

# Rethinking NTN architecture: leveraging 6G-RUPA for scalable and energy-efficient LEO networks

Sergio Giménez-Antón<sup>1,2</sup> · Eduard Grasa<sup>1</sup> · Jordi Perelló<sup>2</sup> · John Day<sup>3</sup> · Maryan Rizinski<sup>3</sup> · Lou Chitkushev<sup>3</sup>

<sup>1</sup>Software Networks Group, i2CAT Foundation, Barcelona, Spain

<sup>2</sup>Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

<sup>3</sup>Boston University, MET College, Boston, MA, USA

<sup>1</sup>{sergio.gimenez, eduard.grasa}@i2cat.net; <sup>2</sup>{sergio.gimenez.anton, jordi.perello}@upc.edu; <sup>3</sup>{day, rizinski, ltc}@bu.edu



## Motivation

- ▶ **LEO is key for 5G and beyond connectivity.** Mega-constellations provide global, low-latency coverage.
- ▶ **Traditional IP addressing conflates identity with location,** forcing address changes during handovers that break transport sessions.
- ▶ **IP-based routing struggles in LEO topology graphs.** Topology changes force frequent recomputation.
- ▶ **Scalability and energy are coupled.** Traditional routing protocols require *global* routing information ( $O(N)$ ) and large forwarding tables; hardware lookups (e.g., TCAM) make table size a direct contributor to on-board energy draw.
- ▶ **A simple solution: topology-aware addressing.** If addresses encode constellation structure, forwarding can be computed locally with small state, improving scalability and energy efficiency.

## Objective

We propose a **topological addressing and routing scheme** that belongs to a broader network architecture called **6G-RUPA**.

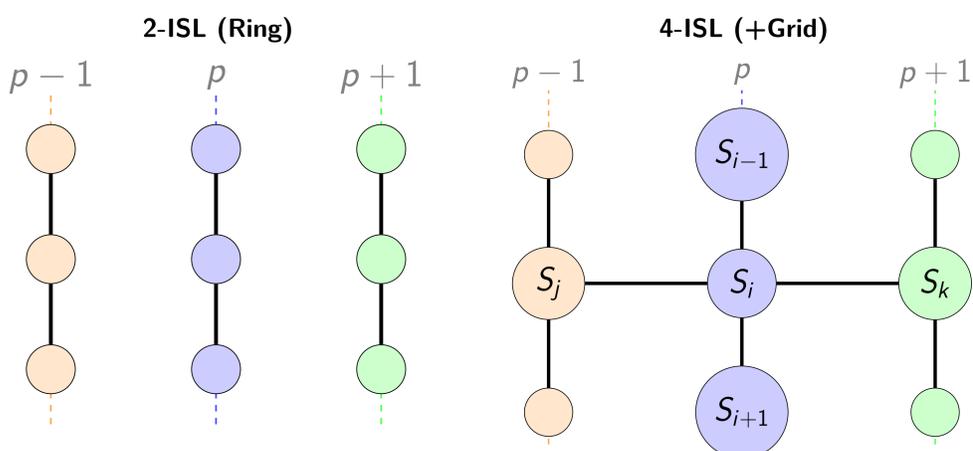
The goal is to use 6G-RUPA's principles to make LEO routing scalable and energy-efficient. Specifically we evaluate the trade-off between:

1. **Forwarding state size** (data-plane memory/lookup complexity),
2. **Routing churn** (how often next hops change over time),
3. **Path stretch** (how far paths deviate from shortest path).

## Constellation Connectivity (Ring vs +Grid)

We propose two simple scenarios.

- ▶ **Ring (2-ISL):** Satellites in the same orbital plane are connected with inter-satellite links.
- ▶ **+Grid (4-ISL):** Same as in 2-ISL, but adds two inter-plane links to adjacent planes, improving reachability and reducing path length.



## 6G-RUPA Topological Addressing (in +Grid)

**6G-RUPA idea.** Use programmable, topology-aware addressing to reduce routing-table lookups.

**Address format:**

$$A = (sh, o, st, x)$$

$sh$ : shell,  $o$ : orbital plane,  $st$ : satellite-in-plane index,  $x$ : endpoint identifier.

**Distance (hierarchical + wrap-around):**

$$f_{dist}(A_i, A_j) = \begin{cases} 0 & \text{if } (sh_i, o_i, st_i) = (sh_j, o_j, st_j) \\ 1 + \min(\Delta st, S_{max} - \Delta st) & \text{if } sh_i = sh_j, o_i = o_j \\ 100 + 10 \cdot \min(\Delta o, O_{max} - \Delta o) & \text{if } sh_i = sh_j, o_i \neq o_j \\ 1000 + 100 \cdot |sh_j - sh_i| & \text{if } sh_i \neq sh_j \end{cases}$$

where  $\Delta st = |st_j - st_i|$  and  $\Delta o = |o_j - o_i|$ .

- ▶ **Forwarding (local rule):** at node  $u$ , evaluate  $f_{dist}(A_n, A_{dst})$  for each one-hop neighbor  $n$  and forward to the arg min.
- ▶ **Why it works (intuition):** weights force the path to (i) reach the right shell, then (ii) the right plane, then (iii) the right satellite index, using wrap-around.

## LEOPath: A LEO Satellite Routing Simulation Framework

**LEOPath** is an open-source simulation framework to analyze and compare routing algorithms in LEO satellite constellations. It enables rapid, reproducible experiments on time-varying topologies (e.g., Starlink-like shells) without packet-level overhead.

**Key features:**

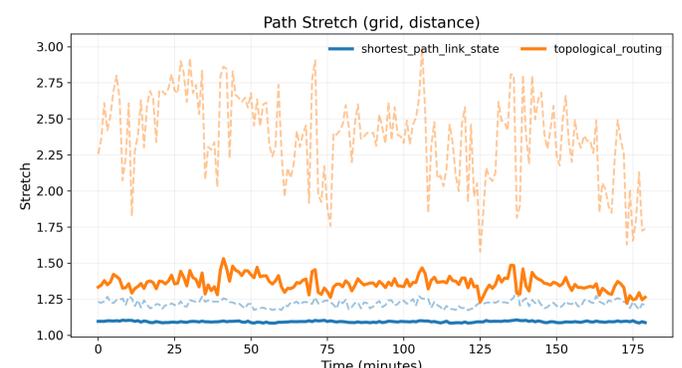
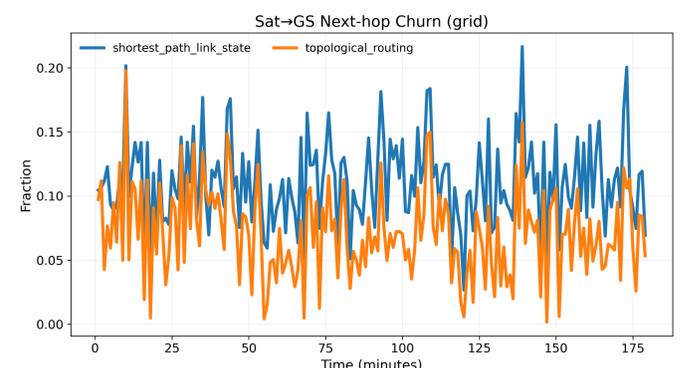
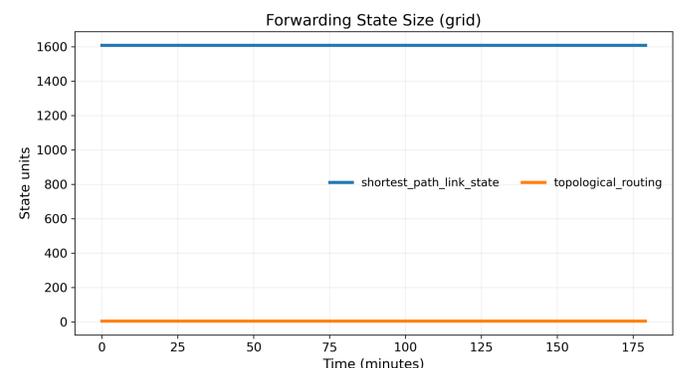
- ▶ Easy to run (container-friendly workflow).
- ▶ Realistic orbit propagation (TLE/SGP4) and link visibility.
- ▶ Dynamic ISL/GSL state and configurable ISL scenarios (Ring, +Grid).
- ▶ Pluggable routing algorithms and attachment strategies.
- ▶ Interactive visualization using cesium.

**Get the simulator:**



## Evaluation: Simulation of a Starlink-like constellation

**Setup:** Starlink first shell ( $22 \times 72$ ,  $N = 1584$ ), 24 ground stations, 3 hours,  $\Delta = 1$  minute; link-state shortest path vs topological routing.



**Conclusion:**

Topology-aware addressing enables per-hop forwarding using only local neighbor state, avoiding  $O(N)$  forwarding tables and reducing TCAM-style lookup energy pressure. In Starlink-like mega-constellations, this substantially cuts forwarding state and improves stability under dynamic ISLs/GSLs compared to link-state shortest-path routing, at the cost of a controlled increase in path stretch.

**Future work:**

- ▶ Evaluate multiple constellations (different shells, altitudes, and ISL policies).
- ▶ Implement and benchmark state-of-the-art NTN routing baselines (e.g., IS-IS/OSPF, segment routing).
- ▶ Extend to multi-shell and failure scenarios (ISL/GSL outages, gateway failures) and quantify end-to-end performance/energy trade-offs.