FLoRaSat 2: Simulating Cross-Linked Direct-to-Satellite IoT LEO Constellations

Alexander Choquenaira-Florez
Benoit Coeugnet
Robin Ohs
Juan A. Fraire
Hervé Rivano

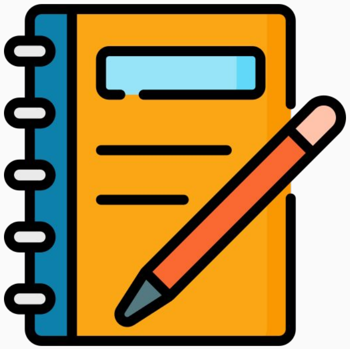
RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité
Inria

NTN-Days

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

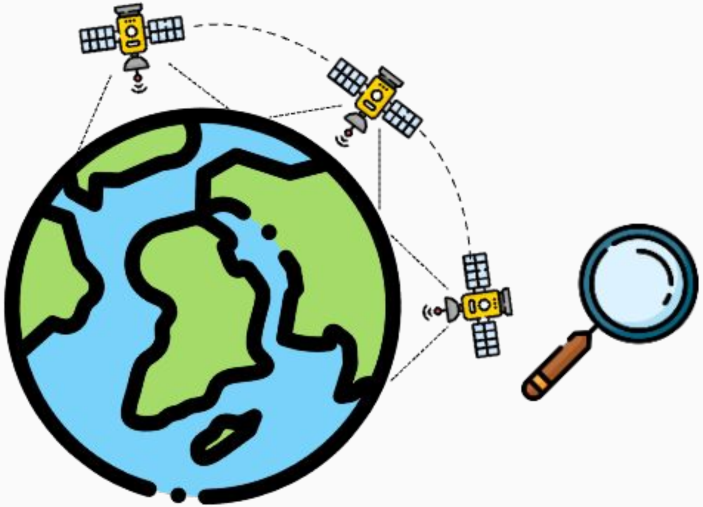
Agenda

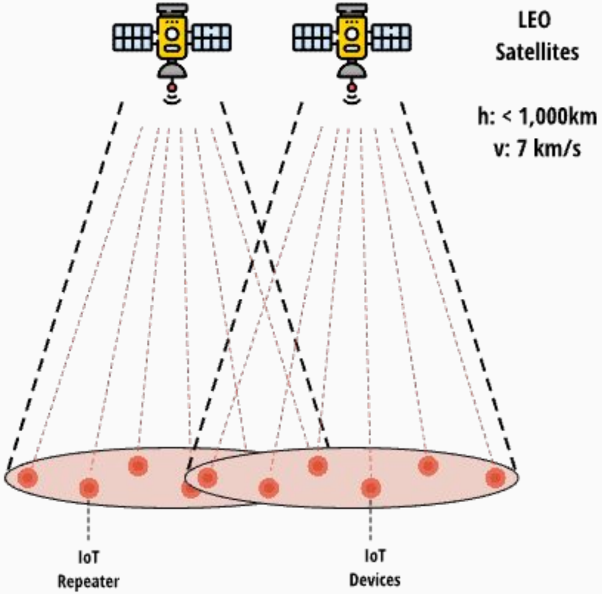
1. Context and Motivation
2. ***FLoRaSat*** evolution
3. Built-in modules
4. Use Case
5. Conclusions



Context and Motivation

DtS-IoT





Applications

Main Actors

kinéis
Lacuna space
Wyld
SATELIOT
Myriota
FOSSA

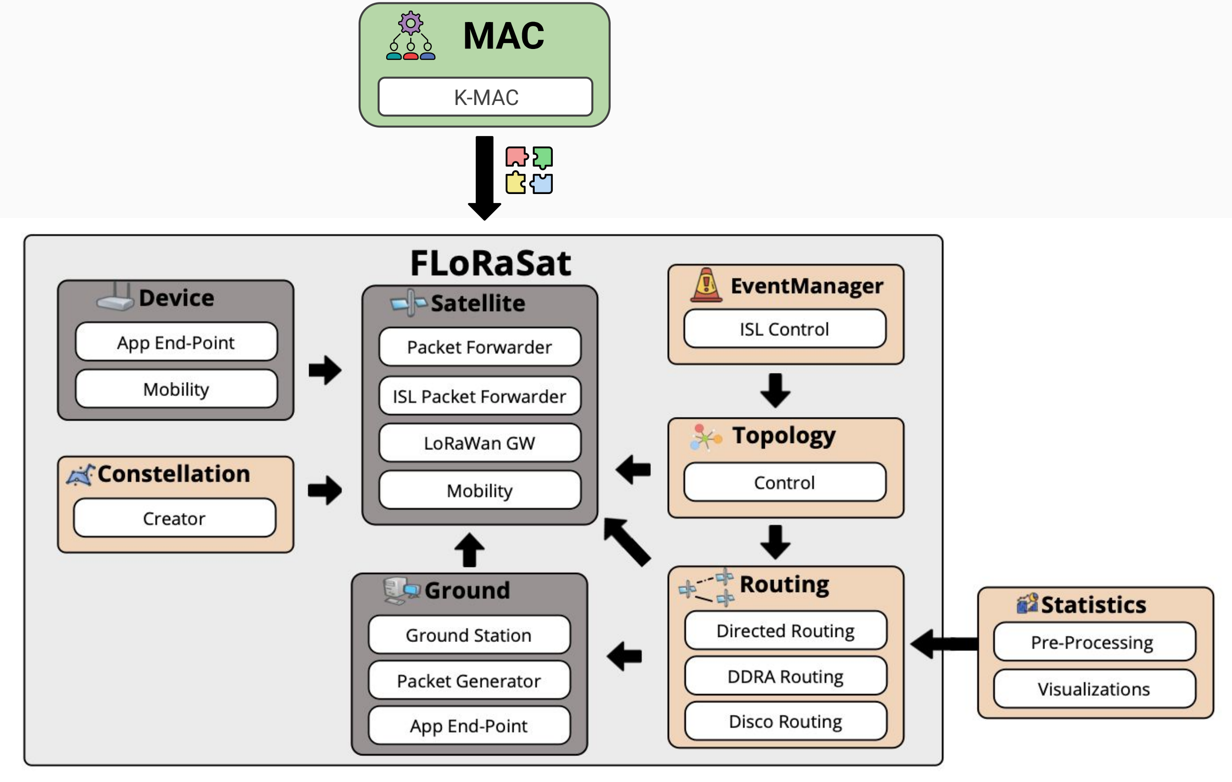
On-ground devices

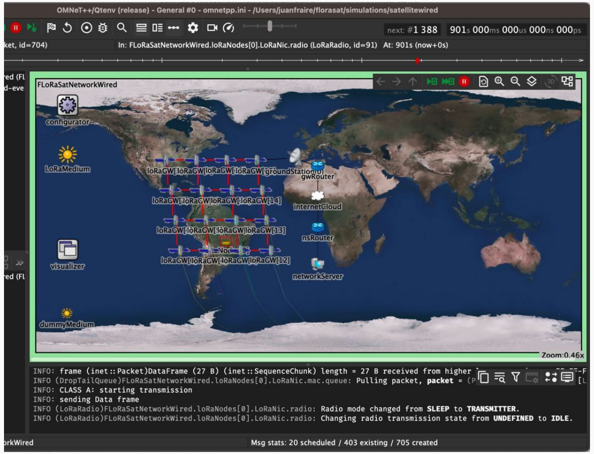
LoRa
NB-IoT

Context and Motivation

	STK	*Hypatia*	*OpenSAND*	*GMAT*	*Orekit*	*SILLEO*	***FLoRaSat***
•Orbital Propagator	✓	✓	✗	✓	✓	✓	✓
•Packet Simulation	✗	✓	✓	✗	✗	✗	✓
•DtS-IoT	✗	✗	✗	✗	✗	✗	✓
•Constellation Grade	✓	✓	✓	✗	✗	✓	✓

FLoRaSat Evolution





Omnet++ GUI

The extended architecture of FLoRaSat. Orange modules were introduced in FLoRaSat 2.

Constellation and Analysis Modules

Topology Control

- Manage communication between satellites and ground modules.
- Continuously updates ISLs and GSLs.

Event Manager

- Enable and disable Inter-Satellite Links.
- Test routing algorithms' adaptability to failures and congestion.

Constellation Creator

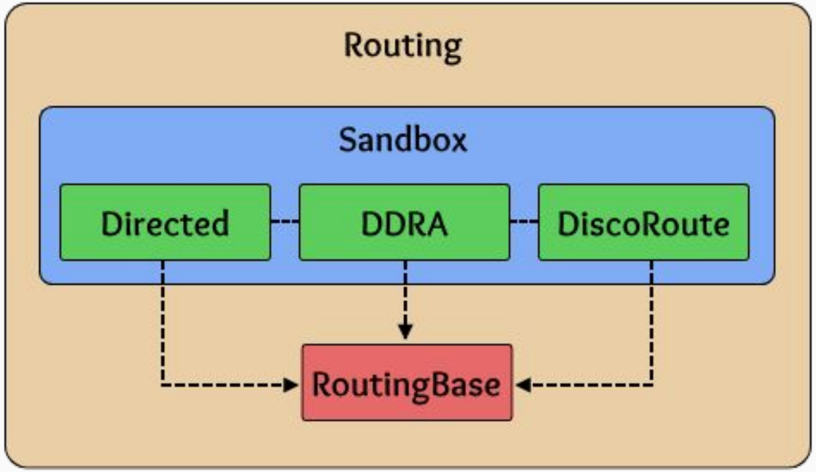
- Supports Walker-Delta and Walker-Star constellations.
- Enables complex constellations like One Web and Starlink.
- Positions validated against STK [3] reference values.

FLoRaSat - CLI

- Command-line interface for analyzing simulation results with metrics and visualization.
- Supports easy integration of additional graphs and metrics.

Routing Sandbox

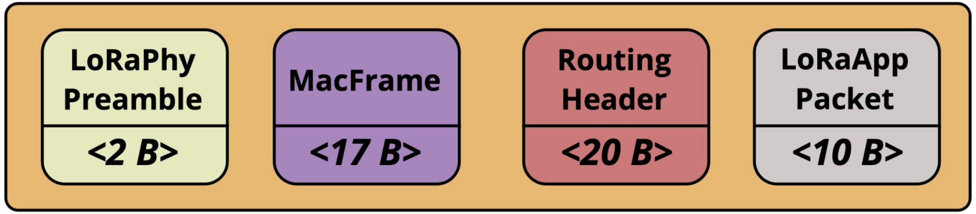
- Based on an abstract class ***RoutingBase*** to facilitate quickly creation and integration.
- Main functions to overwrite:
  - handlePacket()
  - preparePacket()
  - handleTopologyChange()
- Three routing algorithms were implemented: ***Directed, DDRA, Disco.***



Structure of the *Routing* sandbox

MAC Protocols and Integration

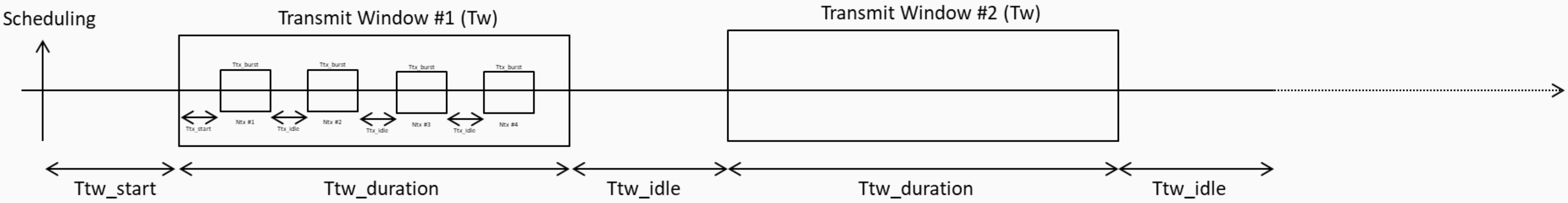
- This integration enables **FLoRaSat** to optimize routing by considering multiple constraints and prioritizing less congested paths to enhance overall network efficiency.
- The process begins with physical and data link layer operations. Before transmission, the routing algorithm determines the optimal route. The resulting structure is shown in the next figure.



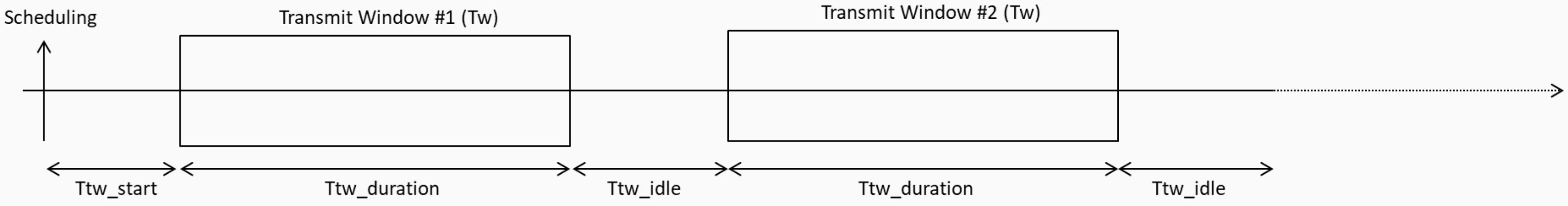
Structure of the *LoRaWAN* packet after processing

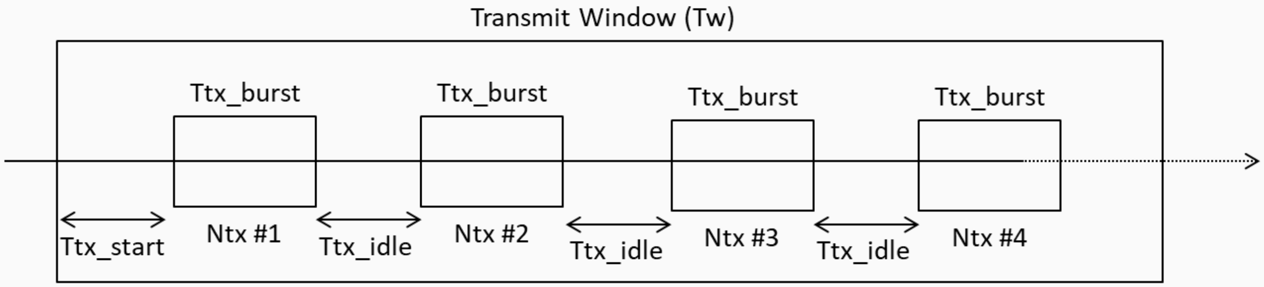
K-MAC

- **Messages** => information
- Multiple **Repetitions** per message => effort
- Message: We only count the first instance of a successfully received re-transmission



K-MAC


Scheduling
Transmit Window #1 (Tw)
Transmit Window #2 (Tw)
Ttw_start
Ttw_duration
Ttw_idle
Ttw_duration
Ttw_idle


Transmit Window (Tw)
Ttx_burst
Ttx_burst
Ttx_burst
Ttx_burst
Ttx_start
Ntx #1
Ttx_idle
Ntx #2
Ttx_idle
Ntx #3
Ttx_idle
Ntx #4

FLoRaSat Demo

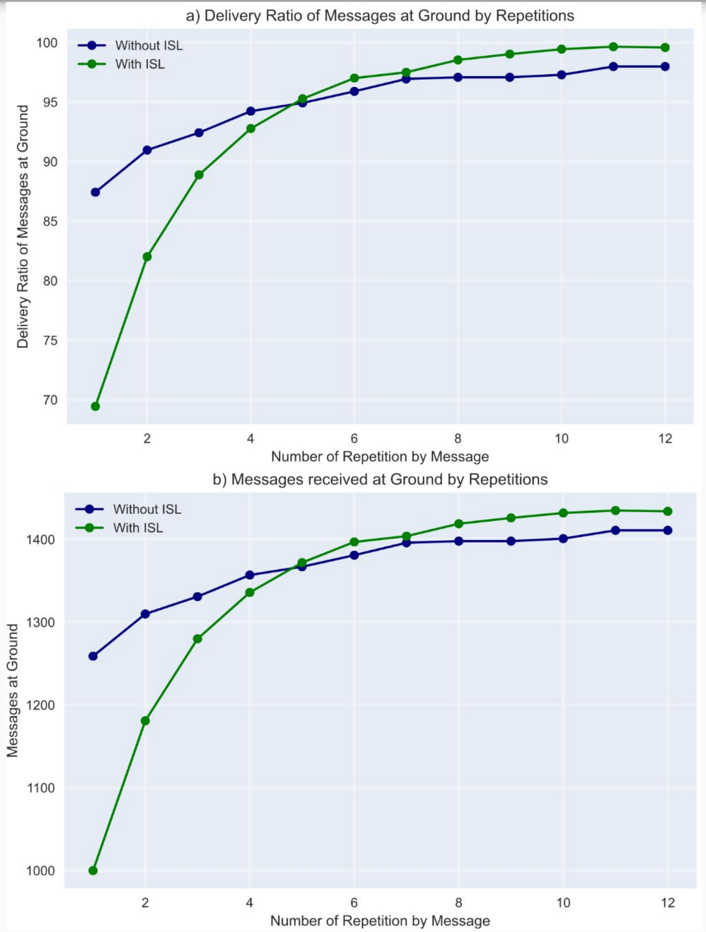
- The demo scenario consists of a Walker-Delta Iridium constellation with 66 satellites, 20 ground stations, and 20 devices transmitting to the **nearest ground station**.
- The scenario is configured with the DDRA routing algorithm and the **K-MAC protocol** (set to 5 repetitions).
- The main objective of this scenario is to **analyze the features** previously introduced.

Use Case

- The use case evaluates the performance of K-MAC by comparing **two scenarios**: one that applies a routing algorithm, and another where the satellites retain the messages until they reach the transmission range (satellites as a store-and-forward node).
- Consists of a Walker-Delta Iridium constellation, which is composed of 66 satellites and 288 devices that **transmit to the nearest ground station**. Depending on the configuration, each message is sent between one and twelve times.
- The main objective of this scenario is to evaluate the impact of **increasing the repetitions** on the MAC scheme.

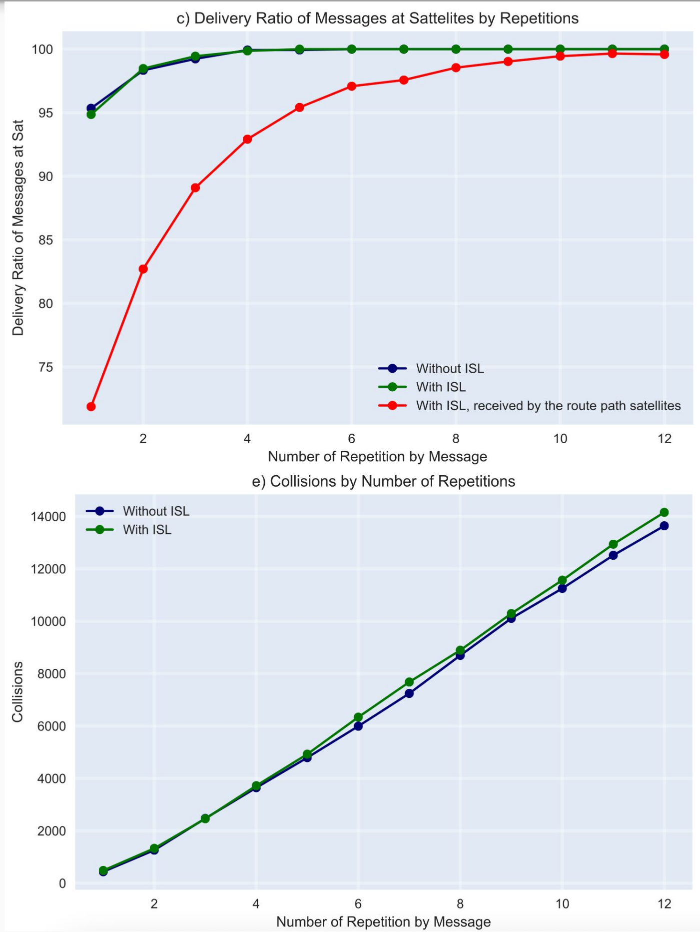
Use Case - Analysis at ground

- In the first repetitions, there was a better performance for the scenario without ISL.
- However, after the 5th repetition, the ISL overpassed the non-ISL's results.
- For lower repetitions, the messages often fail to reach the route path satellite to start the routing process.



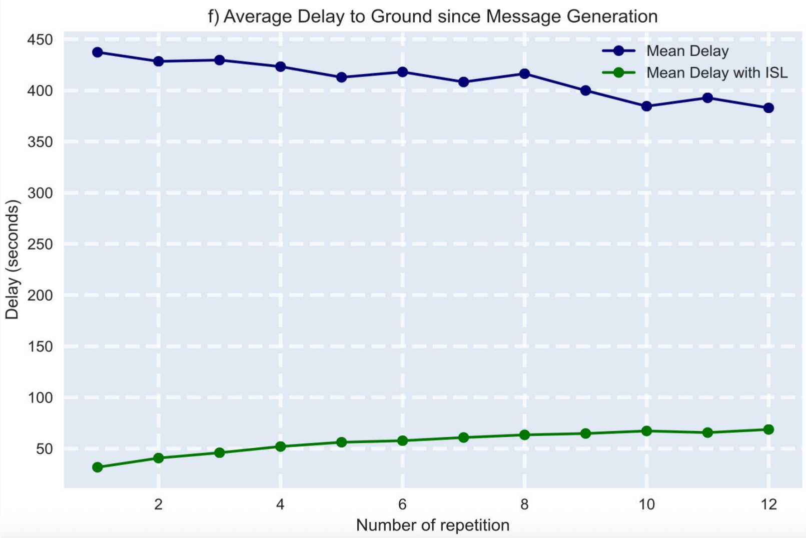
Use Case - Analysis at Sats and Collisions

- The reception rate at satellites is very high (>95%), but few messages reach the route path satellite in first repetitions.
- This can be explained by devices transmitting to all the satellites within range.
- Collisions increase with the repetitions, which is consistent with expectations.



Use Case - Analysis of delay

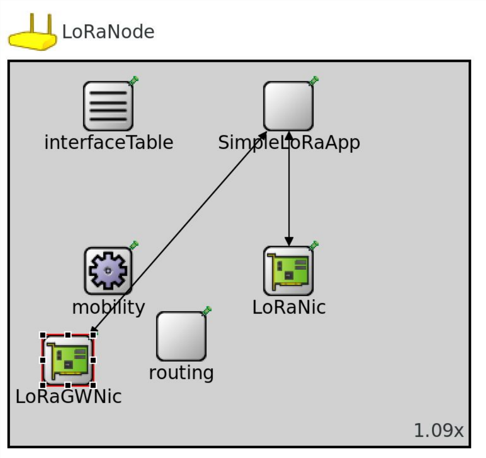
- For the ISL scenario, the delay is significantly lower (~50 - 85 s), which means that the messages arrive faster than in the non-ISL scenario (~450 s).
- This difference highlights the most significant benefit of employing ISLs and routing.

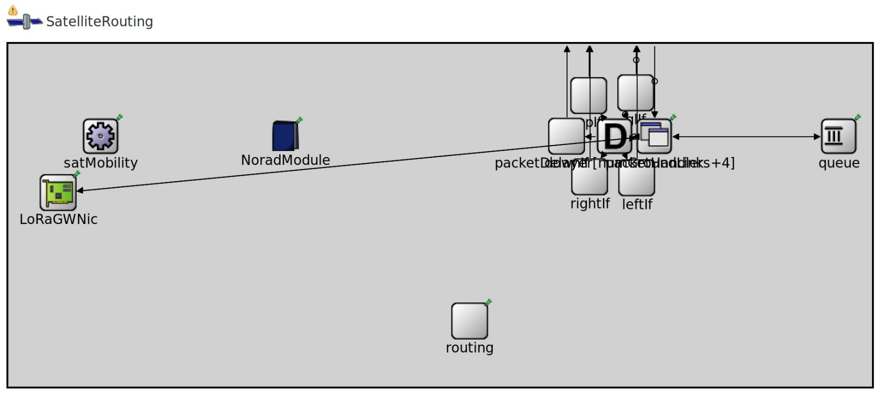


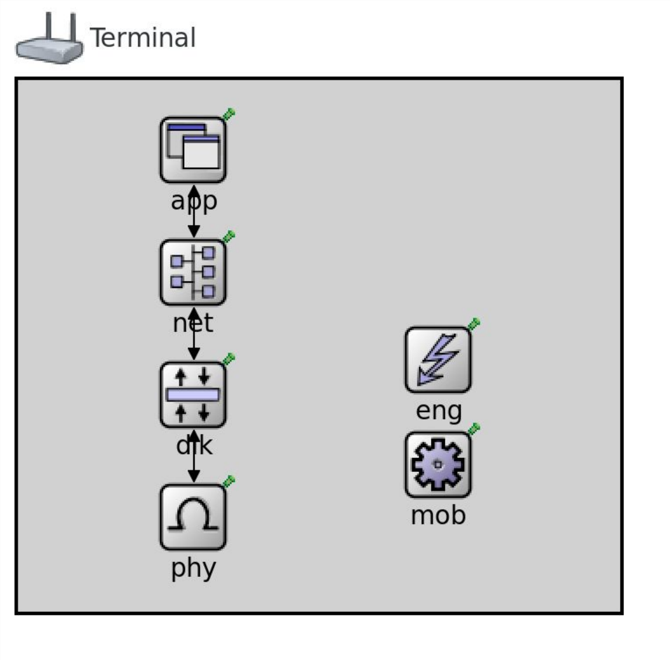
Conclusions and Future Work

- **FLoRaSat 2:** Enhanced original version with new modules, including constellation creation, dynamic topology, graphs, and routing.
- **K-MAC:** A simple repetition-based MAC protocol that addresses critical challenges in DtS-IoT. In this protocol, each device transmits messages multiple times across transmission windows to enhance the probability of successful delivery.
- **Future Work** should focus on the development of a potential dedicated MAC Sandbox that facilitates the seamless implementation, integration, and evaluation of customized MAC protocols, ensuring FLoRaSat's robustness in practical deployments.
- All these enhancements make FLoRaSat 2 a robust and comprehensive **end-to-end simulation platform**.

FLoRaSat v3 ?


LoRaNode
interfaceTable
SimpleLoRaApp
mobility
LoRaNic
routing
LoRaGWNic
1.09x


SatelliteRouting
satMobility
NoradModule
queue
LoRaGWNic
rightIf
leftIf
routing


Terminal
app
net
dlk
phy
eng
mob

Thanks for your attention!

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Inria

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

NTN-Days