

Journées LPWAN 2026

NC-LoRaSim: A No-Code LoRaWAN Simulation Framework for NS-3

Presented by : Ismail BENNIS

MCF, IRIMAS - Université de Haute-Alsace (UHA, France)

Ismael.bennis@uha.fr



2-3 juillet, Clermont-Ferrand

OUTLINE

1) Introduction and Context

- a) IoT & LoRaWAN
- b) Limitations of existing LoRaWAN simulators
- c) Motivation

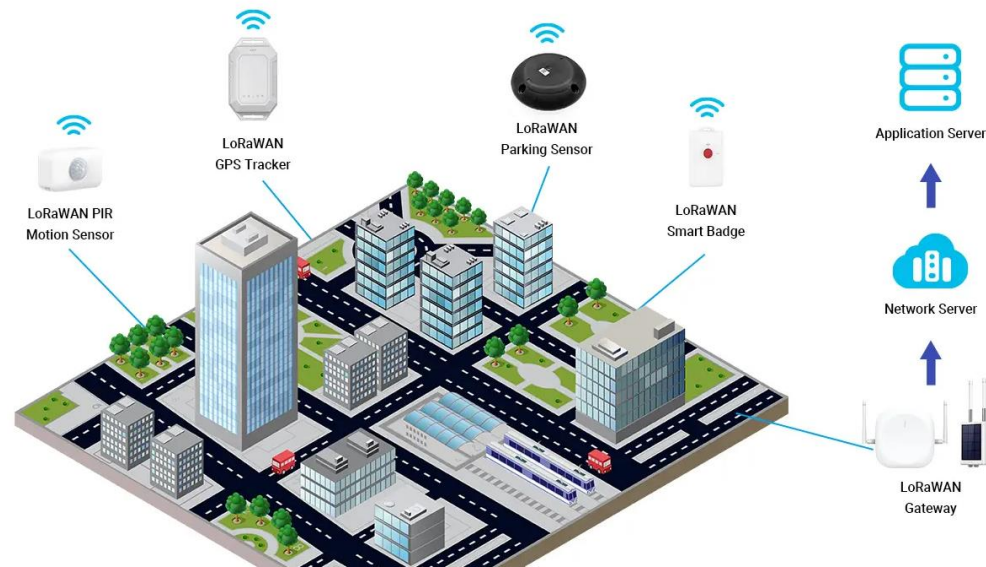
2) Proposed solution : NC-LoRaSim

- a) Architecture and design
- b) Operation modes:
 - i. Manual
 - ii. Automatic
 - iii. Advanced

3) Conclusion

Introduction

- ✓ The Internet of Things (**IoT**) is a reality today:
 - ✓ Smart cities
 - ✓ Smart farming
 - ✓ Smart metering
 - ✓ Environmental and Infrastructure Monitoring
 - ✓ Etc.



- ✓ A keystone solution: **LoRa/LoRaWAN** technology: enable long-range, low-power sensing and low cost.

Introduction

- ✓ Designing LoRa networks needs many parameter sweeps (SF, TP, GWs, placement, traffic load, propagation conditions, etc.) and extensive simulation.
- ✓ Existing tools trade off accessibility and fidelity:

Simulator	Strength	Limitation	Energy	Batch	GUI	Geo
LoRaSim	lightweight Python	simplified PHY/MAC	~	x	x	x
FloRa	OMNeT++ ecosystem	configuration complexity	✓	x	✓	x
LoRaWANSim	MATLAB modeling	proprietary environment	✓	x	x	x
ns-3 LoRaWAN	high fidelity	code-based workflow	~	x	x	x

- ✓ The issue is not only simulation accuracy, is how to make **systematic, reproducible, large-scale** LoRaWAN simulation **feasible** for more users.

Problem Statement & Objectives

✓ Gaps to fill:

- **Accessibility:** simulators require **programming** skills and lack a **user-friendly**, graphical interface to configure or visualize LoRaWAN scenarios.
- **Automation:** the batch execution and parameter sweeping must be **scripted manually**, with no integrated tools to automate or manage large experiment sets.
- **Statistical Rigor:** Simulators do not natively support **repeated runs**, **confidence interval computation**, or built-in **statistical validation** of results.
- **Visualization:** Current tools offer only **static plots** or raw log outputs, without interactive or **map-based visualization** of network behavior.
- **Geographic Realism:** Simulations rely on **abstract coordinates** and idealized **propagation**, with no integration of real-world maps or **obstacles**.

✓ Problem:

- ✓ How to fill these gaps?

Problem Statement & Objectives

✓ NC-LoraSim:

- ✓ Web platform (client-server-data architecture) → No-code, web UI for scenario design.
 - ✓ Three modes: Manual, Automatic (batch), Advanced (light C++ IDE).
 - ✓ OSM integration, multiple propagation models → Realistic environments.
 - ✓ Predefined protocols (LLNRM, GeoNet, ADR) + exportable .cc
 - ✓ Automated statistics with 95% CI of a built-in metrics.
 - ✓ Persistent results storage.
- ✓ M-A. H. Amor and I. Bennis, 23rd International Symposium on Network Computing and Applications (**NCA**), Lisbon, Portugal, 2025, pp. 39-42, <https://doi.org/10.1109/NCA67271.2025.00019>



Targeted Public and Use Cases

Research

- Rapid prototyping of new LoRaWAN protocols.
- Large-scale parameter sweeps.
- Reproducible experiments with confidence intervals.
- OSM-based realistic environments.

Industry

- Virtual testbed before real deployment.
- Coverage and capacity planning.
- QoS prediction (PDR, latency, energy).
- Gateway placement and network optimization.

Education

- Hands-on learning without NS-3 programming.
- Understanding SF, ToA, and propagation effects.
- Interactive visualization and experimentation.
- Progressive learning (Manual → Automatic → Advanced).

- ✓ **NC-LoRaSim** democratizes high-fidelity LoRaWAN simulation for researchers, engineers, and students while preserving scientific rigor and reproducibility.

System Architecture

✓ Client Layer:

- ✓ Provides an intuitive web interface using HTML, CSS, and JavaScript with frameworks (Bootstrap, Leaflet, Chart.js, and Three.js).
- ✓ Users can configure simulation parameters and select real map areas from OSM without programming.

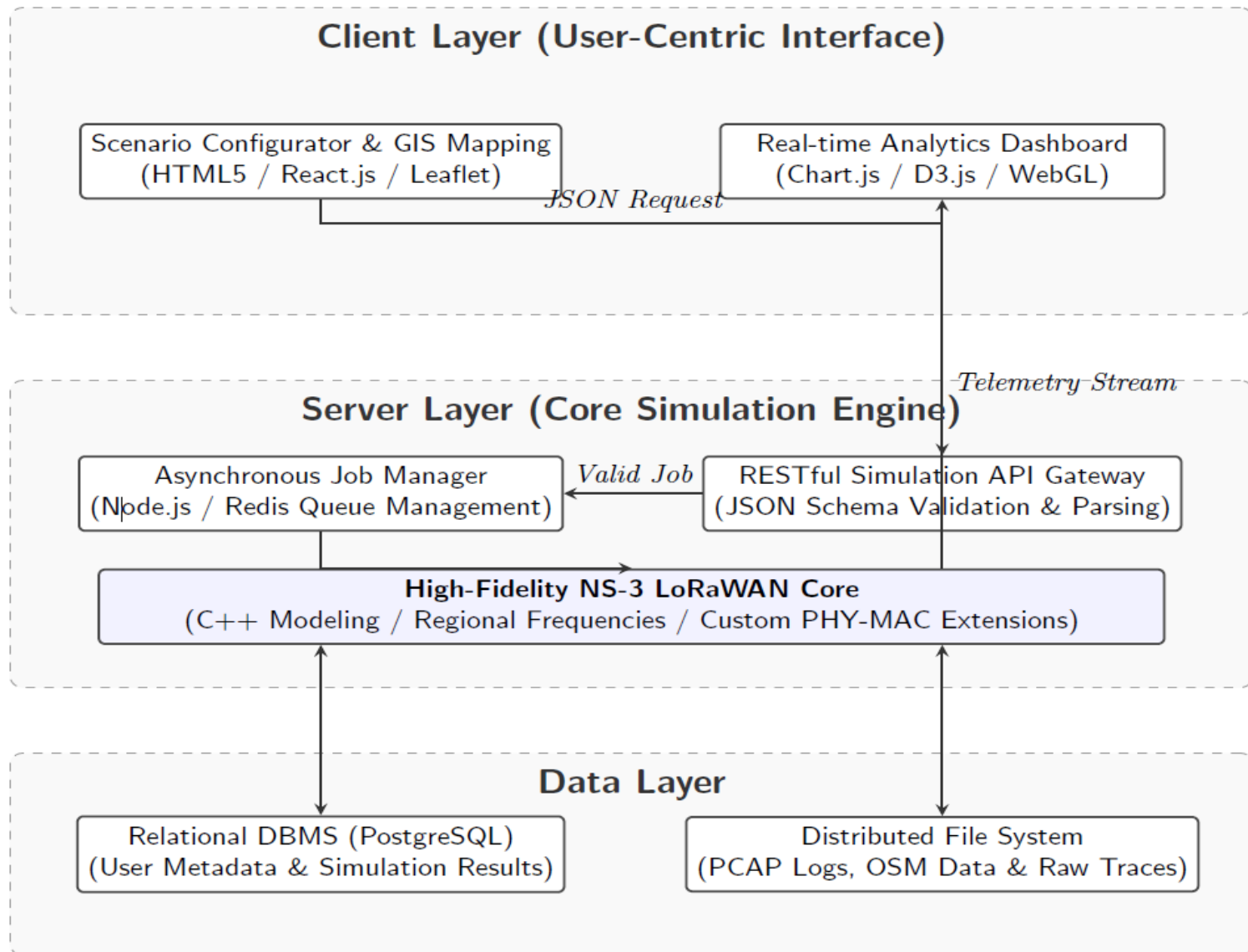
✓ Server Layer:

- ✓ Built on Node.js, it manages simulation execution, **translates** user inputs into **NS-3 commands**, and parses results.
- ✓ Custom NS-3 extensions in C++ and Python to add gateway placement, energy tracking, and realistic geographic modeling.

✓ Data Layer:

- ✓ Uses an SQL database to store configurations, results, and map data, ensuring **persistence**, **reproducibility**, and **reliable** experiment management.

System Architecture



System Setup

NC-LoRaSim Installation

Installation Directory :

Select installation folder...

Browse...

Start Installation

Installation Log:

```
[22:05:48] ✓ NC-LoRaSim application ready
[22:05:48] 1. Click "Browse" to select a folder
[22:05:48] 2. Click "Start Installation"
[22:05:48]
```

✓ includes an automated deployment framework, designed to reduce the configuration overhead

✓ main interface exposes three workflows

NC-LoraSim

Home

Login

Automatic Mode

Manual Mode

Advanced Mode

My Account

NC-LoraSim: LoRaWAN Simulation Platform

Simulation Settings

Choose the simulation mode that best fits your research and development needs.



Automatic Mode

Launch automatic evaluations across multiple network configurations.

- ✓ Sequential testing
- ✓ Comprehensive performance analysis
- ✓ Automatically generated reports

Launch Automatic Simulation



Manual Mode

Full control over all parameters for complex and customized scenarios.

- ✓ Advanced manual configuration
- ✓ Access to all LoRaWAN parameters
- ✓ Customizable scenarios

Launch Manual Simulation



Advanced Mode

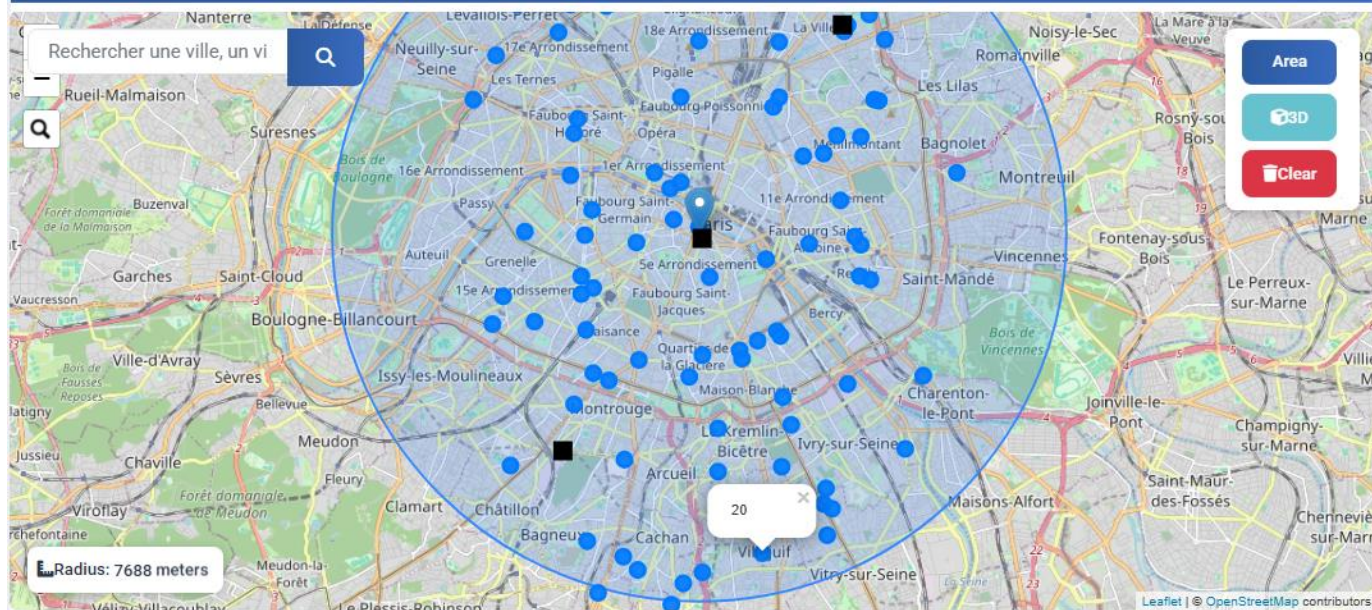
Search and analyze specific configurations from your previous simulations.

- ✓ Advanced search
- ✓ Results filtering
- ✓ Configuration comparison

Access Advanced Mode

NC-LoraSim: Manual Mode

Simulation Area Selection



Simulation Parameters

Sending Frequency (minutes)

Payload (bytes)

Number of Gateways

Number of Nodes

Simulation Radius (m)

Simulation Duration (minutes)

Number of Repetitions

Center Coordinates (Lat,Lng)

Propagation Model

Bandwidth (BW)

Coding Rate (CR)

☒ Include Buildings

☐ Manual Gateway Coordinates Configuration

☐ Manual Spreading Factors Configuration

☐ Manual Transmission Powers Configuration

Point-and-click
scenario
configuration
(nodes, gateways,
coords).

Radio parameters:
BW (125/250/500
kHz), CR
(4/5...4/8), SF, TP.

Propagation
models: free-
space, log-
distance,
shadowing,
Nakagami, hybrid.

Different output
levels: global
results / in-depth
analysis.

Manual Mode – Global Results

- ✓ Total number of transmitted and successfully received packets, the overall PDR, and the 95% CI.
- ✓ **Gateway-level** metrics report the number of packets collected by GW.
- ✓ **Node-level** metrics detail: PDR, energy consumption, spatial coordinates, SF, data rate, and ToA.

Simulation Summary

3000

Total Packets Sent

2121

Total Packets Received

70.7%

Packet Delivery Ratio (PDR)

65.6% – 75.8%

95% Confidence Interval

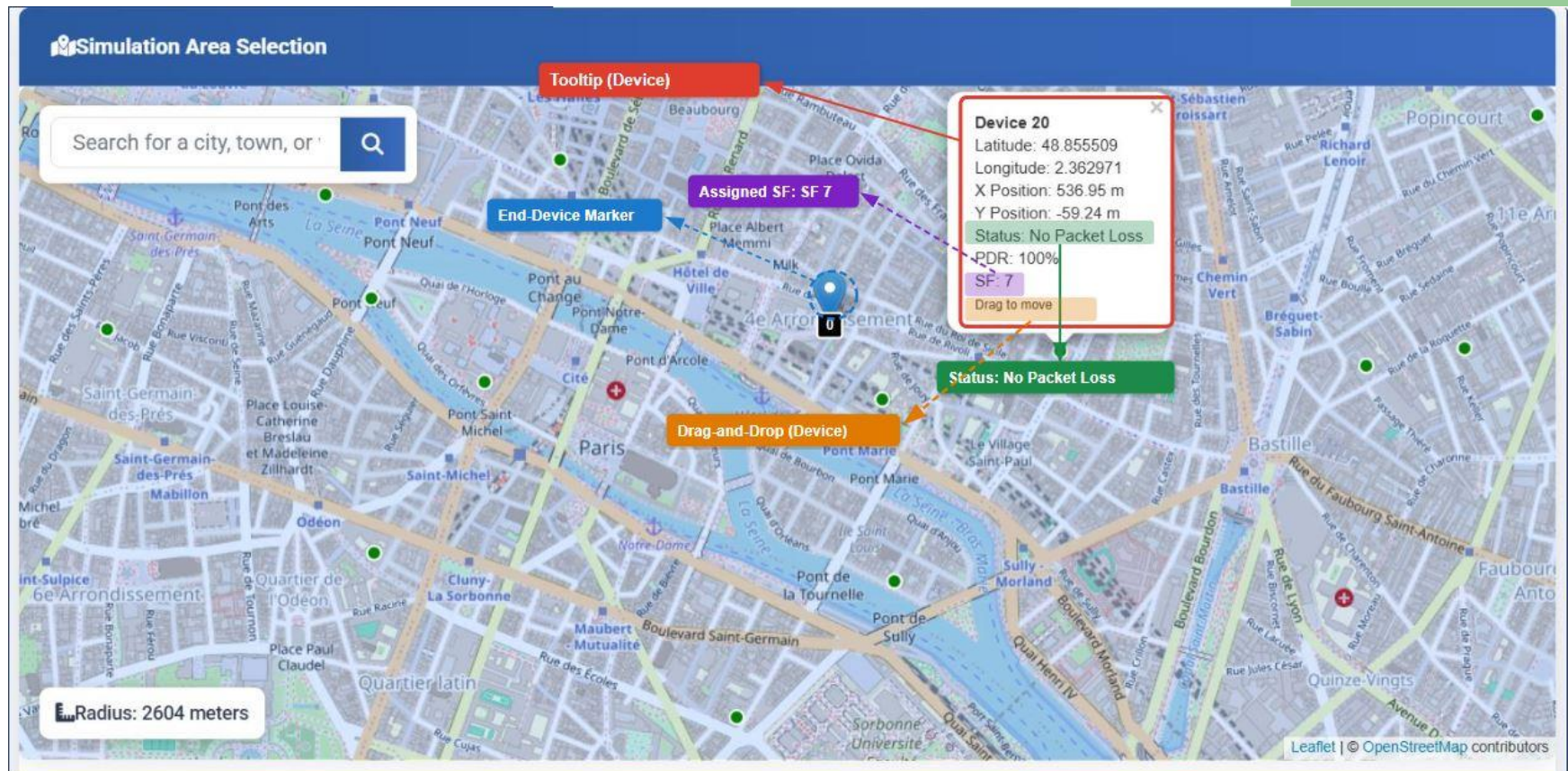
Packets Received per Gateway

Gateway ID	Packets Received	95% Confidence Interval (Packets)
GW0	900	851 – 949
GW1	750	704 – 796
GW2	650	606 – 694
GW3	500	460 – 540
GW4	400	363 – 437

Node Details

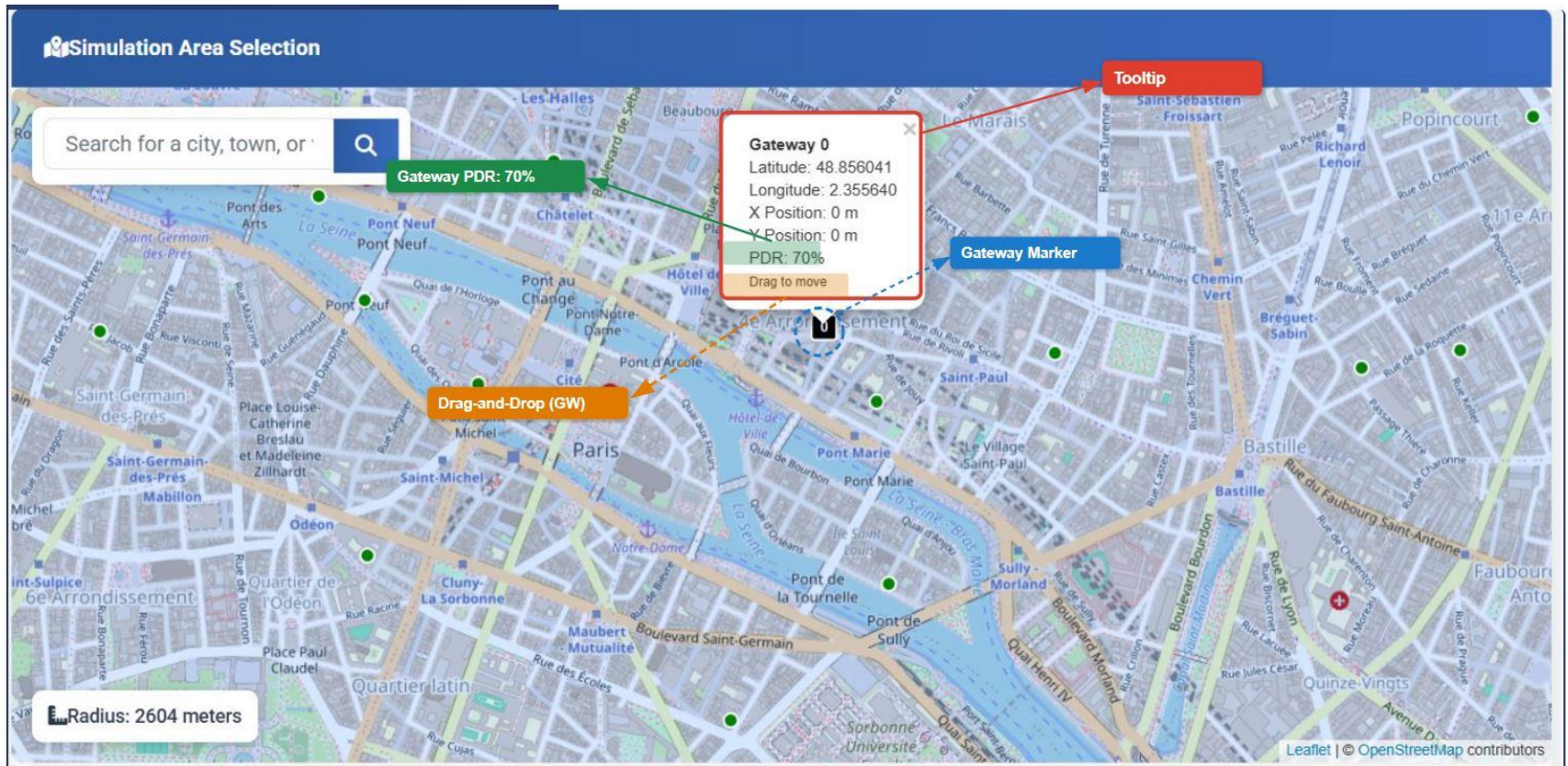
ID	X (m)	Y (m)	SF	Sent	Received	PDR (%)	ToA (ms)	Rate (kbps)	Energy (J)
1	500	7500	7	120	115	95.8%	56	2.86	0.60
2	1520	2000	9	130	120	92.3%	125	1.28	1.16
3	2322	1202	12	100	85	85%	1450	0.11	6.20
4	300	180	10	150	120	80%	450	1.8	0.40

Manual Mode – Interactive Exploration of Local Network Behavior



- ✓ Node and GW-level infos. (Position, SF assignment, loss status, PDR)
- ✓ identification of: coverage holes, overloaded GW, anomalous nodes.

Manual Mode – Interactive Exploration of Local Network Behavior



- ✓ Interactive Deployment Analysis: Drag-and-drop of nodes and GW.
- ✓ Rapid evaluation of coverage gaps and GW diversity.

Manual Mode – in-depth Analysis

LSF Factor

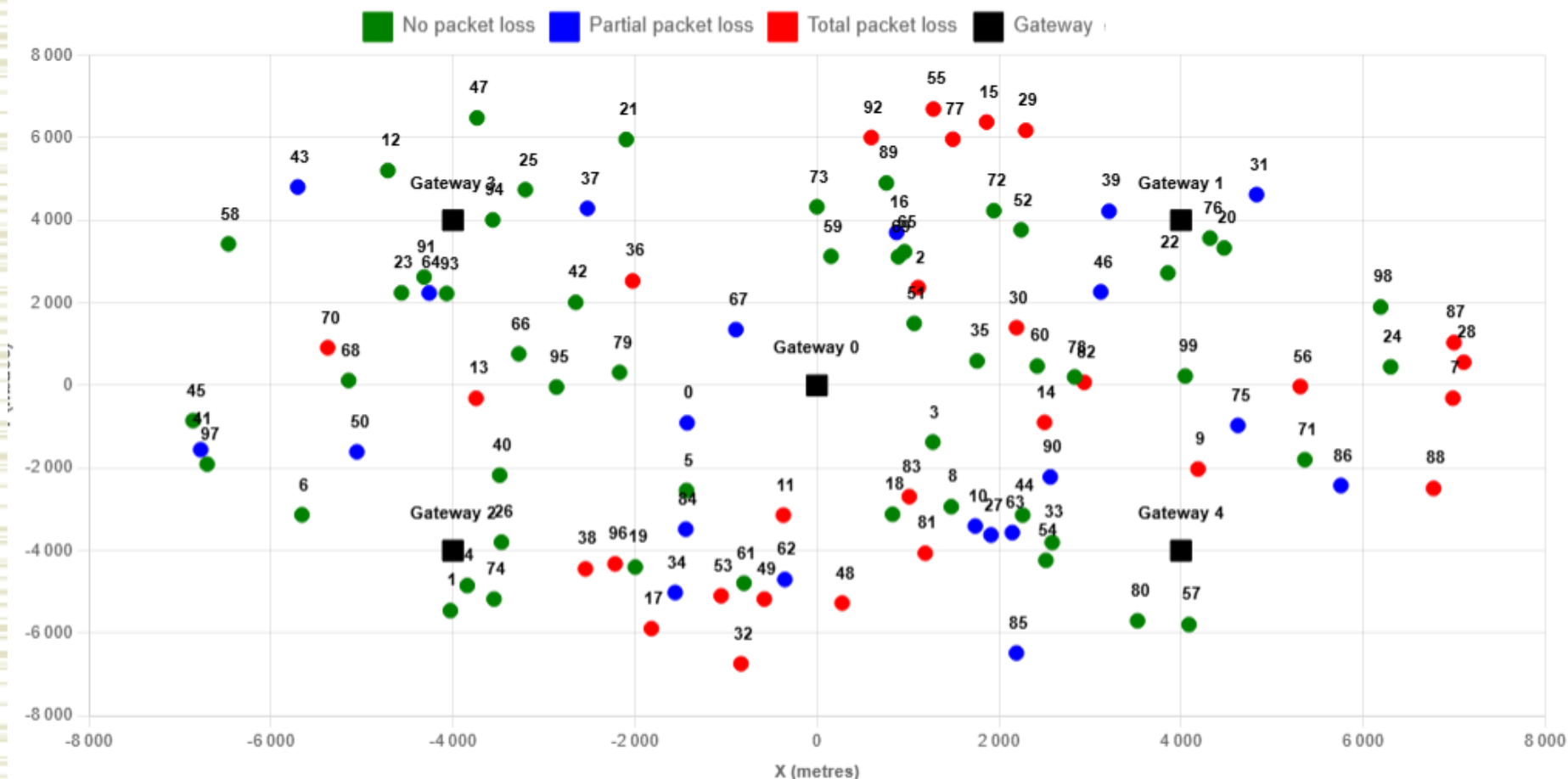
Node Positions

Packet Loss

Rate

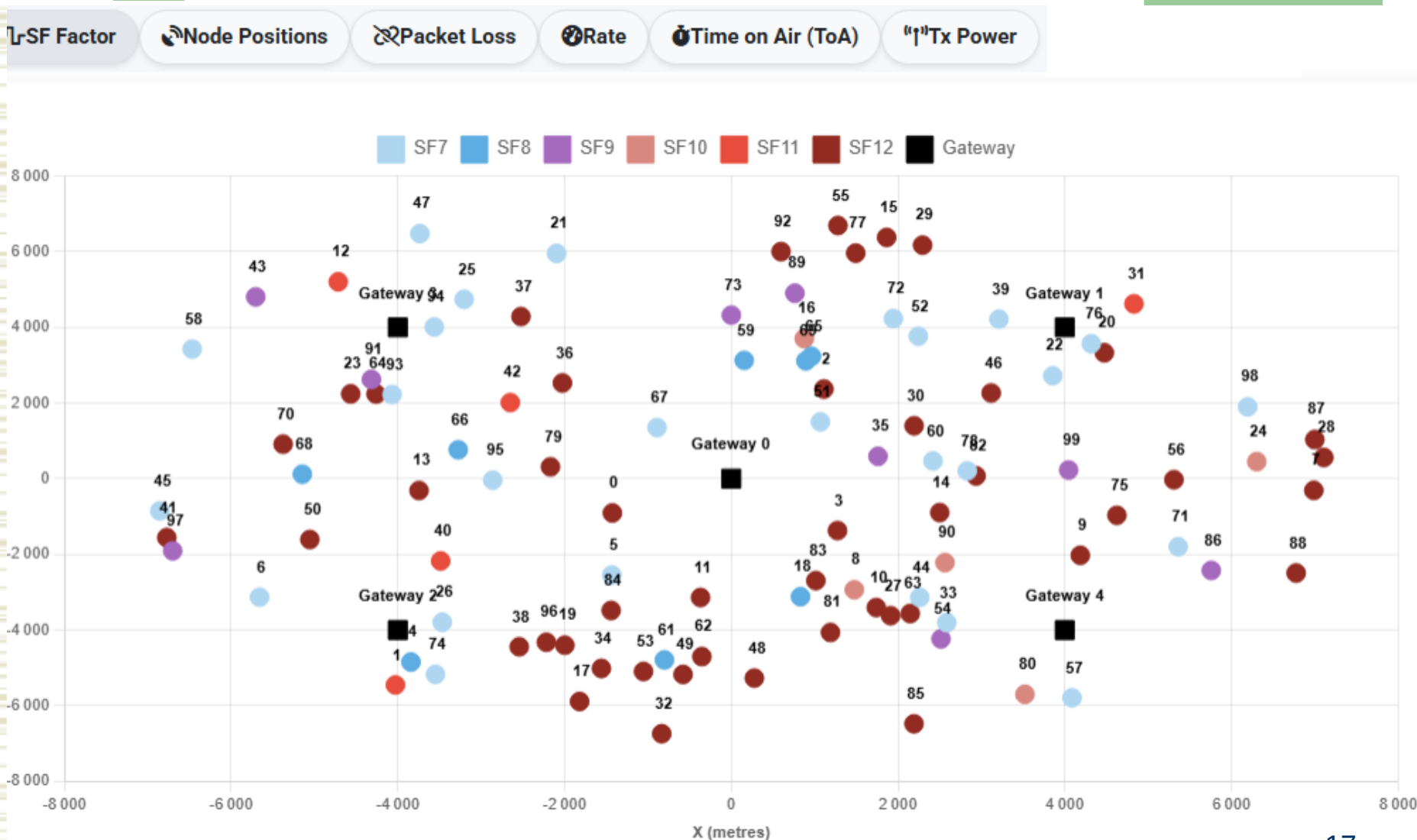
Time on Air (ToA)

Tx Power



Coverage map with packet loss

Manual Mode – in-depth Analysis



Coverage map with SF distribution

- ✓ Simplify and accelerate testing of multiple network parameters.
- ✓ Built-in protocols: LLNRM, GeoNet, ADR; upload custom protocol (.cc).
- ✓ key performance indicators: PDR, energy consumption, and latency.

[Home](#)
[Login](#)
[Automatic Mode](#)
[Manual Mode](#)
[Researcher Mode](#)
[My Account](#)

NC-LoraSim: Automatic Mode

Simulation Area Selection

Show Map & Choose Area

Simulation Parameters

Node List

Gateway List

Simulation Time (min)

Application Periods (min)

Payload Sizes (bytes)

Number of Repetitions

Include Buildings

Poisson Traffic Model

Mechanism

LLNRM
 LLNRM
 GeoNet
 ADR
 New mechanism

Simulation Radius (meters)

Center Coordinates (Lat, Lng)

Propagation Model

Bandwidth (BW)

Coding Rate (CR)

Run Simulation

Retrieve Results

Simulation progress:

10%

Automatic Mode

- ✓ Centralized interface for managing simulation campaigns
- ✓ Define, save, reuse, and modify simulation scenarios
- ✓ Handles scenario generation, execution scheduling, progress monitoring, error logging, and result storage

The screenshot displays the NC-LoRaSim web interface in Automatic Mode. The top navigation bar includes links for Home, Login, Automatic Mode (selected), Manual Mode, Advanced Mode, and My Account. The main heading is "Parallel Simulation Management". Below this is a "Simulation History" section containing two simulation entries: "test" and "test2". Each entry shows its creation date and time, and a status indicator (Running...). Below each entry are three buttons: Access, Stop, and Delete. At the bottom right, there is a green button labeled "Start a New Simulation".

NC-LoRaSim Home Login Automatic Mode Manual Mode Advanced Mode My Account

Parallel Simulation Management

Simulation History

Simulation Name	Status	Created On	Created By	Actions
test	Running...	08/29/2025, 10:17 AM		Access Stop Delete
test2	Running...	12/15/2025, 01:33 PM	Created by User	Access Stop Delete

Start a New Simulation

Automatic Mode - Outputs

Performance Charts

PDR

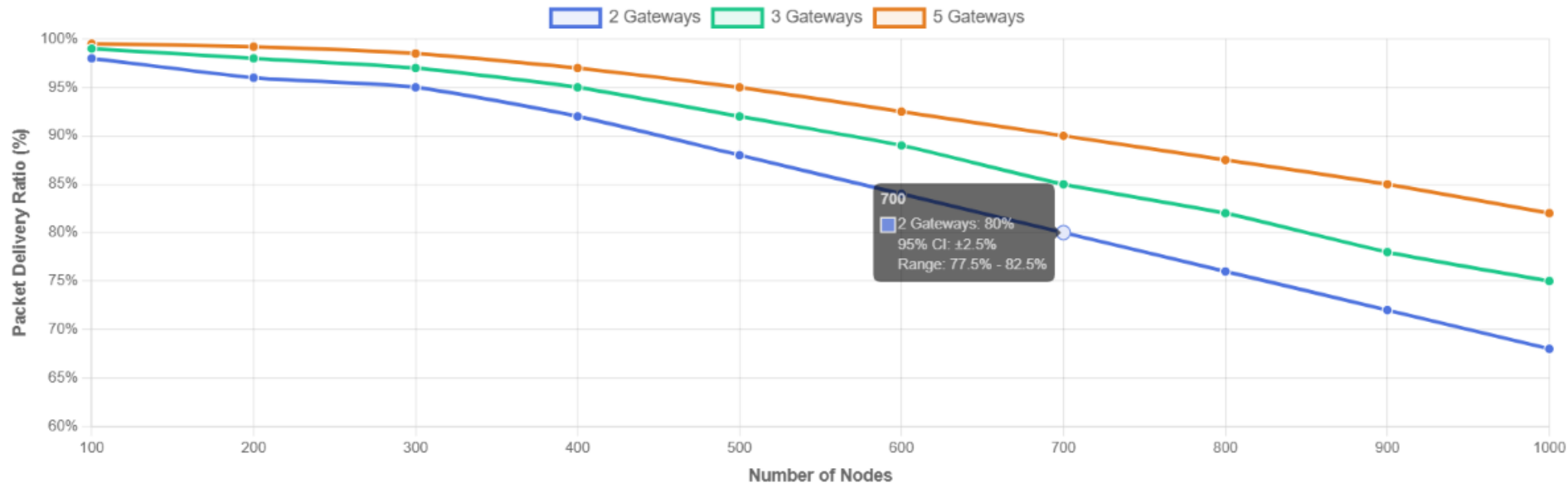
Payload

Gateways

Period

Download Charts

Simulation Time= 240 min | Include Buildings=False | Traffic Poisson=false | Period=12 min | Payload=20 | Nbre of Repetitions=100



Interactive performance graphs with confidence intervals

➔ highlight the impact of key factors on network performance

- ✓ Integrates a lightweight C++ IDE with interactive terminal.
- ✓ Pre-defined functions: **changeSF()**, **changePT()**, **statisticsPDR()**, etc. → enable researchers to focus on the logic of their experiments

 **NC-LoraSim** [Home](#) [Login](#) [Automatic Mode](#) [Manual Mode](#) [Advanced Mode](#) [My Account](#)

NC-LoraSim : Advanced Mode

 **Simulation Area Selection**

Show Map & Choose Area

 **Advanced Code Editor**

Write your code: [Help](#)

C++

```
1 // Network parameters
2 int nDevices = 10; // Number of Nodes
3 int nGateways = 5; // Number of gateways
4 double simulationTime = 120;
5 int appPeriod = 12; // Sending period
6 int payload = 50;
7
8 bool addBuildings = true;
9 bool confirmedup = true;
10
11 int main () {
12     // Your simulation code here
13
14     int indexNode = 3;
15     std::vector<double> pos = GetNodeCoordinates(endDevices, indexNode);
16 }
```

You can use loops `for`, `while`, conditions `if`, functions like `sqrt()` etc.
Use `{}` for blocks and `;` to separate instructions.

[▶ Test Code](#) [↗ Execute Code](#) [⬇ Download Code](#)

>_Output Terminal [Clear](#)

Conclusion

- **NC-LoRaSim** offers lowers entry barrier to realistic LoRaWAN research.
- Combines NS-3 accuracy, no-code GUI, batch statistics and prototyping IDE.

- Available:



- Future Research Directions:
 - ✓ Include mobility scenarios.
 - ✓ Extend analysis with additional metrics.
 - ✓ Validate through a real-world testbed comparisons and large-scale stress tests.