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

Presented by : Ismail BENNIS

Authors: Mohamed-Ali Hadj Amor and Ismail Bennis
IRIMAS - Université de Haute-Alsace (UHA, France)

UNIVERSITÉ HAUTE-ALSACE

November 5-7th, 2025 Instituto Superior Técnico, Lisbon, Portugal

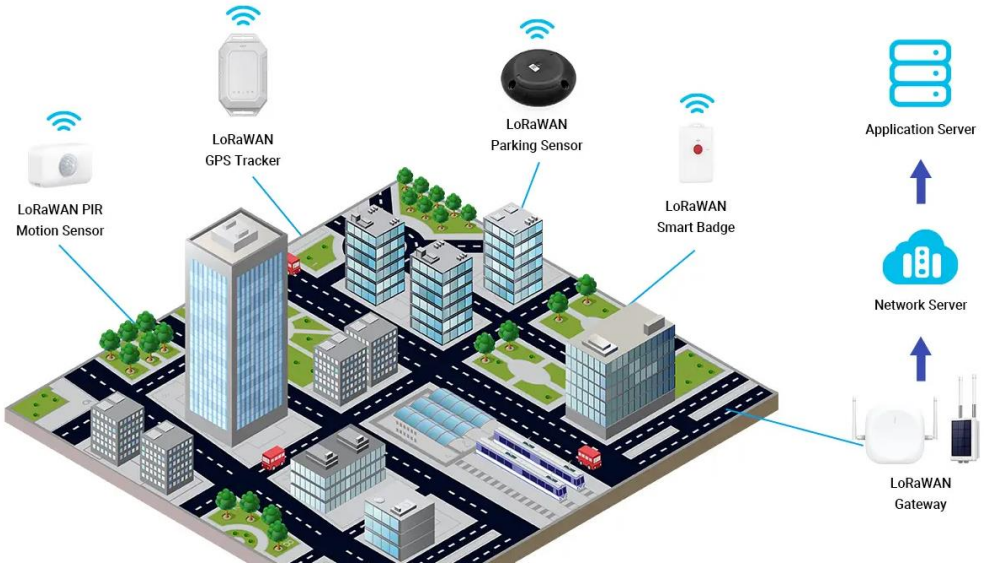
OUTLINE

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

Introduction

- ✓ Designing LoRa networks needs many parameter sweeps (SF, TP, gateways, placement…) and extensive simulation
- ✓ Existing tools trade off accessibility and fidelity:
  - **LoRaSim**, **LoRa-MAB**: lightweight/Python, limited stack (PHY/MAC)
  - **FLoRa**, **LoRaWANSim**: richer, but tool & license limitations
  - **NS-3** LoRa modules: accurate but code-only
- ✓ 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**.

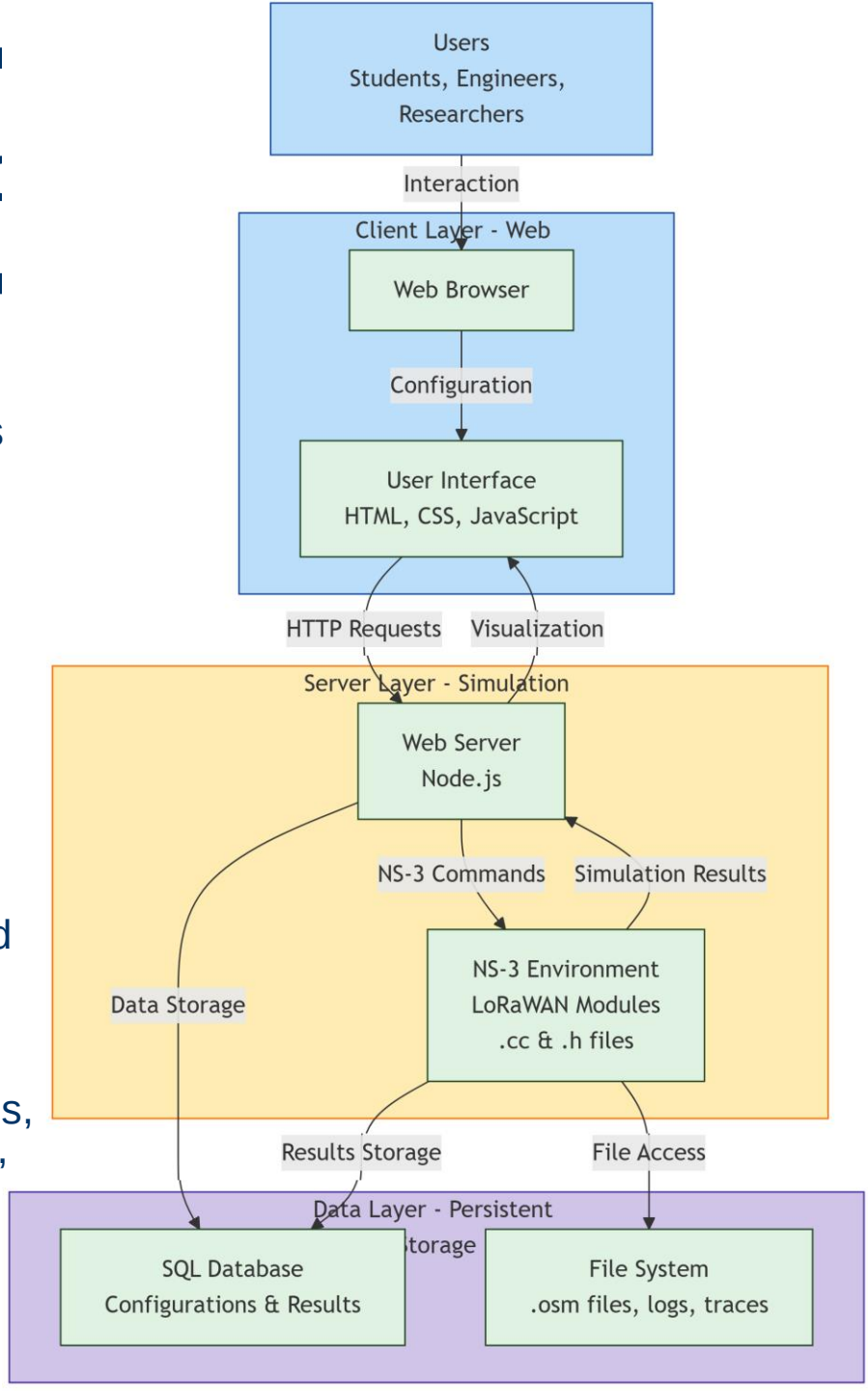
Problem statement & objectives

- Problem:
  - How to fill these gaps?
  - Researchers/teachers lack **accessible**, **reproducible** LoRaWAN simulation tools with NS-3 **fidelity**

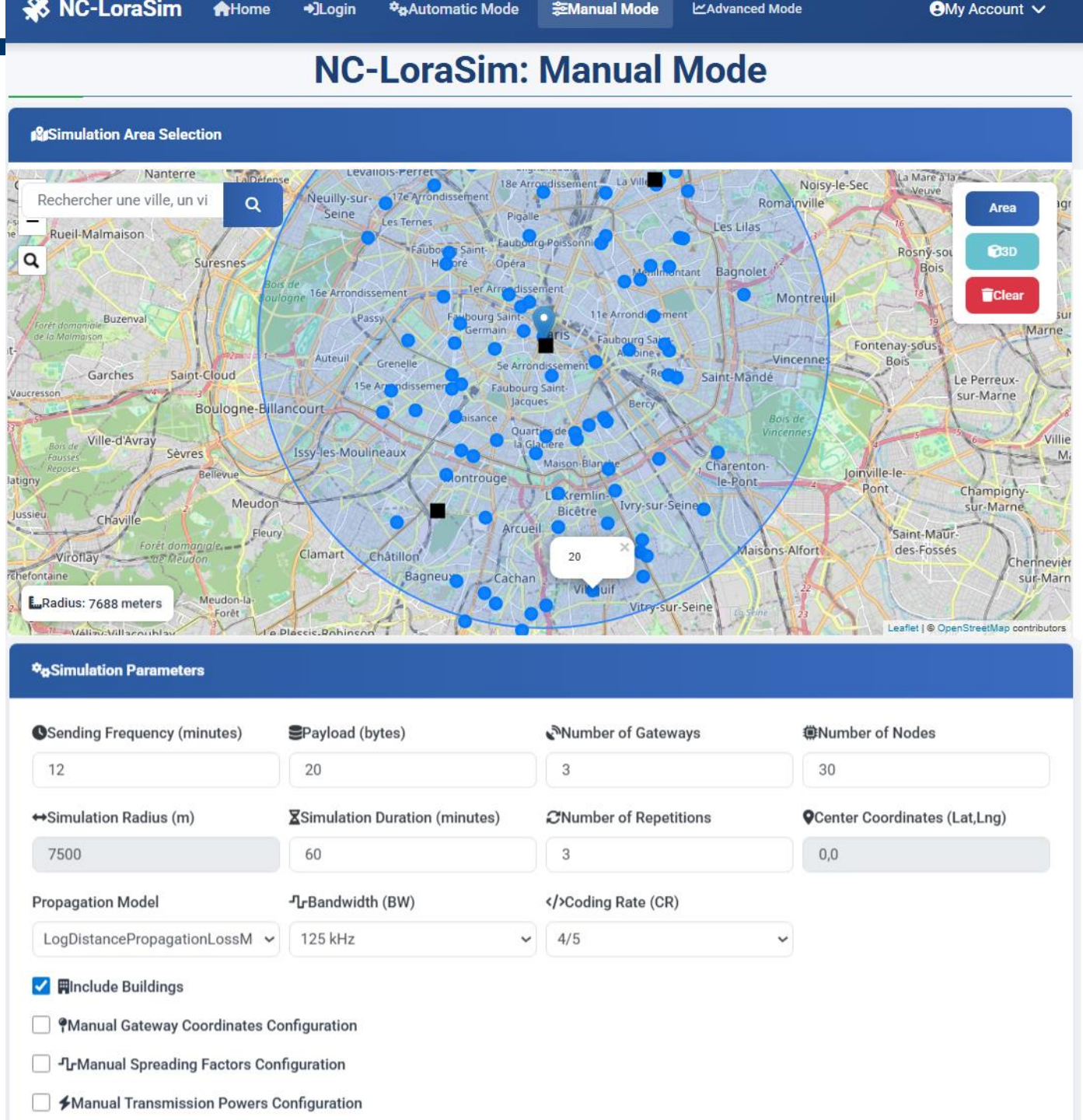
- **NC-LoraSim**:
  - Web platform (client-server-data architecture) ➔ No-code, web UI for scenario design
  - Three modes: Manual, Automatic (batch), Advanced (light C++ IDE)
  - OSM integration, multiple propagation models ➔ Realistic environments
  - Predefined protocols (LLNRM, GeoNet, ADR) + exportable .cc
  - Automated statistics with 95% CI of a built-in metrics
  - Persistent results storage

System arc

- **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 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.



- ✓ 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	2121	70.7%	65.6% – 75.8%
Total Packets Sent	Total Packets Received	Packet Delivery Ratio (PDR)	95% Confidence Interval

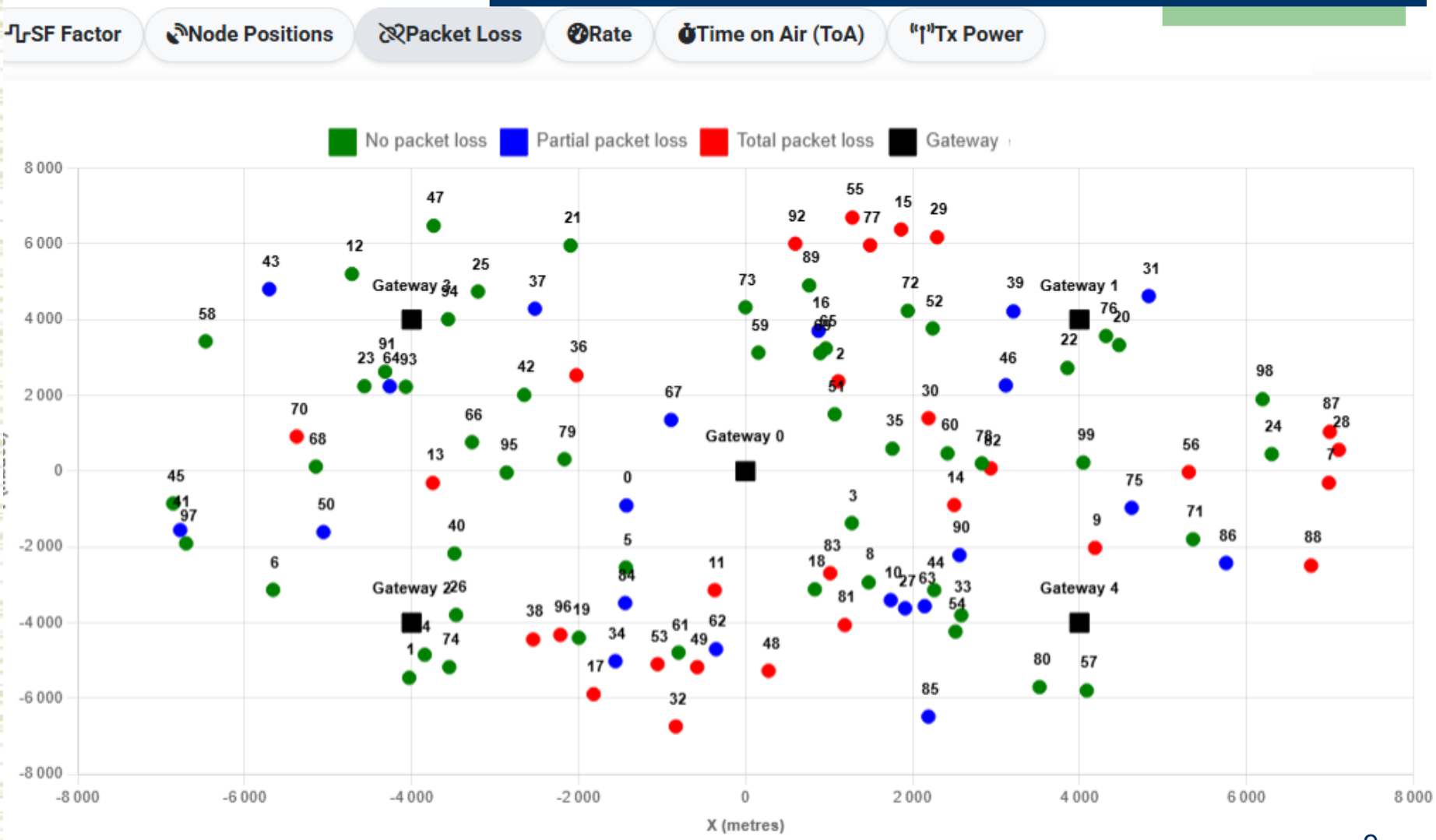
Packets Received per Gateway

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

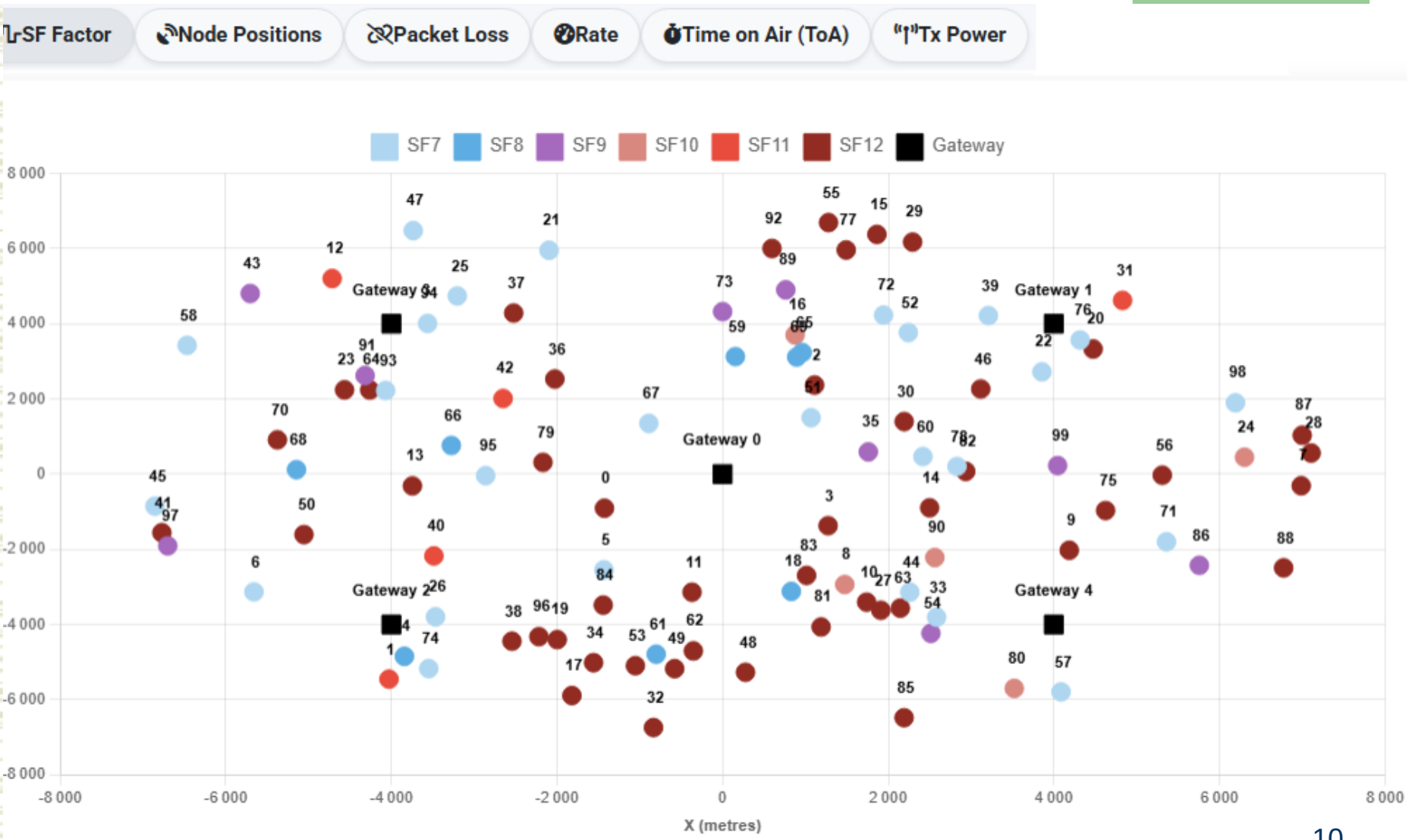
ID	X (m)	Y (m)	SF	Sent	Received	PDR (%)	ToA (ms)	Rate (kbps)	Energy (J)
1	500	7500	7	120	115	95.8%	56	2.86	0.60
2	1520	2000	9	130	120	92.3%	125	1.28	1.16
3	2322	1202	12	100	85	85%	1450	0.11	6.20
4	300	180	10	150	120	80%	450	1.8	0.40

Manual mode – in-depth analysis



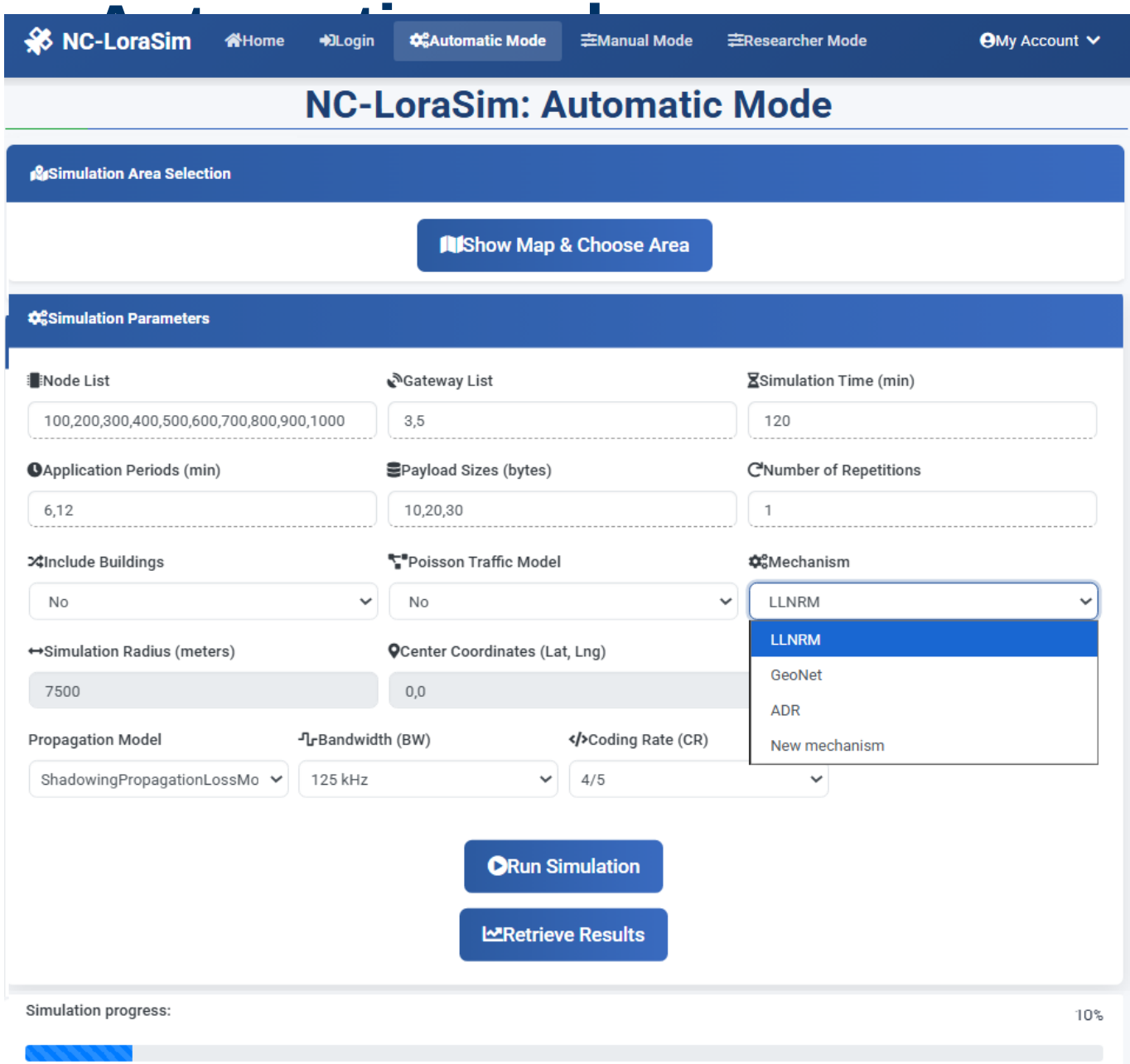
Coverage map with packet loss

Manual mode – in-depth analysis

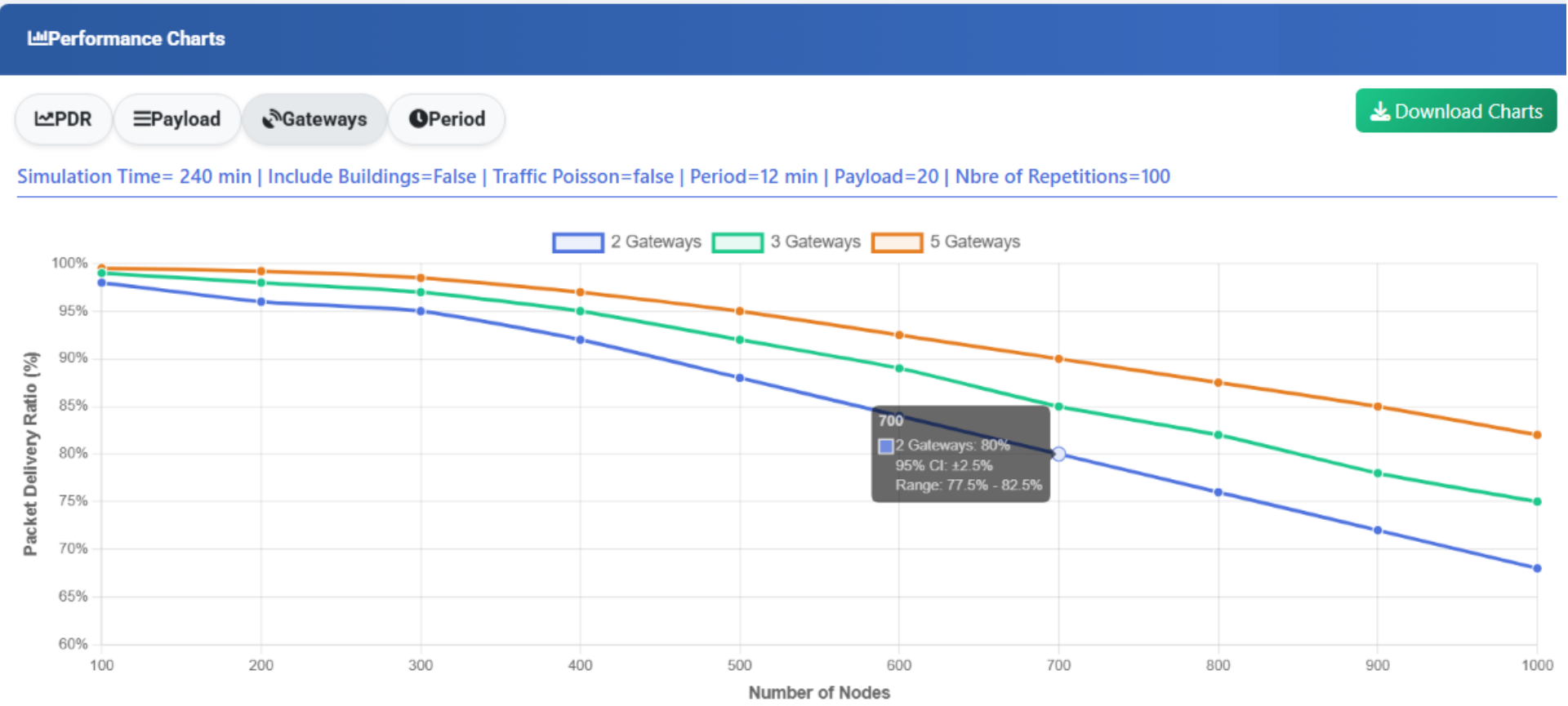


Coverage map with SF distribution

- ✓ 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



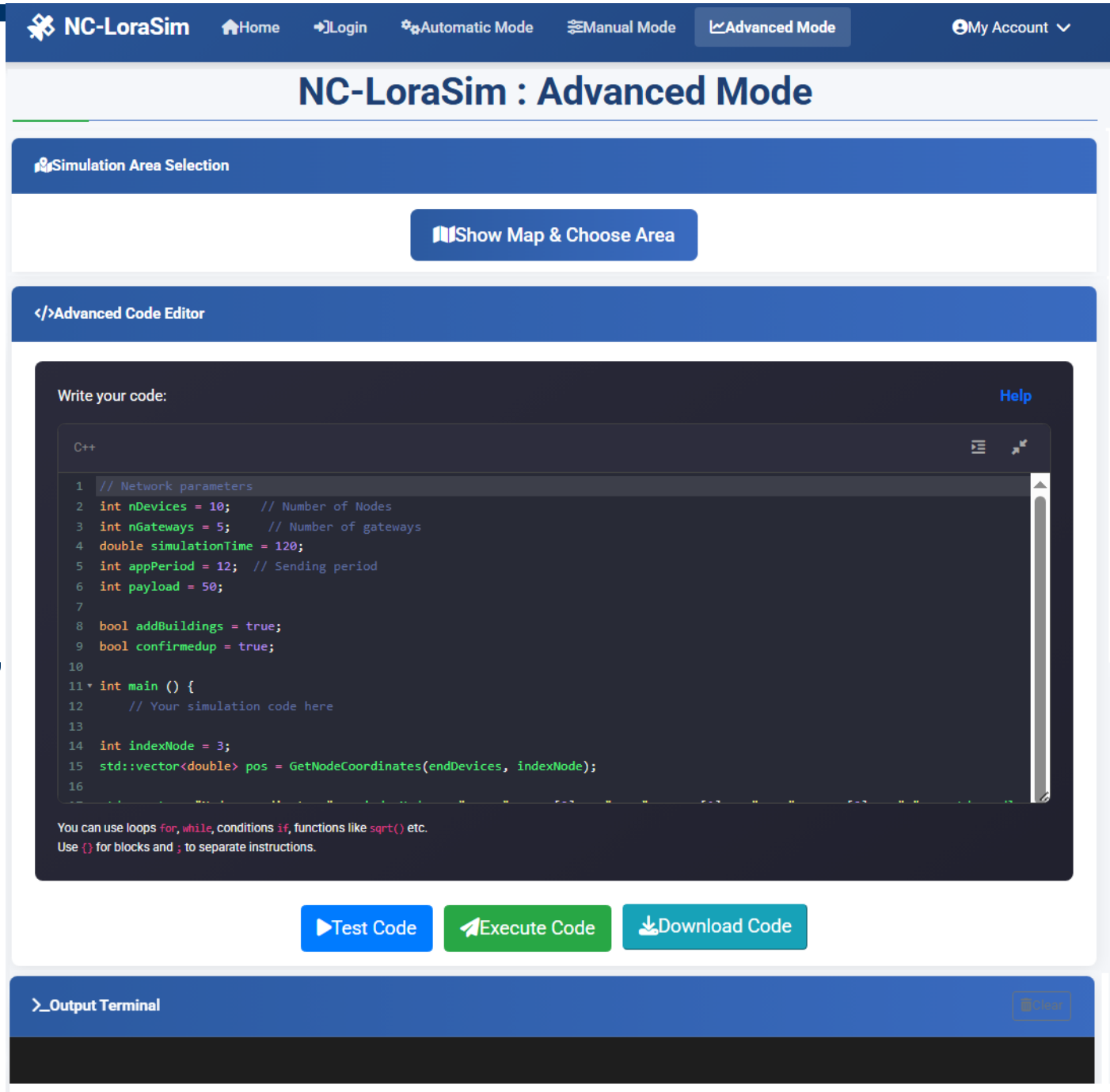
Automatic mode - Outputs



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



Conclusion

- **NC-LoRaSim** offers lowers entry barrier to realistic LoRaWAN research
- Combines NS-3 accuracy, no-code GUI, batch statistics and prototyping IDE
- Repo: https://github.com/HADJAMOR/NC-LoraSim
- Future Research Directions:
  - ✓ Include mobility scenarios.
  - ✓ All-in-one solution: integrating all required dependencies over different platforms
  - ✓ Extend analysis with additional metrics.
  - ✓ Validate through a real-world testbed comparisons and large-scale stress tests