experimental Results

Simulation Parameters

Cell Radius	6400 m (12900 ha)
Number of End-Devices	1000 [x,y:Uniformly Dist, z=1.2m]
GWs	1 [@(x=0 , y=0, y=14m)]
App Packet Size	32 Bytes (41/45B Raw)
Time between packets (min)	20min [Initial Delay = U(0,20)]
Simulation Time	33h20m (100 Packets)
TX Power	14 dBm (Max)
Frequency Carrier	U.Random in {868.1, 868.3 , 868.5}
Bandwidth	125 kHz

Propagation Model

Propagation Loss Model (Loss [dB])	LogDistancePropagationLossModel: Loss [dB] = 120.5 + 10 ∗ 3.76 ∗ log10 (R [km]) + RandomPropagationLossModel [0 , 10]
Building Penetration Loss	(Disabled for now)
Shadowing	(Disabled for now)

- We leverage in vanilla NS-3: very powerful PHY model (Models can be “stacked”)
- LoRa Interference Matrix:
  - Current is Gorsaud (2015) [1] https://github.com/signetlabdei/lorawan/blob/develop/model/lora-interference-helper.cc#L128
  - I did not use the “new” LoRa Colission Matrix (2018) [2]

Three Strategies

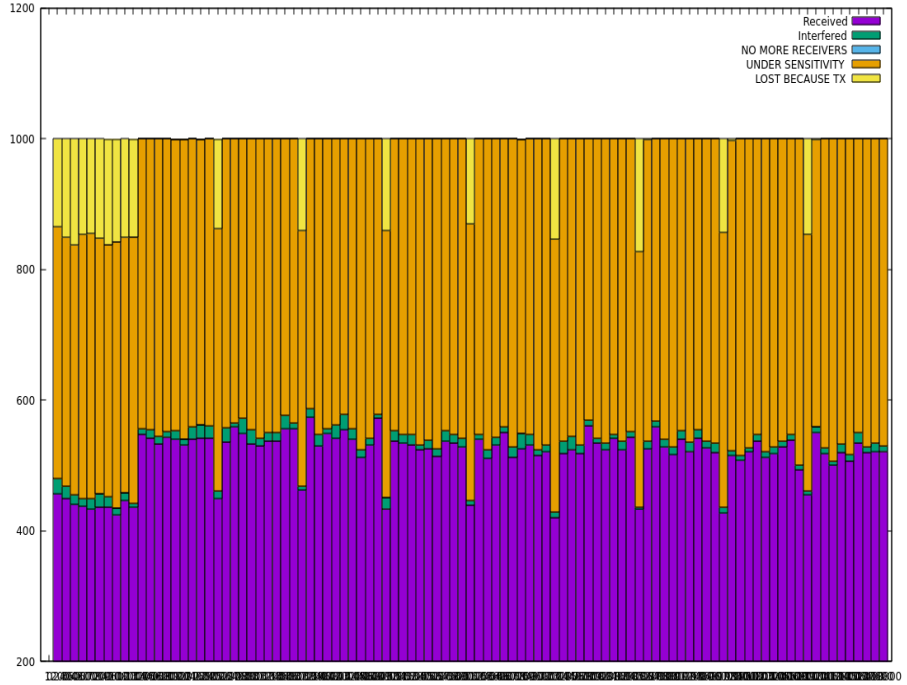
1. LoRaWAN's ADR "The Things Network" (TTN)
   - > All Nodes start in SF=12
   - > SF Down (NS-centric): Uses LinkADRReq/Ans MAC command, w/SNR Avg. of last 10 frames.*
   - > SF Up (EN-centric): LoRaWAN ADR Backoff. (Implem) → Negligible for our test
2. Bandit: Thompson Sampling → raw-PDR reward (all 1)
   - > Cold Arm bootstrap 0 and 1
3. Bandit: Thompson Sampling → Energy-aware reward
   - > A) Cold Arm bootstrap 0 and 1
   - > B) Cold Arm bootstrap 0 and 8

** Can also use Min or Max, It also takes in account all Gateways*

PDR convergence: Energy-aware (A)

T=0

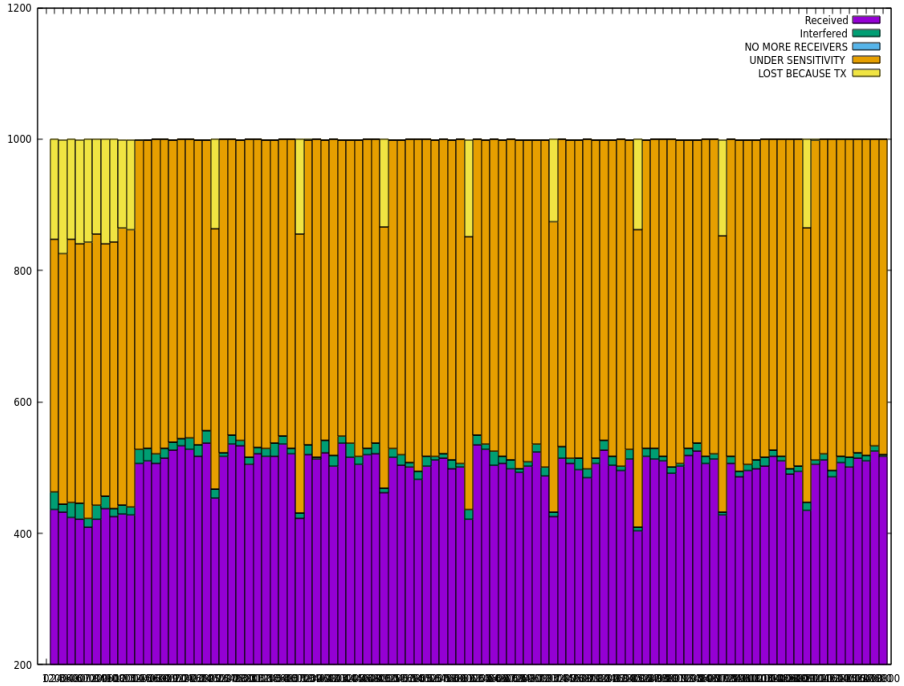
T=100



52.1%

PDR convergence: Energy-aware (B)

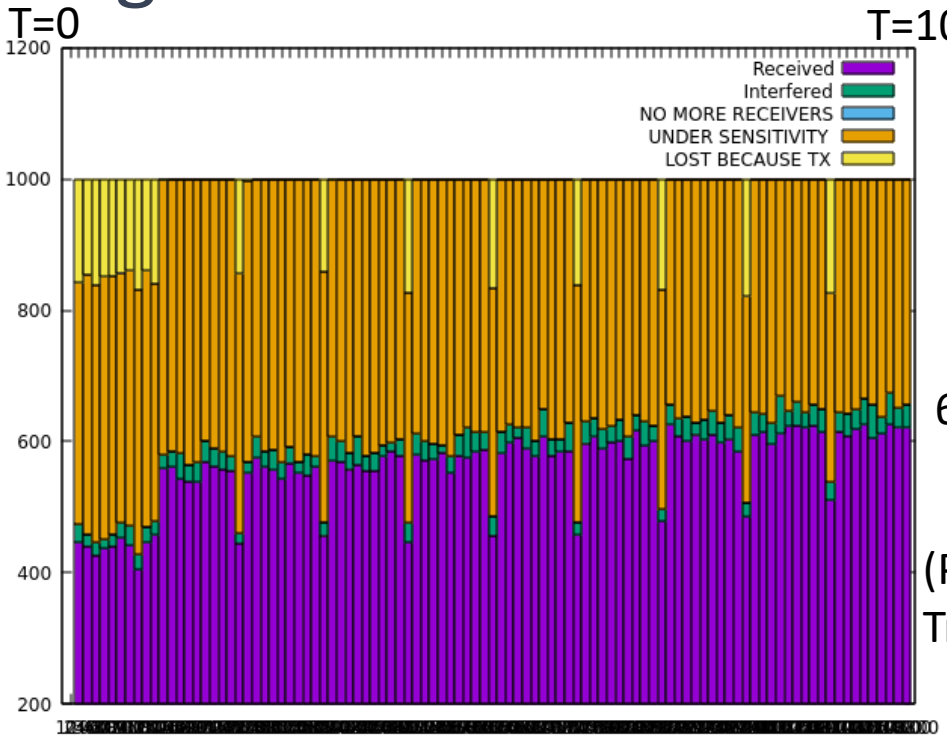
T=0 T=100



51.7%

(PDR seems converged)

PDR convergence: raw-PDR Bandit

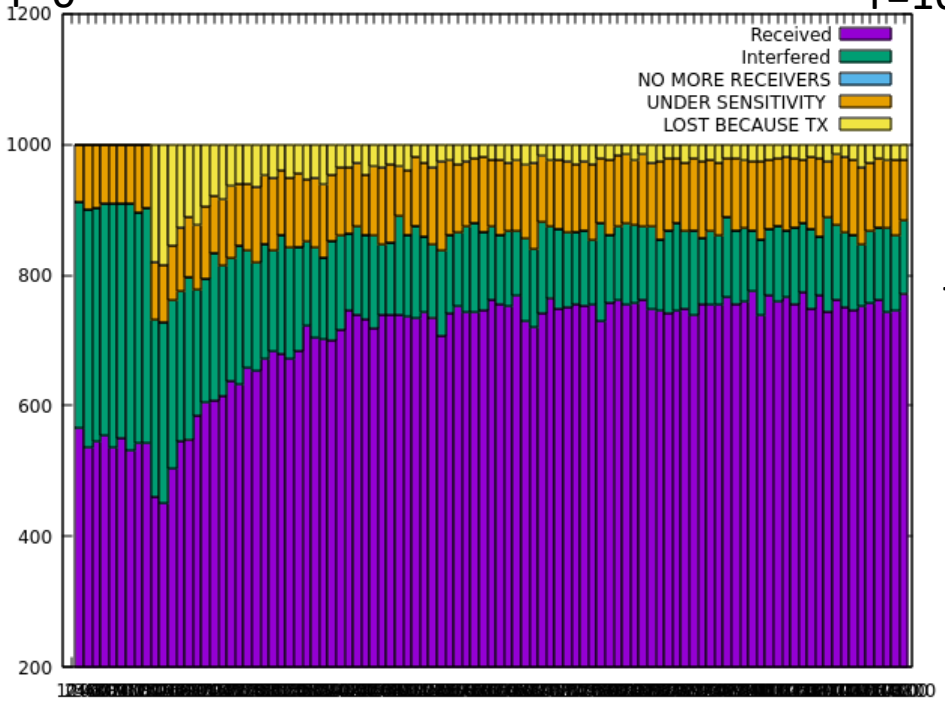


62.1%

(PDR “linear” Upper Trend not finished.. ?)

PDR convergence: ADR

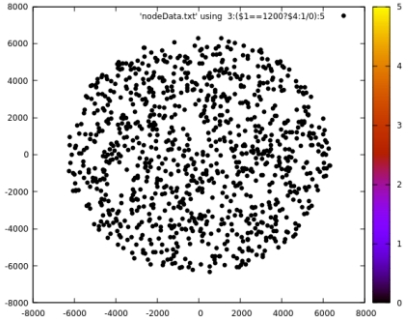
T=0 T=100



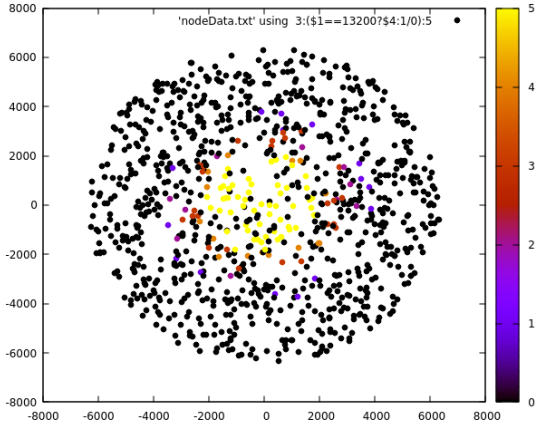
77.1%

(PDR seems converged)

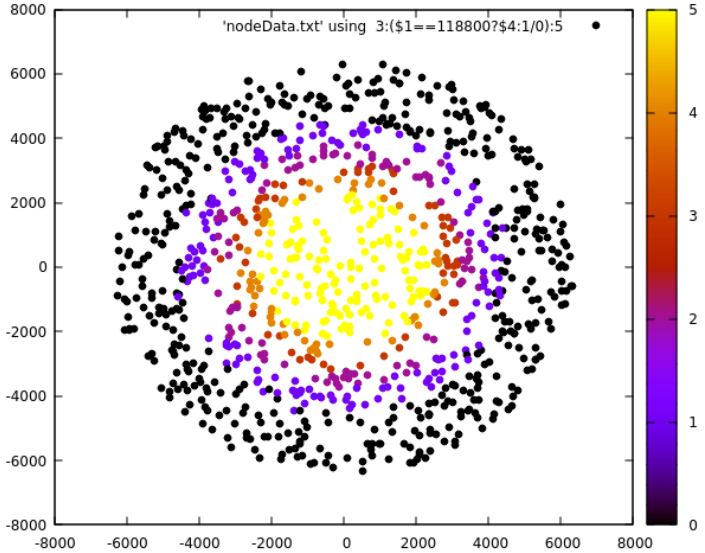
Data Rate Evolution: ADR



T=0

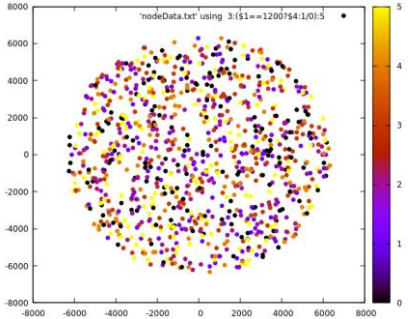


T=11

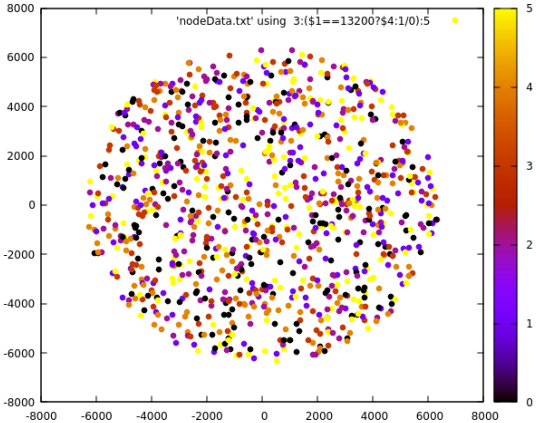


T=100

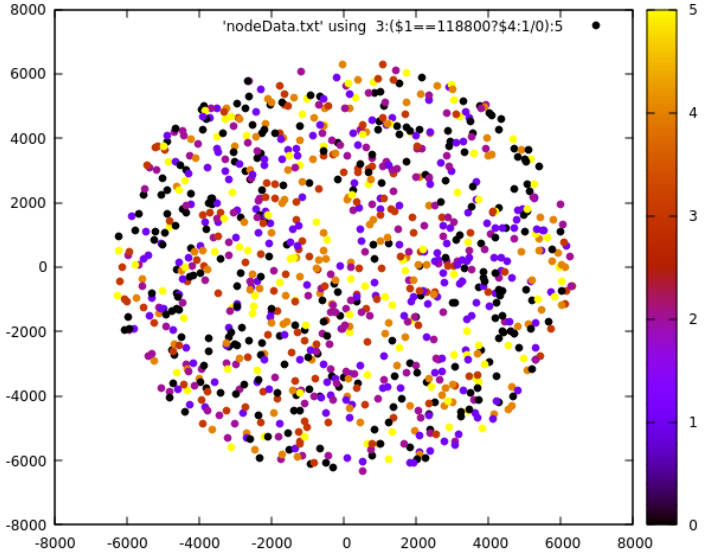
Data Rate Evolution: raw-PDR Bandit



T=0

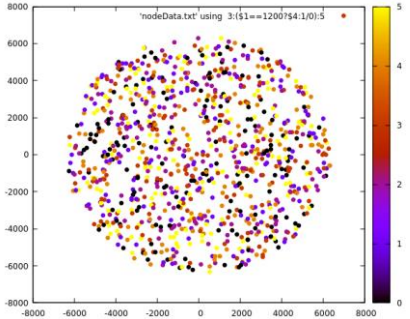


T=11

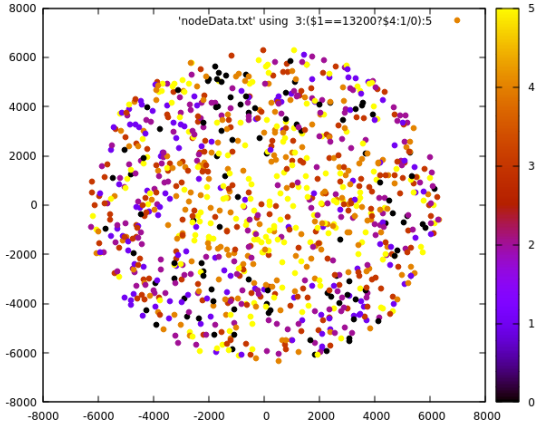


T=100

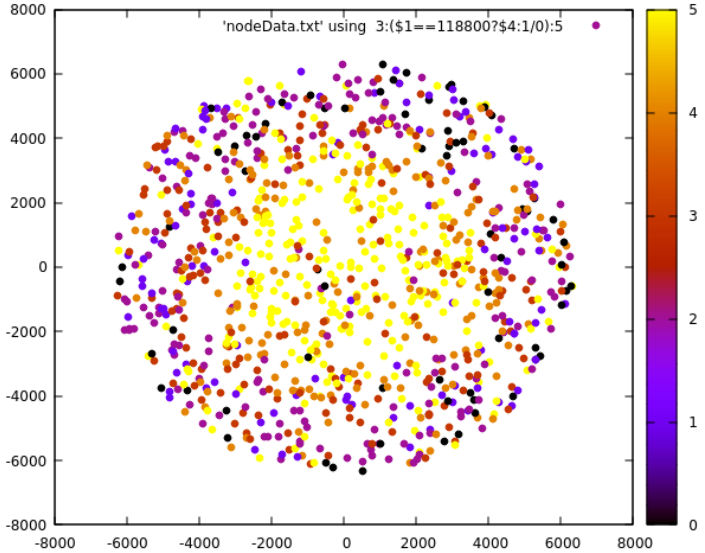
Data Rate Evolution: Energy-aware (A)



T=0

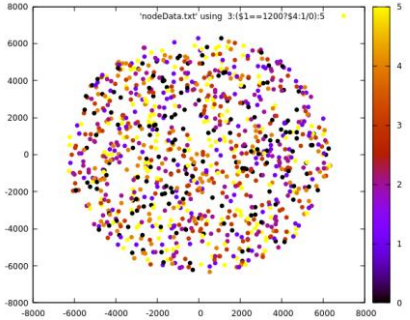


T=11

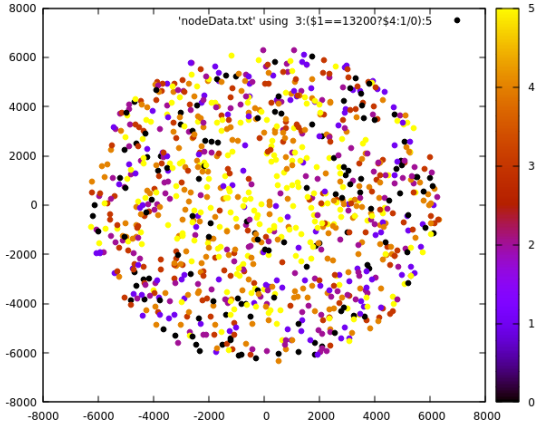


T=100

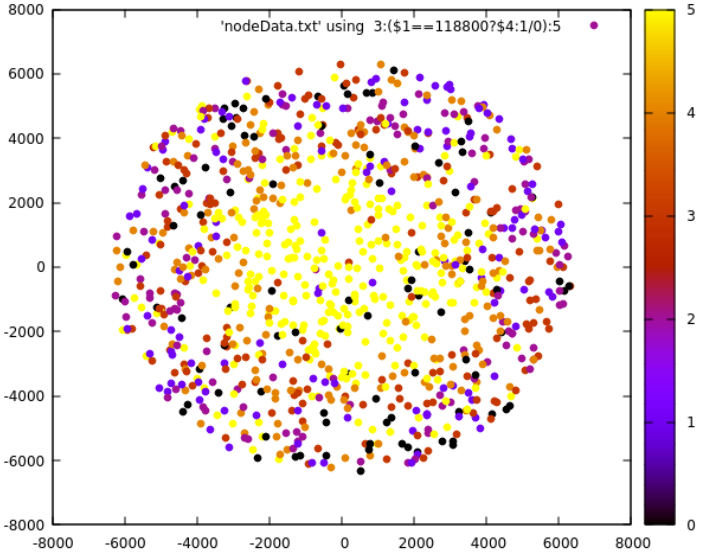
Data Rate Evolution: Energy-aware (B)



T=0



T=11



T=100

Pointers from Preliminary Results

- PDR-centric Reward
  - > See if current Reward Bandit converges to ADR (If it does: good!)
  - > Refine Reward to see if we can approach ADR PDR value (or increase speed) of convergence
- Energy-aware Rewards
  - > The most promising results: From PDR to Energy paradigm (already identif in [3])
  - > Opens up many LoRaWAN improvements: e.g., adaptive packet frequency
  - > **FOCUS ON ENERGY (+ Fairness)**
- Work on automatization of sim (and Metrics/Graphs)
  - > See [4] “A simulation execution manager for ns-3: Encouraging reproducibility and simplifying statistical analysis of ns-3 simulations.” (Davide, 2019)

[3] Kerkouche, Raouf, Réda Alami, Raphaël Féraud, Nadège Varsier, and Patrick Maillé. "Node-based optimization of LoRa transmissions with Multi-Armed Bandit algorithms.“ In 2018 25th International Conference on Telecommunications (ICT), pp. 521-526. IEEE, 2018.

04

Wrap-up and Future

Wrapping Up:

- A Functional Thompson S. Bandit in NS-3 LoRaWAN Simulation
  - https://gitlab.inria.fr/intelligentsia/LoRaWAN-Bandits
  - Scales to N nodes; not tested multiple GWs (could lead to bugs)
- Defined and Implemented the “delayed feedback” MAC command
- Energy-aware Rewards seem promising
- Identified areas to Improve:
  - Refine the “bootstrapping” (including **REWARDS** definition –fn(PDR,Energy)–)
  - Measure how Energy evolves, and notion of Fairness (do all nodes communicate?)
  - Statistics/visual representation

Future

- I am leaving 1 Sept :/
- Advance on Simulations, and tell a nice story:
  - Focus on energy, and that bandits do very well
  - Add UCB Bandits to mix
  - Bootstrapping and Heuristics
  - City Model (buildings), multiple GWs, etc.. Bandits probably will adapt well vs ADR
- Start writing Paper and D3.1, both will map almost 1 to 1.
  - I want to publish this paper ☺
- Clean the source code and document for my relay/replacement

Merci !

renzo.navas@inria.fr

Bibliography

[1] Goursaud, Claire, and Jean-Marie Gorce. "Dedicated networks for IoT: PHY/MAC state of the art and challenges." EAI endorsed transactions on Internet of Things (2015). https://hal.archives-ouvertes.fr/hal-01231221/

interferer desired	7	8	9	10	11	12
7	-6	16	18	19	19	20
8	24	-6	20	22	22	22
9	27	27	-6	23	25	25
10	30	30	30	-6	26	28
11	33	33	33	33	-6	29
12	36	36	36	36	36	-6

Table 1. Cochannel rejection (dB) for all combinations of spreading factor for the desired and interferer user

Bibliography

[2] Croce, Daniele, Michele Gucciardo, Stefano Mangione, Giuseppe Santaromita, and Ilenia Tinnirello. "Impact of LoRa imperfect orthogonality: Analysis of link-level performance." IEEE Communications Letters 22, no. 4 (2018): 796-799. https://doi.org/10.1109/LCOMM.2018.2797057

TABLE II

SIR THRESHOLDS WITH SX1272 TRANSCEIVER

SF_{ref} \ SF_{int}	7	8	9	10	11	12
7	1	-8	-9	-9	-9	-9
8	-11	1	-11	-12	-13	-13
9	-15	-13	1	-13	-14	-15
10	-19	-18	-17	1	-17	-18
11	-22	-22	-21	-20	1	-20
12	-25	-25	-25	-24	-23	1

Bibliography

[3] Kerkouche, Raouf, Réda Alami, Raphaël Féraud, Nadège Varsier, and Patrick Maillé. "Node-based optimization of LoRa transmissions with Multi-Armed Bandit algorithms." In 2018 25th International Conference on Telecommunications (ICT), pp. 521-526. IEEE, 2018.

[4] Magrin, Davide, Dizhi Zhou, and Michele Zorzi. "A simulation execution manager for ns-3: Encouraging reproducibility and simplifying statistical analysis of ns-3 simulations." Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. 2019.

Interesting Pubs (for Future Work…)

[Downlink] Capuzzo, Martina, Davide Magrin, and Andrea Zanella. "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures." In 2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), pp. 1-7. IEEE, 2018. **(Downlink issues and proposals, very interesting! Ex.: Invert RX2 and RX1 params)**

[Jamming] Bolivar, Ivan Marino Martinez. "Jamming on LoRaWAN Networks: from modelling to detection." PhD diss., Institut National des Sciences Appliquées de Rennes, 2021. https://tel.archives-ouvertes.fr/tel-03196484/document