

# TP Simulation d'un Essaim de Nanosatellites

## SN – TP Réseaux Non Terrestres

### Contexte

Dans le cadre des TP du cours *Essaims de Nanosatellites*, vous allez manipuler un outil de simulation développé en Python afin de modéliser et analyser les communications entre 100 nanosatellites en orbite autour de la Lune.

Cet essaim vise à construire un réseau distribué pour des applications d'interférométrie radio. Le système est fortement hétérogène et dynamique, avec des contraintes de connectivité, de congestion et de résilience.

Les simulations sont réalisées via un **notebook interactif** disponible sur le dépôt GitHub : <https://github.com/tp3asn/tp-essaim-nanosatellites>

### Objectifs pédagogiques

- Comprendre les principes de communication dans un essaim de nanosatellites (ISL, portée, dynamique orbitale).
- Analyser la structure du réseau formé par les satellites (graphe dynamique).
- Exploiter des stratégies de découpage de réseau (clustering, sampling).
- Mesurer les performances réseau : degré, densité, chemins, latence, redondance.

### Organisation du TP

Le TP est entièrement encapsulé dans le notebook interactif `simulator.ipynb`, disponible sur GitHub.

**Tout le code est déjà implémenté** : vous n'avez pas à développer d'algorithmes. Votre mission est d'**explorer, analyser, visualiser et comprendre** les mécanismes mis en jeu.

Le TP est découpé en plusieurs parties, directement accessibles dans le notebook `simulator.ipynb`.

#### Partie 1 – Observation dynamique de l'essaim

Vous observerez les trajectoires orbitales de 100 satellites sur 10 000 pas de temps, et visualiserez les configurations d'essaim à différents instants.

**Objectif** : comprendre la mobilité relative et la densité spatiale de l'essaim.

#### Partie 2 – Connectivité et graphes dynamiques

Vous construirez un graphe à partir des connexions inter-satellites selon une portée donnée (par défaut : 30 km). Le graphe change au cours du temps.

**Exploration proposée** :

- Comparer les graphes pour différentes portées (20, 30, 50 km),
- Observer comment la topologie évolue avec le temps,
- Identifier des phases de forte ou faible connectivité.

## Partie 3 – Analyse des métriques de graphe

Les métriques suivantes seront définies dans le sujet (*voir Annexe A*) et calculées automatiquement par le notebook :

- **Degré moyen (Average Degree)**,
- **Coefficient de clustering (ACC)**,
- **Densité du graphe (GD)**,
- **Diamètre du graphe**,
- **Temps inter-contact (ICT)**.

**Objectif :** relier ces métriques à des phénomènes réels : connexions intermittentes, congestion possible, nœuds critiques...

## Partie 4 – Stratégies de subdivision (algorithmes)

Deux stratégies sont fournies dans le code comme vu dans le cours :

- **Clustering** : regrouper les nœuds proches selon leurs connexions locales.
- **Sampling** : créer des sous-ensembles représentatifs de l'ensemble du réseau.

L'implantation de ces algorithmes est donnée en *Annexe B*

**Exploration :** utilisez la fonction de visualisation 2D du graphe *après division* pour observer la répartition des nœuds par groupe. Cela vous permettra de mieux interpréter l'effet de chaque algorithme.

## Partie 5 – Simulation du trafic et redondance

Chaque satellite transmet 5 Gbits à tous les autres. Vous observerez :

- L'impact de la subdivision sur la charge réseau,
- La distribution des flux entre nœuds,
- Les situations de congestion ou d'optimisation.

## Évaluation de l'équité et de la résilience

En complément de l'analyse topologique, ce TP propose d'évaluer :

- **L'équité des algorithmes de division** : chaque stratégie de subdivision doit être analysée en fonction de la distribution des métriques entre les groupes (taille, densité, degré moyen...).
- **La résilience du système** : nous analysons la robustesse du réseau face à des perturbations comme la perte de lien ou de nœud.

Les **métriques de résilience et robustesse** sont décrites en *Annexe C*, notamment :

- Le **taux de redondance utile** (nombre de chemins alternatifs),
- La **efficacité du réseau**
- La **coût de routage**,
- La **criticité des nœuds**

## Utilisation du Notebook

- Lancer : `jupyter notebook simulator.ipynb`
- Installer les dépendances : `pip install -r requirements.txt`
- Modifier uniquement les paramètres définis (portée, stratégie, pas de temps, etc.)
- Utiliser les fonctions de visualisation pour illustrer vos observations.

## Annexe A – Définitions mathématiques des métriques de graphe

### Métriques de graphe – Définitions formelles

Dans ce TP, les graphes inter-satellites sont modélisés comme des graphes non orientés  $G = (V, E)$ , où :

- $V$  est l'ensemble des nœuds (satellites),
- $E$  est l'ensemble des arêtes (liaisons inter-satellites actives à un instant donné).

Les métriques suivantes sont utilisées pour analyser les propriétés structurelles du réseau. Toutes sont calculables via la bibliothèque **NetworkX** en Python, utilisée dans le fichier `metrics.py`.

**1. Degré moyen (Average Degree)** Le degré  $d_i$  d'un nœud  $i \in V$  est le nombre de voisins directs connectés à  $i$ . Le degré moyen est défini comme :

$$AD(G) = \frac{1}{|V|} \sum_{i \in V} d_i$$

Dans NetworkX : `nx.degree(G)`, puis moyenne.

**2. Coefficient de clustering moyen (Average Clustering Coefficient)** Pour chaque nœud  $i$ , le coefficient de clustering  $C_i$  est le ratio entre le nombre de liens entre les voisins de  $i$  et le nombre maximal de tels liens possibles :

$$C_i = \frac{2 \cdot |\{(u, v) \in E : u, v \in N(i)\}|}{d_i(d_i - 1)}$$

où  $N(i)$  est l'ensemble des voisins de  $i$ . Le coefficient de clustering moyen est :

$$ACC(G) = \frac{1}{|V|} \sum_{i \in V} C_i$$

Dans NetworkX : `nx.average_clustering(G)`.

**3. Densité du graphe (Graph Density)** La densité mesure la proportion de connexions réelles par rapport au nombre maximum possible :

$$GD(G) = \frac{2|E|}{|V|(|V| - 1)}$$

Dans NetworkX : `nx.density(G)`.

**4. Diamètre du graphe** Le diamètre est la plus grande longueur de tous les plus courts chemins entre paires de nœuds connectés :

$$Dia(G) = \max_{i, j \in V} d(i, j)$$

où  $d(i, j)$  est la distance (en nombre de sauts) entre  $i$  et  $j$ .

Dans NetworkX : `nx.diameter(G)` (attention : nécessite que  $G$  soit connexe).

**5. Temps inter-contact (Inter-Contact Time, ICT)** Soit un lien inter-satellite  $(i, j)$ , le temps inter-contact est la proportion de temps pendant laquelle ce lien est inactif sur la période d'observation :

$$\text{ICT}_{ij} = \frac{T_{\text{absents}}}{T_{\text{total}}}$$

où  $T_{\text{absents}}$  est le nombre d'instants où  $(i, j) \notin E$ , et  $T_{\text{total}}$  est la durée totale simulée. Cette métrique est spécifique au contexte dynamique du TP, et calculée via des fonctions personnalisées dans le notebook.

Pour aller plus loin, Vous pouvez consulter la documentation officielle NetworkX : [https://networkx.org/documentation/stable/auto\\_examples/index.html](https://networkx.org/documentation/stable/auto_examples/index.html)

## Annexe B – Algorithmes de division de graphe

Dans le cadre de ce TP, trois stratégies de division de l'essaim de satellites sont proposées. Ces algorithmes sont déjà implémentés dans la classe **Swarm** dans le fichier **Swarm\_sim.py**. Chaque stratégie permet de former un découpage de l'essaim en  $x$  sous-groupes.

### 1. RND – Random Node Division

**Principe :** chaque satellite choisit de manière aléatoire un numéro de groupe entre 0 et  $x - 1$ , avec probabilité uniforme. Tous les nœuds ayant choisi le même identifiant sont regroupés.

**Caractéristiques :**

- Algorithme *distribué aléatoire*,
- Complexité  $O(1)$ ,
- Répartition rapide mais sans garantie de connectivité dans les groupes.

**Implémentation simplifiée :**

Algorithme 1 : Random Node Division (RND)	
<b>Data :</b> $G_t = (V_t, E_t)$ ;	/* Graphe représentant l'essaim */
<b>Data :</b> $0 < x \leq  V_t $ ;	/* Nombre de groupes */
<b>for</b> $u \in V_t$ <b>do</b>	
$u.\text{group} \leftarrow \text{choice}([0; x - 1])$ ;	/* Attribution du groupe */
<b>end</b>	

FIGURE 1 – Algorithme RND – Division aléatoire des nœuds [1]

### 2. MIRW – Multiple Independent Random Walks

**Principe :** on sélectionne  $x$  nœuds sources, puis on lance  $x$  marches aléatoires indépendantes dans le graphe. Chaque marche forme un groupe en explorant les voisins libres.

**Caractéristiques :**

- Algorithme *d'exploration ordonnée*,
- Complexité  $O(n/x)$ ,

— Les marches évitent les doublons et assurent une certaine compacité des groupes.  
**Implémentation**

### 3. FFD – Forest Fire Division

**Principe :** basé sur la propagation, comme un feu de forêt. On sélectionne  $x$  sources, puis on propage un feu (groupe) à partir de chaque source avec une probabilité  $p$ , créant ainsi une diffusion épidémique.

**Caractéristiques :**

— Algorithme *d'exploration par propagation*

**Implémentation**

---

**Algorithme 3 :** Forest Fire Division (FFD)

---

```

Data :  $G_t = (V_t, E_t)$  ; /* Graphe représentant l'essaim */
Data :  $0 < x \leq |V_t|$  ; /* Nombre de groupes */
Data :  $0 < p < 1$  ; /* Probabilité de propagation */
 $src \leftarrow \text{sample}(V_t, x)$  ; /* Sélection de  $x$  sources */
 $groups \leftarrow \{\{u\} \mid u \in src\}$  ; /* Formation des groupes */
 $free\_nodes \leftarrow V_t \setminus src$  ; /* Nœuds sans groupe */
 $burning\_nodes \leftarrow src$  ; /* Nœuds actifs */
while  $free\_nodes \neq \emptyset$  do
  for  $u \in burning\_nodes$  do
     $free\_neighbors \leftarrow free\_nodes \cap u.neighbors$ 
    if  $free\_neighbors \neq \emptyset$  then
       $nodes \leftarrow \text{propagate}(free\_neighbors, p)$  ; /* Propagation */
    else
       $nodes \leftarrow \{\text{choice}(free\_nodes)\}$  ; /* Saut aléatoire */
    end
     $group = groups[u]$  ; /* Groupe auquel appartient  $u$  */
     $group \leftarrow group \cup nodes$  ; /* Ajout des nœuds au groupe */
     $burning\_nodes \leftarrow burning\_nodes \cup nodes$  ; /* Nouveaux nœuds actifs */
     $free\_nodes \leftarrow free\_nodes \setminus nodes$ 
  end
end

```

---

FIGURE 3 – Algorithme FFD -exploration [1]

## 1 Annexe C - Métriques pour évaluer la Robustesse et la Résilience du réseau

L'analyse de la fiabilité d'un réseau distribué tel qu'un essaim de nanosatellites repose sur des métriques qui évaluent à la fois sa **robustesse structurelle**, sa **résilience aux pannes**, et son **efficacité de transmission**. Nous présentons ici un résumé des principales métriques utilisées dans ce TP, inspirées des travaux de **Evelyn Akopyan** sur la fiabilité des architectures réseau spatiales [1].

### Métriques de robustesse

— **Robustesse de flux**  $F_t(G)$  : proportion de paires de nœuds connectées par au moins un chemin. Elle indique si l'ensemble du réseau est globalement accessible.

- **Coût de routage**  $R_t(G)$  : somme des longueurs des plus courts chemins entre tous les nœuds. Plus ce coût est faible, plus le réseau est économe en ressources.
- **Efficience réseau**  $\Theta_t(G)$  : moyenne des inverses des longueurs de chemin  $1/d(u, v)$ . Elle mesure la rapidité et compacité des communications.

## Métriques de résilience

- **Redondance des chemins**  $\Psi_t(G)$  : nombre moyen de chemins alternatifs entre les nœuds. Plus il y en a, plus le réseau est tolérant aux coupures.
- **Disparité**  $\Delta_t(G)$  : diversité des sommets traversés dans les différents chemins entre deux nœuds. Une forte disparité diminue le risque de congestion.
- **Criticité**  $C_t(G)$  : ensemble des nœuds très fréquemment traversés dans les chemins (forte centralité). Un réseau fiable minimise le nombre de ces nœuds critiques.

Ces métriques permettent d'évaluer les effets des algorithmes de division (RND, MIRW, FFD) sur la performance et la tolérance aux pannes du système. Pour des détails théoriques et démonstrations formelles, se référer à :

Evelyn Akopyan. *Fiabilité de l'architecture réseau des systèmes spatiaux distribués sur essaims de nanosatellites*. Université de Toulouse, 2024. hal-04796136.

## Références

- [1] E. Akopyan. *Fiabilité de l'architecture réseau des systèmes spatiaux distribués sur essaims de nanosatellites*. Université de Toulouse, 2024. Thèse de doctorat.  
Disponible en ligne : <https://theses.hal.science/tel-04796136>