FLoRaSat : Le simulateur pour la communication LoRa avec les satellites LEO

Juan Andrés A Fraire, Mehdi Mesbah, Pablo Madoery, Oana Iova, Fabrice Valois

FLoRaSat : Le simulateur pour la communication LoRa avec les satellites LEO

Juan A. Fraire[1,2], Mehdi Ait Mesbah[1], Pablo Madoery[2], Oana Iova[1], Fabrice Valois[1]

[1] *Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France*
[2] *CONICET - Universidad Nacional de Córdoba, Argentina*

Le nouveau paradigme de communication, Direct-to-Satellite-IoT (DtS-IoT), est une approche très prometteuse pour le transfert de données depuis/vers des appareils IoT dans des zones reculées où le déploiement d'une infrastructure terrestre n'est pas faisable. Dans ce contexte, les satellites en orbite terrestre basse (LEO) peuvent équiper des passerelles IoT comme celles de LoRa, pour faciliter la collecte de données des capteurs situés sur Terre en ligne de vue directe. Cependant, cette communication est soumise à plusieurs défis : grandes distances de transmission (500 à 2000 km), couverture sporadique et de courte durée (environ 7 minutes), liaisons inter-satellites à faible bande passante, et appareils IoT contraints en énergie. Pour permettre à la communauté de mettre en œuvre et d'étudier des protocoles pour la communication DtS-IoT de façon réaliste, nous proposons un nouvel outil de simulation à événements discrets : FLoRaSat. Le simulateur s'appuie sur Omnet++ et implemente un scénario de communication DtS-IoT de de bout en bout comprenant 16 satellites LEO interconnectés et 1500 nœuds IoT sur Terre qui communiquent à l'aide du protocole LoRaWAN.

Mots-clefs : LoRa, LoRaWAN, Satellite en orbite terrestre basse (LEO), Simulateur, Direct-to-Satellite-IoT, Omnet++

1 Introduction

The recent spur in space projects has revamped the interest in satellite communication. This is especially observed in the Internet of Things (IoT) community that constantly seeks to diversify the application scenarios [CCG+21], but also, from the telecommunication operators that want to expand the IoT network coverage to offer a promising omnipresent world-wide service.

Satellites in Low Earth Orbit (LEO) are deployed at low altitudes ($160-1,000$ km) and move at ~7 km/s, which results in overflights of <10 min over a given spot on the surface. These characteristics make LEO satellites very appealing to the IoT community, as they can be reached by low-power, long-range technologies such as NB-IoT and LoRa. IoT devices with a clear line-of-sight to the sky can directly access the satellite with an on-board gateway, even with low-cost antennas, and without any other modification to the device. This can be extremely useful to applications in less accessible regions (i.e., oceans, mountains, poles) that might not justify or even hinder the deployment of IoT gateways. However, transmission distance and orbital dynamics, combined with highly constrained devices on the ground make Direct-to-Satellite IoT (DTS-IoT) a very challenging problem. In fact, existing IoT medium access control protocols, negotiations schemes, etc. need to be revised and/or extended to scale up to these challenging conditions.

One of the main challenges with DTS-IoT is the evaluation of all these new protocols and network architectures, as real deployments are extremely costly and rigid. So we need realistic simulators implementing both IoT and satellite technologies as detailed as possible, which currently do not exist. The few papers that started looking into DTS-IoT use custom-made simulators based on Matlab that lack accurate modelling of the network layer. To this end, we propose FLoRaSat, a novel event-driven open-source end-to-end simulation tool that leverages OMNET++ and includes a benchmarking DTS-IoT scenario comprising 16 cross-linked LEO satellites and 1500 IoT end-devices.

2 FLoRaSat

In order to realistically represent the environment in which future DTS-IoT systems will operate in, we need (i) an accelerated simulation approach so that we can quickly simulate LEO satellite orbits, (ii) a realistic implementation of the physical, medium access, and network layers, and (iii) an environment that allows running multiple seeds and parametric studies. To this end, we propose FLoRaSat[†] (*Framework for LoRa-based Satellite IoT*), an open-source, event-driven simulator that mimics satellite and IoT systems, and their interactions.

FLoRaSat leverages (i) the well-known OMNeT++ framework[‡] with its INET extension for the classical Internet stack, and (ii) the open-source LoRa-based simulator FLoRa[§] with its basic implementation of the LoRaWAN [SPDF18] protocol. We adapted, extended, and ported the physical layer (LoRa) and the link layer (LoRaWAN) from FLoRa to the DTS-IoT configuration, to which we added a set of new ground-braking features related to the satellite IoT context. We propose a scenario in which end-devices communicate directly with one or more LoRaWAN gateways situated on LEO satellites, which are organised in a constellation. Gateways send beacons at regular time intervals to announce their presence using Class B mode of LoRaWAN, and end-devices buffer data until a satellite is in sight. Satellites also communicate among each other using inter-satellite links. The current version of FLoRaSat includes a constellation of 16 LEO satellites and up to 1500 devices distributed uniformly in a circular region with a 2000 km radius (in South America)[¶]. To simulate this scenario in FLoRaSat, we used a modular approach that allows us to easily modify and plug-in different approaches.

Orbital mechanics. The orbital dynamics elements are in charge of propagating satellites along realistic trajectories. We created an orbital processing pipeline in which the scenarios can be designed using the state-of-the-art Systems Toolkit (STK), from AGI[‖]. Then, we implemented an export script in C# to obtain the position of each satellite in X, Y, Z coordinates in the Earth-centred inertial frame (ECI), as well in Latitude, Longitude and Altitude (LLA) on a second-by-second granularity. Due to the ECI coordinate reference system, IoT devices are seen as fixed points (satellites are the only element with a time-evolving position in ECI), so they do not need to be created in STK. During simulation time, these files are consulted by the satellite mobility module, the inter-satellite distance computation module, and by the visual module.

Inter-satellite routing. Since satellite IoT services need to be deployed in networked constellations to cover large geographical areas, we include an inter-satellite link (ISL) model with a basic LEO-to-LEO routing solution. We propose a grid-shaped LEO constellation, in which a single satellite implements a contact with a ground station (e.g., satellite `[15]` in Figure 1), and we implement a simple *right-up* and *down-left* routing approach for the uplink and downlink traffic flows respectively. More sophisticated routing approaches (e.g., topology-aware Contact-Graph routing) can be easily added in the future.

LoRaWAN Class B. As discussed by Vogelgesang *et al.* [VFH21], a beacon can play an important role in announcing satellite presence and synchronising DTS-IoT devices on ground. In LoRaWAN, this can be achieved leveraging the Class B mode of communication where the gateway periodically sends a beacon to synchronise and ping end-devices. Class A is also supported from the legacy version of FLoRa.

Application model. In the DTS-IoT scenario, IoT devices can not always assume an IoT gateway will be present in the sky. Therefore, the application model was extended to also include a delay-tolerant scheme, where data is buffered until a satellite is present.

All these features are integrated into the FLoRaSat modular architecture depicted in Figure 1. Each of the illustrated elements corresponds to an OMNeT++ compound module. `LoRaNic` and `LoRaGWNic` implement the LoRa and LoRaWAN layers in the device and in the gateway situated inside the satellite. The `packetForwarder` and `ISLpacketForwarder` modules take care of multiplexing and relaying LoRaWAN frames to and from the LoRa radio and the ground station link (available in satellite `[15]`), and the ISLs links (up to 4 depending on the satellite position on the grid), respectively. When forwarded to the ground station, the LoRaWAN frame is encapsulated in an UDP datagram, which is then forwarded

†. Available at: `https://gitlab.inria.fr/jfraire/florasat`.
‡. Home page: `https://omnetpp.org/`
§. Available at `https://flora.aalto.fi/`.
¶. See scenario animation in this video: `https://youtu.be/Xa5EITAMqaQ`
‖. Home page: `https://www.agi.com/products/stk`

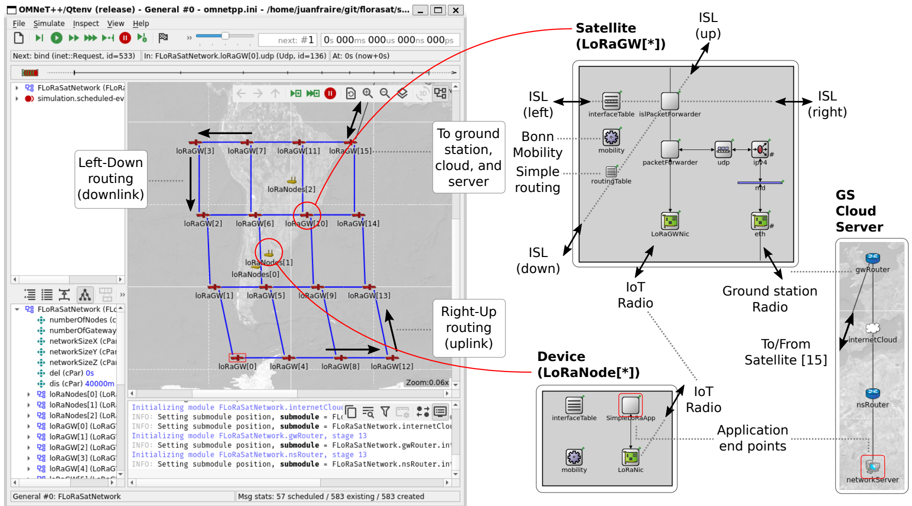

FIGURE 1: FLoRaSat user interface (left) and architecture (right).

to the Internet, to be finally delivered to LoRaWAN network server. Other modules like `mobility`, `interfaces`, and `routingTable` elements leveraged from INET are illustrated in Figure 1. Finally, a visualizer module takes care of the world-map graphics and a `LoRaMedium` component affects LoRa frames with channel impairments (free space loss channel model in this first release).

Metrics. FLoRaSat disposes of a plethora of metrics, such as transmission/reception frame count, collision tracking, or end-to-end latency. We also present LoRaWAN-based DTS-IoT specific metrics in the following section.

3 Preliminary Results

In this section we present preliminary results of the 16-satellite scenario (ISL data rate set to 100 Kbps) for varying device count. Traffic model is set to generate one LoRa frame of 20 B, (spreading factor 12) every 40-120 seconds with a uniform distribution (quite close to the 1% duty cycle limit). The simulation runs for 1 hour, enough for the satellite fleet to pass-by the circular region with up to 1500 devices.

While analysing the first FLoRaSat simulations, we found that LoRaWAN technology operates on tight latency constraint between the server and gateway, when downlink messages are sent to the device (*i.e.*, ACKs). As per LoRaWAN specification, devices will open two one-second reception windows precisely one and two seconds after the uplink attempt concluded. But LoRaWAN gateways are not expected to buffer data, instead, it is the network server who should timely send the downlink data so that it properly fits in the device's reception windows. Because the DTS-IoT scenario involves multiple LEO satellites, and multi-hop ISL routing, accomplishing such a precise timing proved to be challenging. To keep track of this phenomena, we define and study a *Downlink Packet Arrival Delay Since Beginning of the Rx Window* (DPAD) metric.

The left plot in Figure 2 presents the drop rate in the uplink and the downlink (due to collisions), and the resulting transmission failure (when any of the two fails). Results show that 80% drop rate can be expected with 200 nodes under the most demanding traffic load. DPAD for decoded downlink ACKs destined to the local node (i.e., local traffic) ** are presented in the right plot of Figure 2. Results show that 100 kbps in the ISL (and the custom *right-up-down-left* routing) are enough to accommodate a multi-hop delivery of the downlink message without exceeding 0.12 seconds (which fits the 1-second reception window).

Figure 3 presents two example parametric studies with FLoRaSat. The right plot study the varation of DPAD for different ISL data rates. It confirms that DPAD approximates 0.5 seconds for data rates below

∗∗. Given the multi-gateway condition of the scenario, we found that under the most demanding traffic load, multiple nodes might open the reception window simultaneously, while hooking into a downlink message preamble destined to another node. This is peculiar observation possibly affecting terrestrial LoRaWAN we plan to study in further detail in future research.

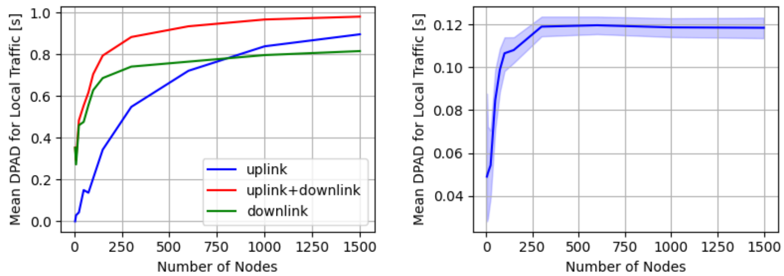

FIGURE 2: Drop Rate (left) and mean DPAD for frames destined to the local node (right) (shadow: 95% CI)

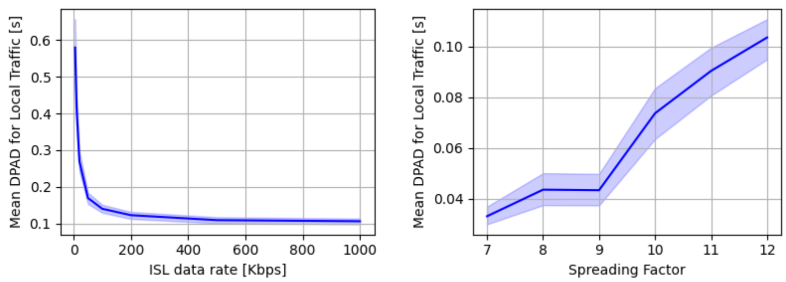

FIGURE 3: Mean DPAD for local traffic for varying ISL data rate (left) and for varying SF (right) (shadow: 95% CI)

10 kbps ††. The right plot of Figure 3 illustrates the impact on DPAD if other spreading factors (SFs) are used in the downlink. Of course, lower SFs (more aggressive modulation) render shorter frames, and thus reduced end-to-end latencies along the multi-hop LEO path.

4 Outlook

Even though the preliminary nature of the results presented for this first version of FLoRaSat, this promising satellite IoT simulator already enabled non-intuitive observations such as 1) the effect of DPAD and 2) the issue of detecting preambles from frames destined to other nodes (in a multi-gateway case). While we keep deepening on these end-to-end satellite IoT aspects, we envision a fruitful development path towards a feature-rich FLoRaSat including the new LoRa modulation (long-range frequency hopping spread spectrum, LR-FHSS) along side with the NB-IoT protocol stack. Furthermore, interesting analysis are already undergoing by the authors, such as LoRaWAN Class-A and Class-B comparison in the DTS-IoT context, the consideration of a GEO relay satellite, and the development of novel scheduling and learning techniques that profit from the predictable and repetitive characteristics of satellite IoT. ‡‡

References

[CCG+21] Marco Centenaro, Cristina E Costa, Fabrizio Granelli, Claudio Sacchi, and Lorenzo Vangelista. A survey on technologies, standards and open challenges in satellite IoT. *IEEE Communications Surveys & Tutorials*, 2021.

[SPDF18] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. Adaptive configuration of lora networks for dense iot deployments. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.

[VFH21] Kai Vogelgesang, Juan Andrés Fraire, and Holger Hermanns. Uplink transmission probability functions for lora-based direct-to-satellite iot: A case study. In *IEEE GLOBECOM*, 2021.

††. Not a surprisingly low data rate for ISLs in resource-constrained CubeSats on link ranges of thousands of kilometres.

‡‡. This research has received financial support from the project ANR-21-CE25-0002-01.