THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :
l'Institut National Polytechnique de Toulouse (Toulouse INP)

Présentée et soutenue le *18/10/2024* **par :**

Evelyne AKOPYAN

Fiabilité de l'Architecture Réseau des Systèmes Distribués sur Essaims de Nanosatellites

---

JURY

Marcelo Dias de Amorim	Directeur de recherche (CNRS), Sorbonne Université	Examinateur
Isabelle Guérin-Lassous	Professeure, Université Claude Bernard Lyon 1	Examinatrice
Razvan Stanica	Maître de conférence (HDR), INSA Lyon	Rapporteur
Nathalie Mitton	Directrice de recherche, INRIA Lille	Rapporteuse
Riadh Dhaou	Professeur, Toulouse INP	Directeur de thèse
Emmanuel Lochin	Professeur, ENAC	Co-directeur de thèse

École doctorale et spécialité :
MITT : Informatique et Télécommunications
Unité de Recherche :
Laboratoire des Télécommunications Spatiales et Aéronautiques (TéSA)
Directeur(s) de Thèse :
Riadh DHAOU et *Emmanuel LOCHIN*
Rapporteurs :
Razvan STANICA et *Nathalie MITTON*

Remerciements

On pense souvent que la thèse est un travail individuel, et qu'on peut se retrouver livré à soi-même pendant plusieurs années. Mais en réalité, c'est tout l'inverse : en plus de l'équipe d'encadrement, d'innombrables interactions contribuent à la thèse, et créent une expérience inoubliable.

J'aimerais donc tout d'abord remercier mes encadrants de thèse, Riadh Dhaou et Emmanuel Lochin, pour m'avoir accompagnée et soutenue au cours de ces trois années qui sont passées bien vite. Vos précieux conseils et vos encouragements m'ont plusieurs fois évité de me retrouver comme un lapin au milieu de l'autoroute. Je remercie également Jacques Sombrin et Bernard Pontet pour leur expertise, leur motivation et l'intérêt qu'ils ont porté à mes travaux de recherche. Sans vous tous, cette aventure n'aurait pas pu avoir lieu.

Je remercie chaleureusement mes rapporteurs de thèse, Razvan Stanica et Nathalie Mitton, pour le temps passé à évaluer mon manuscrit de thèse. Je tiens aussi à remercier Marcelo Dias De Amorim et Isabelle Guérin-Lassous pour avoir complété mon jury en tant qu'examinateurs. Merci à vous quatre d'avoir bravé les conditions météorologiques, quelque peu humides, pour assister à ma soutenance.

Un immense merci à mes équipes TéSA et RMESS, pour m'avoir accueillie dans un cadre de travail particulièrement agréable et convivial. J'ai beaucoup apprécié nos petits moments de la vie de tous les jours, ceux qui étaient productifs comme ceux qui ne l'étaient vraiment pas. J'aimerais en particulier remercier Gaston, Hamish, Youssra, Paul, Thomas, Jade, Antonin, Benoît, et tous les autres pour m'avoir supporté pendant ces années. Et Jihanne, pour tes pâtes au sucre dont je n'ai toujours pas vu la couleur (mais c'est très bien ainsi).

Je tiens également à remercier les doctorants de Manu : Victor, Matthieu, Cyril, Louis et Marina, pour les bons moments qu'on a passé ensemble, et les projets sportifs un peu fous qui en ont découlé, pour le meilleur et surtout pour le pire.

Je remercie aussi mes chiens, Silex et Unix, pour m'avoir apporté un soutien psychologique indéniable, sans même s'en rendre compte. Bons chiens.

Pour finir, j'aimerais remercier mes parents, Arthur et Eugénie, ainsi que ma soeur Olga, pour leur soutien inconditionnel au cours de ces années de thèse. Un immense merci à mon copain, Luc, pour m'avoir suivie à Toulouse, pour m'avoir encouragée et pour avoir toujours cru en moi. Je n'y serais pas arrivée sans vous.

Table des matières

Liste des figures **vi**

Liste des tableaux **vii**

Liste des abréviations **ix**

1 Introduction **1**

1.1 Contexte scientifique 1

1.1.1 Un petit pas pour l'observation de l'Espace... 1

1.1.2 ... Un grand pas pour les Systèmes Spatiaux Distribués . . . 3

1.2 Contributions de la thèse 5

1.2.1 Problèmes rencontrés 5

1.2.2 Notre solution : diviser pour mieux régner 6

1.3 Liste des publications 8

2 État de l'art **11**

2.1 Les architectures de DSS 12

2.1.1 Constellation 12

2.1.2 Trailing 17

2.1.3 Satellite fractionné 18

2.1.4 Essaim 20

2.1.5 Récapitulatif 25

2.2 Sur Terre, la grande famille des MANET 26

2.2.1 Présentation 26

2.2.2 Cas particuliers de réseaux MANET 27

2.2.3 Topologies des MANET 32

2.2.4 Stratégies de routage 34

2.2.5 L'intérêt des MANET pour les essaims de satellites 37

2.3 Comment fiabiliser les systèmes distribués 39

2.3.1 Les quatre types de pannes 40

2.3.2 Mécanismes de robustesse 41

2.3.3 Mécanismes de résilience 44

2.3.4 Conclusion 46

3 Caractérisation du réseau de nanosatellites 49
3.1 Principe de fonctionnement 50
3.2 Description du modèle 50
3.2.1 Modélisation par la théorie des graphes 51
3.2.2 Hypothèses de communication 52
3.2.3 Description du dataset 53
3.3 Propriétés du réseau 55
3.3.1 Mobilité de l'essaim 55
3.3.2 Densité des noeuds 57
3.3.3 Disponibilité des ISL 61
3.4 Conclusions 64

4 Division équitable de graphe 65
4.1 Motivations 66
4.2 Algorithmes de division 67
4.2.1 Random Node Division (RND) 67
4.2.2 Multiple Independent Random Walks (MIRW) 69
4.2.3 Forest Fire Division (FFD) 70
4.2.4 Le problème du clustering 71
4.3 Evaluation de l'équité de division 72
4.3.1 Critères d'évaluation 72
4.3.2 Protocole d'évaluation 73
4.3.3 Résultats 74
4.4 Conclusions 77

5 Fiabilité du système distribué 79
5.1 Appréciation de la fiabilité réseau 80
5.1.1 Métriques de robustesse 80
5.1.2 Métriques de résilience 82
5.2 Impact de la division sur la fiabilité réseau 84
5.2.1 Protocole expérimental 85
5.2.2 Fiabilité du graphe non divisé 86
5.2.3 Résultats 88
5.3 Conclusion 94

6 Conclusion 97

Bibliographie 101

Liste des figures

1.1 Vue aérienne des interféromètres ALMA et VLTI 2
1.2 Schémas de l'échange de données dans les interféromètres 3
1.3 Schéma du mode opératoire de l'essaim en orbite 4

2.1 Orbite et zone de couverture des satellites Iridium (source : [89]) . . 15
2.2 Orbite des satellites Galileo (source : [2]) 16
2.3 Cartographie par la constellation Pléiades 17
2.4 Satellites des missions PRISMA et TANDEM-X 18
2.5 Satellite fractionné du projet ANDESITE, lancé en 2020 19
2.6 L'essaim OLFAR en orbite lunaire (source : [28]) 22
2.7 Exemple de topologie urbaine VANET 28
2.8 Exemple de topologie FANET pour la surveillance des forêts 29
2.9 Exemple de topologie SANET près d'une côte 30
2.10 Niveaux de maillage d'une topologie de MANET 32

3.1 Représentation de la trajectoire de l'essaim en orbite lunaire et ses variations de topologies 54
3.2 Vitesse de déplacement de l'essaim sur une révolution 56
3.3 Evolution de la mobilité relative des satellites sur une révolution . . 57
3.4 Analyse de l'évolution de la force des sommets 58
3.5 Analyse de l'évolution du coefficient de clustering local 60
3.6 Distribution des ICT théoriques et réels sur une révolution 61
3.7 Distribution du temps d'isolement des noeuds sur une révolution . . 62
3.8 Evolution du nombre de composantes connexes au cours du temps . 63

4.1 Illustration des techniques de clustering, d'échantillonnage et de division de graphe 68
4.2 RMSE de la distribution des métriques de densité de noeuds 75
4.3 RMSE de la distribution de la taille des groupes 76
4.4 Evolution du nombre de flux en fonction du nombre de groupes . . . 77

5.1 Calcul de la disparité de la paire de sommets $(0, 5)$ 84
5.2 Evolution temporelle des métriques de fiabilité du graphe de référence 86
5.3 Impact sur la robustesse de flux 89
5.4 Impact sur le coût de routage 90
5.5 Impact sur l'efficience réseau 91

5.6 Impact sur la redondance des chemins 91
5.7 Impact sur la disparité des chemins 92
5.8 Impact sur la criticité des noeuds 93

Liste des tableaux

2.1 Classification des satellites par masse 12
2.2 Liste des principales constellations en orbite terrestre 14
2.3 Liste des principales exigences du système NOIRE 24
2.4 Comparaison des caractéristiques des architectures de DSS 26
2.5 Comparaison des VANET, FANET et SANET avec les essaims de satellites 31
2.6 Récapitulatif des stratégies de routage MANET 37

3.1 Paramètres orbitaux de l'essaim de nanosatellites 53
3.2 Paramètres généraux du dataset 55

4.1 Paramètres de la simulation pour l'évaluation de l'équité 74

5.1 Paramètres de la simulation pour l'évaluation de l'impact 86

Liste des abréviations

ABR *Associativity-Based Routing*, Routage Basé sur l'Associativité

AIS *Automatic Identification System*, Système d'Identification Automatique

ALMA *Atacama Large (sub-)Millimeter Array*, Grand Réseau d'Antennes (sub-)Millimétriques d'Atacama

AODV *Ad-hoc On-demand Distance Vector*, Vecteur de Distance Ad-hoc sur Demande

BFT *Byzantine Fault Tolerance*, Tolérance aux Pannes Byzantines

BS *Base Station*, Station de Base

DARPA *U.S. Defense Advanced Research Projects Agency*

DSDV *Destination-Sequenced Distance Vector*, Vecteur de Distance Séquencé par Destination

DSR *Dynamic Source Routing*, Routage de Source Dynamique

DSS *Distributed Space System*, Système Distribué Spatial

DTN *Delay Tolerant Network*, Réseau Tolérant à la Latence

DTN *Delay-Tolerant Network*, Réseau Tolérant à la Latence

FANET *Flying Ad-hoc Network*, Réseau Ad-hoc Volant

FFD *Forest Fire Division*, Division par Feu de Forêt

FSR *Fisheye State Routing*, Routage "Oeil de Poisson" par Etat

GEO *Geostationary Orbit*, Orbite Géostationnaire

GNSS *Global Navigation Satellite System*, Système de Positionnement par Satellites

ICT *Inter-Contact Time*, Temps d'Inter-Contact

IGSO *Inclined Geosynchronous Orbit*, Orbite Géosynchrone Inclinée

ISL *Inter-Satellite Link*, Liaison Inter-Satellite

LEO *Low Earth Orbit*, Orbite Terrestre Basse

LoS *Line of Sight*, Ligne de Vue directe

MANET *Mobile Ad-hoc Network*, Réseau Ad-hoc Mobile

MEO *Medium Earth Orbit*, Orbite Terrestre Moyenne

MIRW *Multiple Independent Random Walks*, Multiple Marches Aléatoires Indépendantes

MLE *Maximum Likelihood Estimator*, Estimateur de Vraisemblance Maximale

NOIRE Nanosatellites pour un Observatoire Interférométrique Radio dans l'Espace

OLFAR *Orbiting Low Frequency Antennas for Radio Astronomy*, Antennes Orbitales Basse Fréquence pour l'Astronomie Radio

OLSR *Optimized Link-State Routing*, Routage par Etat des Liens Optimisé

RFI *Radio-Frequency Interferences*, Interférences Radioélectriques

RMSE *Root-Mean-Square Error*, Racine de l'Erreur Quadratique Moyenne

RND *Random Node Division*, Division de Noeuds Aléatoire

SANET *Sea Ad-hoc Network*, Réseau Ad-hoc Maritime

SNR *Signal to Noise Ratio*, Rapport Signal-à-Bruit

SSO *Sun-Synchronous Orbit*, Orbite Héliosynchrone

TORA *Temporally Ordered Routing Algorithm*, Algorithme de Routage Temporellement Ordonné

UAV *Unmanned Aerial Vehicle*, Véhicule Aérien sans Pilote (drone)

UT *Unit Telescope*, Télescope Unitaire

V2I *Vehicle to Infrastructure*, Communication Véhicule à Infrastructure

V2V *Vehicle to Vehicle*, Communication Véhicule à Véhicule

VANET *Vehicular Ad-hoc Network*, Réseau Ad-hoc Véhiculaire

VLTI *Very Large Telescope Interferometer*, Interféromètre du Très Grand Télescope

WSN *Wireless Sensor Networks*, Réseaux de Capteurs Sans-fil

ZHLS *Zone-based Hierarchical Link State*, Etat des Liens Hiérarchique basé sur la Zone

ZRP *Zone Routing Protocol*, Protocole de Routage par Zone

Chapitre 1

Introduction

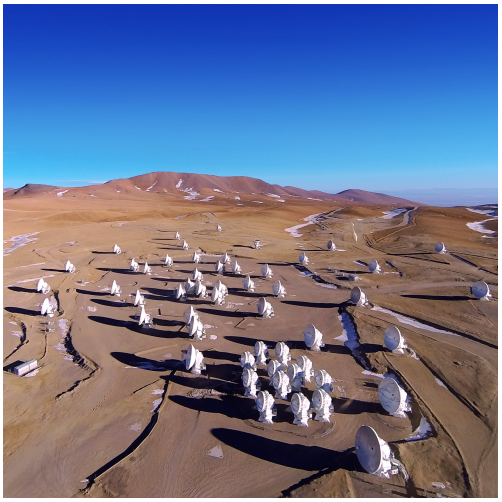
1.1 Contexte scientifique

Depuis la nuit des temps, l'observation de l'Espace nous met des étoiles plein les yeux. Les instruments pour l'observer ont connu une évolution fulgurante : nous souhaitons apporter notre humble pierre à cet édifice scientifique.

1.1.1 Un petit pas pour l'observation de l'Espace...

Depuis de nombreuses années, le domaine de l'observation spatiale s'intéresse de près aux signaux de très basses fréquences, classés en dessous de 100 Méga-Hertz. En effet, ces signaux permettent de cartographier l'Espace lointain, et fournissent d'importantes informations quant à la naissance et aux premiers jours de l'Univers [25]. A ce jour, les télescopes observant ces signaux utilisent des techniques d'interférométrie et se trouvent à la surface de la Terre : on peut notamment citer l'interféromètre ALMA (*Atacama Large (sub-)Millimeter Array*, le Grand Réseau d'Antennes (sub-)Millimétriques d'Atacama) et le VLTI (*Very Large Telescope Interferometer*, l'Interféromètre du Très Grand Télescope), situés tous deux dans le désert d'Atacama au Chili (voir Figure 1.1). Contrairement aux télescopes classiques, les interféromètres sont composés de plusieurs Télescopes Unitaires (UT) qui peuvent combiner leurs données pour former, virtuellement, un télescope plus large, offrant ainsi une précision inégalée pour l'observation des faibles signaux nous parvenant de l'Espace lointain. Par exemple, le VLTI comporte 4 UT, et l'ALMA dispose d'un vaste champ de 66 UT.

Pour générer des images à partir des signaux basses fréquences, l'interféromètre oriente tous ses UT vers un point de l'Espace, puis chaque UT capte un signal qui est transmis à l'unité de calcul de l'interféromètre, généralement appelée *corrélateur*. Le corrélateur combine ensuite tous les signaux reçus par les UT en effectuant des corrélations croisées, d'où son nom. La matrice de corrélations croisées ainsi obtenue donne l'image basse fréquence observée par l'interféromètre. Bien que le fonctionnement puisse paraître assez simple au premier abord, la difficulté de la génération d'images repose sur deux facteurs : la synchronisation des UT et la gestion



(a) ALMA (crédit : ALMA (ESO/NAOJ/NRAO), A. Marinkovic/X-Cam)

(b) VLTI (crédit : ESO)

FIGURE 1.1 – Vue aérienne des interféromètres ALMA et VLTI

des interférences terrestres. En effet, les UT doivent être parfaitement synchrones pour que le corrélateur puisse combiner leurs signaux, au risque d'obtenir des décalages importants sur l'image. On parle tout de même d'une précision de l'ordre de 10^{-12} seconde ! Heureusement, les technologies actuelles permettent déjà d'atteindre de tels niveaux de précision.

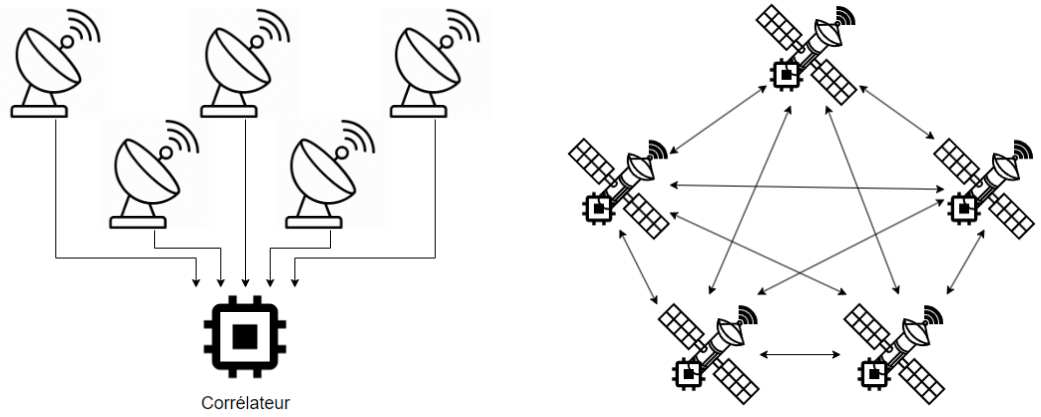
L'autre source d'erreur provient des signaux en eux-mêmes. En effet, les signaux très basses fréquences sont particulièrement sensibles aux Interférences Radioélectriques (*Radio-Frequency Interferences*, RFI) massivement produites par la Terre, ainsi qu'à l'ionosphère : ils sont donc difficilement observables en l'état. C'est la raison pour laquelle les interféromètres se situent dans des zones désertiques et de préférence en haute altitude. Ainsi, le désert d'Atacama est un candidat idéal pour l'installation d'ALMA et du VLTI : ces derniers culminent respectivement à 5058 mètres et 2635 mètres d'altitude, dans l'une des régions les plus arides sur Terre. Cependant, l'observation des signaux en dessous de 10 Méga-Hertz reste difficile, voire impossible en raison de leur absorption par les molécules d'eau. Ainsi, même s'ils sont équipés de modèles atmosphériques pour corriger les signaux reçus, les interféromètres terrestres restent tristement limités par l'environnement.

Pour pallier à ce second problème, des chercheurs du CNES proposent d'observer les signaux très basses fréquences directement depuis l'espace, dans l'objectif de se défaire des effets néfastes des RFI et de l'ionosphère ; mais pour cela, il est nécessaire d'adapter la technologie des interféromètres terrestres à l'environnement spa-

tial. Ainsi, l'étude NOIRE (Nanosatellites pour un Observatoire Interférométrique Radio dans l'Espace) suggère de déployer un Système Spatial Distribué (*Distributed Space System*, DSS) constitué d'un essaim de nanosatellites en orbite autour de la Lune, qui fonctionnerait tel un interféromètre spatial distribué [30]. L'orbite lunaire de l'essaim est l'élément clé permettant de le préserver des effets des RFI et de l'ionosphère.

1.1.2 ... Un grand pas pour les Systèmes Spatiaux Distribués

Le fonctionnement global d'un essaim de nanosatellites est théoriquement similaire au fonctionnement d'un interféromètre terrestre. Chaque nanosatellite embarque les instruments nécessaires pour former un UT. Le nombre de nanosatellites dans l'essaim correspond donc au nombre d'UT de l'interféromètre spatial : plus l'essaim dispose de nanosatellites, plus l'image produite sera détaillée et précise. Le contexte spatial apporte évidemment des spécificités au fonctionnement de cet interféromètre, la plus marquante étant l'absence de corrélateur centralisé : l'idée est de distribuer la puissance de calcul pour que chaque nanosatellite puisse lui-même corréler les signaux reçus.



(a) Interféromètre terrestre (type ALMA) (b) Interféromètre spatial (essaim)

FIGURE 1.2 – Schémas de l'échange de données dans les interféromètres

Cela entraîne une deuxième différence de fonctionnement par rapport à un interféromètre terrestre, car les nanosatellites font à la fois office d'UT et de corrélateur. Et puisque les UT doivent envoyer toutes leurs données au corrélateur pour calcul, dans notre cas, chaque nanosatellite doit échanger ses données avec tous les autres membres de l'essaim, comme illustré sur la Figure 1.2. On peut cependant légitimement se demander pourquoi le calcul est effectué à bord, et non au niveau d'un corrélateur centralisé du segment sol : ce choix est fait en raison des ressources

limitées à bord d'un nanosatellite. En effet, il est moins énergivore d'effectuer des corrélations à bord puis d'envoyer l'image à la Station de Base (*Base Station*, BS) terrestre via un mécanisme de formation de faisceau, plutôt que d'envoyer les signaux bruts individuels captés par les nanosatellites.

La deuxième spécificité, ou plutôt contrainte, est liée au mode opératoire en orbite. En effet, l'observation des signaux et l'envoi de l'image à la BS terrestre ne peuvent se faire que dans des configurations précises. La Figure 1.3 illustre le mode opératoire simplifié de l'essaim en orbite. Une mission spatiale telle que proposée par l'étude NOIRE opère en 4 phases majeures, qui dépendent de l'intensité des RFI perçues (identifiée par le code couleur sur la figure) et de la distance de l'essaim par rapport à la Terre :

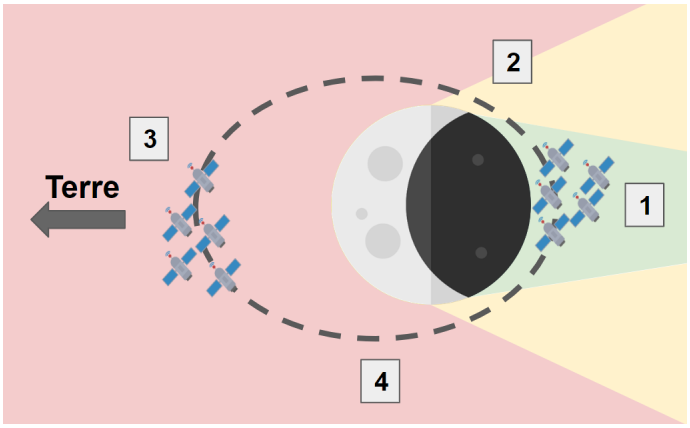


FIGURE 1.3 – Schéma du mode opératoire de l'essaim en orbite

1. Phase d'observation : lorsque l'essaim survole la face cachée de la Lune, il est totalement invisible depuis la Terre : les RFI perçues sont minimales (zone verte) et l'essaim est à la distance maximale de la Terre. Cette configuration est optimale pour que les nanosatellites puissent observer les signaux de l'Espace lointain.
2. Phase de calcul : dès que l'essaim sort de la zone d'observation optimale, il commence à calculer l'image globale à partir des signaux perçus lors de la phase précédente. Pour cela, chaque nanosatellite doit échanger ses données d'observation avec le reste de l'essaim, puis combiner ces données en effectuant des corrélations croisées de chaque signal récupéré. Cette matrice de corrélations croisées correspond à l'image interférométrique globale perçue par l'essaim.
3. Phase de transfert : l'essaim transmet l'image interférométrique globale à la BS terrestre dès qu'il est au plus proche de la Terre sur son orbite.

4. Phase de repos : lors de cette phase, les nanosatellites restent passifs pour récupérer de l'énergie. Ils peuvent également se reconfigurer et apporter des correctifs à leur trajectoire si cela a été demandé par la BS terrestre lors de la phase précédente.

Chacune de ces phases constitue un sujet de recherche à part entière. Aussi, cette thèse se concentre uniquement sur la phase 2, à savoir la phase de calcul. L'objectif est de réussir à transférer un volume très important de données d'observation au sein d'un système distribué, mobile dans l'Espace et ne disposant d'aucune infrastructure additionnelle : ce type de système correspond à un système spatial distribué sur Réseau Ad-hoc Mobile (*Mobile Ad-hoc Network*, MANET).

1.2 Contributions de la thèse

Un interféromètre spatial distribué sur un essaim de nanosatellites constitue un système complexe et peu commun. Sur le plan de la communication intra-essaim, plusieurs problèmes peuvent se poser quant à la mise en place et la configuration du réseau. Nous présentons brièvement la solution proposée dans cette thèse pour résoudre ces problèmes.

1.2.1 Problèmes rencontrés

Lors de la phase de calcul, les nanosatellites commencent par échanger les données collectées lors de la phase précédente avec tous les membres de l'essaim. Ensuite, chaque nanosatellite effectue des corrélations croisées des signaux reçus grâce à son corrélateur embarqué, et génère ainsi une image basse fréquence de l'Espace. Ce principe de fonctionnement est plutôt basique et peut sembler sans risque, si l'on ne prend pas garde au volume colossal de données à transférer ! Maintenant, quitte à faire un doctorat en télécommunications spatiales, autant le prendre en compte et nous intéresser aux problèmes que cela peut engendrer.

En termes de communication, la configuration d'un essaim de nanosatellites en interféromètre spatial peut s'avérer compliquée en raison de l'absence d'infrastructure dans l'Espace. Par conséquent, la communication dans l'essaim se base exclusivement sur des Liaisons Inter-Satellites (*Inter-Satellite Links*, ISL) sans-fil. Bien que les ondes se propagent dans un environnement vide et en ligne de vue directe (*Line of Sight*, LoS), la capacité des ISL reste limitée. De plus, les nanosatellites étant mobiles en orbite et disposant d'une puissance d'émission limitée, rien ne garantit la stabilité d'un ISL dans le temps : un éloignement trop important des nanosatellites peut casser la liaison en pleine communication.

Par ailleurs, en raison de cette puissance d'émission limitée, les nanosatellites ne peuvent pas établir un contact direct avec tous les autres membres de l'essaim. Les paquets de données doivent donc être propagés en multi-saut et routés par les nanosatellites. Cependant, la transmission des paquets de données est très énergivore en raison de leur volume. Si la majorité des transmissions est concentrée sur quelques nanosatellites, ces derniers seront très susceptibles de tomber en panne d'énergie, et cela peut engendrer de graves perturbations dans le réseau. Par conséquent, on s'assurera de la mise en place de mécanismes de robustesse sur les nanosatellites pour éviter de perdre des UT.

Enfin, et comme énoncé plus haut, les données d'observation spatiale peuvent être particulièrement volumineuses et atteindre plusieurs Gigabits de données brutes par nanosatellite en fonction de la fréquence d'échantillonnage des antennes. La propagation simultanée des données dans l'essaim peut poser de sérieux problèmes de congestion sur des ISL à capacité limitée, voire conduire à des pertes de paquets. Ces perturbations se traduisent par une diminution de la précision de l'image basse fréquence : chaque paquet perdu représente une information qui ne fera pas partie de l'image finale.

En résumé, les principales contraintes liées au fonctionnement de l'essaim de nanosatellites reposent sur des questions de *fiabilité*, c'est-à-dire de robustesse et de résilience face aux nombreuses pannes encourues. Existe-t-il des solutions pour améliorer la fiabilité de l'essaim, et ainsi prolonger sa durée de vie ?

1.2.2 Notre solution : diviser pour mieux régner

En politique, il existe depuis longtemps une stratégie permettant de rendre gérable une situation qui semble pourtant hors de contrôle. Cette méthode consiste à affaiblir les forces adverses en créant des divisions et des oppositions, afin d'éviter de faire face à un bloc uni qui représenterait une menace trop importante : de là vient la fameuse maxime attribuée au roi Philippe de Macédoine, père d'Alexandre le Grand : "*divide ut regnes*", "diviser pour régner".

Cette stratégie politique peu scrupuleuse a inspiré une méthode de conception d'algorithmes informatiques, dont le concept est de diviser un gros problème en plusieurs petits problèmes, puis de les régler un à un afin de résoudre le problème initial. Dans cette lignée, nous proposons une méthodologie basée sur la division du réseau de nanosatellites en plusieurs sous-réseaux, l'objectif étant de limiter la propagation des paquets de données au nécessaire. Cette méthodologie est présentée en détails au travers des différents chapitres de cette thèse.

Avant de commencer, nous effectuons une étude approfondie sur les travaux

existants. Le Chapitre 2 consiste en un état de l'art général sur les différentes architectures de DSS, dont les essaims, ainsi que sur les réseaux ad-hoc mobiles auxquels ils s'apparentent. Ce chapitre présente également des mécanismes de fiabilisation applicables à notre cas d'étude.

Les chapitres suivants détaillent les travaux techniques réalisés au cours de la thèse. Dans un premier temps, il convient d'analyser le DSS étudié afin de comprendre ses dynamiques et ses points faibles. Le Chapitre 3 présente une caractérisation détaillée de l'essaim de nanosatellites et des principales propriétés liées à son modèle de mobilité, la densité de ses noeuds et la disponibilité de ses ISL. Ce chapitre met en avant la forte hétérogénéité de l'essaim de nanosatellites en termes de répartition des ressources.

Le Chapitre 4 présente le coeur du sujet, à savoir le mécanisme de division équitable de réseau. Cette division consiste à agréger les noeuds du réseau en plusieurs groupes, sans pour autant les isoler les uns des autres. Nous présentons d'abord trois algorithmes de division : un algorithme de sélection aléatoire, et deux algorithmes d'exploration. Ensuite, nous les analysons en termes d'équité pour déceler celui qui a les meilleures performances sur la répartition équitable des ressources, et sous quelles conditions. Les performances de ces algorithmes sont également comparées à un algorithme de clustering populaire dans la littérature afin d'appuyer nos propos.

Enfin, le Chapitre 5 met en lien les résultats présentés au chapitre précédent avec une application concrète. Factuellement, quel est l'impact d'une division de réseau sur la fiabilité du système ? Nous proposons de répondre à cette question en étudiant la robustesse (capacité à résister aux pannes) et la résilience (capacité à maintenir un niveau de fonctionnement correct en présence de pannes) du DSS. En analysant l'évolution de métriques de robustesse et de résilience avant et après division, nous démontrons que la division équitable de réseau permet de grandement améliorer la fiabilité du système, et ainsi prolonger la durée de vie de l'essaim de nanosatellites.

1.3 Liste des publications

[12] Evelyne Akopyan, Riadh Dhaou, Emmanuel Lochin, Bernard Pontet, Jacques Sombrin. *On the Network Characterization of Nanosatellite Swarms.* In : The 28th IEEE Symposium on Computers and Communications (ISCC), pp.944-949, 2023.

Cet article, publié en conférence internationale à comité de relecture, adresse le problème de la caractérisation du réseau de l'essaim de nanosatellites étudié dans cette thèse. En effet, il existe très peu d'études sur les propriétés réseau des essaims de nanosatellites dans la littérature. Cette contribution vise à combler cette lacune, et constitue le socle des travaux présentés dans cette thèse. Le Chapitre 3 reprend les travaux de notre article de manière plus approfondie.

[11] Evelyne Akopyan, Riadh Dhaou, Emmanuel Lochin, Bernard Pontet, Jacques Sombrin. *Fair Network Division of Nanosatellite Swarms.* In : IEEE 97th Vehicular Technology Conference (VTC2023-Spring), pp.1-5, 2023.

Cet article, publié également en conférence internationale à comité de relecture, présente les mécanismes de division équitable de réseau constituant la principale contribution de cette thèse. Dans la littérature, il existe de nombreux algorithmes de clustering permettant de réduire la consommation énergétique globale d'un système distribué ; cependant, le clustering n'assure pas la répartition équitable des ressources, contrairement à la division réseau présentée dans cet article.

[13] Evelyne Akopyan, Riadh Dhaou, Emmanuel Lochin, Bernard Pontet, Jacques Sombrin. *Division Réseau Equitable dans les Essaims de Nanosatellites.* In : AlgoTel-CoRes, 2024.

Cet article est une version plus récente de l'article précédent, et a été publié en conférence nationale à comité de relecture dans l'objectif de présenter les récents travaux à la communauté réseau française. Il présente de nouveaux résultats sur l'équité de la division : le Chapitre 4 détaille les algorithmes de division proposés, et démontre leur intérêt face aux algorithmes existants.

[14] Evelyne Akopyan, Riadh Dhaou, Emmanuel Lochin, Bernard Pontet, Jacques Sombrin. *Resilient and Robust Strategies for Swarming Ad-hoc Networks.* Soumis à Computer Networks, 2024.

Cet article a été récemment soumis au journal international à comité de relecture *Computer Networks.* Il détaille l'impact de la division équitable de réseau sur la fiabilité du DSS, et le généralise aux systèmes similaires basés sur des réseaux MA-

NET. L'intérêt majeur de cette contribution est de fournir une étude approfondie sur l'effet d'un mécanisme (ici, la division) en analysant conjointement la robustesse *et* la résilience du système. Dans la littérature, ces deux aspects sont généralement étudiés séparément, alors qu'ils influent l'un sur l'autre. Le Chapitre 5 présente les travaux associés à cet article.

Chapitre 2

État de l'art

Sommaire

2.1 Les architectures de DSS 12
2.1.1 Constellation 12
2.1.2 Trailing 17
2.1.3 Satellite fractionné 18
2.1.4 Essaim 20
2.1.5 Récapitulatif 25
2.2 Sur Terre, la grande famille des MANET 26
2.2.1 Présentation 26
2.2.2 Cas particuliers de réseaux MANET 27
2.2.3 Topologies des MANET 32
2.2.4 Stratégies de routage 34
2.2.5 L'intérêt des MANET pour les essaims de satellites 37
2.3 Comment fiabiliser les systèmes distribués 39
2.3.1 Les quatre types de pannes 40
2.3.2 Mécanismes de robustesse 41
2.3.3 Mécanismes de résilience 44
2.3.4 Conclusion 46

Passionnée par tout ce qui a trait au spatial depuis mon enfance, j'ai été captivée par le sujet proposé dans cette thèse. L'idée qu'il serait possible d'obtenir des images encore plus détaillées de l'Espace était très plaisante... Mais parfois, il faut redescendre sur Terre et se poser les bonnes questions : pourquoi étudier les essaims, alors qu'il existe déjà de nombreux DSS parfaitement fonctionnels ? Cette nouvelle formation a-t-elle un intérêt ? Si tel est le cas, par où commencer la recherche ? Si je me cantonnais uniquement aux publications sur les essaims de satellites, cet état de l'art serait tristement court. Et enfin : est-il seulement possible de rendre un tel système fiable ? Je vous propose de vous plonger dans ce chapitre pour découvrir ce qui rend les essaims si spéciaux, si singuliers, et évoquer les systèmes existants dont il est possible de s'inspirer. Car en réalité, avec un peu d'imagination, on peut trouver une multitude de solutions pour fiabiliser un essaim de nanosatellites.

2.1 Les architectures de DSS

On appelle DSS tout système constitué d'au moins deux satellites qui fonctionnent de concert pour accomplir une tâche commune, qui serait difficilement réalisable par un satellite isolé (dit *monolithique*). Un DSS a de nombreux avantages par rapport à un système spatial monolithique : tout d'abord, l'ouverture des satellites peut être combinée afin d'augmenter drastiquement la résolution globale du système, ce qu'il n'est possible de faire que si l'on dispose de plusieurs satellites. La combinaison des satellites permet ainsi d'améliorer la qualité des observations en augmentant le rapport signal-à-bruit (*Signal to Noise Ratio*, SNR) des signaux reçus [4]. La multiplicité des satellites et leur miniaturisation permettent également de prolonger la durée de vie de la mission en diminuant la fraction massique du propergol du système, c'est-à-dire la proportion massique de "carburant" nécessaire à envoyer les satellites sur leur orbite [84].

Avant de continuer sur les architectures de DSS, il convient de faire un point sur la dénomination des satellites : ces derniers sont classifiés selon leur masse. La miniaturisation des satellites consiste donc à réduire leurs dimensions, et donc leur masse, en vue de réduire le coût de production [56]. En effet, plus le satellite est lourd, plus il coûte cher à la fabrication et au lancement. En contrepartie, plus le satellite est léger, plus sa charge utile (instruments embarqués, puissance de calcul) est limitée. La masse et le nombre de satellites sont donc des paramètres primordiaux et indissociables dans le développement d'une architecture de DSS. Le Tableau 2.1 décrit les différentes classes de satellites en fonction de leur masse.

Les DSS peuvent être catégorisés de plusieurs façons en fonction des caractéristiques que l'on cherche à mettre en avant. Dans notre cas, nous classons les DSS en fonction de l'homogénéité des satellites et de la manière dont ils se déplacent les uns par rapport aux autres. On obtient ainsi quatre grandes catégories d'architectures de DSS : les constellations, les formations en trailing, les satellites fractionnés, et enfin les essaims.

Classe	**Femto**	**Pico**	**Nano**	**Micro**	**Mini**	**Gros**
Masse (kg)	0.01-0.1	0.1-1	1-10	10-100	100-1000	>1000

TABLE 2.1 – Classification des satellites par masse

2.1.1 Constellation

La constellation est sûrement la première topologie qui vient en tête lorsqu'on parle d'architecture de DSS. Une constellation est constituée d'un ensemble de sa-

tellites placés sur des orbites distinctes et se déplaçant de manière coordonnée. Le principal intérêt de l'architecture en constellation est la large zone de couverture assurée par les satellites : cette architecture est généralement utilisée pour assurer un service stable sur de grandes étendues géographiques, comme par exemple la géolocalisation ou l'accès à Internet. Cette couverture quasi-globale de la Terre s'explique par la complémentarité des orbites des satellites composant la constellation. Par conséquent, ces satellites sont fortement mobiles les uns par rapport aux autres, mais de manière déterministe : les trajectoires de tous les satellites sont connues à l'avance et peuvent être anticipées.

Les constellations déployées en orbite terrestre sont nombreuses et variées : elles se distinguent par rapport à leur mission, l'altitude de déploiement, au nombre de satellites opérationnels ainsi qu'à la classe (masse) de ces satellites. Le Tableau 2.2 répertorie les principales constellations déployées en orbite terrestre. Une enquête bien plus exhaustive est réalisée en 2020 afin de répertorier les constellations déjà déployées et celles en projet [35]. Cette enquête classifie les constellations selon les critères mentionnés ci-dessus, et met en avant les enjeux actuels et futurs des technologies de constellations. Il est intéressant de noter que les constellations ayant la même fonction ont des caractéristiques similaires :

— les constellations assurant un service de télécommunications sont généralement déployées en orbite terrestre basse (*Low Earth Orbit*, LEO) et possèdent une large flotte de petits satellites ;
— les constellations GNSS sont déployées en orbite terrestre moyenne (*Medium Earth Orbit*, MEO) voire géostationnaire (*Geostationary Orbit*, GEO), parfois inclinée (*Inclined Geosynchronous Orbit*, IGSO). Elles disposent de 20 à 40 satellites de masse avoisinant la tonne ;
— les constellations d'observation de la Terre sont déployées en orbite héliosynchrone (*Sun-Synchronous Orbit*, SSO) et sont constituées de quelques gros satellites.

Les caractéristiques de ces architectures ne sont pas choisies au hasard : intéressons-nous d'un peu plus près aux besoins des télécommunications, de la géolocalisation et de l'observation de la Terre à travers les constellations Iridium, Galileo, et Pléiades.

Iridium, la constellation de la téléphonie

La constellation Iridium est une pionnière dans le domaine des DSS. Le projet est lancé à la fin du XXe siècle [1] par des ingénieurs de l'entreprise Motorola dans le

1. En raison de l'ancienneté de la constellation Iridium, ses satellites sont progressivement remplacés à partir de 2016 par des satellites nouvelle génération baptisés Iridium Next [53].

Fonction	Constellation	Orbite	Nombre de satellites	Classe
Téléphonie et accès Internet	**Iridium**	LEO	66	Mini
	Iridium NEXT	LEO	75	Mini
	OneWeb	LEO	648	Mini
	Starlink	LEO	>6000	Mini/gros
Géolocalisation (GNSS)	GPS	MEO	31	Gros
	Galileo	MEO	24	Mini
	BeiDou	MEO/IGSO/GEO	35	Mini/gros
	GLONASS	MEO	24	Mini
Observation de la Terre	RADARSAT	SSO	3	Gros
	Pléiades	SSO	2	Gros
	A-train	SSO	3	Mini/gros

TABLE 2.2 – Liste des principales constellations en orbite terrestre

but de mettre en place un service de téléphonie par satellite [40]. La constellation est composée de 66 satellites d'environ 500 kilogrammes chacun, répartis tout autour de la Terre en orbite basse à une altitude d'environ 780 kilomètres (voir Figure 2.1).

Expliquons maintenant les choix dans l'architecture d'Iridium. D'abord, les satellites étant déployés en orbite basse, ils nécessitent moins d'énergie et de carburant pour y être placés que sur une orbite plus haute. Ensuite, et toujours en raison de la faible altitude, les satellites ont besoin de moins de puissance à bord pour assurer une bonne qualité de service, ce qui limite leur masse au lancement. Ainsi, la constellation en LEO garantit une large bande passante et une très faible latence, ce qui est essentiel pour un service de télécommunications. Enfin, l'orbite de la constellation Iridium est quasi-polaire, c'est-à-dire que les satellites passent très près des pôles, permettant ainsi de balayer la quasi-totalité de la surface de la Terre. En contrepartie, les satellites ont, individuellement, une plus faible zone de couverture au sol : c'est pour cette raison que les constellations de télécommunications déploient de larges flottes de satellites en orbite afin d'assurer une couverture globale de la Terre.

Le service de téléphonie assuré par Iridium est donc bien plus stable et plus accessible qu'un service de téléphonie terrestre, qui se repose historiquement sur des liaisons câblées : en effet, un service au sol est plus difficile à mettre en place dans des zones reculées ou difficilement accessibles (comme des régions montagneuses). La constellation Iridium a ainsi apporté un réel progrès dans l'accessibilité aux télécommunications à l'échelle mondiale.

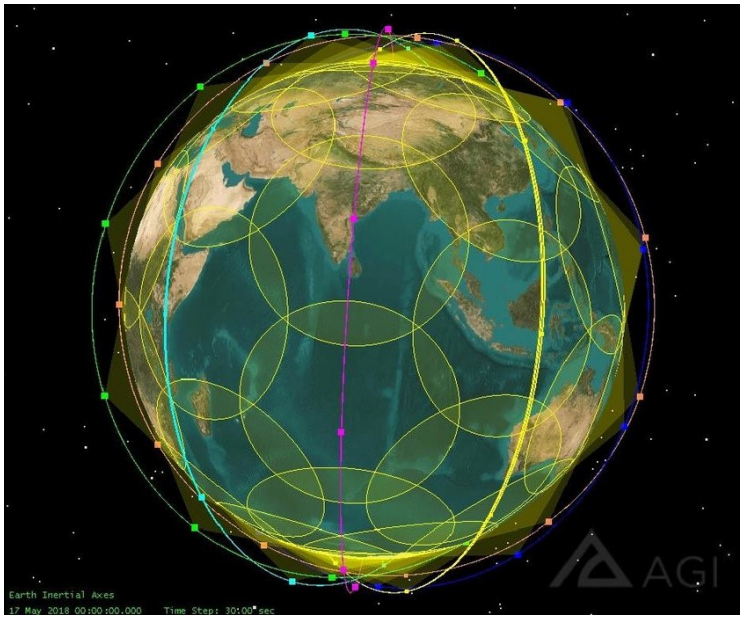


FIGURE 2.1 – Orbite et zone de couverture des satellites Iridium (source : [89])

Galileo, la constellation de la navigation

La constellation Galileo est un ambitieux projet de l'Agence Spatiale Européenne (ESA) qui permettra, à terme, de se géolocaliser précisément à la surface de la Terre [5]. Cette constellation se compose de 24 satellites opérationnels (plus 6 de rechange) d'environ 700 kilogrammes chacun, déployés en MEO à une altitude d'environ 23200 kilomètres (voir Figure 2.2). Il s'agit avant tout d'une volonté de l'Union Européenne de disposer de son propre système GNSS afin de ne pas être dépendante du GPS américain [3]. Et oui, même l'Espace est politisé.

Contrairement aux constellations de télécommunications, les satellites Galileo sont déployés à une altitude bien plus élevée qui correspond à une période orbitale d'environ 14 heures : cela signifie qu'un satellite met 14 heures à effectuer une révolution complète autour de la Terre. L'intérêt de cette altitude est d'avoir une large zone de couverture au sol par satellite, expliquant pourquoi les systèmes GNSS ne déploient que 20 à 40 satellites en orbite. On peut alors se demander pourquoi Galileo ne déploie pas un nombre plus faible de satellites, mais en orbite encore plus haute, comme l'orbite géostationnaire (environ 36000 kilomètres) : la raison repose sur la masse des satellites, qui serait nécessairement bien plus élevée, entraînant une explosion des coûts de lancement pour peu de gain de couverture.

Les zones de couverture des constellations GNSS sont fortement superposées afin de pouvoir localiser un utilisateur au sol : c'est le principe de la triangulation. Ainsi, un utilisateur se repérant grâce à ce système est visible en permanence par au

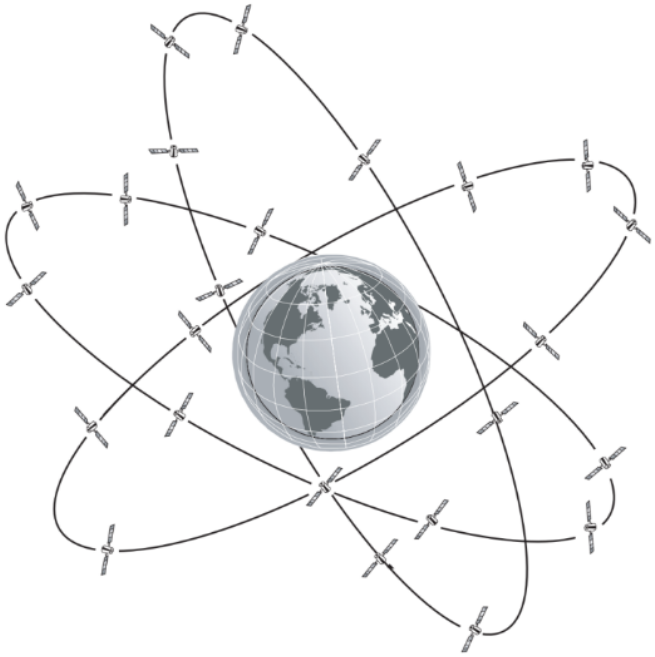


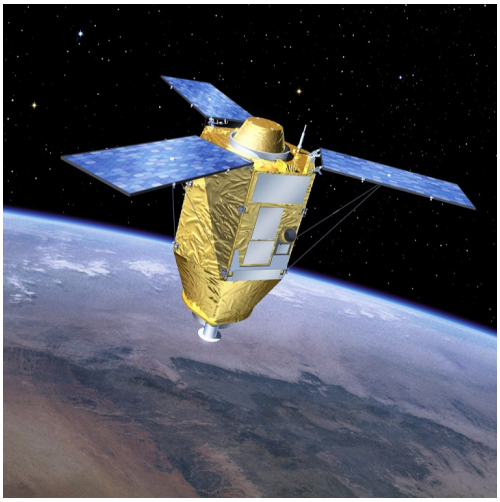
FIGURE 2.2 – Orbite des satellites Galileo (source : [2])

moins 4 satellites Galileo (ou tout autre système GNSS). Les constellations GNSS demeurent à ce jour les systèmes les plus fiables pour se repérer à la surface de la Terre.

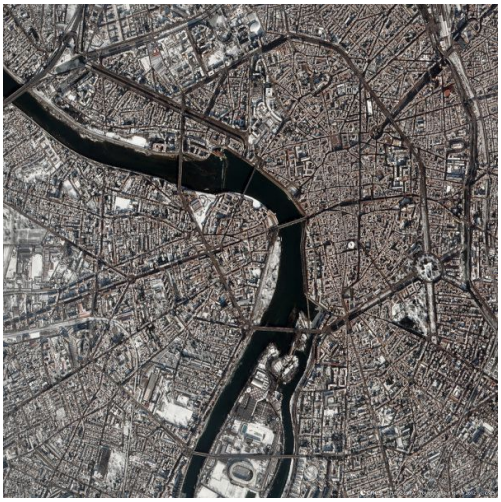
Pléiades, la constellation de l'observation

Pléiades est une bien petite constellation : seuls deux satellites en font partie, dénommés de manière originale Pléiades 1A et Pléiades 1B. Le projet Pléiades est initié par le CNES et a pour objectif de cartographier la surface de la Terre et de fournir des images à très haute résolution [43] (voir Figure 2.3). Les deux satellites, pesant chacun près d'une tonne, sont déployés sur la même orbite en SSO à environ 700 kilomètres d'altitude, à exactement 180 degrés l'un de l'autre.

L'orbite héliosynchrone (SSO) est très prisée des missions d'observation de la Terre : en effet, un satellite placé en SSO survole toujours un point de la Terre à la même heure, et donc à un niveau d'éclairement constant. Cela permet à un satellite de cartographier précisément une zone en effectuant plusieurs survols, sans tenir compte de l'heure, et de détecter si des changements notables se sont produits, notamment dans le cas de la surveillance de l'évolution des catastrophes naturelles. La dualité des satellites Pléiades permet à la constellation de survoler quotidiennement une même zone pour effectuer un suivi précis. Contrairement aux constellations Iri-



(a) Satellite Pléiades 1A (crédit : CNES, illustration Pierre Carril, 2005)



(b) Toulouse vue par Pléiades (crédit : CNES, 2012)

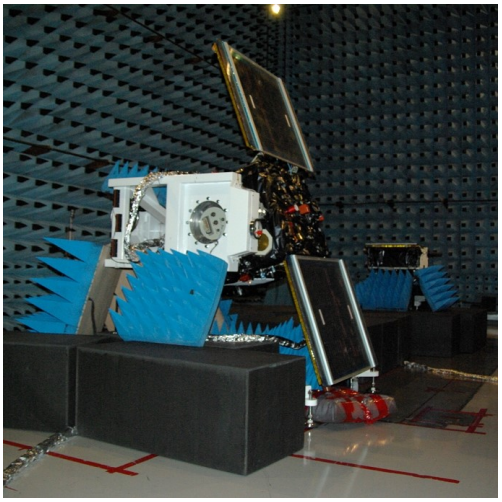
FIGURE 2.3 – Cartographie par la constellation Pléiades

dium et Galileo, la masse des satellites Pléiades est majoritairement impactée par les instruments télescopiques embarqués, et non par le carburant nécessaire à les mettre en orbite.

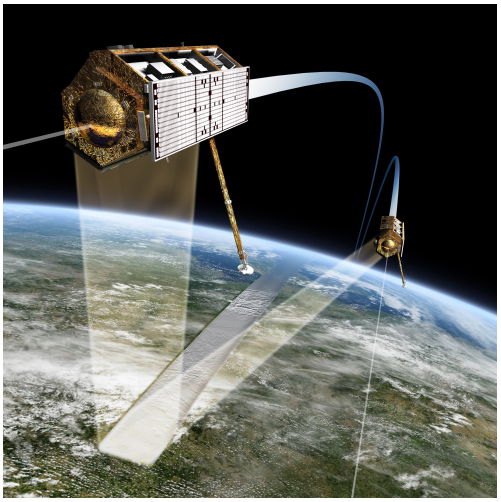
2.1.2 Trailing

Une formation en trailing désigne une architecture de DSS dite *figée*, où chaque satellite vole de manière très contrôlée afin de maintenir une formation. Généralement, les satellites d'un système en trailing sont placés sur la même orbite et se suivent à distance constante. La mobilité inter-satellite est donc très faible, voire pratiquement nulle. L'intérêt de cette architecture est d'améliorer la résolution temporelle d'un service, par exemple en survolant un même point à intervalle de temps constant. Il est également possible d'observer un même point avec des angles de vue différents, ce qui apporte des informations supplémentaires par rapport à une observation classique. Le vol en formation trailing est donc particulièrement prisé des missions d'observations de la Terre pour la météorologie, la surveillance de la propagation des feux de forêts, ou encore l'observation en 3D des ouragans.

Plusieurs missions de vol en trailing ont été lancées au cours des années. En 2015, des chercheurs ont réalisé une enquête sur les missions utilisant des petits satellites [20], c'est-à-dire ceux n'excédant pas 1000 kilogrammes (femto, pico, nano, micro et mini-satellites). Les plus connues sont les missions PRISMA [79] et TANDEM-X [57], qui ont chacune déployé deux satellites en trailing autour de la Terre étant capables de contrôler leur trajectoire et leur distance relative (voir Figure 2.4).



(a) PRISMA (crédit : CNES)



(b) TANDEM-X (crédit : DLR)

FIGURE 2.4 – Satellites des missions PRISMA et TANDEM-X

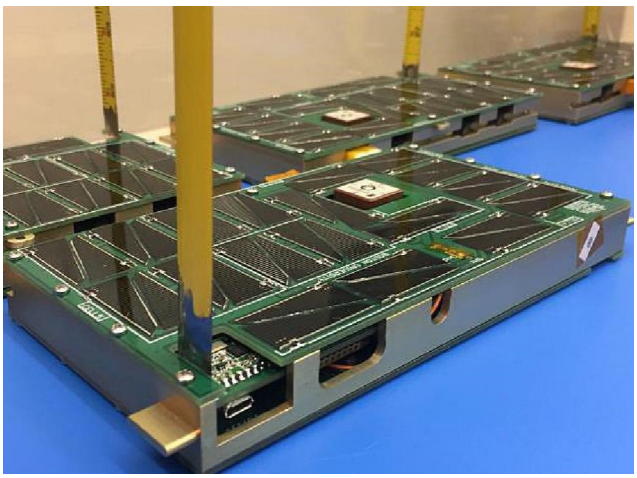
La mission PRISMA avait pour but de faire de la démonstration de technologies de vol en formation et de rendez-vous, ainsi que des expérimentations de manoeuvrabilité dans l'Espace. Deux satellites ont été déployés le 15 juin 2010 en LEO et se sont basés sur leurs coordonnées GSP respectives ainsi que sur des capteurs radio-fréquence de vol en formation (développés par le CNES) pour maintenir leur formation dans l'Espace avec une précision de l'ordre de la dizaine de mètres. Dans le cas de la mission TANDEM-X, lancée le 21 juin 2010, les deux satellites ont été déployés pour faire de l'interférométrie radar ; ils ont réussi à maintenir une distance relative très faible (200 mètres) en se basant sur les signaux GPS, à l'image de PRISMA, démontrant ainsi l'efficacité de leur système de contrôle d'orbite. Dans la même période, en 2012, la mission AeroCube-4 a également suscité de l'intérêt : en déployant trois nanosatellites en LEO, les opérateurs ont démontré la capacité du système à maintenir sa formation à toute épreuve. Les satellites ont ainsi su modifier leur trajectoire pour maintenir leur formation [42].

2.1.3 Satellite fractionné

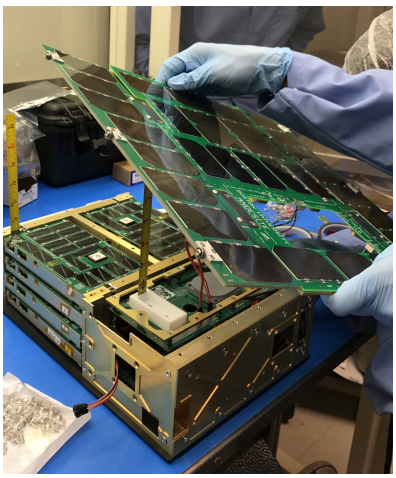
Malgré son nom trompeur, un satellite fractionné est en réalité un système constitué de plusieurs satellites embarquant à leur bord des modules divers et variés. Pour plus de clarté dans cette section, nous appellerons *satellite* le système de satellite fractionné, et *module* les satellites modulaires composant ledit satellite fractionné.

Les modules communiquent entre eux afin de réaliser une tâche complexe, et sont

également mobiles les uns par rapport aux autres dans une formation de cluster. Contrairement aux autres architectures de DSS, les modules du satellite fractionné ne sont pas du tout homogènes, car ayant chacun un but bien précis dans l'architecture [62]. Par exemple, un satellite peut disposer d'un module d'observation par capteurs, un module de traitement d'images, un module d'ordonnancement, etc. Par conséquent, la panne d'un module est moins impactante que sur un satellite monolithique, car elle n'entraîne que la perte d'une fonctionnalité localisée qui peut facilement être remplacée. D'ailleurs, les satellites fractionnés sont généralement dotés de mécanismes leur permettant de résister aux pannes et de maintenir un niveau de fonctionnement correct à tout instant [33]. A l'inverse, en cas de panne sur un satellite monolithique embarquant toutes les fonctionnalités à bord, il faudrait intervenir manuellement pour réparer la panne du composant, ce qui est moins envisageable et bien plus onéreux.



(a) Modules de magnétométrie (crédit : université de Boston)



(b) Vaisseau mère (crédit : université de Boston)

FIGURE 2.5 – Satellite fractionné du projet ANDESITE, lancé en 2020

Le système de satellite fractionné reste un concept jusqu'à ce que la DARPA (*U.S. Defense Advanced Research Projects Agency*) lance son programme System F6 (*Future Fast, Flexible, Fractionated, Free-Flying Spacecraft united by Information eXchange*) en 2009. L'objectif de ce progamme est de démontrer l'intérêt, pour certaines missions, de remplacer un satellite monolithique par un satellite fractionné équivalent, qui serait notablement plus flexible et robuste aux pannes [26]. Malheureusement, le projet est interrompu en 2013, avant qu'une démonstration du satellite en vol n'ait pu être réalisée.

Le concept de satellite fractionné n'a pas été abandonné pour autant. Afin d'étudier les phénomènes plasmiques des aurores boréales, l'université de Boston a initié

le projet ANDESITE (*Ad-Hoc Network Demonstration for Extended Satellite-Based Inquiry and Other Team Endeavors*) qui consiste en un satellite fractionné mesurant les champs magnétiques terrestres [73]. Le système, lancé en orbite le 13 juin 2020, comprend un nanosatellite "vaisseau mère" et 8 picosatellites additionnels embarquant les magnétomètres, pour une masse totale d'environ 6 kilogrammes (voir Figure 2.5). Les mesures captées par les picosatellites sont envoyées au vaisseau mère, qui à son tour les envoie à la Terre via la constellation de communication GlobalStar.

Plus récemment, des chercheurs de l'université de Toronto ont mené une étude sur la mise en place d'un satellite fractionné pour l'observation à très haute résolution de la surface terrestre [16]. Ce satellite, composé de 12 modules de nanosatellites, pourrait être déployé en LEO afin de cartographier la Terre à l'image d'un satellite Sentinel-2 [2]. Leur étude démontre la faisabilité d'un tel projet, ainsi que son intérêt principal : améliorer la scalabilité et la robustesse du système, et réduire la masse et le coût du système total tout en répondant aux critères d'une mission telle que Sentinel-2. Cependant, ce projet est à ce jour en phase de simulation : aucune démonstration n'a été faite pour le moment.

2.1.4 Essaim

L'architecture en essaim est au coeur du travail de cette thèse. Un essaim de satellites fait référence à un DSS composé de plusieurs satellites placés sur des orbites proches mais distinctes. Les satellites se déplacent de manière groupée mais non contrôlée, à l'instar d'un essaim d'abeilles ou d'un banc de poissons, par exemple. Ainsi, le mouvement de l'essaim dans son ensemble est prédictible, mais celui des satellites, individuellement, l'est beaucoup moins, ce qui diffère complètement du principe de fonctionnement d'une architecture en trailing ou d'une constellation. De plus, la mobilité intra-essaim reste modérée en comparaison à la mobilité relative des satellites d'une constellation. Enfin, l'architecture en essaim diffère d'un satellite fractionné sur un point central : les satellites composant l'essaim sont tous strictement identiques et homogènes, et remplissent tous la même fonction. Pour toutes ces raisons, l'essaim de satellites est considéré comme une architecture de DSS à part entière.

Grâce à son vol groupé, l'architecture en essaim présente l'avantage de fournir un service robuste et résilient à très haute résolution. Cependant, ce service est très localisé, avec une faible zone de couverture. Par conséquent, il s'agit d'une architecture de choix pour des missions scientifiques telles que l'interférométrie basse

2. La mission Sentinel-2 fait référence à une constellation de 2 gros satellites déployés en SSO afin de cartographier la Terre dans le cadre du programme Copernicus.

fréquence, qui a besoin de prendre des observations à très haute résolution pendant une durée limitée en orbite, comme expliqué dans la Section 1.1.2 du chapitre introductif. Pour mieux comprendre l'intérêt d'une architecture en essaim, nous proposons d'analyser deux projets d'interférométrie basés sur des essaims de satellites : les projets OLFAR et NOIRE.

OLFAR, le DSS pour l'interférométrie basse fréquence

Porté par l'Université Technologique de Delft et l'Université de Twente aux Pays-Bas, le projet OLFAR (*Orbiting Low Frequency Antennas for Radio Astronomy*) est l'un des premiers à s'intéresser aux architectures en essaim pour faire de l'interférométrie basse fréquence dans l'Espace. Les chercheurs de ce projet s'intéressent aux très basses fréquences, en-dessous de 30 Méga-Hertz, qui sont difficilement observables depuis la surface de la Terre (voir Section 1.1.1). Leur idée consiste alors à les observer depuis l'Espace, et plus précisément depuis une orbite lunaire, en utilisant un satellite ayant un diamètre d'ouverture compris entre 10 et 100 kilomètres pour pouvoir atteindre une résolution suffisante. Une telle contrainte technique n'est réalisable que par un DSS capable de combiner virtuellement les ouvertures de ses satellites : leur choix s'est donc naturellement porté sur une architecture en essaim [39]. Plus précisément, l'architecture envisagée serait composée de 50 nanosatellites, car une telle flotte serait en mesure de produire d'excellentes observations spatiales pour un coût de production et de lancement abordable (voir Figure 2.6).

De 2010 à 2014, le projet OLFAR donne lieu à de nombreuses publications sur les spécificités du système requis. Les chercheurs se sont par exemple intéressés aux contraintes techniques des antennes pour établir les ISL dans le réseau [97] : comme les positions et alignements des nanosatellites ne sont pas contrôlés, leur idée est de placer une antenne sur chaque face du nanosatellite (6 antennes au total) afin de pouvoir émettre dans toutes les directions. Un mécanisme de formation de faisceau est ajouté pour régler la puissance de chaque antenne afin d'optimiser l'émission du signal dans la bonne direction. Les auteurs précisent néanmoins que leur solution permet de satisfaire en théorie les contraintes techniques (débit, efficience, etc.), mais risque d'être moins efficace en pratique.

Un autre aspect important développé dans le projet OLFAR est le partage de charge dans le réseau. En effet, les chercheurs se sont vite rendus compte des problèmes de congestion et de perte de paquets que pourrait causer la transmission des données entre les satellites de l'essaim (voir Section 1.2.1). Ils proposent donc un mécanisme de regroupement hiérarchique adaptatif basé sur une technique de clus-

tering [29] : leur algorithme consiste à élire des *cluster heads* (chefs de cluster), qui à leur tour sélectionnent dynamiquement les satellites faisant partie de leur cluster. L'algorithme prend en compte les métriques d'énergie et de puissance disponibles afin d'optimiser la consommation énergétique du système. Les auteurs parviennent ainsi à limiter la transmission des données, mais au détriment de la redondance de l'image interférométrique observée. Ce compromis ne nous est pas étranger, et sera également étudié dans le cadre de cette thèse dans les Chapitres 4 et 5.

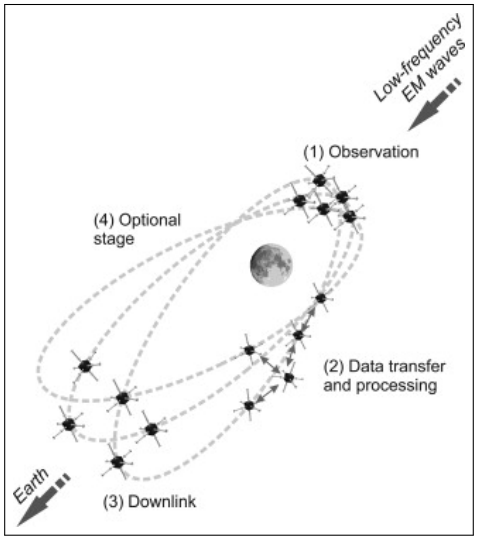


FIGURE 2.6 – L'essaim OLFAR en orbite lunaire (source : [28])

Les chercheurs se sont également intéressés au problème de la communication entre l'essaim et la BS terrestre. Ils démontrent qu'une communication fiable entre l'essaim et la Terre est réalisable si l'on met à profit la pluralité des satellites composant l'essaim [28]. Ainsi, il est possible de transmettre des signaux à la Terre en se basant sur la diversité spatiale des satellites, ou en adoptant la stratégie d'un réseau d'antennes. Les auteurs montrent que le réseau d'antennes permet d'obtenir le meilleur SNR sur la transmission, mais est bien plus complexe à mettre en place. Ils préfèrent donc se baser sur la diversité spatiale, qui permet malgré tout d'obtenir un SNR convenable.

Pour finir, les chercheurs se sont interrogés sur la fiabilité intrinsèque d'une architecture en essaim de nanosatellites [38]. En effet, un nanosatellite étant par définition un satellite léger, il embarque très peu de composants redondants, et n'est donc pas fiable d'un point de vue individuel. En revanche, les auteurs démontrent que la fiabilité du DSS dépend uniquement du nombre (et donc la redondance) de nanosatellites, et non de la qualité et la fiabilité intrinsèque des satellites. Ils expliquent ainsi que, pour assurer un débit fiable dans un essaim de nanosatellites, il est plus pertinent d'ignorer un satellite en panne, et donc d'abandonner sa charge de travail, plutôt que d'essayer de transférer sa charge de travail sur des satellites fonctionnels.

En résumé, le projet OLFAR a largement contribué à faire avancer la recherche sur les essaims de nanosatellites pour l'interférométrie basse fréquence. Malheureusement, ce projet n'est plus financé depuis 2014. Le projet NOIRE nous permet à présent de continuer la recherche sur ces essaims dans un contexte similaire.

NOIRE, le retour du DSS pour l'interférométrie basse fréquence

Quelques années plus tard, des chercheurs de plusieurs universités, instituts et centres de recherche français et internationaux s'associent et forment l'équipe NOIRE (Nanosatellites pour un Observatoire Interférométrique Radio dans l'Espace). Leur objectif est de mener plusieurs études afin de déterminer si l'utilisation d'un essaim de nanosatellites est une option viable pour faire de l'interférométrie basse fréquence depuis l'Espace. Un rapport exhaustif, publié en 2018, s'intéresse aux principaux aspects du système requis [30] : le principe de fonctionnement et l'architecture de l'essaim, la synchronisation des horloges, la localisation en milieu dépourvu de GNSS, le choix de l'orbite, la communication entre les noeuds et avec la BS, le traitement des données, la propulsion, la gestion de l'énergie, et enfin la compatibilité électromagnétique. Cette étude préliminaire conclut qu'un essaim de satellites pourrait être envisagé en tenant compte des exigences listées dans le Tableau 2.3, mais des recherches supplémentaires sont nécessaires sur plusieurs points, notamment la synchronisation des horloges et la localisation des satellites.

Plusieurs thèses sont en cours de réalisation pour répondre à certains de ces points, notamment au laboratoire TéSA. En effet, la possibilité de faire de l'interférométrie dans l'Espace est fortement dépendante de la précision des horloges des nanosatellites, ainsi que de leur capacité à se localiser dans l'Espace. L'une des thèses mentionnées se concentre donc sur les mécanismes de synchronisation d'horloges en milieu spatial, tandis que la deuxième étudie les méthodes de localisation en milieu dépourvu de système GNSS.

Parlons d'abord de la synchronisation des horloges. Dans un essaim de nanosatellites placé en orbite lunaire, les instruments embarqués sont placés dans un environnement hostile, causant leur détérioration au cours du temps sans possibilité de réparation. Les horloges des nanosatellites sont également impactées, ce qui réduit petit à petit leur précision. Or, afin de pouvoir corréler les signaux captés par les membres de l'essaim, ces signaux doivent être correctement horodatés. Il est donc nécessaire de mettre en place un mécanisme de synchronisation des horloges qui permette de corriger les erreurs des horloges défaillantes : ce mécanisme doit cependant être distribué afin de pouvoir être déployé dans un DSS ayant peu de contact avec la Terre. C'est ainsi que des chercheurs proposent la mise en place d'un système basé sur un Estimateur de Vraisemblance Maximale (MLE) qui permette de synchroniser les horloges des nanosatellites de manière distribuée, en tenant compte de la distribution des valeurs d'horloges et en détectant les valeurs aberrantes [63].

La seconde thèse concerne la localisation précise des satellites dans l'essaim, qui est le deuxième prérequis pour pouvoir faire de l'interférométrie dans l'Espace. En

	Paramètre	Description
Architecture	Classe de satellites	Nanosatellites
	Nombre de satellites	50
	Homogénéité	Hardware et fonctions
	Topologie	Dynamique car mobilité relative des satellites
Orbitographie	Position	Orbite lunaire
	Demi-grand axe	<5000 km
	Contrôle de la trajectoire	Faible
Interférométrie	Corrélateur	Distribué dans l'essaim
	Localisation	Basée sur les distances relatives. Précision requise : 1 cm
	Synchronisation des horloges	Non centralisée. Précision requise : <5 ns
	Sensibilité des capteurs	Très élevée. Compatibilité électromagnétique requise
Réseau	Antennes d'émission	Omnidirectionnelles
	Connectivité	Possible entre tous les satellites et avec la Terre
	Liaison essaim-Terre	Synthétiser les antennes pour meilleur débit
	Liaisons intra-essaim	ISL avec voisins directs, multi-saut pour les autres
	Routage	Routes instables. Envisager le routage MANET

TABLE 2.3 – Liste des principales exigences du système NOIRE

effet, contrairement aux UT des interféromètres terrestres, les UT embarqués sur satellite sont mobiles dans l'Espace et les uns par rapport aux autres. Cela engendre une difficulté majeure dans la phase de calcul, car le corrélateur doit connaître la position exacte (à 1 centimètre près) des satellites qui observent les signaux, et ce dans un environnement qui ne dispose d'aucun système GNSS. Les satellites doivent donc être en mesure de se repérer dans l'Espace en se basant sur leur distance relative par rapport aux autres satellites de l'essaim, ainsi que par rapport aux objets célestes comme la Lune, la Terre ou le Soleil. Dans ce contexte, les auteurs de [23] proposent un nouveau modèle de signal permettant de précisément estimer la latence et l'effet Doppler afin d'améliorer drastiquement la précision des applications telles que la localisation des nanosatellites dans l'essaim.

Pour finir, la troisième thèse basée sur l'étude NOIRE est celle que vous êtes en train de lire, bien entendu. Afin que les horloges puissent être synchronisées et les nanosatellites localisés, ils doivent être en mesure de communiquer des informations

dites de *signalisation*. Afin de pouvoir générer une image de corrélations croisées, les nanosatellites doivent être en mesure de communiquer des informations dites de *données*. Dans tous les cas, la fiabilité de la communication est un élément central dans cette architecture de DSS. Les informations de données sont les plus critiques d'un point de vue réseau, car plus volumineuses et donc celles pouvant potentiellement avoir le plus grand impact sur la connectivité du DSS. Notons que cette thèse ne traite que de la communication intra-essaim et de la fiabilité de l'architecture proposée pour transmettre des données dans le réseau. Les potentiels protocoles de routages sont, quant à eux, discutés dans la Section 2.2.4.

2.1.5 Récapitulatif

Besoins en télécommunications, GNSS, observation de la Terre et des objets célestes, missions scientifiques... les applications des DSS sont extrêmement variées. Pour réaliser ces applications, il existe de nombreux types d'architectures de DSS, avec chacunes ses spécificités, ses avantages et ses inconvénients. Il convient de choisir l'architecture la plus appropriée à l'utilisation que l'on en souhaite : les constellations sont parfaites pour assurer des services stables à la surface de la Terre (ou de tout autre objet céleste), les formations type trailing permettent de faire d'excellentes observations d'évolutions temporelles, les satellites fractionnés peuvent accomplir les mêmes tâches que les satellites monolithiques tout en offrant une meilleure résilience et des coûts de production plus attractifs. Enfin, les essaims de satellites permettent d'assurer un service à très haute résolution de manière périodique et pour un coût de production abordable. Ces principales caractéristiques sont rappelées dans le Tableau 2.4.

D'après les travaux énoncés dans cette section, il est évident que les essaims de satellites constituent un type d'architecture de DSS à part entière. Les technologies de routage et de fiabilité développées pour les constellations ou les formation en trailing ne sont donc pas applicables aux essaims, car ces architectures sont bien trop différentes dans leur construction. On peut, à la rigueur, apparenter un essaim et un satellite fractionné, mais l'hétérogénéité des noeuds de ce dernier est une trop grande différence à prendre en compte pour pouvoir appliquer les mêmes technologies aux deux architectures.

Les essaims de satellites nécessitent encore de nombreux travaux de recherche avant de pouvoir être opérationnels. Cette thèse étant orientée sur le réseau intra-essaim, nous allons nous intéresser, à travers la suite de cet état de l'art et des prochains chapitres, aux mécanismes pouvant être mis en place pour assurer une communication fiable entre les nanosatellites composant l'essaim.

Architecture	Constellation	Trailing	Satellite fractionné	Essaim
Mobilité inter-noeuds	Elevée	Nulle	Modérée	Modérée
Nature des noeuds	Homogène	Hétérogène	Hétérogène	Homogène
Contrôle de la trajectoire des noeuds	Elevé	Elevé	Faible	Faible
Complexité des noeuds	Elevée	Elevée	Modérée	Faible
Redondance des noeuds	Faible	Faible	Faible	Elevée
Impact de la perte d'un noeud	Baisse de la qualité de service	Arrêt de la mission	Perte d'une fonctionnalité	Baisse de la résolution

TABLE 2.4 – Comparaison des caractéristiques des architectures de DSS

2.2 Sur Terre, la grande famille des MANET

Un essaim de satellites est, certes, un système distribué composé de satellites orbitant dans l'Espace. Mais les similarités avec les autres architectures de DSS, notamment du point de vue réseau, s'arrêtent là. Il est donc difficile, voire impossible, de transposer les technologies développées pour les autres types de DSS sur une architecture en essaim. Fort heureusement, il existe bien assez d'architectures de réseaux terrestres similaires et de recherches associées pour y trouver de l'inspiration.

Dans cette section, nous présentons les architectures réseau les plus proches de ce que pourrait être un essaim, mais sur Terre. Comme préconisé dans l'étude NOIRE [30], nous allons nous intéresser de près aux Réseaux Ad-hoc Mobiles (MANET) et leurs dérivés, afin de comprendre leur fonctionnement et identifier les problématiques transposables aux essaims de nanosatellites.

2.2.1 Présentation

On appelle MANET un réseau qui n'a pas d'infrastructure dédiée (*ad-hoc*) et dont les noeuds peuvent être en mouvement (*mobile*) [1]. Dans un MANET, les noeuds communiquent avec leurs voisins directs (c'est-à-dire les noeuds les plus proches) et routent les paquets en multi-saut depuis la source jusqu'à la destination. Ce type de réseau nous intéresse tout particulièrement pour deux raisons :

1. l'absence d'infrastructure, donc de routeur dédié ou de point d'accès, implique que tous les noeuds puissent router des paquets dans le réseau ;

2. la mobilité des noeuds engendre une instabilité des routes, qui doivent être établies de manière dynamique et ré-évaluées régulièrement.

Bien que le terme MANET fasse généralement référence à un réseau terrestre, on peut tout de même considérer que le réseau d'un essaim de satellites y soit similaire : il n'a pas d'infrastructure additionnelle, et les satellites sont mobiles les uns par rapport aux autres et font office de source, destination et routeurs de paquets, comme décrit dans la Section 1.1 du chapitre introductif. Il est donc très judicieux d'analyser les principes de routage et de fiabilité mis en place dans les MANET, qui font l'objet de nombreuses études scientifiques, afin de pouvoir potentiellement les transposer aux réseaux d'essaims.

2.2.2 Cas particuliers de réseaux MANET

Il existe de nombreuses applications concrètes de MANET, civiles comme militaires. On peut par exemple penser à la mise en réseau d'unités de secours lors d'une catastrophe naturelle [19], aux réseaux de patrouilles militaires [81, 95], ou encore aux réseaux de capteurs mobiles [67, 81]. Parmi ces applications, on retrouve trois cas particuliers de MANET qui pourraient nous intéresser : il s'agit des réseaux véhiculaires (*Vehicular Ad-hoc Networks*, VANET), des réseaux aériens (*Flying Ad-hoc Networks*, FANET) et des réseaux maritimes (*Sea Ad-hoc Networks*, SANET). Nous allons étudier en détail leurs spécificités en termes de densité et de mobilité des noeuds, afin d'y trouver de possibles rapprochements avec les essaims de nanosatellites.

Réseaux ad-hoc véhiculaires (VANET)

Un VANET est un réseau ad-hoc dont les noeuds sont des véhicules (voitures, camions, bus, etc.) connectés entre eux pour échanger des données afin de sécuriser et optimiser le trafic routier, et limiter son impact environnemental [9]. Les véhicules peuvent communiquer entre eux (*vehicle to vehicle*, V2V) ainsi qu'avec des éléments d'infrastructure routière (*vehicle to infrastructure*, V2I) grâce à des équipements embarqués, composés de modules de communications et de capteurs [41] (voir Figure 2.7). La recherche sur les VANET est particulièrement prisée par la communauté scientifique en raison des répercussions positives qu'elle pourrait avoir sur notre vie de tous les jours, notamment en diminuant le nombre d'accidents de la route et en désengorgeant les villes, bien qu'il reste encore de nombreux aspects à améliorer [60].

Si vous vous êtes déjà baladé en ville, il est aisé de comprendre qu'un VANET est un réseau particulièrement dense : plusieurs centaines de noeuds peuvent en

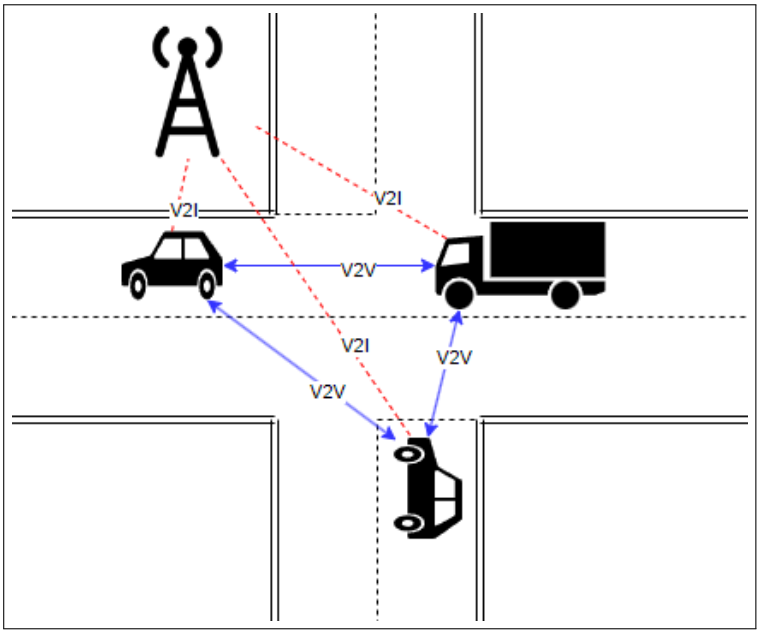


FIGURE 2.7 – Exemple de topologie urbaine VANET

faire partie, du moment que l'on interconnecte les véhicules. De plus, ces noeuds sont hétérogènes (type de véhicule, élément d'infrastructure) et fortement mobiles les uns par rapport aux autres ; du moins, tant qu'ils ne sont pas pris dans un embouteillage. Cependant, leur mobilité reste prévisible, car les véhicules sont tenus d'emprunter les routes existentes, et sont équipés d'un module de géolocalisation (type GPS ou Galileo, obligatoire en VANET). On peut donc connaître la position de chaque noeud de manière précise à chaque pas de temps. Les VANET sont donc assez différents d'un essaim de nanosatellites en termes de dynamique et de mobilité.

Réseaux ad-hoc volants (FANET)

Un FANET est un réseau ad-hoc dont les noeuds sont des véhicules aériens sans pilote (*Unmanned Aerial Vehicles*, UAVs), plus communément appelé drones. Ces UAVs interagissent entre eux au sein d'une flotte et volent généralement en formation, à l'instar des constellations ou des satellites en trailing [32]. Il faut cependant qu'au moins un des UAVs soit connecté à une station de base, qui peut être fixe ou bien mobile (par exemple sur un hélicoptère, un avion ou un satellite). Un FANET peut être déployé pour accomplir des missions très diverses, comme la surveillance et le contrôle de champs agricoles [98], le sauvetage de personnes en détresse [61], ou encore la détection et la surveillance des feux de forêts [64] (voir Figure 2.8).

Un FANET peut être un réseau très dense en fonction de la zone à couvrir et de la mission : des centaines de drones peuvent être déployés en maillage serré. Les noeuds

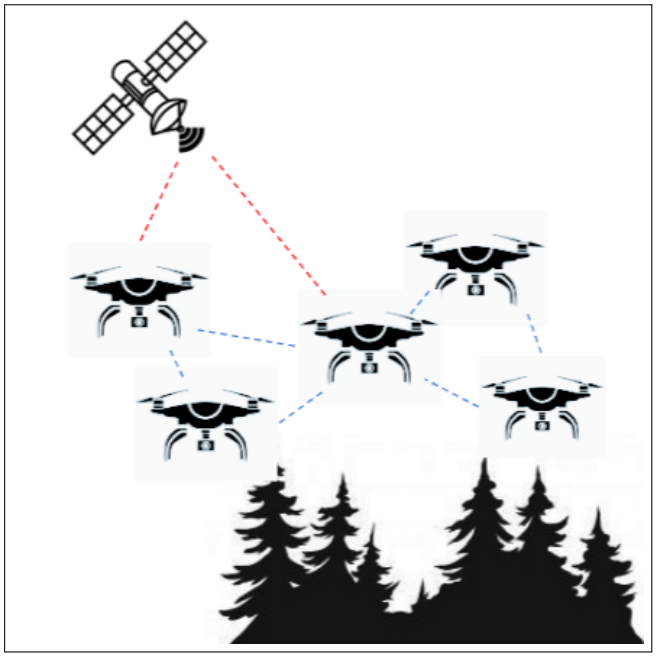


FIGURE 2.8 – Exemple de topologie FANET pour la surveillance des forêts

d'un FANET ne réalisent pas forcément les mêmes tâches : à l'instar d'un satellite fractionné, certains noeuds peuvent accomplir des tâches bien spécifiques, auquel cas ils auront possiblement besoin de se déplacer au sein de la flotte. Par conséquent, la mobilité relative des noeuds d'un FANET peut être particulièrement élevée, et les informations de mobilité (coordonnées, temps de vol, conditions climatiques, etc.) doivent être transmises à tous les membres du réseau pour éviter les collisions et les pannes [8]. La mobilité d'un FANET est donc déterministe, puisque l'opérateur (via la station de base) du réseau dispose d'un contrôle total sur le positionnement des UAVs et leur déplacement dans la flotte [3]. On est donc, encore une fois, dans une configuration bien différente d'un essaim de satellites du point de vue de la mobilité.

Réseaux ad-hoc maritimes (SANET)

S'agissant sûrement du réseau le moins connu des trois, le SANET est un réseau ad-hoc composé de bateaux, navires, sous-marins et balises maritimes. L'intérêt et le rôle majeur du SANET est d'améliorer la connectivité en mer : en effet, la connexion satellitaire peut parfois être difficile à mettre en place au-dessus d'un

3. Il est courant de retrouver l'appellation d'essaim pour faire référence à une flotte de drones. Cette appellation est cependant erronée : la mobilité d'un essaim est prévisible, mais pas celle de ces noeuds. Cela reviendrait à essayer de prédire la position précise d'une abeille dans un essaim d'insectes !

océan en fonction de la zone où ce trouve le bateau [31]. Un ou plusieurs noeuds du SANET peuvent être connectés à une station de base fixe (sur la côte) ou mobile (satellite) afin de transférer les données (voir Figure 2.9). Le SANET est particulièrement prisé pour la détection et la surveillance de catastrophes naturelles, comme les séismes, tsunamis et éruptions volcaniques [86], ou plus généralement pour la météorologie [15].

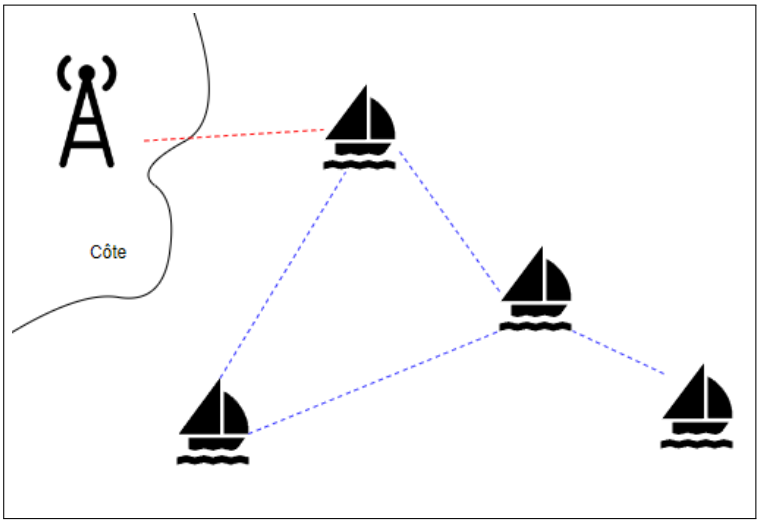


FIGURE 2.9 – Exemple de topologie SANET près d'une côte

Un SANET peut être composé de noeuds hétérogènes, mais cela n'a pas beaucoup d'impact sur son modèle de mobilité. En effet, les noeuds d'un SANET se déplacent à allure plus faible par rapport aux noeuds d'un VANET ou d'un FANET, et leur mobilité relative est elle aussi modérée. Par conséquent, les changements de topologie dans un SANET sont assez lents par rapport aux autres types de réseaux [31]. Il s'agit également d'un réseau peu dense, puisque le réseau couvre généralement une très grande surface avec un nombre de noeuds limité. Enfin, la position des noeuds du SANET peut être connue à l'avance (en raison de la trajectoire fixée des bateaux), mais le positionnement peut s'avérer peu précis [4] : on peut donc estimer qu'un noeud se trouve dans une zone donnée, mais il est plus compliqué d'obtenir sa localisation exacte. En résumé, et contre toute attente, le SANET est un réseau dont les dynamiques de mobilité sont assez proches d'un essaim de nanosatellites, à la vitesse de déplacement près.

4. La localisation des navires en mer n'est généralement pas assez précise avec un GNSS seul : la plupart du temps, les navires utilisent des données d'identification additionnelles (*Automatic Identification System*, AIS) pour pouvoir gagner en précision.

Récapitulatif

Les MANET se déclinent sous de nombreuses formes en fonction de ce que l'on souhaite interconnecter, et dans quel but. On retrouve ainsi les réseaux véhiculaires (VANET), qui visent à optimiser le trafic routier ; les réseaux de drones (FANET), souvent employés pour de la surveillance et la détection ; et enfin les réseaux maritimes (SANET), très prisés pour la détection de catastrophes et la météorologie [99]. Ces trois familles de réseaux ad-hoc ont chacune leurs spécificités en matière de mobilité et de densité de noeuds : ainsi, les VANET et FANET sont généralement plus denses et dynamiques que les SANET, dont la topologie évolue plus lentement au cours du temps. Les caractéristiques de ces trois réseaux, ainsi que des essaims de nanosatellites, sont résumées dans le Tableau 2.5.

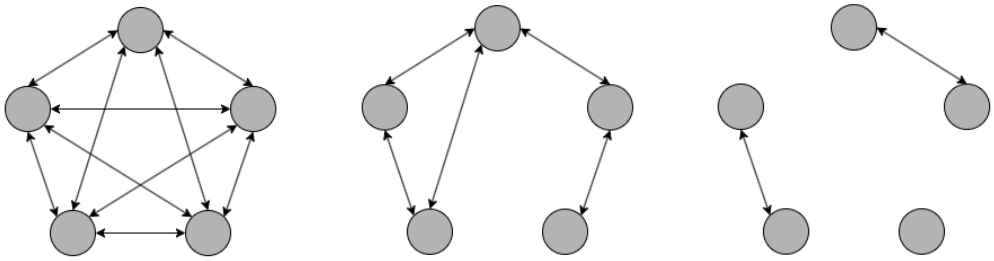
Caractéristique	**VANET**	**FANET**	**SANET**	**Essaim**
Densité des noeuds	Elevée	Elevée	Modérée	Modérée
Vitesse des noeuds	Modérée à élevée	Elevée	Modérée	Très élevée
Mobilité relative	Elevée	Elevée	Modérée	Modérée
Evolution de la topologie	Rapide	Rapide	Lente	Lente
Localisation des noeuds	Précise	Précise	Assez précise	Approximative

TABLE 2.5 – Comparaison des VANET, FANET et SANET avec les essaims de satellites

Dans un essaim de (nano)satellites, les noeuds se déplacent à vitesse très élevée par rapport aux réseaux terrestres. Cependant, c'est le modèle de mobilité interne qui nous intéresse et qui a le plus d'importance : la vitesse relative des noeuds, ainsi que leur densité, sont des éléments clés pour pouvoir établir une connexion entre les membres du réseau. D'après le Tableau 2.5, il est évident que le SANET est le type de réseau se rapprochant le plus d'un essaim de satellites, en raison de la mobilité relative des noeuds, de la vitesse d'évolution de la topologie, ainsi que de la capacité à localiser et contrôler les noeuds du réseau. Et paradoxalement, celui dont l'essaim de satellites est le plus distinct est le FANET qui, mis à part la vitesse de déplacement des noeuds, est différent de l'essaim sur tous les autres points. Par la suite, et dans les travaux futurs, il sera intéressant de faire des parallèles entre les essaims de satellites et les SANET, qui sont finalement bien plus semblables qu'on pourrait le croire. On a désormais les éléments de réponse à la question : "Quel est le point commun entre un bateau et un nanosatellite ?"

2.2.3 Topologies des MANET

Puisque les noeuds sont mobiles, la topologie d'un MANET est également vouée à évoluer au cours du temps. Il reste néanmoins possible de la caractériser à chaque instant en analysant son niveau de maillage, c'est-à-dire la quantité de liens disponible, schématisé sur la Figure 2.10. L'étude de la topologie du réseau est primordiale, car elle est particulièrement impactante quant au choix de la stratégie de routage à adopter. En effet, une topologie où tous les noeuds peuvent se joindre directement (dite topologie à maillage *complet*) n'aura pas les mêmes exigences de routage ou d'allocation de ressources qu'une topologie présentant des noeuds isolés (topologie à maillage *déconnecté*).



(a) Maillage complet (b) Maillage partiel (c) Maillage déconnecté

FIGURE 2.10 – Niveaux de maillage d'une topologie de MANET

Une topologie de MANET peut donc avoir trois niveaux de maillage : complet, partiel ou déconnecté. Un maillage *complet* désigne une topologie où tous les noeuds sont les voisins directs de tous les autres. Mathématiquement, cela signifie qu'il existe un nombre de liens m proportionnel au carré du nombre de noeuds n :

$$m = m_{\max}$$

$$m_{\max} = \frac{n(n-1)}{2} \tag{2.1}$$

Il s'agit du nombre maximal de liens uniques pouvant être établis dans un réseau, en supposant que les noeuds reliés par un lien soient obligatoirement distincts et que les liens soient bi-directionnels (donc duplex). L'avantage évident de cette topologie est la rapidité de la transmission des informations : en effet, le multi-saut n'est pas nécessaire car les paquets n'ont qu'un saut à faire pour atteindre leur destination. L'énorme inconvénient de cette topologie réside dans sa difficulté à passer à l'échelle : puisque le nombre de liens croît comme le carré du nombre de noeuds, si

le nombre de noeuds est élevé, le nombre de liens à établir explose. Si l'on reprend les paramètres de l'étude NOIRE (Tableau 2.3), dans un essaim de 50 nanosatellites, il faudrait établir 1225 liens en tout, ce qui représente une quantité astronomique de liens par rapport au nombre de noeuds. A l'inverse, le maillage complet serait une topologie viable dans un réseau comportant peu de noeuds (comme représenté sur la Figure 2.10a, où il n'y a que 5 noeuds, et donc 10 liens). En résumé, le maillage complet est très efficace dans une petite topologie, mais ne passe pas à l'échelle d'un essaim de nanosatellites.

L'opposé du maillage complet est le maillage *déconnecté*. Dans une topologie déconnectée, certains noeuds, ou groupes de noeuds, sont isolés car aucun lien ne les unit au reste du réseau. Il n'est donc pas possible d'établir des routes entre chaque paire de noeuds. On se retrouve alors avec un nombre de liens m inférieur au nombre de liens $m_{\min}$ nécessaire pour connecter le réseau :

$$m < m_{\min}$$

$$m_{\min} = n - 1 \tag{2.2}$$

On dit alors que le réseau est fragmenté en *composantes connexes*, où les noeuds d'une même composante connexe sont connectés entre eux, mais sont isolés des autres composantes connexes. Par exemple, le réseau représenté sur la Figure 2.10c est constitué de 3 composantes connexes, comprenant chacune 1 à 2 noeuds. Dans un cas extrême, il se peut qu'un réseau ne comporte plus aucun lien, ce qui en fait une topologie à maillage complètement déconnecté, constituée d'autant de composantes connexes qu'il y a de noeuds dans le réseau. Dans une situation réelle, cela arrive notamment lors d'une panne de courant, où tous les membres du réseau se retrouvent déconnectés pendant un certain temps.

Mais dans les faits, pour un essaim de nanosatellites, il serait rare de se retrouver dans l'une des situations extrêmes décrites plus haut. Très souvent, la topologie d'un MANET correspond plutôt à un niveau de maillage intermédiaire, dit *partiel*. Dans une topologie à maillage partiel, les noeuds sont tous reliés entre eux par des routes : il n'y a qu'une seule composante connexe, mais il n'y a pas pour autant des liens entre chaque paire de noeuds. Un exemple de maillage partiel est représenté sur la Figure 2.10b. Mathématiquement parlant, dans une topologie à n noeuds, le nombre de liens m est défini comme suit :

$$m_{\min} \leq m < m_{\max}$$

$$n - 1 \leq m < \frac{n(n-1)}{2} \tag{2.3}$$

La topologie à maillage partiel est celle souhaitée dans le cadre du projet NOIRE pour un essaim de nanosatellites : d'après le Tableau 2.3, tous les noeuds doivent être capables de communiquer avec les autres, mais en faisant du multi-saut pour ne pas avoir à établir trop de liens. Cela correspond précisément à la définition d'une topologie à maillage partiel.

Notons également qu'il s'agit de la topologie la plus adaptée pour intégrer des protocoles de routage type MANET, car une topologie à maillage complet n'en a tout simplement pas besoin (mais cela pose d'autres problèmes tout aussi importants, comme l'allocation de ressources), et une topologie déconnectée ne permet pas d'atteindre tous les noeuds. Dans ce dernier cas, il est possible de mettre en place un routage opportuniste pour atteindre les noeuds quand ceux-ci seront à nouveau connectés, ce qui correspond à une stratégie de routage type DTN (*Delay Tolerant Network*). Cependant, ce type de routage n'étant pas préconisé par l'étude NOIRE, nous ne nous y attarderons pas dans cette thèse.

2.2.4 Stratégies de routage

Puisque les MANET sont largement étudiés par la communauté scientifique depuis de nombreuses années, ils disposent d'un vaste éventail de protocoles de routage [1, 68, 75, 51]. Cependant, nous n'allons pas nous intéresser aux protocoles en tant que tels, mais bien aux *stratégies* de routage qu'il est possible de mettre en place dans les MANET. On appelle stratégie de routage le principe fondamental sur lequel se base un protocole. Nous pouvons alors regrouper les protocoles de routage en trois catégories, en fonction de leur stratégie : on retrouve des protocoles de routage dits réactifs, proactifs, et hybrides.

Routage réactif

Le principe de la stratégie de routage réactive, comme son nom l'indique, est de réagir à la demande du réseau. Ainsi, les routes ne sont établies que lorsqu'un noeud source demande à joindre une destination. Généralement, lorsqu'un noeud source veut communiquer avec une destination, il cherche d'abord une route valide dans sa table de routage (la validité de la route peut être compromise par l'évolution de la topologie). S'il ne trouve pas de route valide, il enclenche un mécanisme de découverte de routes. La manière dont les routes sont découvertes est propre à chaque protocole de routage, dont les plus répandus sont AODV (*Ad-hoc On-demand Distance Vector*) [77, 10], DSR (*Dynamic Source Routing*) [50], TORA

(*Temporally Ordered Routing Algorithm*) [74] et ABR (*Associativity-Based Routing*) [91].

Le routage réactif dans les réseaux MANET peut prendre différentes formes, mais comprend généralement au moins une phase de découverte de routes et une phase de maintien (ou reconstruction) de ces routes. La manière dont ces routes sont découvertes et mises à jour dépend du protocole de routage. Le principal avantage du routage réactif est qu'il n'utilise la bande passante que de manière ponctuelle, puisqu'il réagit à la demande du réseau : il s'adapte donc très bien au moindre changement de topologie. En revanche, la découverte des routes peut introduire une latence non négligeable, et dans certains cas, surcharger le réseau avec des paquets de signalisation.

Routage proactif

La stratégie de routage proactive consiste à maintenir les tables de routage à jour sans attendre la demande d'un noeud source pour établir des routes. Le routage proactif se base généralement sur un mécanisme d'inondation de réseau afin de répandre les informations de routage à tous les noeuds. Les principaux protocoles de routage proactif sont OLSR (*Optimized Link State Routing protocol*) [34, 66], DSDV (*Destination-Sequenced Distance Vector*) [45], et FSR (*Fisheye State Routing*) [76].

Les protocoles de routage proactif se basent sur l'analyse périodique de l'état des liens ou des vecteurs de distance pour obtenir des routes optimales entre chaque paire de noeuds. L'intérêt majeur du routage proactif est qu'il évite la latence générée par le routage réactif lors de la découverte des routes, puisque les routes sont connues à l'avance. Cependant, la mise à jour régulière des tables de routage génère une quantité de trafic non négligeable lorsque le réseau est dense, ou lorsque la bande passante est limitée.

Routage hybride

Les stratégies de routage réactives et proactives, présentés ci-dessus, semblent opposées par nature. Il est cependant possible de les combiner pour contrer les faiblesses l'une de l'autre : on parle alors de stratégie de routage *hybride*. Les principaux protocoles de routage hybride sont ZRP (*Zone Routing Protocol*) [22], ZHLS (*Zone-based Hierarchical Link State*) [49] et AntHocNet [37].

Le routage hybride définit des zones stratégiques dans le réseau, dits *clusters*, au sein desquels les noeuds communiquent de manière régulière. Les noeuds d'un même cluster adaptent donc une stratégie plutôt proactive pour communiquer entre eux,

puisqu'il y aura fréquemment de la demande. En revanche, les communications entre différents clusters sont plus rares : elles demandent donc la mise en place d'une stratégie réactive pour établir des routes seulement lorsque cela est nécessaire. L'intérêt du routage hybride est donc sa haute adaptabilité face à un réseau hétérogène (en termes de demande de communication), qui se traduit par l'optimisation conjointe du nombre de paquets de signalisation à transmettre et de la mise à jour des routes. Son principal défaut est son efficacité en pratique, qui dépend grandement du protocole utilisé et de la dynamique de la topologie du réseau.

Récapitulatif

Les réseaux type MANET peuvent adopter plusieurs stratégies de routage, qui doivent être choisies en fonction des caractéristiques du réseau. Combien y a-t-il de noeuds dans le réseau ? Les paires de noeuds communiquent-elles régulièrement, ou de manière sporadique ? Est-ce que tous les noeuds communiquent à la même fréquence, ou y a-t-il des noeuds communiquant plus souvent que d'autres ? Toutes ces questions doivent être minutieusement étudiées pour sélectionner la meilleure stratégie de routage. Le choix du protocole viendra naturellement une fois la stratégie définie.

Le réseau d'un essaim de nanosatellites peut avoir des caractéristiques totalement différentes en fonction de son dimensionnement. Prenons le cas d'un essaim de 50 nanosatellites : ce nombre de noeuds est suffisant pour une mission scientifique, tout en étant assez faible pour ne pas affecter l'essaim de potentiels problèmes d'échelle. Ainsi, si les noeuds du réseau communiquent de manière régulière et que la topologie ne change pas brutalement au cours du temps, il est conseillé d'adopter une stratégie de routage proactive, car les routes seront fréquemment utilisées et maintenues à jour. Si, au contraire, les noeuds communiquent de manière plus sporadique, et que des changements de topologie peuvent se produire de manière brusque, il est préférable d'adopter une stratégie de routage réactive, qui saura s'adapter à l'évolution de la topologie tout en limitant la fréquence de mise à jour des routes à son strict minimum. Enfin, si la communication dans le réseau est hétérogène ou que la topologie elle-même est hétérogène, il convient d'adopter une stratégie de routage hybride qui s'adaptera aux noeuds en fonction de leur demande en communication. Le Tableau 2.6 résume les avantages et inconvénients de chaque stratégie de routage, ainsi que le type de réseau visé par chaque stratégie.

Quelle serait donc la stratégie de routage la plus adaptée à notre essaim de nanosatellites ? Pour répondre à cette question, il convient d'abord d'analyser les caractéristiques du réseau afin d'en comprendre les dynamiques. Ces caractéris-

Stratégie de routage	Réactive	Proactive	Hybride
Latence de découverte des routes	Elevée	Faible	Faible
Surcharge du réseau	Modérée	Elevée	Modérée
Réactivité au changement de topologie	Elevée	Modérée	Modérée
Topologie cible	Homogène, variable	Homogène, stable	Hétérogène
Demande en communication cible	Sporadique	Régulière	Hétérogène
Exemples de protocoles	AODV, DSR, TORA, ABR	OLSR, DSDV, FSR	ZRP, ZHLS, AntHocNet

TABLE 2.6 – Récapitulatif des stratégies de routage MANET

tiques pourront également varier en fonction du dimensionnement de l'essaim et de la mission scientifique qu'il doit accomplir. Pour plus d'informations sur les caractéristiques de notre essaim pour l'interférométrie basse fréquence, nous vous donnons rendez-vous au Chapitre 3.

2.2.5 L'intérêt des MANET pour les essaims de satellites

Dans cette section, nous avons passé en revue les principales caractéristiques des MANET, car la recherche sur ce domaine est largement documentée par la communauté scientifique, par opposition à la recherche sur les essaims de nanosatellites. Les MANET peuvent se décliner en plusieurs types afin de répondre au mieux à une problématique donnée : on retrouve ainsi les réseaux véhiculaires (VANET) permettant d'optimiser et sécuriser le trafic routier, les réseaux de drones (FANET) pour la détection et la surveillance, et enfin les réseaux maritimes (SANET) pour la surveillance et la météorologie. En termes de dynamique et de mobilité, un essaim de nanosatellites se rapproche le plus d'un SANET en raison de la mobilité modérée de ses noeuds les uns par rapport aux autres, de sa densité relativement faible et des changements de topologie lents (notons que ces propriétés sont étudiées bien plus en détail dans le Chapitre 3). Il serait donc intéressant de faire des parallèles entre la recherche sur les essaims de nanosatellites et celle sur les SANET, car les technologies développées pour l'une seraient potentiellement applicables à l'autre, en effectuant quelques adaptations évidemment, puisqu'un satellite ne se comporte pas comme un bateau.

Il est primordial d'étudier la topologie du réseau de l'essaim, ainsi que ses dynamiques, pour pouvoir établir un plan de connexion. Nous avons vu qu'il existe trois niveaux de maillage dans une topologie MANET : le maillage complet correspond à

une interconnexion totale de l'ensemble des noeuds, le maillage déconnecté implique que certaines paires de noeuds ne sont pas connectées, et le maillage partiel correspond à toutes les topologies entre ces deux cas extrêmes. Dans une situation réelle, la topologie d'un MANET peut être hétérogène, et donc présenter des groupes de noeuds à maillage complet aux côtés de noeuds déconnectés. Toutes ces spécificités sont à prendre en compte pour adapter au mieux une stratégie de routage. Dans un essaim de nanosatellites, nous cherchons à obtenir une topologie à maillage partiel, c'est-à-dire que toutes les paires de noeuds doivent être joignables en faisant du multi-saut, donc du routage. Afin de garantir le maillage partiel, nous pouvons jouer sur plusieurs paramètres :

1. Adapter le débit : afin de relier des noeuds éloignés, il est possible de réduire le débit afin d'émettre plus loin, mais plus longtemps. Cette solution est économique en termes d'énergie, mais ne permet pas de garantir une qualité de service suffisante pour la mission scientifique.
2. Adapter la puissance d'émission : si un noeud est trop éloigné, il est possible d'augmenter la puissance d'émission afin d'augmenter la portée de connexion. Cette solution garantit la qualité de service nécessaire, mais impacte fortement la consommation énergétique du réseau.

Dans les deux cas, il faudra faire un compromis entre la qualité de service et la consommation énergétique. Nous choisissons la deuxième option, qui consiste à adapter la portée de connexion avec les noeuds distants, au détriment de la consommation énergétique. Voilà pourquoi, dans la suite de ce manuscrit, nous apporterons une attention toute particulière à la mise en place de mécanismes d'économie d'énergie afin de pallier à ce problème. Ces mécanismes permettront de garantir la qualité de service nécessaire à l'interférométrie spatiale, tout en limitant la consommation énergétique de l'essaim : on parle de mécanismes de *fiabilité*. La recherche associée à la fiabilisation des systèmes distribués est présentée dans la Section 2.3, et notre solution est détaillée dans les Chapitres 4 et 5.

Une fois la fiabilité du système assurée, il sera possible d'envisager des solutions de routage dans l'essaim de nanosatellites. Les MANET disposent de trois stratégies de routage, chacune adaptée à des configurations spécifiques. La stratégie réactive établit des routes à la demande du réseau, et s'adapte aux réseaux homogènes à communication sporadique dont la topologie change rapidement. La stratégie proactive maintient les tables de routage à jour dans un réseau où les noeuds communiquent régulièrement, et s'adapte aux réseaux homogènes dont la topologie évolue lentement. Enfin, la stratégie hybride maintient les tables de routage à jour entre les noeuds fortement connectés, et s'adapte à la demande entre les

noeuds qui communiquent moins souvent : cette stratégie est donc particulièrement adaptée aux réseaux où la communication est hétérogène. Dans cette thèse, nous n'oublierons pas de proposer la stratégie de routage la plus adaptée à un essaim de nanosatellites fonctionnant comme un radiotélescope spatial distribué. Le choix et la détermination du protocole de routage optimal sera, quant à lui, laissé à de futurs travaux.

2.3 Comment fiabiliser les systèmes distribués

En tant que DSS, un essaim de nanosatellites est sujet aux mêmes failles que n'importe quel système distribué. Afin d'assurer un service correct et donc une qualité de service suffisante, le système doit être *fiable*, c'est-à-dire fonctionner correctement le plus longtemps possible (idéalement tout au long de la mission). Dans un système monolithique, il faut s'assurer que chaque composant soit en mesure d'éviter et de tolérer les pannes ; dans un système distribué, il faut additionnellement s'assurer que chaque membre du système puisse gérer les pannes. Or, les membres d'un essaim de nanosatellites sont particulièrement rudimentaires, avec peu de composants à bord : fort heureusement, leur fiabilité individuelle n'influe que très peu sur la fiabilité globale du système, qui dépend davantage de la redondance du nombre de nanosatellites dans l'essaim [38], comme nous l'avons vu dans la Section 2.1.4. Par conséquent, nous n'allons pas étudier la fiabilisation des nanosatellites en termes de composants (*hardware*), mais bien la fiabilisation de l'essaim d'un point de vue global.

Quand on parle de fiabilité, on fait généralement référence à deux concepts distincts, à savoir la *robustesse* du système et sa *résilience*. La robustesse définit la capacité d'un système à résister aux pannes, ou à les éviter : il s'agit de mécanismes de prévention. La résilience, quant à elle, caractérise la capacité d'un système à maintenir un certain niveau de fonctionnement (correct ou dégradé) une fois que les pannes ont eu lieu, et que le niveau de robustesse n'était pas suffisant. Un exemple assez parlant est celui d'une autoroute : le nombre de voies sur une autoroute est dimensionné de sorte à contenir le trafic afin d'éviter les embouteillages (robustesse). Mais si un accident se produit, entraînant un embouteillage, les secours peuvent emprunter la bande d'arrêt d'urgence pour rétablir le trafic au plus vite (résilience). Ainsi, un nombre de voies bien dimensionné couplé à la présence d'une bande d'arrêt d'urgence permettent d'assurer la fiabilité de l'autoroute.

Dans cette partie, nous allons étudier les mécanismes de robustesse et de résilience proposés dans la littérature pour fiabiliser les systèmes distribués selon certains types de pannes. Mais au fait, de quelles pannes parle-t-on ?

2.3.1 Les quatre types de pannes

Les pannes pouvant affecter un système, informatique ou autre, peuvent être classifiées selon divers critères. La classification la plus répandue de nos jours est celle prenant en compte la durée (permanente ou transitoire) et la raison (intentionnelle ou non) de la panne [6]. Il existe ainsi quatre types de pannes :

- la panne franche : cette panne bloque de façon permanente le fonctionnement d'un système ou d'un élément. Par exemple, un nanosatellite vidé de son énergie est dit en panne franche, car il ne peut plus fonctionner dans l'état ;
- la panne par omission : cette panne indique que des paquets (de signalisation ou de données) sont perdus en entrée ou en sortie d'un élément. Par exemple, un paquet perdu en raison d'une congestion du réseau constitue une panne par omission. Il s'agit d'une panne transitoire ;
- la panne temporelle : cette panne indique que le temps de réponse d'un élément est anormalement long, et que les délais sont dépassés. Par exemple, des paquets accumulés dans une file d'attente constituent une panne temporelle. Celle-ci est généralement transitoire ;
- la panne byzantine : cette panne fait référence au contenu du paquet, et non à sa forme ou à son émission. La panne byzantine représente les comportements aberrants, voire malicieux, d'un élément du système. Par exemple, une valeur extrême relevée par un capteur est une panne byzantine, car la forme du paquet est correcte, mais pas son contenu. Une intrusion de sécurité ou un virus informatique constituent là aussi des pannes byzantines. Ces pannes peuvent être transitoires ou permanentes en fonction de leur gravité.

Les mécanismes de fiabilité à mettre en place doivent être adaptés au(x) type(s) de pannes que l'on souhaite contrer [6]. Prenons l'exemple d'un serveur Internet : il est possible de le doter de batteries de secours pour contrer les pannes de courant (résilience face à un panne franche), ainsi que d'un pare-feu pour n'autoriser qu'un certain type de trafic, et donc se protéger des virus (robustesse face à une panne byzantine). Il est alors évident que ces deux types de pannes n'ont rien à voir et doivent être traitées distinctement.

A quelles pannes peut donc s'exposer un essaim de nanosatellites ? Et bien, dans le cadre d'une mission d'interférométrie type NOIRE, la panne la plus évidente est la panne franche liée à la perte d'énergie. En effet, comme nous l'avons précisé dans la section précédente, un compromis doit être établi entre la qualité de service et la consommation énergétique : dans notre cas, nous choisissons d'accorder la priorité à la qualité de service, au détriment de la consommation énergétique. Les pannes franches par perte d'énergie seront donc le principal type de panne que l'on pourra

rencontrer dans notre contexte : le nanosatellite fonctionne correctement, ou il ne fonctionne pas du tout.

Les pannes temporelles et par omission sont, quant à elles, liées à la surcharge du réseau : leur présence est tout à fait réaliste, mais elle détériore la qualité de service. Puisque, dans cette thèse, nous considérons que la qualité de service est respectée pour l'application de l'interférométrie, ces types de pannes seront étudiés dans l'état de l'art, mais pas dans les chapitres suivants.

Pour finir, les pannes byzantines sont étudiées avec tant de ferveur par la communauté scientifique qu'un domaine entier de la recherche leur est consacré : la Tolérance aux Pannes Byzantines (*Byzantine Fault Tolerance*, BFT). Cela s'explique par le fait que ces pannes soient au coeur de la plupart des problèmes de sécurité, de conformité et d'intrusion des systèmes informatiques [100]. Dans les faits, ces pannes peuvent affecter un essaim de nanosatellites, notamment sur l'exactitude des données renvoyées par les capteurs. Elles seront rapidement étudiées dans cet état de l'art, mais comme dit plus haut, seules les pannes franches seront considérées dans les chapitres suivants.

2.3.2 Mécanismes de robustesse

Les mécanismes de robustesse constituent le premier rempart face aux pannes d'un système. Comme énoncé plus haut, mais il est important de le rappeler :

Définition 1 *La **robustesse** d'un système est sa capacité à résister aux pannes et à les éviter.*

Dans un système complexe, les mécanismes de robustesse peuvent être mis en place à plusieurs niveaux : cela va de la conception du hardware à la stratégie de routage. Dans le cas d'un essaim de nanosatellites, la robustesse se manifeste par une bonne gestion de l'énergie afin d'éviter les pannes franches, ce qui impacte directement la durée de vie du système. La mise en place d'un consensus vérifiant l'exactitude des données relevées par les capteurs constitue également un mécanisme de robustesse, face aux pannes byzantines cette fois.

La majorité des mécanismes de robustesse peuvent être classés dans deux catégories : les mécanismes par *prédiction*, et les mécanismes par *anticipation*. Intéressons-nous de plus près à quelques exemples de ces mécanismes et à leur implémentation dans les systèmes courants.

Robustesse par prédiction

La robustesse par prédiction repose sur le principe de détection des pannes avant qu'elles ne surviennent : l'objectif est de les éviter. Généralement, on entraîne des modèles de *machine learning* (ou de *deep learning*) afin qu'ils puissent prévoir les zones critiques du réseau. Par exemple, il est possible de prévoir une congestion si l'on détecte un remplissage important des files d'attentes. Il convient ensuite de prendre les mesures nécessaires pour que ces pannes ne surviennent pas. L'intérêt de la robustesse par prédiction est la précision de détection des algorithmes utilisés : cela rend le système particulièrement réactif à l'état du réseau. En revanche, le coût calculatoire peut être élevé.

Dans le cadre des FANET, des chercheurs se sont intéressés à un mécanisme permettant aux flottes de drones autonomes d'optimiser leurs trajets en tenant compte de la capacité limitée de leurs batteries [17, 27]. Tout comme pour les essaims de nanosatellites, la gestion de l'énergie est centrale dans le déploiement d'une flotte de drones, car sa durée de vie en dépend [21]. Ainsi, les auteurs de [17] proposent un mécanisme en plusieurs phases, dont la première consiste à établir un modèle de la consommation d'énergie type d'un drone par des algorithmes de machine learning. Ensuite, la planification des trajets de chaque drone s'effectue en se basant sur ce modèle et sur les conditions réelles, alternant entre des phases de vol et des phases de rechargement. Il s'agit, en réalité, d'une application concrète du célèbre problème du voyageur de commerce, puisque chaque drone tente d'optimiser son trajet pour se recharger le moins souvent possible. Les auteurs prouvent alors l'utilité et l'efficacité de leur approche pour la gestion de l'énergie de la flotte, notamment au travers d'expériences terrain.

La robustesse réside aussi dans la capacité du système à contrôler et limiter la congestion du réseau, cette dernière pouvant entraîner un grand nombre de pannes transitoires telles que les pannes par omission (pertes de paquets) et les pannes temporelles (files d'attentes saturées). Dans les réseaux de capteurs sans-fil (*Wireless Sensor Networks*, WSN), le nombre élevé de noeuds dans le réseau fait du contrôle de congestion une priorité [18]. Pour éviter la surcharge du réseau, il est possible de mettre en place des techniques de machine learning afin de prédire la congestion, et re-distribuer le trafic en conséquence pour l'éviter [52, 94]. Les modèles de machine learning peuvent toutefois se tromper lors de la détection de la congestion (plus précisément, la classification des erreurs), mais il est possible d'implémenter des algorithmes permettant de paramétrer au mieux les modèles de classification afin de réduire ces erreurs, et donc d'optimiser le trafic. Pour aller plus loin, il est également possible de mettre en place des techniques de deep learning dans un réseau,

comme par exemple un mécanisme d'attention, qui permet là aussi de détecter la saturation des files d'attentes [78].

En résumé, être capable de prédire et détecter les pannes futures est la raison d'être des mécanismes de robustesse par prédiction, et ils s'avèrent particulièrement efficaces lorsqu'implémentés dans des systèmes complexes tels que les FANET ou les WSN, dont les noeuds sont très sensibles aux pertes d'énergie. En revanche, ces modèles doivent d'abord être entraînés sur des bases de données : ils ne peuvent pas être déployés directement dans le réseau. Si l'on souhaite des mécanismes plus simples et légers, il faut se tourner vers la robustesse par anticipation.

Robustesse par anticipation

La robustesse par anticipation regroupe en réalité des mécanismes très simples, mais efficaces. L'objectif n'est pas d'éviter les pannes, car on sait qu'elles surviendront : l'objectif est d'y résister malgré tout. Par exemple, dans un MANET disposant d'un protocole de routage multi-saut, on sait que les noeuds routant le plus de paquets sont ceux qui risquent de tomber en panne d'énergie le plus vite. L'idée est donc de répartir la charge de routage dès le départ, avant que des déséquilibres trop importants n'apparaissent. Contrairement à la robustesse par prédiction, l'anticipation est très peu coûteuse en termes de calcul, mais elle est forcément moins réactive. Elle convient donc aux réseaux de noeuds disposant d'une faible puissance de calcul, et dont la topologie n'évolue pas de manière brutale. Le choix dépend donc, comme toujours, de l'application et du cahier des charges.

Dans les WSN, il est courant de faire du *clustering* de noeuds afin d'ordonner le trafic et l'optimiser : cela permet de limiter le nombre de transmissions, et donc de limiter la consommation énergétique du système [36, 54]. L'optimisation est assez intuitive : imaginez devoir relier toutes les villes et les villages de France entre eux par des routes. Etablir une route unique entre chaque paire de villes n'est ni optimal, ni même réalisable. C'est pour cela que les routes allant plus ou moins dans la même direction sont agrégées et hiérarchisées (départementales, nationales, autoroutes). En plus de fluidifier le trafic, cela simplifie le réseau et rend les déplacements plus efficaces. Sans grande surprise, cela fonctionne de manière similaire avec les réseaux informatiques comme les MANET ou les WSN [87, 93], où la simplification du réseau permet de réduire la charge globale, et donc de réduire le risque de panne.

Enfin, et nous l'avons vu dans la Section 2.1.4, des mécanismes de robustesse ont déjà été pensé par des membres du projet OLFAR, applicables aux essaims de nanosatellites en mission d'interférométrie [29]. Leur proposition part de l'idée que tous les noeuds n'ont pas forcément besoin de communiquer de bout-en-bout avec l'en-

semble des noeuds de l'essaim. Par conséquent, les auteurs proposent un algorithme de clustering de noeuds, hiérarchique, permettant de regrouper les noeuds : la communication inter-groupes ne s'opère qu'à travers les noeuds de haute hiérarchie (les *cluster heads*) afin de limiter le trafic. Il est alors démontré que le regroupement des noeuds permet de réduire le nombre de paquets transmis, et donc de limiter la consommation énergétique de l'essaim, prolongeant ainsi sa durée de vie. Cependant, cette agrégation n'est pas sans conséquence sur la résilience du système, puisque qu'elle tend à réduire la redondance globale de l'essaim. Il est donc nécessaire de faire un compromis entre l'efficience énergétique et la redondance du système, car ces deux aspects sont antagonistes par définition [5].

2.3.3 Mécanismes de résilience

Par opposition à la robustesse, la résilience d'un système est mise à l'épreuve dès lors que des pannes surviennent, et que le niveau de robustesse du système n'était pas suffisamment élevé pour prévenir leur apparition. La définition de la résilience n'étant pas forcément consensuelle au sein de la communauté scientifique [7, 47, 65], nous l'énonçons comme suit :

Définition 2 *La **résilience** d'un système est sa capacité à maintenir un niveau de fonctionnement (correct ou dégradé) en présence de pannes* [6].

Concrètement, les mécanismes de résilience permettent de tolérer les pannes [7]. Attention, ils n'ont pas vocation à les corriger : les mécanismes permettant de remettre le système en état de marche sont dit de *correction* de pannes, et non de *tolérance* aux pannes. Pour reprendre l'exemple de l'autoroute cité plus haut, la résilience de cette autoroute réside dans la présence d'une bande d'arrêt d'urgence, et non du déploiement des secours (mécanisme de correction). Par conséquent, les mécanismes de résilience implémentés dans un système ne sont censés se déclencher qu'*après* l'apparition d'une panne.

Dans le cadre des MANET, il n'existe malheureusement que peu de mécanismes répondant à cette définition : en effet, les algorithmes et protocoles dits tolérants aux pannes sont souvent implémentés de sorte à éviter une panne au moment où celle-ci est prédite, ce qui rentre donc dans la catégorie de la robustesse. La résilience des MANET se manifeste généralement par la redondance des routes disponibles, ainsi

5. Pour augmenter l'efficience énergétique, il faut émettre le moins de paquets possible. Pour augmenter la redondance, il faut émettre le plus de paquets possible. Que choisir ?

6. Il n'est pas rare de trouver des articles scientifiques utilisant les termes fiabilité, robustesse, résilience, tolérance, correction ou résistance de manière interchangeable. Par conséquent, les définitions que nous donnons ici ne sont peut-être pas consensuelles, mais elles sont clairement définies et respectées rigoureusement tout au long du manuscrit.

qu'à la manière de choisir une route alternative si un noeud tombe en panne via un protocole de routage résilient [88]. Le choix de la nouvelle route est une décision sensible, car il faut veiller à toujours minimiser la consommation énergétique lors de la transmission.

En dehors des protocoles de routage, il est possible d'améliorer la résilience des MANET en réfléchissant à leur architecture. Par exemple, lors d'une mission de sauvetage après une catastrophe (avalanche, éboulement), un réseau FANET peut être déployé pour détecter les personnes disparues ou prises au piège. Cependant, la communication avec ces personnes peut être difficile à mettre en place si la bande passante des drones est faible. Pour pallier à ce problème, il est possible d'intégrer, en amont, des modules de communication supplémentaires sur les téléphones, qui ne se déclenchent que lorsqu'aucune autre forme de communication n'est possible [69]. Ce réseau hybride, composé de noeuds mobiles au sol (personnes à sauver) et de noeuds mobiles volants (drones), dispose d'une meilleure connectivité et permet ainsi de sauver les personnes plus rapidement.

Une autre approche vis-à-vis de l'architecture consiste à détecter et réduire le nombre de noeuds critiques. Dans un réseau, un noeud est dit critique si son importance dans la connectivité du réseau est plus élevée que celle des autres. Généralement, la métrique associée à cette notion d'importance est la centralité intermédiaire (*betweenness centrality*), qui est une mesure de la proportion de plus courts chemins passant par ce noeud. Un noeud à centralité élevée sera plus important qu'un noeud à centralité faible, et sera donc considéré comme critique dans le réseau. Le problème des noeuds critiques est leur vulnérabilité face aux pannes [59, 96] : premièrement, si une grande quantité de trafic transite par un seul noeud, il y a un risque non trivial de créer un goulot d'étranglement et de perdre des paquets (pannes temporelles et par omission) ; deuxièmement, en raison de la grande quantité de paquets à router, ce dernier est plus susceptible de se vider de son énergie (panne franche) ; troisièmement, une attaque ciblée contre un noeud critique aura bien plus d'impact négatif sur la connectivité du réseau (panne byzantine). Bref, moins il y a de noeuds critiques dans un réseau, plus celui-ci est résilient.

Dans un contexte légèrement différent, la résilience est primordiale dans la planification des réseaux de transports. Et oui, à force de faire des analogies avec les autoroutes, il fallait bien qu'on en parle ! La fiabilité des réseaux de transports, qu'ils soient routiers, ferroviaires ou en commun, repose en grande partie sur la capacité de ces réseaux à tolérer les incidents : ainsi, il existe de nombreuses métriques et indicateurs permettant d'évaluer leur niveau de résilience [44, 85, 90]. Un moyen simple et efficace d'évaluer la résilience d'un réseau de transports est d'analyser la présence et le nombre de chemins alternatifs viables, à savoir la redondance [48].

La redondance permet de quantifier tout ce que l'on souhaite : dans un essaim de satellites par exemple, on cherchera à quantifier le nombre de plus courts chemins entre chaque paire de noeuds. L'avantage des métriques de résilience est qu'elles peuvent s'adapter à presque tous les contextes.

Enfin, comment parler de résilience sans mentionner le plus résilient des systèmes : notre planète. La nature a trouvé un mécanisme très efficace pour assurer sa résilience, à savoir la survie du plus adapté. Ainsi, du côté de la biosphère, les écosystèmes se doivent d'être résilients pour pouvoir survivre et prospérer [83]. Notre société, ainsi que tout notre modèle socio-économique, se doivent d'être résilients pour pouvoir faire face à toutes les crises [24], au risque de s'effondrer. La résilience de notre système sociétal doit même s'étendre jusqu'à la planification architecturale des zones urbaines, afin d'assurer une efficacité et une sobriété énergétique [55]. Et pour finir, avec le changement climatique et l'intensification des événements météorologiques extrêmes, il est primordial que nous, les humains, parvenions à devenir résilient face aux variations de températures afin de pouvoir nous adapter à l'environnement [72] : ce sont les lois de la nature. La notion de résilience n'est donc pas seulement un concept scientifique théorique : en réalité, elle constitue un objectif dans notre vie de tous les jours, parfois sans même nous en rendre compte.

2.3.4 Conclusion

Les systèmes complexes peuvent être sujets à de nombreux imprévus et incidents lors de leur fonctionnement. Il est donc primordial de les rendre fiable, afin qu'ils puissent garantir une certaine qualité de service. La fiabilité des systèmes complexes repose sur deux notions principales, à savoir la robustesse et la résilience : la robustesse s'assure que les pannes n'apparaissent pas (ou peu), et la résilience permet tout de même d'assurer le service, même si une panne survient.

Les mécanismes de robustesse fonctionnent de deux manières distinctes : il est possible de prédire les pannes afin de les éviter (robustesse par prédiction), ou bien de partir du principe que des pannes auront tout de même lieu, et tenter de baisser leur probabilité d'apparition (robustesse par anticipation). La prédiction est généralement basée sur des méthodes de machine learning (ou dérivés) et accorde une grande précision à la détection des pannes, tandis que l'anticipation permet de réduire le risque de panne tout en ayant un coût calculatoire faible.

Les mécanismes de résilience, quant à eux, dépendent fortement de l'architecture du système. L'absence de noeuds critiques couplé à une importante redondance de chemins alternatifs sont des éléments clés de la tolérance aux pannes d'un réseau, car ils assurent une distribution équitable de la charge du réseau ainsi qu'une conti-

nuité du flux, même en présence de noeuds et routes défaillants. La nature est un excellent exemple de système complexe résilient : après d'innombrables catastrophes naturelles, il y a toujours de la vie sur Terre.

Un système doit donc disposer d'au moins un de ces mécanismes pour être fiable. En revanche, la mise en place conjointe de mécanismes de robustesse et de résilience peut être compliquée. En effet, certains mécanismes peuvent être tout simplement antagonistes : c'est le cas de l'efficacité énergétique, dont le but est d'émettre le moins de paquets possibles, et de la redondance des paquets, dont le but est au contraire d'en émettre le plus possible. Mais heureusement, la nature est bien faite, et il convient de s'en inspirer : à l'échelle d'un écosystème fiable, on observe un équilibre naturel entre l'efficacité énergétique et la redondance des éléments [92]. Cela signifie que pour être viable, le système n'essaie pas de maximiser l'une ou l'autre de ces métriques, mais bien de parvenir à un compromis.

C'est cet équilibre entre robustesse et résilience que l'on cherchera à atteindre dans les travaux de cette thèse, en proposant un mécanisme hybride, basé sur de la robustesse par anticipation couplée à de la résilience, dans l'objectif de fiabiliser le réseau de l'essaim de nanosatellites afin qu'il puisse remplir sa mission au mieux, et le plus longtemps possible.

Chapitre 3

Caractérisation du réseau de nanosatellites

Sommaire

3.1 Principe de fonctionnement 50
3.2 Description du modèle 50
3.2.1 Modélisation par la théorie des graphes 51
3.2.2 Hypothèses de communication 52
3.2.3 Description du dataset 53
3.3 Propriétés du réseau 55
3.3.1 Mobilité de l'essaim 55
3.3.2 Densité des noeuds 57
3.3.3 Disponibilité des ISL 61
3.4 Conclusions 64

Comme énoncé en introduction, configurer un essaim de nanosatellites en interféromètre spatial placé en orbite lunaire est un défi de taille, notamment en raison de l'important volume de données à transmettre au sein du système pour établir l'image d'observation de l'Espace lors de la phase de calcul. Aussi, la première étape de cette thèse consiste à caractériser le réseau et d'en comprendre les principales dynamiques et propriétés.

Ce chapitre reprend nos travaux publiés dans [12] et s'organise de la façon suivante : la Section 3.1 rappelle brièvement le mode opératoire particulier de l'interféromètre spatial, puis la Section 3.2 présente le modèle de l'essaim étudié. La Section 3.3 détaille les différentes propriétés réseau extraites à partir de nos données. Enfin, la Section 3.4 conclut ce chapitre sur les caractéristiques de l'essaim de nanosatellites.

3.1 Principe de fonctionnement

La caractéristique principale d'un interféromètre classique, qu'il soit terrestre ou spatial, est sa dépendance vis-à-vis de ses UT pour générer des images d'observation de l'Espace. Il est en effet nécessaire de disposer d'au moins 2 UT afin de faire de l'interférométrie de signaux spatiaux. Chaque UT apporte une information unique : ainsi, plus un interféromètre dispose d'UT, plus l'image générée est riche et détaillée.

Dans un premier temps, chaque UT collecte des données brutes d'observation de signaux spatiaux. Ensuite, le fonctionnement diffère en fonction de l'interféromètre utilisé. Dans le cas d'un interféromètre *terrestre*, les données d'observation sont centralisées puis inter-corrélées pour générer une image d'observation spatiale. Un interféromètre terrestre fonctionne donc comme un système distribué statique avec une capacité de centralisation.

En revanche, il est impossible de mettre en place cette centralisation dans un interféromètre *spatial* qui est, d'une part, mobile en orbite lunaire, et d'autre part, constitué d'UT embarqués sur des nanosatellites aux capacités de calcul strictement identiques et particulièrement limitées par rapport à un UT terrestre. Par conséquent, dans le cas d'un interféromètre spatial, chaque nanosatellite doit partager ses données et récupérer celles de tous les autres membres de l'essaim pour réaliser localement des corrélations croisées des signaux obtenus.

Le transfert de ces données peut poser problème dans un contexte spatial, du fait de la mobilité et du caractère distribué d'essaim de nanosatellites. En effet, les données générées à partir d'observations de l'Espace peuvent être particulièrement volumineuses en fonction de la fréquence d'échantillonnage des UT, et donc atteindre plusieurs Gigabits en taille sur chaque UT. La transmission simultanée d'un aussi grand volume de données peut rapidement saturer les canaux de communication des nanosatellites et entraîner des pertes de paquets dans le réseau. Ces pertes de paquets ont de graves conséquences sur la qualité de l'image générée : concrètement, chaque paquet perdu réduit la précision de l'image. Afin de prévenir les risques de congestion et de pertes de paquets, il convient dans un premier temps d'étudier en détail le réseau et d'en extraire les principales propriétés.

3.2 Description du modèle

La première étape pour étudier un essaim de nanosatellites est de le modéliser correctement. Nous faisons le choix de le représenter par la théorie des graphes, qui est une manière très récurrente de représenter les réseaux ad-hoc dans la littérature [80]. Cette partie introduit les notations employées dans notre modélisation, les

hypothèses de communication dans l'essaim de nanosatellites ainsi que le dataset utilisé dans l'intégralité de cette thèse.

3.2.1 Modélisation par la théorie des graphes

Le réseau étudié est un essaim composé de nanosatellites en orbite lunaire. Nous modélisons cet essaim par un graphe temporel $G_t = (V_t(G), E_t(G))$, ou plus simplement $G_t = (V_t, E_t)$. Ce graphe est composé d'un ensemble de sommets V_t représentant les nanosatellites, de taille $|V_t| = n(t)$, et d'un ensemble d'arêtes E_t représentant les ISL, de taille $|E_t| = m(t)$. La composante temporelle établit la topologie du graphe à chaque instant t et permet de tenir compte de la mobilité de l'essaim.

Nous avons vu dans le Chapitre 2 que la topologie idéale pour faire de l'interférométrie est une topologie à maillage partiel, ce qui implique qu'il n'y ait pas de noeud isolé du réseau. Il faut donc forcer le graphe à être connexe. Pour cela, l'essaim est modélisé comme un graphe **pondéré**, où chaque arête est caractérisée par un poids w et s'écrit donc $e_t(u, v, w)$. Nous expliquerons la manière dont le poids est défini dans la Section 3.2.2.

Ensuite, pour prendre en compte le caractère distribué du système, il est nécessaire de quantifier les sommets qui ont besoin de communiquer entre eux. Pour rappel, chaque nanosatellite doit échanger ses données avec tous les autres membres de l'essaim. Soit $N_G(t)$ le nombre de paires de sommets nécessitant une liaison bout à bout à l'instant t, appelé *nombre de flux* :

$$N_G(t) = n(t) \times (n(t) - 1)$$

Les ISL étant des liens duplex, le graphe G_t est **non-orienté** : une arête entre deux sommets n'a pas de direction. Soit $e_t(u, v, w)$ l'arête reliant les sommets u et v à l'instant t. Alors :

$$e_t(u, v, w) = e_t(v, u, w)$$

Par conséquent, le nombre de flux devient :

$$N_G(t) = \frac{n(t) \times (n(t) - 1)}{2} \tag{3.1}$$

Le nombre de flux a une grande importance dans la suite du manuscrit pour quantifier les transmissions inter-satellites ainsi que la stabilité des ISL.

3.2.2 Hypothèses de communication

Le réseau de l'essaim de nanosatellites est un MANET, et ne dispose pas d'infrastructure externe par définition. La connectivité dans le réseau est donc exclusivement assurée par les ISL. Plusieurs hypothèses sont avancées afin de modéliser le système le plus fidèlement possible.

On commence par établir des hypothèses simplificatrices sur le fonctionnement des nanosatellites. Tout d'abord, les nanosatellites de l'essaim possèdent trois rôles : ils agissent à la fois en tant qu'émetteur (appelé *source*), récepteur (appelé *destination*) et transmetteur (appelé *routeur*). La transmission des paquets dans le réseau s'opère en plusieurs sauts si les noeuds source et destination sont trop éloignés. De plus, les nanosatellites sont équipés d'antennes omnidirectionnelles : cette hypothèse est réaliste dans un contexte spatial car elle facilite la communication dans l'essaim en négligeant les problématiques liées à l'orientation des antennes. Ainsi, un nanosatellite peut joindre tous ses voisins avec une seule émission :

Hypothèse 1 *Dans un essaim de nanosatellites, les paquets sont émis en broadcast et en multi-saut.*

Ensuite, on s'intéresse aux ISL du réseau. Comme indiqué en Section 3.2.1, les canaux de communication entre les nanosatellites sont des liens duplex : les nanosatellites peuvent communiquer dans les deux sens sans contrainte supplémentaire. On suppose également que les ISL sont fiables, c'est-à-dire que les paquets envoyés sur un ISL sont forcément reçus :

Hypothèse 2 *Soit u et v deux sommets du graphe G_t. S'il existe une arête $e_t(u, v, w)$ entre ces sommets, alors la probabilité de perdre un paquet sur cette arête est nulle.*

Pour finir, les nanosatellites peuvent communiquer entre eux s'ils sont à portée l'un de l'autre. Cette portée peut être adaptative si les nanosatellites sont très éloignés, afin d'éviter de se retrouver avec des noeuds isolés. En revanche, une plus grande portée coûtera plus cher en termes d'énergie, puisque l'énergie d'émission est proportionnelle au carré de la distance. Dans notre cas, il est possible de doubler la portée de connexion pour connecter le graphe, mais les ISL ainsi établis auront un coût multiplié par 4. On appelle *portée de connexion* la distance maximale à laquelle les nanosatellites peuvent communiquer sans augmentation du coût de transmission, notée R_G :

Hypothèse 3 *Soit u et v deux sommets du graphe G_t. Si la distance euclidienne entre u et v est inférieure à R_G, alors il existe une arête de poids unitaire $e_t(u, v, 1)$.*

Si la distance est dans l'intervalle $(R_G, 2R_G]$*, alors il existe une arête de poids quadruple* $e_t(u, v, 4)$*. Sinon, les deux sommets ne sont pas connectés.*

Nous faisons le choix d'une portée de connexion par palier. Il serait possible de considérer une portée entièrement adaptative, où le poids d'une arête serait quadratiquement proportionnel à la distance euclidienne entre deux sommets, mais cela complexifie les calculs sans apporter un grand intérêt. Notons aussi que, dans notre logique, il serait possible de tripler la portée de connexion et d'avoir un poids égale à $3^2 = 9$: nous choisissons d'arrêter la transmission au-delà de $2R_G$ car cela permet d'ores et déjà d'avoir un graphe connexe la grande majorité du temps (voir la Section 3.3.3). Si malgré cette condition, un noeud se retrouve quand même isolé, il sera simplement considéré comme étant en panne.

Les hypothèses avancées dans cette section sont générales. Elles restent valables dans toute la suite du manuscrit, sauf mention contraire.

3.2.3 Description du dataset

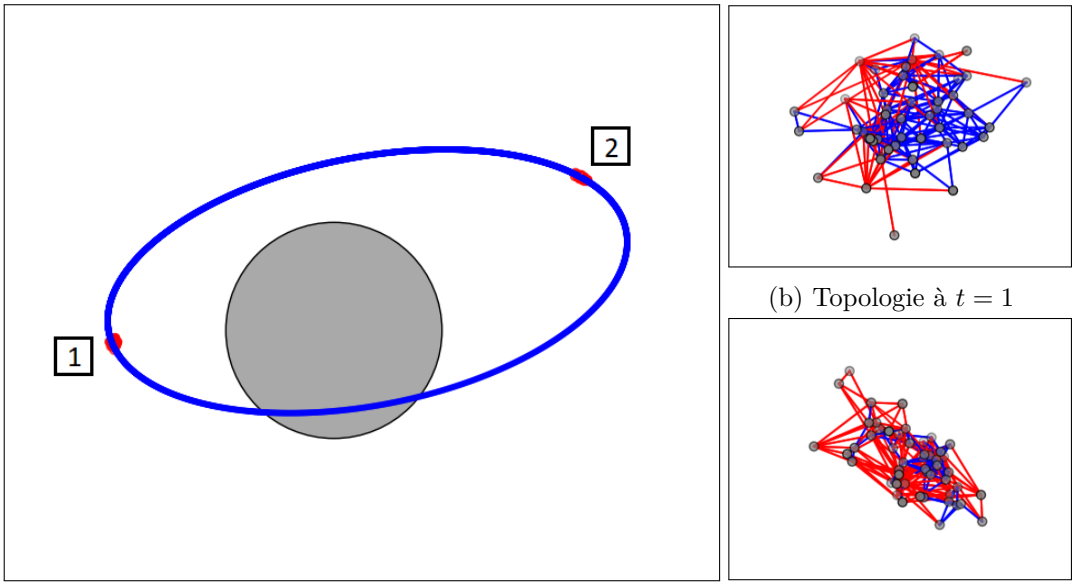
A ce jour, il n'y a aucun essaim de nanosatellites déployé en orbite lunaire, ni même terrestre. Le travail de cette thèse repose donc sur des données synthétiques générées par un simulateur de trajectoire en orbite lunaire développé pour le CNES [71]. Les paramètres orbitaux de l'essaim sont présentés dans le Tableau 3.1 : on commence par générer la trajectoire d'un seul nanosatellite en suivant les paramètres initiaux (colonne "Valeur initiale"), puis on génère les trajectoires des autres nanosatellites en injectant du bruit dans les paramètres initiaux.

Paramètre orbital	**Valeur initiale**	**Ecart-type** (σ)	**Bruit à l'échelle** (5σ)
Demi-grand axe	3416.853 km	0 km	0 km
Excentricité	0.359146	0.0015	0.0075
Inclinaison	90.188 deg	0.1 deg	0.5 deg
Longitude du noeud ascendant	237.721 deg	0.1 deg	0.5 deg
Argument du périastre	292.930 deg	0.1 deg	0.5 deg
Instant de passage au périastre	0	6	30

TABLE 3.1 – Paramètres orbitaux de l'essaim de nanosatellites

Le dataset utilisé dans cette thèse est constitué des traces de trajectoire des nanosatellites de l'essaim. Puisqu'il n'y a pas de système de géolocalisation type GNSS en orbite lunaire, la position des nanosatellites s'obtient de manière plus

rudimentaire : chaque trace contient les positions (x, y, z) des nanosatellites dans un système de coordonnées centré sur la Lune.



(a) Trajectoire de l'essaim autour de la Lune

(b) Topologie à $t = 1$

(c) Topologie à $t = 2$

FIGURE 3.1 – Représentation de la trajectoire de l'essaim en orbite lunaire et ses variations de topologies

L'essaim étudié est composé de 50 nanosatellites placés sur des orbites proches les unes des autres. Le nombre de nanosatellites est constant dans le temps, alors on simplifiera les notations suivantes dans le graphe :

$$|V_t| = n(t) = n \quad \text{et} \quad N_G(t) = N_G$$

Les positions des nanosatellites sont relevées toutes les 10 secondes, pour une durée totale d'environ 28 heures. L'essaim réalise une révolution complète autour de la Lune en 5 heures, ce qui nous donne 5 révolutions complètes dans le dataset. La portée de connexion est identique pour chaque nanosatellite, et est fixée à 30 km. Cette valeur de portée est réaliste dans le contexte spatial car les nanosatellites sont en LoS dans un environnement vide. Les paramètres généraux du dataset sont rappelés dans le Tableau 3.2.

La Figure 3.1 représente la trajectoire de l'essaim autour de la Lune, ainsi que les variations de topologies observées à deux endroits de l'orbite. Les arêtes des graphes présentées en Figures 3.1b et 3.1c sont obtenues par palier, comme énoncé par l'Hypothèse 3 : les arêtes bleues sont de poids $w = 1$ et les arêtes rouges sont de poids $w = 4$. L'évolution de la topologie de l'essaim est étudiée en profondeur dans la section suivante.

Paramètre	Notation	Valeur
Nombre de nanosatellites	n	50
Fréquence d'échantillonnage	f_s	0.1 Hz
Durée d'échantillonnage	T_s	27.8 h
Période de révolution	T_r	5 h
Portée de connexion	R_G	30 km

TABLE 3.2 – Paramètres généraux du dataset

3.3 Propriétés du réseau

Le fonctionnement de l'essaim de nanosatellites est modélisé dans un simulateur assez basique nommé *SwarmPy* : il s'agit d'un module de simulation fait main, programmé en Python3. Il permet notamment d'analyser des datasets de déplacement d'essaims de nanosatellites et de visualiser des résultats et des topologies (comme sur la Figure 3.1). Tous les résultats présentés dans ce manuscrit sont générés depuis ce simulateur [1]. Dans cette section, nous analysons un ensemble de métriques tantôt physiques, tantôt empruntées à la théorie des graphes, afin d'extraire des propriétés concernant la mobilité de l'essaim, la densité des noeuds et la disponibilité des ISL. Ces trois propriétés sont primordiales pour comprendre les dynamiques globales du système étudié.

3.3.1 Mobilité de l'essaim

L'essaim de nanosatellites est mobile autour de la Lune sur une orbite définie par les paramètres du Tableau 3.2. Dans un premier temps, nous allons nous intéresser à l'évolution de la vitesse de déplacement de l'essaim en orbite. Ensuite, nous allons étudier l'évolution des distances inter-satellites au cours du temps, afin de détecter une possible mobilité intra-essaim. La durée d'échantillonnage du dataset est trop courte pour observer des changements notables entre les différentes révolutions de l'essaim : nous allons donc restreindre notre analyse à une seule période de révolution.

La vitesse instantanée de l'essaim est calculée à partir de la distance moyenne parcourue par chaque nanosatellite par pas de temps, les pas de temps étant définis par la fréquence d'échantillonnage f_s. La Figure 3.2 présente l'évolution de la vitesse instantanée de l'essaim (courbe bleue, axe de gauche) en fonction du temps et de son éloignement par rapport à la Lune (courbe grise, axe de droite) sur une révolution

1. Les résultats sont reproductibles à partir du projet Github à l'adresse `https://github.com/eakopyan/SwarmPy_Simulator`.

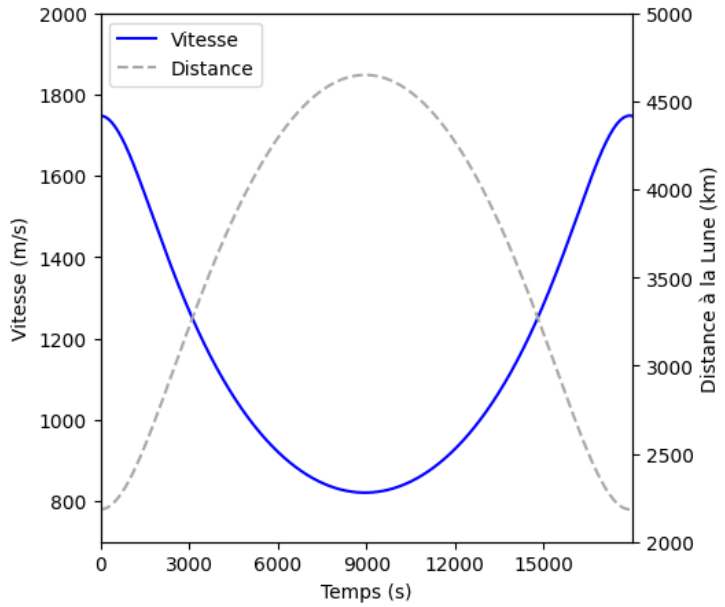


FIGURE 3.2 – Vitesse de déplacement de l'essaim sur une révolution

complète. Nous pouvons aisément voir que la vitesse de l'essaim n'est pas constante, et qu'elle est inversement proportionnelle à l'éloignement de l'essaim par rapport à la Lune : plus l'essaim en est proche, plus il se déplace rapidement. Sa vitesse varie ainsi du simple au double entre 821 m/s et 1748 m/s, pour une vitesse moyenne de **1161 m/s** (soit 4178 km/h).

Nous nous intéressons à présent à la mobilité inter-satellites en termes de vitesse et distance relatives, dont l'évolution par pas de temps sur une révolution est présentée sur la Figure 3.3. D'abord, il est évident que les distances inter-satellites ne sont pas constantes dans le temps : on observe deux minima locaux et deux maxima locaux, dont la présence n'est pas liée à l'éloignement de l'essaim par rapport à la Lune. La vitesse relative des satellites est presque constante (75 m/s), sauf lorsque l'essaim se rapproche de la Lune, auquel cas elle augmente drastiquement ; la vitesse relative moyenne est d'environ **102 m/s** (soit 368 km/h). Cette vitesse peut sembler élevée, mais elle est en réalité inférieure au dixième de la vitesse moyenne de l'essaim. On peut donc en déduire que l'essaim présente une mobilité relative modérée, c'est-à-dire que les nanosatellites sont constamment mobiles les uns par rapport aux autres avec peu de variations brusques. A partir de ces résultats, nous pouvons déduire la première propriété de l'essaim de nanosatellites sur la mobilité :

Propriété 1 *L'essaim de nanosatellites suit un double modèle de mobilité, constitué de sa mobilité par rapport à la Lune et de la mobilité de ses nanosatellites les uns par rapport aux autres.*

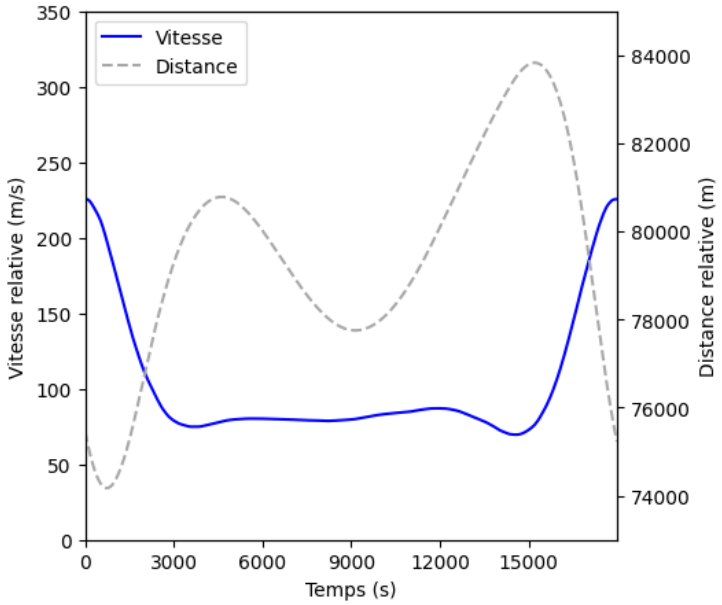


FIGURE 3.3 – Evolution de la mobilité relative des satellites sur une révolution

Cette propriété impacte directement l'ensemble d'arêtes E_t du graphe temporel, car elle stipule que ces ensembles évoluent et ne sont pas stables au cours d'une révolution :

$$\exists\,(i,j) \leq T_r \;\mid\; E_i \neq E_j \tag{3.2}$$

La Propriété 1 démontre ainsi que les ISL sont bien des liens établis dynamiquement dans le temps.

3.3.2 Densité des noeuds

Afin d'évaluer la densité des noeuds dans l'essaim, nous analysons deux métriques issues de la théorie des graphes : la force des sommets (équivalent pondéré du degré [2]) et leur coefficient de clustering.

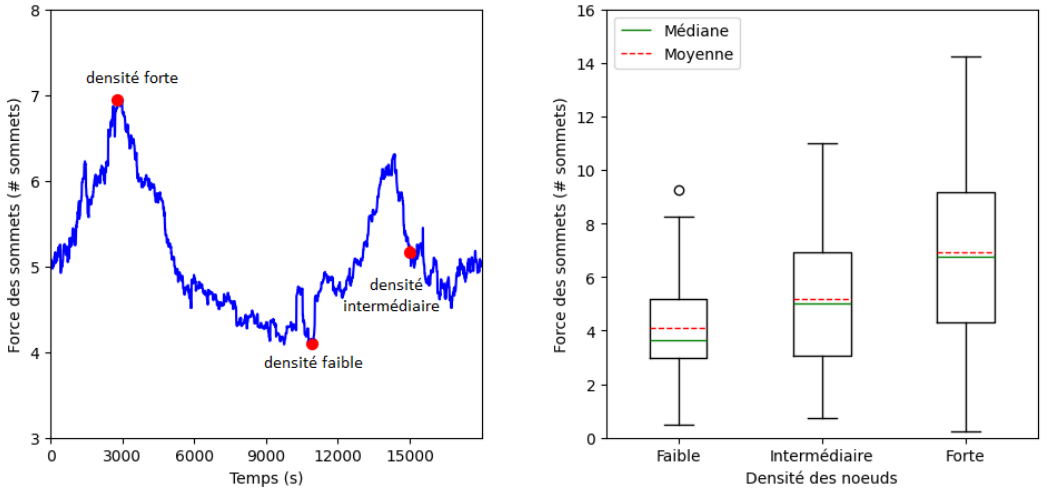
Dans un graphe G_t pondéré non-orienté, la *force* d'un sommet u correspond au nombre de sommets reliés à u par une arête à l'instant t, pondéré par l'efficacité énergétique de ces arêtes, et se note $s_t(u)$. Concrètement, dans un essaim de nanosatellites, la force d'un noeud correspond au nombre de voisins directs que ce noeud peut contacter, pondéré par l'efficacité de la transmission. La force est une manière simple et efficace de caractériser la densité de l'essaim : plus la force moyenne est

2. Cela n'a pas beaucoup de sens d'étudier le degré dans un graphe pondéré, puisque le degré ne tient pas compte du poids des arêtes.

élevée, plus la topologie de l'essaim est dense. De même, plus la distribution des forces dans une même topologie est étendue, plus la densité est hétérogène.

NB : Attention, on parle bien d'*efficacité* énergétique d'arête et non de *poids* : une arête de poids $w = 4$ est, en pratique, une arête qui sera très coûteuse à la transmission, et qui ne sera que rarement utilisée. Un noeud relié qu'avec des arêtes de poids élevé a une force faible, car la transmission lui coûte plus cher que la moyenne. La métrique prise en compte dans le calcul de la force, donc l'efficacité, est définie comme l'inverse du poids $c = \frac{1}{w}$.

La Figure 3.4 représente l'évolution des forces des sommets du graphe temporel sur une période de révolution. La Figure 3.4a présente l'évolution de la force moyenne de l'essaim au cours du temps : on constate que la force moyenne n'est pas constante et que la densité de l'essaim change au cours d'une révolution. Il est intéressant de noter que l'évolution de la force n'est pas corrélée à l'évolution des distances inter-satellites présentée sur la Figure 3.3, bien qu'elle soit calculée à partir de ces distances d'après l'Hypothèse 3. Cela démontre que l'analyse des distances seule est insuffisante pour caractériser ce système.



(a) Evolution de la force moyenne au cours du temps

(b) Distribution des forces en fonction du niveau de densité

FIGURE 3.4 – Analyse de l'évolution de la force des sommets

A partir de cette figure, nous pouvons extraire trois instants (ou topologies) d'intérêt : une topologie à densité faible (représentée par des forces basses), une topologie à densité forte (forces élevées) et une topologie à densité intermédiaire (valeur moyenne des forces sur la révolution complète, soit environ 5 sommets). Ces

trois topologies d'intérêt sont analysées plus précisément dans la Figure 3.4b, qui représente les distributions des forces en fonction de la densité de chaque topologie.

Les distributions sont modélisées par des boîtes à moustaches classiques : chaque boîte s'étend du premier quartile (Q_1) de la distribution au troisième quartile (Q_3), et est coupée par la médiane (courbe verte) et la moyenne (courbe rouge). La hauteur de la boîte correspond à l'écart inter-quartile ($IQR = Q_3 - Q_1$). La longueur des bornes, ou moustaches, correspond à toutes les valeurs supérieures à $Q_1 - 1.5IQR$ pour la borne inférieure, et inférieures à $Q_3 + 1.5IQR$ pour la borne supérieure. Toutes les valeurs en dehors de cette boîte à moustache sont considérées comme des valeurs extrêmes, représentées par des ronds.

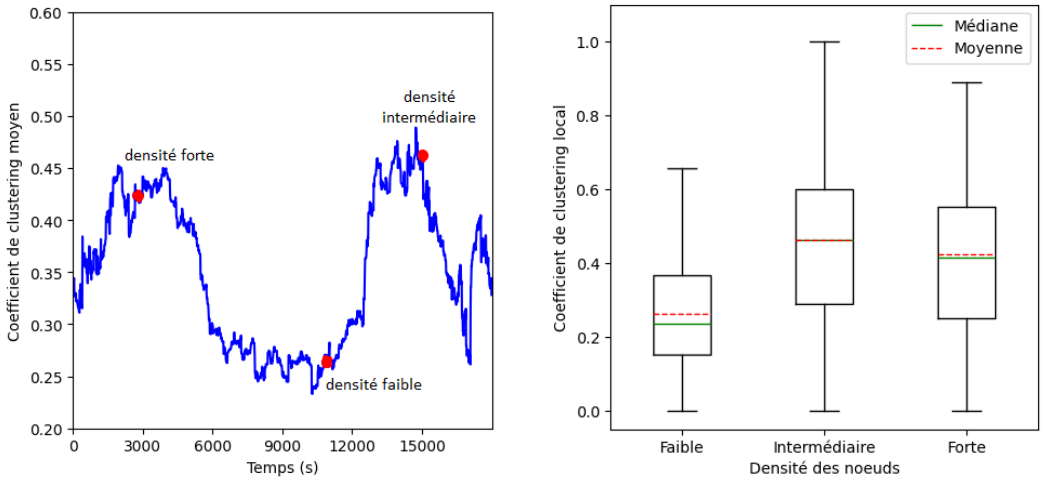
La Figure 3.4b présente donc la distribution des forces pour chacune des trois topologies sélectionnées : topologie à densité faible, intermédiaire et forte. On constate que la distribution statistique s'étend à mesure que la densité de la topologie augmente, ce qui indique que la distribution des forces (et donc la densité) a tendance à devenir de plus en plus hétérogène. Aussi, on n'observe jamais de sommet à la force nulle, donc complètement isolé du reste de l'essaim, et ce, peu importe son niveau de densité. Il se peut qu'un noeud isolé apparaisse à un autre moment, mais ces distributions suggèrent que cet événement reste rare.

La deuxième métrique étudiée pour caractériser la densité des noeuds est le *coefficient de clustering*, qui est un indicateur du regroupement des noeuds dans un réseau. Ce coefficient peut être calculé de deux manières en fonction de si l'on cherche à caractériser le réseau global ou les noeuds individuellement. Dans notre étude, nous nous intéressons au coefficient de clustering *local*, qui donne une appréciation du regroupement des voisins d'un noeud donné. Ainsi, dans un graphe G_t non-orienté, le coefficient de clustering local d'un sommet u, noté $\alpha_t(u)$, est la proportion de paires de voisins de u connectés entre eux par des arêtes, et pondéré par l'efficacité énergétique de ces arêtes (de manière analogue à la force). Il existe plusieurs formules mathématiques pour calculer le coefficient de clustering dans un graphe pondéré, en fonction des propriétés que l'on cherche à mettre en valeur [82] : dans notre cas, nous choisissons la formule présentée dans [70] qui définit le coefficient de clustering local, pour un sommet u, comme suit :

$$\alpha_t(u) = \frac{1}{\delta_t(u)(\delta_t(u)-1)} \sum_{v,k} (c_{uv} c_{uk} c_{vk})^{1/3} \tag{3.3}$$

où $\delta_t(u)$ est le degré, donc le nombre de voisins non pondéré du sommet u, v et k sont les voisins du sommet u, et c_{uv} est l'efficacité de l'arête entre u et v, donc l'inverse du poids. Cette formule est privilégiée dans notre étude car ce coefficient prend des valeurs entre 0 et 1 : il vaut 0 si le sommet n'est relié qu'à un voisin

ou moins, et 1 si tous les voisins du sommet sont interconnectés par des arêtes de coût égal à 1. Plus le coefficient est élevé, plus les voisins du sommet en question sont regroupés. Le coefficient de clustering moyen correspond à la moyenne des coefficients de clustering locaux du graphe.



(a) Evolution du coefficient de clustering au cours du temps

(b) Distribution des coefficients de clustering locaux en fonction de la densité

FIGURE 3.5 – Analyse de l'évolution du coefficient de clustering local

La Figure 3.5 présente l'évolution des coefficients de clustering sur une révolution ; les résultats sont présentés de manière similaire à la force. La Figure 3.5a montre l'évolution du coefficient de clustering du graphe au cours du temps. On remarque que le coefficient varie de manière assez similaire à la force, mais il n'y est pas exactement corrélé. La Figure 3.5b présente la distribution des coefficients de clustering locaux en fonction du niveau de densité de la topologie (il s'agit des mêmes topologies que celles étudiées pour la force). Il est très clair sur la figure qu'une force élevée en moyenne n'implique pas forcément des coefficients de clustering locaux élevés. Cela signifie que la force à elle seule ne permet pas de déduire l'état de connectivité du voisinage des sommets : un sommet peut parfois avoir moins de voisins, mais qui soient plus fortement connectés entre eux.

Ces résultats appuient le caractère très hétérogène du réseau en termes de densité de noeuds : en effet, peu importe le niveau de densité global de la topologie, il existe toujours des zones où les noeuds sont particulièrement regroupés, et à l'inverse, des zones où les noeuds sont potentiellement isolés du réseau. Nous pouvons alors déduire la deuxième propriété du réseau, caractérisant sa densité :

Propriété 2 *La densité d'un essaim de nanosatellites est très hétérogène : sa topologie peut comporter des cliques ainsi que des zones faiblement connectées.*

3.3.3 Disponibilité des ISL

Pour terminer la caractérisation de l'essaim de nanosatellites, nous nous intéressons à la disponibilité des ISL au cours d'une révolution. Pour cela, nous analysons trois métriques : le temps d'inter-contact (ICT), le temps d'isolement et le nombre de composantes connexes.

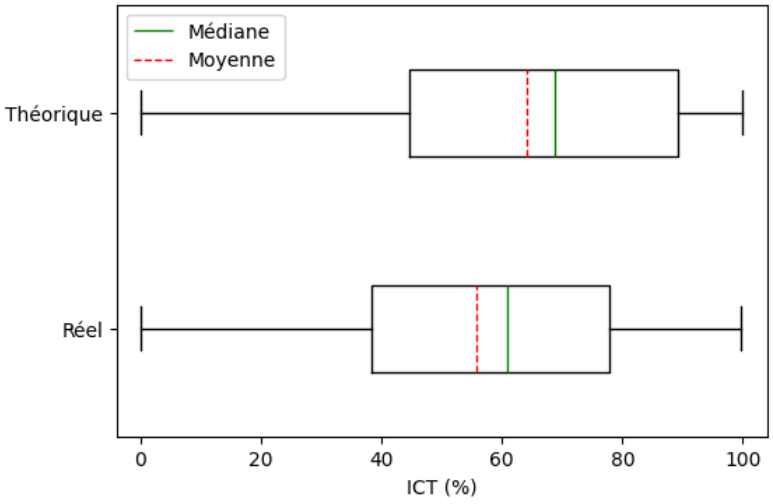


FIGURE 3.6 – Distribution des ICT théoriques et réels sur une révolution

Le temps d'inter-contact, ou ICT, est une mesure de l'indisponibilité d'un ISL. Il se définit pour une paire de noeuds $(u, v) \in V_t^2$ comme la proportion de temps pendant lequel ils ne sont pas en contact. La Figure 3.6 présente la distribution en boîtes à moustaches des ICT de toutes les paires de noeuds dans l'essaim : on constate que les ICT sont plutôt élevés en théorie (en moyenne 64% du temps passé en inter-contact). Il est important de noter qu'en théorie, les ICT sont calculés pour toutes les paires de noeuds du réseau, c'est-à-dire pour N_G paires de noeuds, correspondant au nombre de flux du réseau. Mais en pratique, certains noeuds ne rentrent jamais en contact direct, en raison de leur positionnement dans l'essaim. Par conséquent, deux distributions sont affichées sur la figure : la moyenne de la distribution *théorique* correspond à la moyenne des ICT calculée sur N_G paires de noeuds, et la moyenne de la distribution *réelle* des ICT est celle calculée sur les paires de noeuds existantes, c'est-à-dire qui établissent au moins un contact pendant la période de révolution. Dans les deux cas, cette moyenne est élevée (environ 56% de temps d'inter-contact en situation réelle), ce qui signifie qu'en moyenne, un ISL est disponible moins de **45%** du temps. Il est également intéressant de noter qu'il existe des paires de noeuds connectés en permanence (0% de temps d'inter-contact), indiquant la présence d'ISL permanents.

La deuxième métrique d'intérêt est le temps d'isolement, qui correspond à la durée pendant laquelle un noeud est complètement isolé du réseau. La différence par rapport à l'ICT est que le temps d'isolement ne caractérise pas un ISL, mais un noeud. Afin de pouvoir combiner ces deux métriques, qui ont chacune leur intérêt, le temps d'isolement est exprimé en pourcentage de temps, comme pour le calcul de l'ICT. La Figure 3.7 présente la distribution du temps d'isolement des noeuds de l'essaim sur une révolution, dans le cas où la portée peut s'étendre à $2R_G$ (60 km) et dans le cas théorique où elle serait limitée à R_G (30 km).

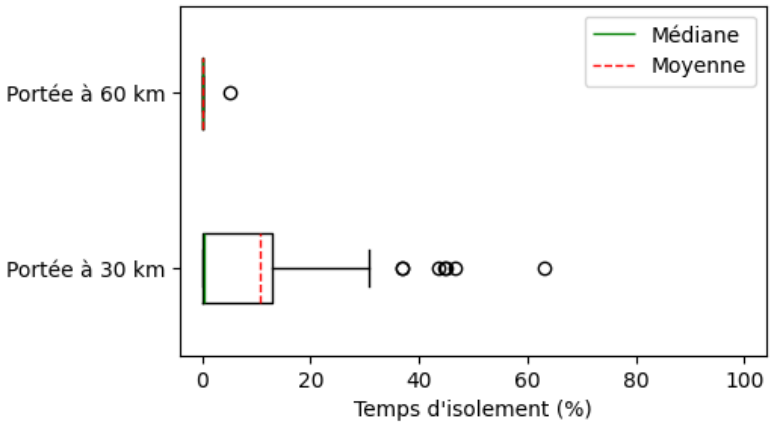


FIGURE 3.7 – Distribution du temps d'isolement des noeuds sur une révolution

Les résultats sur le temps d'isolement viennent nuancer ceux obtenus pour les ICT, qui pouvaient laisser croire que les noeuds étaient, finalement, assez peu connectés au réseau. En réalité, le cas théorique d'une portée fixe à 30 km est ajouté à la Figure 3.7 pour montrer l'intérêt majeur d'avoir une portée adaptative : si on augmente la portée à $2R_G$, on diminue drastiquement le nombre de noeuds isolés ainsi que leur temps d'isolement. La présence d'un noeud isolé devient alors un événement extrêmement rare.

La dernière métrique à laquelle on s'intéresse est le nombre de composantes connexes. Les composantes connexes font partie de la théorie des graphes, et sont définies comme les ensembles de sommets connectés entre eux par des chemins. Dans un graphe connexe, il n'y a qu'une seule composante connexe ; de même, un sommet isolé représente à lui seul une composante connexe du graphe. Par opposition au coefficient de clustering, le nombre de composantes connexes indique le niveau de dispersion des noeuds dans le réseau.

La Figure 3.8 montre l'évolution du nombre total de composantes connexes au cours d'une révolution, dans le cas d'une portée fixe (30 km) et d'une portée adaptative (60 km). Le cas théorique de la portée fixe est à nouveau ajouté pour

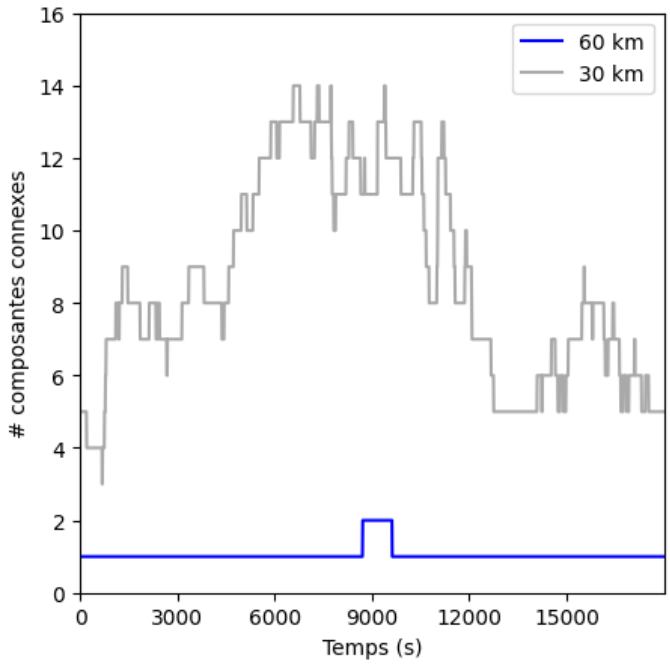


FIGURE 3.8 – Evolution du nombre de composantes connexes au cours du temps

montrer l'impact positif de la portée adaptative par palier sur la connectivité du graphe. En effet, dans le cas d'une portée adaptative, une seule composante s'isole de la composante principale pendant un court moment, s'apparentant ainsi à une panne transitoire. Si la portée était fixe, on remarquerait que les composantes connexes seraient plus nombreuses quand la densité du réseau serait faible. De plus, le réseau ne serait jamais entièrement connexe, puisqu'il y aurait toujours 3 à 14 composantes connexes au cours du temps.

A partir des résultats sur les ICT, le temps d'isolement et le nombre de composantes connexes, nous pouvons déduire la dernière propriété de l'essaim de nanosatellites, concernant la disponibilité de ses ISL :

Propriété 3 *La topologie de l'essaim de nanosatellites est connexe en quasi-permanence, et l'évolution des ISL est très dynamique.*

Cette propriété a des implications fortes sur la manière de propager les paquets dans l'essaim : puisque les ISL ne sont pas stables, les tables de routage des nanosatellites devront être régulièrement mises à jour, et ce malgré la vitesse modérée de changement de topologie. Il serait donc judicieux d'envisager une stratégie de routage **hybride** pour s'adapter à l'hétérogénéité du réseau et aux changements fréquents de topologie.

3.4 Conclusions

La configuration d'un essaim de nanosatellites en radiotélescope spatial distribué peut s'avérer compliquée en fonction des propriétés du réseau. Ce chapitre présente une méthodologie simple et détaillée pour caractériser le réseau de l'essaim, basée sur une modélisation par la théorie des graphes. En se basant sur des données de trajectoire des nanosatellites autour de la Lune et sur quelques hypothèses de communication, il nous est possible d'extraire les principales propriétés de l'essaim, qui sont rappelées ici :

— Propriété 1 : **Mobilité**. L'essaim suit un double modèle de mobilité comprenant sa mobilité en orbite lunaire et la mobilité intra-essaim de ses nanosatellites.

— Propriété 2 : **Densité**. La densité des noeuds dans l'essaim est périodique et fortement hétérogène : l'essaim présente en permanence des zones de faible et forte densités.

— Propriété 3 : **Disponibilité**. Le réseau est connexe, mais l'établissement des ISL est très dynamique et hétérogène : malgré la présence de quelques ISL permanents, la majorité des ISL de l'essaim est peu stable dans le temps.

L'hétérogénéité des noeuds et des liens est de loin la caractéristique qui ressort le plus de notre analyse. Cette hétérogénéité pose un problème majeur dans un DSS type essaim : en effet, comme rappelé en introduction et en Section 3.1, la capacité de calcul et la puissance disponible est identique pour tous les nanosatellites de l'essaim. Or, si leur répartition spatiale est hétérogène, certains nanosatellites seront surexploités par rapport aux autres ; de même, certains nanosatellites seront sous-exploités car trop éloignés du coeur de réseau. Si la charge de travail se retrouve centralisée sur une poignée de nanosatellites, ces derniers épuiseront leur énergie bien plus vite que les autres et risqueraient de tomber en panne prématurément, ce qui aurait pour conséquence de détériorer la qualité de l'image d'observation spatiale.

L'analyse des propriétés du réseau nous a permis de mettre en évidence une faille liée à la répartition de la charge de travail. Plusieurs solutions existent pour re-distribuer cette charge : la solution adoptée dans cette thèse est la re-distribution par division de graphe. Nous verrons dans les chapitres suivants que la division équitable de graphe permet de répartir efficacement la charge de calcul et de limiter la consommation énergétique des nanosatellites de l'essaim, dans le but de préserver la qualité des images d'observation et de prolonger la durée de vie du système.

Chapitre 4

Division équitable de graphe

Sommaire

4.1 Motivations 66
4.2 Algorithmes de division 67
4.2.1 Random Node Division (RND) 67
4.2.2 Multiple Independent Random Walks (MIRW) 69
4.2.3 Forest Fire Division (FFD) 70
4.2.4 Le problème du clustering 71
4.3 Evaluation de l'équité de division 72
4.3.1 Critères d'évaluation 72
4.3.2 Protocole d'évaluation 73
4.3.3 Résultats 74
4.4 Conclusions 77

Dans un essaim de nanosatellites, la transmission des paquets de données est très énergivore. Il faut donc s'assurer que la charge du réseau soit équitablement répartie entre les noeuds de l'essaim, afin d'éviter des pannes d'énergie qui baisseraient la qualité de l'image d'observation spatiale. Dans ce but, nous proposons d'effectuer une division équitable du réseau en plusieurs groupes de nanosatellites afin de limiter le nombre de paquets à transmettre, et répartir équitablement la charge du réseau. Notre objectif est de définir l'algorithme qui donne la division la plus équitable possible en tenant compte de l'évolution spatio-temporelle de l'essaim.

Ce chapitre reprend nos travaux publiés dans [11, 13] et s'organise de la manière suivante : la Section 4.1 rappelle les motivations de nos travaux et leur intérêt ; la Section 4.2 présente trois algorithmes de division de graphe, ainsi qu'un algorithme de clustering utilisé pour comparaison. Enfin, la Section 4.3 présente l'évaluation des techniques de division de graphe et leurs avantages par rapport au clustering. La conclusion de nos travaux est détaillée dans la Section 4.4.

4.1 Motivations

Dans le chapitre précédent, nous avons analysé la dynamique de l'essaim de nanosatellites, et déduit ses principales propriétés en termes de mobilité, densité des noeuds et disponibilité des ISL. La caractéristique qui est le plus ressortie de cette étude est l'importante hétérogénéité de l'essaim. Dans un système distribué où la charge de travail doit être répartie de manière homogène, l'hétérogénéité vient évidemment poser de nombreux problèmes. En effet, les nanosatellites disposent d'une énergie limitée à bord, et strictement identique pour chacun d'entre eux : par conséquent, ceux qui transmettent un plus grand nombre de paquets (en émission en tant que source, ou en retransmission en tant que routeur) se vident plus rapidement de leur énergie, et ont un risque accru de tomber en panne prématurément. Cela peut causer une réaction en chaîne où les noeuds centraux, c'est-à-dire ceux qui routent le plus de paquets, tombent en panne franche, entraînant une déconnexion globale du réseau. Il est donc primordial d'anticiper ces pannes et de distribuer la charge du réseau équitablement entre les noeuds de l'essaim : cela revient à homogénéiser un système hétérogène.

Il existe plusieurs solutions pour éviter la surcharge du réseau : nous avons déjà vu dans le Chapitre 2 qu'il est possible de regrouper les nanosatellites d'un essaim en clusters, afin de limiter la charge du réseau [29]. L'approche du clustering est également populaire dans les WSN, qui peuvent faire face à des problèmes similaires de surcharge du réseau [36, 54]. Cependant, le clustering ne permet pas de répartir la charge équitablement, ou du moins, il ne *garantit* pas une répartition équitable.

La contribution majeure de cette thèse est donc de proposer une nouvelle approche, qui permette de limiter la charge du réseau dans un système distribué fortement hétérogène, tout en la répartissant équitablement entre les noeuds. Cette approche est inspirée des techniques de clustering ainsi que des techniques d'échantillonnage de graphe : il s'agit de la *division de graphe*. Il est certes important de limiter la charge du réseau, mais cela n'empêchera pas les noeuds centraux de consommer plus d'énergie que les autres : c'est pour cette raison qu'il est vital d'ajouter une notion d'*équité* dans la répartition de la charge, afin de réduire davantage les risques de panne.

L'objectif de ce chapitre est de présenter quelques algorithmes de division de graphe, et d'en prouver l'efficacité en termes d'équité de répartition. Les algorithmes se basent sur des stratégies variées, comme la sélection aléatoire ou l'exploration de graphe. Nous nous intéresserons aux performances de ces algorithmes sur deux aspects, à savoir la conservation des propriétés du réseau, et la répartition équitable de la charge de routage. Ces algorithmes devront tenir compte de l'évolution

spatio-temporelle de l'essaim afin de garantir la qualité de service optimale dans cet interféromètre spatial distribué, et ainsi fiabiliser le système.

4.2 Algorithmes de division

La division de graphe consiste à attribuer un numéro à chaque sommet du graphe afin de former des groupes. Dans un premier temps, les sommets d'un même groupe ne communiquent qu'entre eux. Cependant, les groupes ne sont pas isolés les uns des autres : dans un second temps, les groupes communiquent entre eux pour s'échanger leurs données internes. L'objectif de la division est de simplifier, en quelques sortes, la topologie du graphe en agrégeant les sommets dans des groupes.

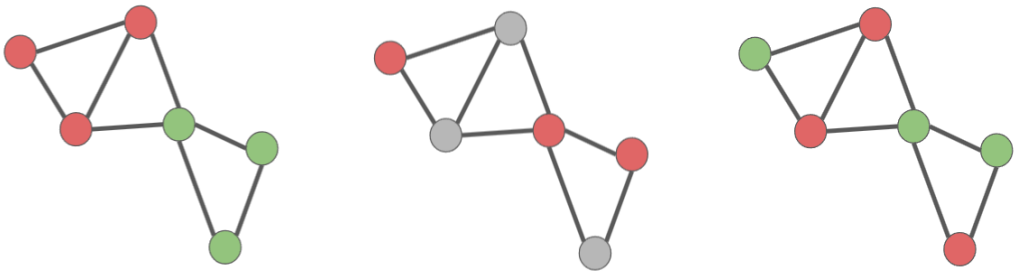
La particularité de la division de graphe est la manière de sélectionner les sommets faisant partie d'un même groupe. En effet, les sommets ne sont pas choisis en fonction de leur similarité mutuelle, mais en fonction de leur représentativité du graphe global. Cela diffère totalement du principe du clustering, dont l'objectif est de regrouper les sommets afin qu'ils soient homogènes au sein du groupe, mais donnant lieu à des groupes hétérogènes entre eux. Ici, les sommets d'un même groupe sont hétérogènes (si le graphe original est hétérogène), mais les groupes entre eux sont homogènes.

Cette différence vient de l'inspiration de la division de graphe des techniques d'échantillonnage, dont le but est de sélectionner un ensemble de sommets intrinsèquement hétérogène, mais représentatif du graphe original. En réalisant plusieurs échantillons dans un même graphe, on obtient donc des échantillons homogènes. Concrètement, il est possible d'assimiler la division de graphe à de l'échantillonnage total, c'est-à-dire créer autant d'échantillons que nécessaire afin que chaque sommet face partie d'un échantillon. Les groupes obtenus par division de graphe ne sont pas forcément connexes, mais la connexité n'est pas nécessaire, puisque les noeuds peuvent router tous les paquets du réseau : seuls les identifiants du groupe source et destination sont restrictifs quant aux paquets pouvant être envoyés.

C'est donc tout naturellement que les algorithmes de division de graphe sont dérivés d'algorithmes d'échantillonnage. Trois algorithmes sont proposés dans ce chapitre : il s'agit de Random Node Division (RND), Multiple Independent Random Walks (MIRW), et Forest Fire Division (FFD).

4.2.1 Random Node Division (RND)

L'algorithme RND (en français Division de Noeuds Aléatoire) est l'un des algorithmes de division de graphe les plus simples. Basé sur le principe de l'algorithme



(a) Clustering (2 groupes) (b) Echantillonnage (1 échantillon) (c) Division (2 groupes)

FIGURE 4.1 – Illustration des techniques de clustering, d'échantillonnage et de division de graphe

d'échantillonnage RNS (*Random Node Sampling* [46]), RND permet d'attribuer aléatoirement un numéro de groupe à chaque sommet du graphe. Par exemple, si l'on veut effectuer une division en x groupes, alors chaque sommet choisit aléatoirement un numéro entre 0 et $x-1$ avec une probabilité uniforme. Les sommets ayant sélectionné le même numéro font partie du même groupe. RND est décrit en pseudo-code dans l'Algorithme 1.

Algorithme 1 : Random Node Division (RND)

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

RND est un algorithme distribué dit de *sélection aléatoire*, c'est-à-dire que les sommets choisis à un instant t ne dépendent pas des sommets choisis à l'instant $t-1$. L'intérêt de RND est sa complexité imbattable de $O(1)$, puisque tous les sommets sélectionnent leur numéro de groupe simultanément. RND fonctionne par un mécanisme aléatoire : on peut donc espérer qu'il ait de bonnes performances en termes d'équité dans un graphe ayant un grand nombre de sommets [1].

1. Il faudrait au moins 30 sommets dans le graphe pour que la loi des grands nombre s'applique.

4.2.2 Multiple Independent Random Walks (MIRW)

Le deuxième algorithme proposé est MIRW (en français Multiples Marches Aléatoires Indépendantes), basé sur l'algorithme d'échantillonnage du même nom [2] [46]. Le principe de MIRW est d'effectuer plusieurs marches aléatoires en parallèle. Pour rappel, une marche aléatoire consiste à sélectionner aléatoirement, à chaque pas, un voisin du sommet visité. Le trajet ainsi effectué forme la marche aléatoire. Dans MIRW, on commence par sélectionner x sommets source, puis on effectue en parallèle des marches aléatoires depuis ces sommets. Quand il n'y a plus de sommet disponible, les marches aléatoires définissent les x groupes. MIRW est décrit en pseudo-code dans l'Algorithme 2.

Algorithme 2 : Multiple Independent Random Walks (MIRW)

```
Data : G_t = (V_t, E_t) ;                    /* Graphe représentant l'essaim */
Data : 0 < x ≤ |V_t| ;                       /* Nombre de groupes */
src ← sample(V_t, x) ;                       /* Sélection de x sources */
groups ← {{u} | u ∈ src} ;                   /* Formation des groupes */
free_nodes ← V_t\src ;                       /* Noeuds sans groupe */
while free_nodes ≠ ∅ do
    for walk ∈ groups do
        u = walk[−1] ;                       /* Dernier noeud assigné */
        free_neighbors ← free_nodes ∩ u.neighbors
        if free_neighbors ≠ ∅ then
            v ← choice(free_neighbors) ;     /* Marche aléatoire */
        else
            v ← choice(free_nodes) ;         /* Saut aléatoire */
        end
        walk ← walk ∪ {v} ;                  /* Ajout du noeud au groupe */
        free_nodes ← free_nodes\{v}
    end
end
```

MIRW est un algorithme dit d'*exploration*, car contrairement à RND, les sommets sélectionnés à un instant t dépendent bien de ceux visités à l'instant $t-1$. Le point faible des marches aléatoires est qu'elles peuvent se retrouver "coincées" si aucun sommet n'est disponible à un instant t : MIRW surpasse ce problème en effectuant un saut aléatoire dans le graphe, reprenant ainsi la marche à un autre endroit à partir d'un sommet disponible. On peut alors dire que MIRW est un algorithme d'exploration ordonnancé, car on sait qu'un seul et unique sommet sera choisi à chaque itération par chaque marche, et qu'il y aura au maximum x som-

2. L'algorithme d'échantillonnage forme un unique échantillon à partir des multiples marches aléatoires ; notre version de MIRW utilise chaque marche pour constituer un groupe distinct.

mets sélectionnés par itération de l'algorithme. Par conséquent, la complexité de l'algorithme est de $O(n/x)$.

4.2.3 Forest Fire Division (FFD)

Le dernier algorithme proposé dans ce chapitre est FFD (en français Division par Feu de Forêt), basé sur l'algorithme d'échantillonnage du même nom [46], similairement à MIRW. Le principe de FFD est de sélectionner x sommets source, comme MIRW, puis de propager les groupes à partir de ces sources de façon épidémique, à la manière d'un feu de forêt. FFD nécessite donc un argument supplémentaire lors de son initialisation, qui est la probabilité p de propager le feu à un sommet voisin. Les feux ainsi propagés forment x groupes distincts dès qu'il n'y a plus aucun sommet disponible. FFD est décrit en pseudo-code dans l'Algorithme 3.

Algorithme 3 : Forest Fire Division (FFD)

```
Data : G_t = (V_t, E_t) ;                  /* Graphe représentant l'essaim */
Data : 0 < x ≤ |V_t| ;                              /* Nombre de groupes */
Data : 0 < p < 1 ;                        /* Probabilité de propagation */
src ← sample(V_t, x) ;                        /* Sélection de x sources */
groups ← {{u} | u ∈ src} ;                     /* Formation des groupes */
free_nodes ← V_t\src ;                            /* Noeuds sans groupe */
burning_nodes ← src ;                                  /* Noeuds actifs */
while free_nodes ≠ ∅ do
    for u ∈ burning_nodes do
        free_neighbors ← free_nodes ∩ u.neighbors
        if free_neighbors ≠ ∅ then
            nodes ← propagate(free_neighbors, p) ;        /* Propagation */
        else
            nodes ← {choice(free_nodes)} ;                /* Saut aléatoire */
        end
        group = groups[u] ;                   /* Groupe auquel appartient u */
        group ← group ∪ nodes ;               /* Ajout des noeuds au groupe */
        burning_nodes ← nodes ;                /* Nouveaux noeuds actifs */
        free_nodes ← free_nodes\nodes
    end
end
```

FFD est lui aussi un algorithme d'exploration, mais légèrement différent de MIRW car le nombre de sommets sélectionnés à chaque instant est aléatoire : en effet, on peut *espérer* que chaque sommet sélectionne une proportion p de ses voisins disponibles, mais rien ne le garantit. On dit que FFD est un algorithme d'exploration par *propagation* : un sommet peut par exemple ne sélectionner aucun voisin lors

d'une itération, ce qui "éteint" le feu et achève la formation du groupe. Il peut aussi sélectionner tous ses voisins, résultant dans des tailles de groupes très différentes. Par conséquent, la complexité de l'algorithme est difficile à évaluer, mais dans le pire scénario, la complexité de FFD s'élève à $O(n-x)$, le pire scénario étant $x-1$ groupes qui s'arrêtent juste après l'initialisation, et 1 groupe qui continue la propagation en ne sélectionnant qu'un seul sommet à chaque itération.

FFD et MIRW sont des algorithmes centralisés : contrairement à RND, on ne peut pas les implémenter tels quels dans un essaim décentralisé. Il est possible d'imaginer des versions distribuées de ces algorithmes, mais cela n'est pas nécessaire dans notre travail car cela revient à définir un protocole, et non un algorithme, dont le principe de fonctionnement reste identique aux algorithmes que nous proposons dans cette thèse.

4.2.4 Le problème du clustering

Jusqu'ici, nous avons présenté des algorithmes de division de graphe qui permettraient de répartir équitablement la charge du réseau et de limiter la consommation énergétique. Mais il est important de noter que dans la littérature, le partitionnement de graphe est généralement fait grâce à des algorithmes de clustering, dont le but est également de réduire la consommation énergétique. Alors, pourquoi ne pas les utiliser ?

Et bien, comme énoncé plus haut, le principe du clustering n'est pas d'obtenir des groupes similaires, mais de regrouper les sommets similaires entre eux. Intuitivement, la répartition des sommets risque d'être hétérogène, or c'est justement ce que l'on cherche à éviter pour, au contraire, atteindre l'équité. Cependant, il reste intéressant de comparer les performances de nos algorithmes de division à des techniques de clustering afin d'en justifier l'intérêt et l'utilité. Pour cette raison, l'algorithme *k-means* est également implémenté et étudié afin d'évaluer ses performances par rapport aux algorithmes de division proposés.

K-means (en français k-moyennes) est un algorithme de machine learning et de clustering, très populaire dans les problèmes d'optimisation [58]. Son principe est de sélectionner x sommets, qui deviendront des chefs de clusters, et de créer des clusters autour de ces sommets de sorte que la distance entre les sommets d'un même cluster soit minimale. L'optimisation réside dans le choix des chefs de clusters à l'initialisation : plusieurs itérations sont réalisées afin de trouver les chefs de clusters qui minimiseront la distance au sein des clusters. Il est pertinent de comparer les performances de nos algorithmes de division par rapport à k-means, car ce dernier permet de paramétrer le nombre de groupes souhaité, de la même manière que

RND, MIRW et FFD. Ils ont donc un mode opératoire assez similaire. En revanche, l'algorithme est assez complexe en temps. Soit n le nombre de sommets dans le graphe, d la dimension des données, x le nombre de clusters et i le nombre maximal d'itérations de k-means : alors la complexité de k-means est égale à $O(i \times x \times n \times d)$. Par conséquent, dans un cas réel, on lui préférera sûrement un algorithme moins complexe en temps de calcul.

4.3 Evaluation de l'équité de division

Dans la section précédente, nous avons présenté trois algorithmes de division de graphe : RND, MIRW et FFD. A présent, nous allons évaluer leurs performances en termes d'équité de partage de charge et de consommation énergétique. Les critères d'évaluation sont présentés dans la Section 4.3.1, et le protocole d'évaluation est décrit dans la Section 4.3.2. Enfin, la Section 4.3.3 présente les résultats des performances de RND, MIRW et FFD, ainsi que leur comparaison avec les performances de l'algorithme de clustering k-means dans le but de démontrer l'intérêt de la division de graphe par rapport aux techniques de clustering.

4.3.1 Critères d'évaluation

L'objectif de cette étude est d'analyser les capacités des algorithmes de division de graphe à produire un partitionnement équitable du graphe original. Nous considérons qu'une division est équitable si les groupes produits préservent les caractéristiques du graphe de référence, distribuent la charge du réseau de manière homogène, et limitent la consommation énergétique globale du réseau. Nous allons donc nous intéresser à plusieurs métriques :

- Propriétés du réseau : pour évaluer la préservation des propriétés, nous analysons la densité des noeuds dans les groupes produits en termes de distribution des *forces* et des *coefficients de clustering* ;
- Charge du réseau : la répartition de la charge du réseau est étudiée à travers la distribution de la *taille* des groupes, c'est-à-dire le nombre de sommets faisant partie de chaque groupe i et notée n_i ;
- Consommation énergétique : nous analysons le nombre de paquets à émettre dans le réseau, appelé *nombre de flux*.

Le nombre de flux, noté $N_G(t)$ ou plus simplement N_G en l'absence de panne, a été défini dans la Section 3.2.1 et correspond au nombre de paires source-destinations dans un graphe. De fait, le nombre de flux représente aussi le nombre de paquets à émettre entre toutes les sources et toutes les destinations, en négli-

geant les retransmissions dues au multi-saut. Nous rappelons que le nombre de flux dans un graphe pondéré, non-orienté et non divisé est défini par l'Equation 3.1 :

$$N_G(t) = \frac{n(t) \times (n(t) - 1)}{2} \text{ soit } N_G = \frac{n(n-1)}{2}$$

où $n = |V|$ est le nombre de sommets dans le graphe G. Dans un graphe divisé, les paquets doivent être transférés entre les sommets d'un même groupe dans un premier temps, puis une fois entre les différents groupes. Soit $G_t^* = (V_t^*, E_t)$ le graphe divisé en x groupes. Les groupes sont définis comme suit :

$$V_t^* = \{V_t^0, V_t^1, ..., V_t^{x-1}\} \text{ et } \bigcup_{i<x} V_t^i = V_t$$

En l'absence de panne, le nombre de sommets reste constant. Dans ce cas, le nombre de flux du graphe pondéré, non-orienté et divisé en x groupes devient :

$$N_G^* = x(x-1) + \sum_{i<x} \frac{n_i(n_i - 1)}{2} \tag{4.1}$$

où le produit représente le nombre de paquets à émettre entre les groupes, et la somme représente le nombre de paquets à émettre au sein des groupes.

Une division équitable doit donc préserver les propriétés du réseau, en générant des groupes dont la force et le coefficient de clustering moyens soient aussi proches que possible de la force et du coefficient de clustering moyens du graphe de référence. Les groupes générés doivent également être de taille égale pour distribuer au mieux la charge du réseau. Enfin, le graphe de référence doit être divisé de sorte à limiter au maximum la consommation énergétique du réseau, en réduisant le nombre de flux moyen sur une période de révolution.

4.3.2 Protocole d'évaluation

L'évaluation des performances est réalisée grâce au module Python3 *swarm_sim* introduit dans le Chapitre 3. Dans un premier temps, il est nécessaire de calculer les métriques de référence auxquelles seront comparés les résultats des performances des algorithmes. Ainsi, la force et le coefficient de clustering moyens sont calculés à chaque instant $t < T_r$ sur le graphe de référence G_t (rappelons que la topologie du graphe évolue au cours du temps). La taille des groupes de référence est égale au nombre total de sommets divisé par le nombre de groupes n/x. Enfin, le nombre de flux de référence est donné par l'Equation 3.1.

A l'initialisation t_0, le graphe est divisé en x groupes avec chacun des algorithmes : RND, MIRW, FFD et k-means. Ensuite, à chaque instant $t \geq t_0$, nous

calculons les valeurs moyennes des métriques de chaque groupe. Nous comparons les distributions de la force moyenne, du coefficient de clustering moyen et de la taille des groupes [3] par rapport aux valeurs de référence à l'aide de la RMSE (*Root-Mean-Square Error*, en français Racine de l'Erreur Quadratique Moyenne). La RMSE est un excellent indicateur dans notre situation, car elle donne une information sur la dispersion d'une distribution par rapport à une valeur de référence : plus la RMSE est faible, plus la distribution est resserrée autour de la valeur de référence. Soit $Y = \{y_1, y_2, ..., y_x\}$ la distribution d'une métrique avec valeur de référence y_0 : alors la RMSE de cette distribution est calculée de la sorte :

$$\mathrm{RMSE}(Y) = \sqrt{\frac{1}{x}\sum_{i=1}^{x}(y_i - y_0)^2} \tag{4.2}$$

Enfin, le nombre de flux du graphe divisé, donné par l'Equation 4.1, est directement comparé à sa valeur de référence.

L'étape précédente est répétée un grand nombre de fois afin d'obtenir une large variété de partitionnements. Cette comparaison est effectuée pour chaque algorithme, et pour chaque nombre de groupes possible : l'idée est aussi d'étudier l'impact du nombre de groupes x sur les performances des algorithmes en termes d'équité de division (et de clustering pour k-means). Les paramètres de la simulation sont listés dans le Tableau 4.1.

Paramètre	**Notation**	**Valeur**
Durée de simulation	T_r	5 h
Nombre de sommets	n	50
Portée de connexion	R_G	30 km
Extension de portée	$2R_G$	60 km
Nombre de groupes	x	$[1 : n]$
Nombre de répétitions	$iter$	30
Probabilité de propagation (FFD)	p	0.7

TABLE 4.1 – Paramètres de la simulation pour l'évaluation de l'équité

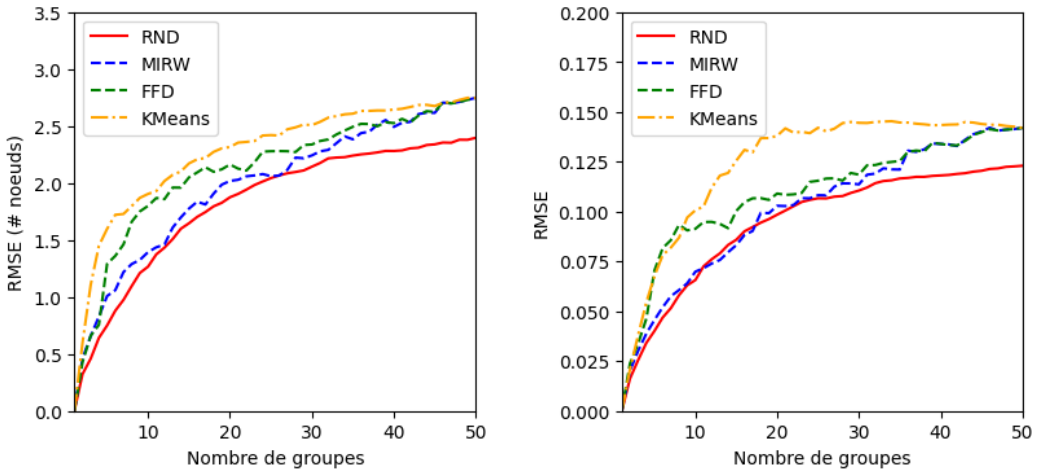
4.3.3 Résultats

La Figure 4.2 présente la RMSE des métriques de densité de noeuds, obtenue avec chacun des quatre algorithmes, en fonction du nombre de groupes x. On constate des résultats similaires sur la force (Figure 4.2a) et le coefficient

3. Puisque la division n'est faite qu'une seule fois à l'initialisation, la taille des groupes est constante dans le temps : sa RMSE n'est alors calculée qu'à l'initialisation.

de clustering (Figure 4.2b) : en effet, l'algorithme RND est celui donnant quasi-systématiquement la RMSE la plus faible, et donc la distribution la plus proche de la valeur de référence. A l'inverse, l'algorithme k-means donne la RMSE la plus élevée, indépendamment du nombre de groupes. Ce résultat valide notre intuition quant à l'équité de partitionnement des algorithmes de clustering. Les algorithmes MIRW et FFD donnent des résultats intermédiaires, mais MIRW est bien plus proche des performances de RND que l'est FFD.

Ensuite, il est intéressant de noter que la RMSE augmente avec le nombre de groupes, indépendamment de l'algorithme choisi. Elle est évidemment égale à 0 pour une division en 1 groupe, puisque ce groupe correspond au graphe de référence. En revanche, lorsque le nombre de groupe est égal à n, on constate que les algorithmes MIRW, FFD et k-means convergent : cela s'explique simplement par le fait que ces trois algorithmes commencent par initier $x = n$ sommets source, et qu'il y a donc exactement 50 groupes de 1 sommet. Ce n'est pas le cas de RND, puisque les sommets choisissent un numéro de groupe aléatoirement entre 0 et $n - 1$: il est donc possible que certains sommets choisissent le même numéro de groupe. En conclusion, RND est l'algorithme permettant au mieux de conserver les propriétés du réseau.



(a) Force moyenne par groupe (b) Coefficient de clustering moyen

FIGURE 4.2 – RMSE de la distribution des métriques de densité de noeuds

Analysons maintenant la taille des groupes obtenus : la RMSE de la distribution de la taille des groupes, en fonction du nombre de groupes, est présentée sur la Figure 4.3. Cette figure donne deux résultats majeurs : le premier est, à nouveau, la convergence des algorithmes MIRW, FFD et k-means lorsque x tend vers n, comme expliqué plus haut. Le deuxième est l'excellente performance de MIRW, en parti-

culier lorsque n est divisible par x. Ce résultat est dû au caractère ordonnancé de l'algorithme MIRW, qui choisit exactement un sommet par groupe par itération, jusqu'à épuisement des sommets disponibles, contrairement aux trois autres algorithmes pour lesquels le nombre de sommets sélectionnés n'est pas garanti. MIRW surpasse donc de loin les algorithmes RND, FFD et k-means en termes de distribution des sommets par groupe, et donc de répartition de charge du réseau.

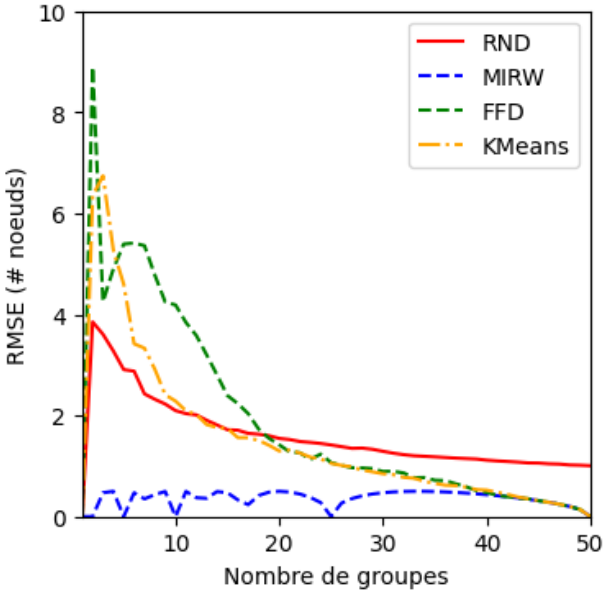
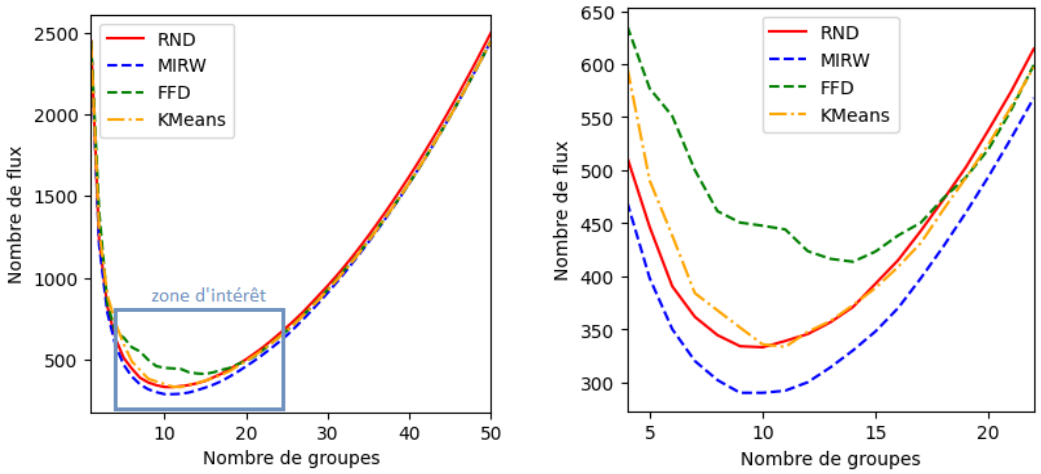


FIGURE 4.3 – RMSE de la distribution de la taille des groupes

Enfin, penchons-nous sur la consommation énergétique : la Figure 4.4 présente l'évolution du nombre de flux en fonction du nombre de groupes. D'après les Equations 3.1 et 4.1, le nombre de flux d'un graphe divisé est égal au nombre de flux du graphe de référence lorsque $x = 1$ ou $x = n$, ce qu'on observe sur la Figure 4.4a. En revanche, le partitionnement de graphe permet de réduire drastiquement ce nombre de flux. On observe alors facilement sur la Figure 4.4b que les quatre algorithmes atteignent une valeur optimale de nombre de flux pour des valeurs de x comprises entre 10 et 15 groupes, environ. Le nombre de flux se retrouve alors presque divisé par dix, ce qui signifie que le partitionnement de graphe, que ce soit par des techniques de division ou de clustering, permet de limiter fortement le nombre de paquets à transmettre, et donc de consommer moins d'énergie.

Il faut cependant noter que, malgré de très bonnes performances de la part des quatre algorithmes, MIRW est encore une fois celui qui obtient les meilleurs résultats, pour une division en $x = 10$ groupes. RND et k-means ont des performances assez similaires et atteignent également leur valeur optimale pour une division en

$x = 10$ groupes. FFD, quant à lui, est légèrement moins performant que les autres, mais atteint tout de même une valeur optimale pour $x = 14$ groupes. En résumé, les quatre algorithmes permettent de limiter la consommation énergétique du réseau, mais MIRW est le plus efficace d'entre eux.



(a) Vision globale (b) Zoom sur la zone d'intérêt

FIGURE 4.4 – Evolution du nombre de flux en fonction du nombre de groupes

En conclusion, les algorithmes de division, et en particulier RND, obtiennent les meilleures performances sur la conservation des propriétés du graphe de référence. Cependant, MIRW est de loin le meilleur choix lorsque l'on veut obtenir une répartition équitable de la charge du réseau, ainsi que limiter sa consommation énergétique, et ce, peu importe le nombre de groupes choisi.

4.4 Conclusions

Dans un essaim de nanosatellites fonctionnant comme un système distribué, il est primordial d'équilibrer la charge du réseau pour améliorer les performances de communication et la gestion de l'énergie. Dans ce chapitre, nous avons proposé un mécanisme de division de graphe permettant partitionner le graphe en plusieurs groupes, et de limiter les transmissions de paquets de données aux seuls paquets dont les sommets source et destination font partie du même groupe. Cette limitation a pour effet de réduire la consommation énergétique des sommets liée à la transmission des données, à condition que la division du graphe soit équitable.

Nous présentons et analysons les performances de trois algorithmes de division : l'algorithme de sélection aléatoire RND, et les algorithmes exploratoires MIRW et FFD. Leurs performances sont comparées sur leur capacité à préserver les propriétés

du réseau en termes de densité de noeuds, répartir les sommets équitablement entre les groupes pour distribuer la charge du réseau, et limiter le nombre de paquets à transmettre pour limiter la consommation énergétique globale de l'essaim.

Les résultats obtenus montrent que RND est l'algorithme permettant au mieux de préserver les propriétés du réseau. De plus, RND est particulièrement peu complexe en temps de calcul, ce qui en fait un choix très intéressant pour un système distribué tel qu'un essaim de nanosatellites. Cependant, MIRW est de loin le meilleur algorithme pour la répartition équitable des sommets dans les groupes ainsi que l'optimisation de la consommation énergétique, et ce, peu importe le nombre de groupes souhaité lors de la division.

Les performances des algorithmes de division ont également été comparées à celle de l'algorithme de clustering k-means. Intuitivement, un algorithme de clustering n'est pas équitable, par nature, dans son partitionnement du réseau. Les résultats nous ont effectivement démontré que k-means offrait les pires performances en termes de conservation des propriétés du réseau, et n'était pas très équitable dans sa répartition de la charge du réseau. En revanche, il reste très efficace pour limiter la consommation énergétique globale, ce qui corrobore les résultats obtenus dans la littérature quant à l'intérêt du clustering dans les réseaux denses. Cependant, les performances des algorithmes de divisions tels que MIRW et RND restent meilleures, rendant ces algorithmes plus intéressants tout en étant moins complexes à l'implémentation.

Nous avons ainsi démontré l'intérêt de la division de graphe dans un système distribué tel qu'un essaim de nanosatellites. Mais à ce stade, on ignore encore l'impact réel de cette division sur la fiabilité de l'essaim : la division permet-elle de fiabiliser le système, ou au contraire, le rend-t-elle plus sensible aux pannes ? Dans le prochain chapitre, nous étudierons l'impact de la division de graphe sur la fiabilité du système du point de vue de sa robustesse, et de sa résilience.

Chapitre 5

Fiabilité du système distribué

Sommaire

5.1 Appréciation de la fiabilité réseau 80
5.1.1 Métriques de robustesse 80
5.1.2 Métriques de résilience 82
5.2 Impact de la division sur la fiabilité réseau 84
5.2.1 Protocole expérimental 85
5.2.2 Fiabilité du graphe non divisé 86
5.2.3 Résultats 88
5.3 Conclusion 94

L'objectif final de cette thèse est de fiabiliser l'essaim de nanosatellites en le rendant plus robuste et résilient face aux pannes qui pourraient survenir dans l'Espace. Nous avons précédemment présenté des algorithmes de division de graphe, qui permettent de répartir équitablement la charge du réseau et de limiter la consommation énergétique du système. Mais quel est leur impact réel sur la fiabilité de l'essaim ? La division de graphe permet-elle d'améliorer la robustesse et la résilience du système, ou au contraire, le rend-elle plus sensible aux pannes ?

Ce chapitre reprend nos travaux soumis au journal *Computer Networks* [14], et est structuré de la manière suivante : la Section 5.1 présente un ensemble de métriques permettant de caractériser la fiabilité d'un système ; ces métriques sont évaluées dans le graphe de référence et le graphe divisé par les algorithmes RND, MIRW, FFD et k-means dans la Section 5.2, afin de comprendre le réel impact de la division de graphe sur le système. Enfin, la Section 5.3 conclut le dernier chapitre de cette thèse.

5.1 Appréciation de la fiabilité réseau

Comme énoncé dans le Chapitre 2, tout système distribué complexe doit être fiable afin de pouvoir garantir une bonne qualité de service. On dit qu'un système est fiable s'il démontre une capacité à résister aux pannes, ou à les éviter : il s'agit de la robustesse. Le système doit également être en mesure de garantir un certain niveau de fonctionnement (correct ou dégradé) en présence de pannes, si la robustesse n'était pas au niveau : il s'agit de sa résilience.

Dans le Chapitre 3, nous avons analysé les propriétés du réseau et avons décelé une forte hétérogénéité dans l'essaim de nanosatellites, que ce soit en termes de densité de noeuds ou de disponibilité des liens. Or l'hétérogénéité peut poser de graves problèmes dans un système distribué, car la charge du réseau risque de ne pas être répartie équitablement entre les noeuds, ce qui augmente le risque de pannes franches dues à la perte d'énergie. Nous avons ainsi présenté, dans le Chapitre 4, quelques algorithmes basés sur la division de graphe permettant de re-distribuer la charge du réseau, et limiter la consommation énergétique. De fait, la division de graphe permet de prolonger la durée de vie du système en limitant le risque de panne d'énergie.

Cependant, la fiabilité réseau regroupe de nombreux critères autres que la consommation énergétique, qui peuvent parfois être antagonistes : c'est par exemple le cas de la redondance et de l'efficacité énergétique. Il est alors important d'évaluer l'impact global de la division de graphe sur la fiabilité du réseau, en termes de robustesse ainsi que de résilience. Nous détaillons dans cette section les principales métriques de robustesse et de résilience qui constitueront, dans la suite du chapitre, les critères d'évaluation de la fiabilité d'un système distribué tel que l'essaim de nanosatellites.

5.1.1 Métriques de robustesse

Afin d'évaluer le niveau de robustesse du système, nous allons analyser les trois métriques suivantes : la robustesse de flux, le coût de routage, et l'efficience réseau. La combinaison de ces trois métriques permet de décrire précisément la dynamique de transmission de paquets dans le réseau.

Robustesse de flux

La robustesse de flux, notée $F_t(G)$, indique la proportion de noeuds joignables dans le réseau en multi-saut. Mathématiquement, la robustesse de flux d'un graphe non divisé G (ou divisé G^*) à un instant t correspond à la proportion de paires

de sommets source-destination reliés par au moins un chemin, et se définit comme suit :

$$F_t(G) = \frac{f_t(G)}{N_G} \tag{5.1}$$

où $f_t(G)$ est le nombre de paires uniques de sommets reliés par un chemin à un instant t, et N_G est le nombre de flux défini dans l'Equation 3.1, c'est-à-dire le nombre maximal de paires de sommets dans le graphe. Dans un graphe divisé, il faut simplement remplacer N_G par N_G^* tel que défini dans l'Equation 4.1. La robustesse de flux vaut 1 lorsque le graphe est connexe, ce qui sera le cas la plupart du temps d'après les résultats sur la disponibilité des liens. Si elle est inférieure à 1, cela indique que des sommets ne sont pas joignables, et qu'il y a donc une panne dans le réseau.

Coût de routage

Le coût de routage, noté $R_t(G)$, indique le nombre total de paquets à transmettre dans le réseau en tenant compte des retransmissions dues au multi-saut : il s'agit d'une estimation précise de la charge du réseau. Concrètement, le coût de routage correspond à la somme de la longueur de tous les plus courts chemins entre toutes les sources et toutes les destinations. Dans un graphe divisé, les sources et destinations sont restreintes aux sommets d'un même groupe dans un premier temps, ce qui limite le coût de routage par rapport au même graphe non divisé. Intrinsèquement lié à l'efficacité énergétique, le coût de routage se définit à chaque instant t comme suit :

$$R_t(G) = \sum_{u,v \in V_t^2} d(u,v) \tag{5.2}$$

où $d(u,v)$ correspond à la longueur du plus court chemin entre les sommets u et v. L'objectif d'un essaim de nanosatellites fiable est de réduire au maximum ce coût de routage, afin de limiter la consommation énergétique et prolonger sa durée de vie.

Efficience réseau

L'efficience réseau, notée $\Theta_t(G)$, caractérise la longueur des routes entre les noeuds du réseau. Il s'agit d'un coefficient compris entre 0 et 1, qui tend vers 1 lorsque les routes sont courtes, et vers 0 à mesure que les routes s'allongent. Typiquement, dans un graphe à maillage complet tel que décrit dans la Section 2.2.3,

tous les noeuds sont interconnectés : les routes sont donc très courtes, puisqu'il n'y a même pas de multi-saut, et l'efficience réseau vaut 1. Dans un graphe à maillage partiel tel que décrit dans cette thèse, l'efficience réseau sera certainement moins élevée.

Afin de calculer l'efficience réseau d'un graphe, il convient d'abord de calculer l'efficience de chaque paire de sommets du graphe, notée $\theta(u, v)$, de la manière suivante :

$$\theta(u, v) = \frac{1}{d(u, v)}$$

L'efficience de la paire vaut 0 s'il n'y a pas de chemin entre u et v, et 1 si ces sommets sont adjacents. L'efficience réseau du graphe G à un instant t correspond alors à la moyenne des efficiences des paires :

$$\Theta_t(G) = \frac{1}{2N_G} \sum_{u,v \in V_t^2, u \neq v} \theta(u, v) \tag{5.3}$$

Dans un graphe divisé, l'efficience réseau ne prend en compte que les paires source-destinations définies par les groupes et devient :

$$\Theta_t(G^*) = \frac{1}{2N_G^*} \sum_{V_t^i \in V_t^*} \left(\sum_{u,v \in (V_t^i)^2} \theta(u, v) \right) \tag{5.4}$$

Idéalement, dans un système robuste, l'efficience réseau doit être proche de 1, indiquant qu'il y a peu de retransmissions nécessaires pour le multi-saut.

5.1.2 Métriques de résilience

Nous proposons trois métriques permettant d'évaluer le niveau de résilience de l'essaim de nanosatellites : la redondance des chemins, leur disparité, et le nombre de noeuds critiques. La combinaison de ces métriques permet de caractériser la capacité des noeuds à transférer des paquets de données en présence de pannes.

Redondance

La redondance des chemins, notée $\Psi_t(G)$, estime la présence et la quantité de chemins alternatifs dans le graphe. La définition d'un chemin alternatif peut varier en fonction du domaine d'application : dans notre cas, nous appelons chemins alternatifs l'ensemble des plus courts chemins entre un sommet source et un sommet destination. En effet, le réseau est tellement dense qu'on observe généralement plusieurs plus courts chemins entre chaque paire source-destination. Mathématique-

ment, la redondance des chemins d'un graphe G à un instant t est définie comme la moyenne du nombre de chemins alternatifs entre chaque paire source-destination :

$$\Psi_t(G) = \frac{1}{2N_G} \sum_{u,v \in V_t^2} |P_{uv}| \tag{5.5}$$

où P_{uv} est l'ensemble des plus courts chemins entre les sommets u et v. Notons qu'il serait possible d'étendre la définition des chemins alternatifs aux chemins étant un peu plus longs que le plus court chemin, mais puisque notre graphe est très dense, cela n'est pas nécessaire. En revanche, c'est une pratique courante dans les applications aux réseaux de transports en commun [48].

Une forte redondance de chemins est un élément clé dans un réseau résilient, car elle implique que les paquets peuvent être routés dans le réseau malgré la présence de pannes, puisqu'il y a plusieurs routes disponibles.

Disparité

La disparité des chemins, notée $\Delta_t(G)$, est une métrique complémentaire à la redondance qui permet d'évaluer la variété des chemins alternatifs disponibles. Concrètement, la disparité quantifie le nombre de sommets différents dans un ensemble de plus courts chemins. Dans un premier temps, il est nécessaire de calculer la disparité de chaque paire du graphe, définie pour une paire de sommets u et v comme suit :

$$\delta(u,v) = \frac{1}{2|P_{uv}|} \sum_{p_i,p_j \in P_{uv}^2} \frac{|p_i(u,v) \oplus p_j(u,v)|}{d(u,v) - 1}$$

où le numérateur représente le nombre de sommets distincts entre les plus courts chemins p_i et p_j, normalisé par le nombre de sommets intermédiaires entre u et v. La disparité de la paire vaut 1 si p_i et p_j n'empruntent aucun sommet en commun, et 0 s'il n'y a qu'un seul plus court chemin, voire pas de chemin du tout. Le calcul de la disparité des paires est illustré sur la Figure 5.1. La disparité des chemins du graphe G est alors simplement définie comme la moyenne des disparités des paires du graphe à un instant t :

$$\Delta_t(G) = \frac{1}{2N_G} \sum_{u,v \in V_t^2} \delta(u,v) \tag{5.6}$$

Une disparité élevée est signe d'un réseau résilient : plus les sommets traversés sont différents, moins il y a de risque qu'une transmission soit impactée par une panne sur l'un des chemins alternatifs.

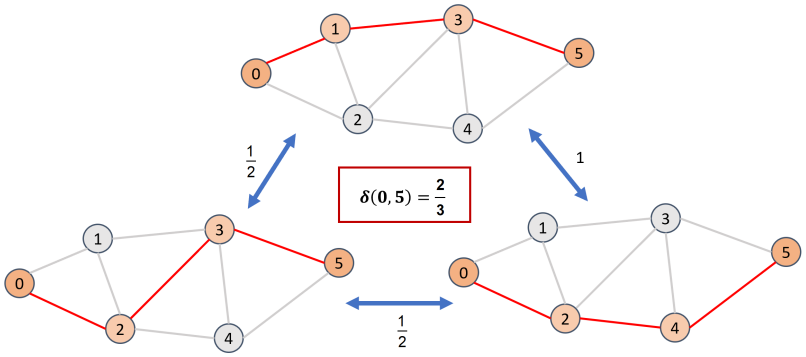


FIGURE 5.1 – Calcul de la disparité de la paire de sommets $(0, 5)$

Criticité

La criticité des noeuds, notée $C_t(G)$, indique la présence de sommets critiques dans le graphe. Un sommet est dit critique s'il est compris dans une part importante des plus courts chemins entre toutes les paires source-destination du graphe. La criticité du graphe dépend donc d'une autre métrique, à savoir la centralité intermédiaire, qui évalue justement la proportion de plus courts chemins passant par un sommet i :

$$BC_t(i) = \frac{1}{2N_G} \sum_{u,v \in V_t^2} \frac{|P_{uv}(i)|}{|P_{uv}|} \tag{5.7}$$

où $P_{uv}(i)$ correspond à l'ensemble des plus courts chemins entre les sommets u et v passant par i. L'ensemble des sommets critiques du graphe, appelé plus simplement criticité, comprend alors l'ensemble des sommets dont la centralité intermédiaire à un instant t est supérieure à une valeur seuil ϵ :

$$C_t(G) = \{u \in V_t \,|\, BC_t(u) \geq \epsilon\} \tag{5.8}$$

Plus la criticité est faible, moins il y a de noeuds critiques dans le réseau, et moins il y a de risque que ces noeuds tombent en panne pour cause de répartition inégale de la charge.

5.2 Impact de la division sur la fiabilité réseau

Dans cette section, nous évaluons l'impact de la division de graphe sur la fiabilité du système en analysant l'évolution des métriques de robustesse et de résilience, avant et après la division. Pour caractériser la robustesse, nous nous intéressons à

la robustesse de flux, au coût de routage et à l'efficience réseau. Pour la résilience, nous analysons la redondance des chemins, leur disparité et la criticité des noeuds. La Section 5.2.1 décrit le protocole expérimental mis en place dans notre analyse. La fiabilité du graphe de référence, non divisé, est étudiée dans la Section 5.2.2. Pour finir, les résultats quant à l'impact de la division par les algorithmes RND, MIRW, FFD et k-means sont présentés dans la Section 5.2.3.

5.2.1 Protocole expérimental

Pour comprendre l'influence de la division de graphe sur la fiabilité du système, il est tout d'abord nécessaire d'évaluer le niveau de fiabilité du graphe temporel de référence, non divisé, en termes de robustesse et de résilience à l'aide des métriques présentées ci-dessus. Pour cela, nous effectuons une analyse temporelle sur une période de révolution complète : à chaque instant $t < T_r$, nous calculons les métriques de robustesse et de résilience sur le graphe non divisé, d'après les Equations 5.1, 5.2, 5.3, 5.5, 5.6 et 5.8. Ces premiers résultats constituent les performances de référence du système en termes de robustesse et de résilience.

Nous analysons ensuite l'impact de la division de graphe et du clustering sur ces performances de référence. Pour cela, à l'initialisation t_0, nous commençons par effectuer une division en x groupes à l'aide de chacun des algorithmes : RND, MIRW, FFD et k-means. Les groupes ainsi constitués sont immuables, ce qui signifie que les sommets ne peuvent pas changer de groupe en cours de simulation. Ensuite, à chaque instant $t < T_r$, nous calculons la distribution des métriques de robustesse et de résilience obtenues pour chaque groupe à l'aide des Equations 5.1, 5.2, 5.4, 5.5, 5.6 et 5.8. Puisque les algorithmes se basent sur des processus aléatoires, cette étape est répétée un grand nombre de fois afin d'obtenir une estimation précise des performances des algorithmes de division.

Pour finir, la valeur moyenne des distributions obtenues à chaque instant est comparée aux performances de référence afin de mettre en lumière l'impact réel de la division de graphe sur la fiabilité du système. La distribution exacte des métriques sur le graphe divisé est également comparée aux performances de référence à trois instants précis, correspondant à des topologies de densité faible, intermédiaire et élevée, afin d'obtenir plus de détails sur la dispersion des métriques au sein des groupes.

Les paramètres de la simulation sont listés dans le Tableau 5.1. Notons que l'on choisit d'effectuer une division en $x = 10$ groupes, car d'après les résultats obtenus dans la Section 4.3.3 sur le nombre de flux, la division en 10 groupes est énergétiquement optimale pour la majorité des algorithmes.

Paramètre	Notation	Valeur
Durée de simulation	T_r	5 h
Nombre de sommets	n	50
Portée de connexion	R_G	30 km
Extension de portée	$2R_G$	60 km
Nombre de groupes	x	10
Nombre de répétitions	$iter$	30
Probabilité de propagation (FFD)	p	0.7
Valeur seuil de criticité	ϵ	5%

TABLE 5.1 – Paramètres de la simulation pour l'évaluation de l'impact

5.2.2 Fiabilité du graphe non divisé

Dans un premier temps, nous nous intéressons au niveau de fiabilité de l'essaim de nanosatellites non divisé. La Figure 5.2 présente l'évolution temporelle des métriques de fiabilité calculées sur le graphe de référence selon le protocole expérimental décrit ci-dessus.

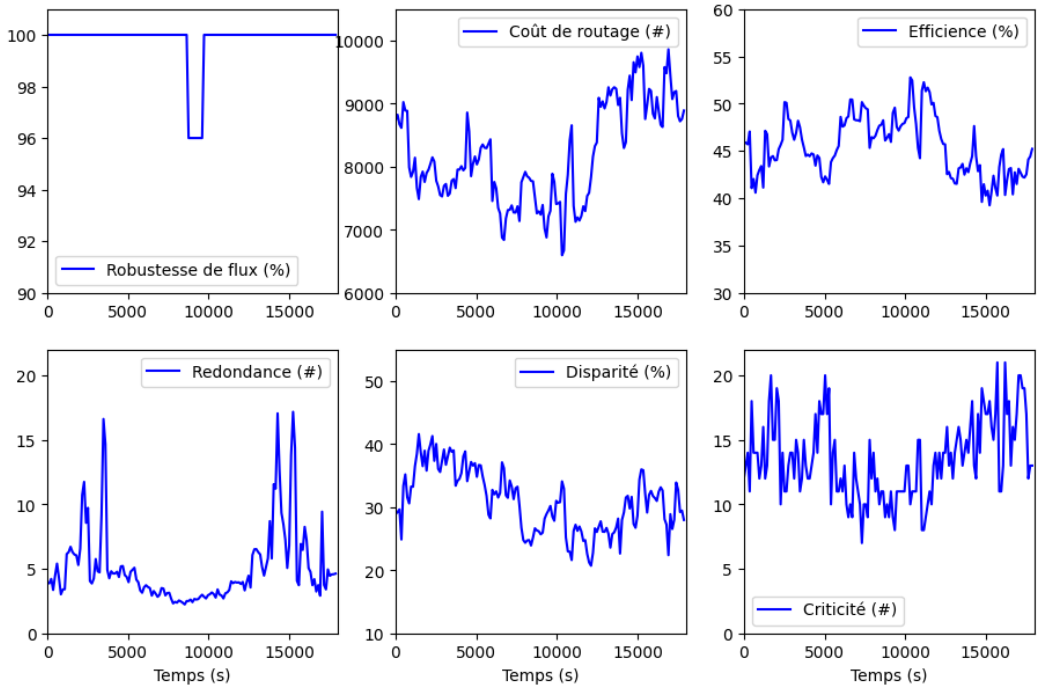


FIGURE 5.2 – Evolution temporelle des métriques de fiabilité du graphe de référence

La première rangée montre les métriques de robustesse, à savoir la robustesse de flux (exprimée en pourcentage de paires reliées), le coût de routage (en nombre de

transmissions de paquets), et l'efficience réseau (en pourcentage). On constate que la robustesse de flux est égale à 100% la majorité du temps, ce qui signifie que toutes les paires source-destinations du graphes sont reliées par au moins un chemin. La robustesse de flux descend un court instant à 96%, ce qui correspond au moment où des sommets se retrouvent isolés du reste du graphe, créant ainsi une seconde composante connexe. Ces sommets sont incapables de communiquer avec les autres membres du réseau, et nous pouvons donc considérer qu'ils sont en panne. Au sujet du coût de routage et de l'efficience réseau, il est difficile de corréler leur évolution à une quelconque propriété du réseau : nous dirons simplement qu'ils fluctuent au cours du temps. En revanche, les échelles sont plus parlantes : le coût de routage varie en moyenne autour de 8000 transmissions au total, et l'efficience réseau est d'environ 45%, impliquant que les routes entre les paires de sommets sont plutôt courtes (une efficience de 50% correspond à des routes de longueur 2, ce qui est très peu). Nous pouvons déduire de ces résultats que le réseau dispose d'un niveau de robustesse correct, et qu'il est donc en mesure de résister aux pannes. Il est cependant très consommateur en énergie en raison du grand nombre de transmissions nécessaires à l'échange des données.

La seconde rangée de la figure présente les métriques de résilience, à savoir la redondance des chemins (exprimée en nombre de chemins alternatifs entre chaque paire), leur disparité (en pourcentage de différents sommets traversés), et enfin la criticité des noeuds (en nombre de sommets critiques). Nous pouvons d'abord remarquer que la redondance des chemins semble corrélée à la densité des noeuds du graphe, et plus précisément à sa force moyenne. Ainsi, on observe une forte augmentation du nombre de chemins alternatifs lorsque le graphe est dense, ce qui est assez intuitif. Notons également que la redondance du graphe de référence est élevée, avec 5 chemins alternatifs en moyenne pour chaque paire source-destination (et on ne considère que les plus courts chemins). Cependant, la disparité de ces chemins n'est pas très élevée : elle est de 31% en moyenne. Cela signifie que les chemins alternatifs disponibles passent par 69% de sommets en commun, ce qui n'est pas très résilient, car une panne sur l'un de ces sommets communs risque d'impacter plusieurs routes à la fois. Enfin, la criticité des noeuds est assez élevée : sur 50 sommets, on en observe 14 en moyenne par lesquels transitent plus de $\epsilon = 5\%$ du trafic total. Si le trafic était correctement réparti, chaque sommet devrait faire transiter $1/50 = 2\%$ du trafic total. Ces sommets critiques transmettent chacun bien plus de données que ce qui est attendu, et risque de tomber en panne prématurément. En résumé, et malgré une forte redondance de chemins, le graphe de référence est peu résilient. Si une panne survient, il risque d'avoir des difficultés à assurer une bonne qualité de service.

L'étude des métriques de fiabilité nous démontre ainsi que l'essaim de satellites, non divisé, dispose d'un niveau de fiabilité assez médiocre. Sa robustesse est correcte, ce qui signifie qu'il y a peu de risque qu'une panne survienne, en dehors des pannes franches liées à la consommation énergétique (ce qui est problématique). Si une panne apparaît tout de même, l'essaim sera fortement impacté, car sa résilience est faible : il ne sera pas en mesure d'assurer un fonctionnement correct. A présent, il est temps de découvrir si la division de graphe permet d'améliorer la fiabilité de ce système.

5.2.3 Résultats

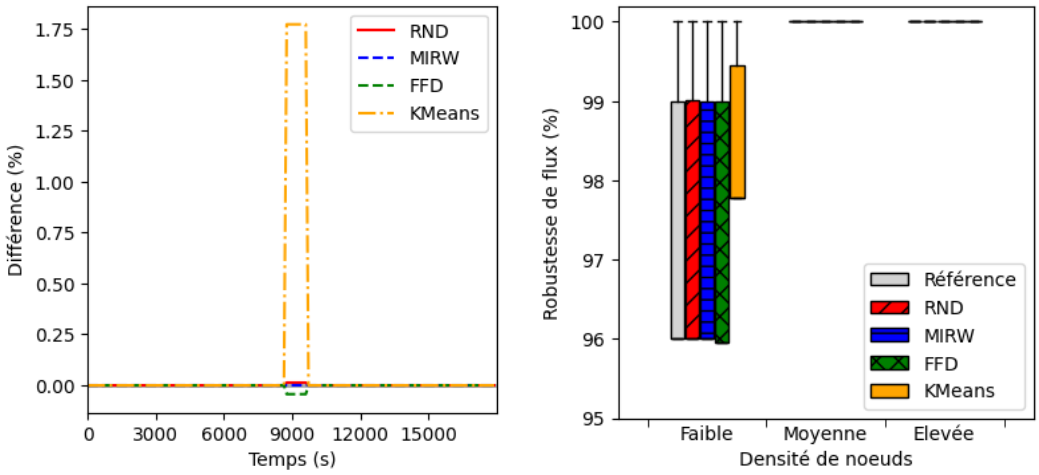
Notre objectif est d'analyser l'impact de la division de graphe sur la robustesse et la résilience du système. Pour cela, nous étudions les performances des trois algorithmes présentés dans le Chapitre 4, à savoir RND, MIRW et FFD. Ces algorithmes sont également comparés à l'algorithme de clustering k-means, qui est basé sur des techniques de machine learning.

Deux types de figures sont présentées dans cette section : des courbes temporelles et des boîtes à moustache. Les courbes temporelles ne représentent pas l'évolution des métriques pour chaque algorithme de division, mais l'*écart* des valeurs obtenues avec chaque algorithme par rapport aux métriques de référence. L'idéal serait d'augmenter les valeurs de robustesse de flux, d'efficience réseau, de redondance et de disparité ; et de diminuer les valeurs de coût de routage et de criticité des noeuds. Les boîtes à moustaches représentent, quant à elles, la distribution des métriques en fonction de trois niveaux de densité de noeuds, tel que décrit dans la Section 5.2.1, afin d'observer l'influence de la densité sur le niveau de fiabilité.

Impact sur la robustesse

L'impact de la division et du clustering sur la robustesse de flux est présenté sur la Figure 5.3. Puisque la robustesse de flux est assez stable sur le graphe de référence, les algorithmes de division n'ont pas beaucoup d'influence sur cette métrique. En revanche, k-means permet de limiter la baisse de robustesse lorsque certains sommets se déconnectent du graphe. Cette légère amélioration, de l'ordre de 2%, indique que les paires source-destinations obtenues avec k-means sont moins impactées par la déconnexion du graphe. Cela signifie que les groupes obtenus avec k-means ont tendance à former des composantes connexes au lieu de s'étendre sur le graphe, ce qui est consistant avec la définition du clustering.

Les résultats sur le coût de routage sont présentés sur la Figure 5.4, et sont déjà bien plus intéressants. Il faut prendre garde à l'échelle de l'évolution temporelle

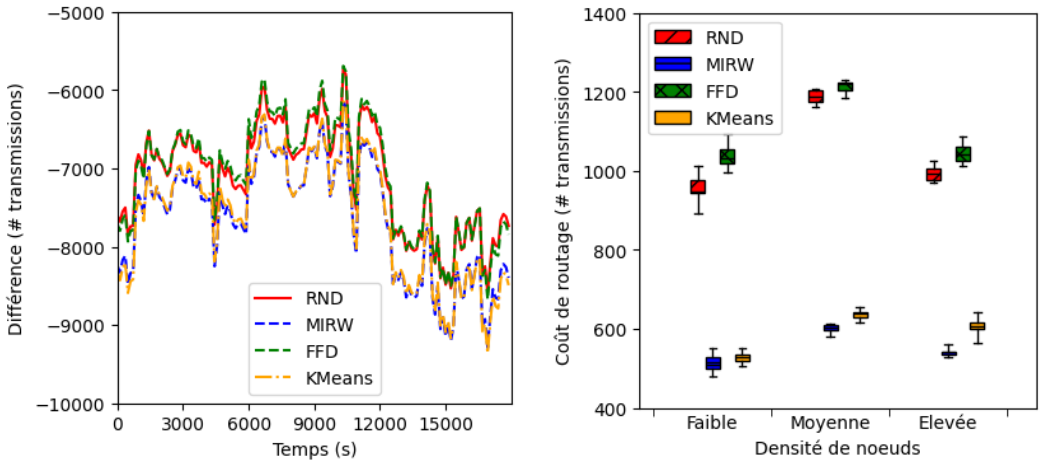


(a) Ecart moyen en fonction du temps (b) Distribution en fonction de la densité

FIGURE 5.3 – Impact sur la robustesse de flux

(Figure 5.4a) pour constater que chaque algorithme fait drastiquement chuter le coût de routage moyen du graphe ! En effet, nous pouvons observer une réduction moyenne de l'ordre de 7500 transmissions, sur la moyenne de 8000 transmissions sur le graphe de référence : en d'autres termes, un graphe divisé nécessite près de 10 fois moins de transmissions de paquets pour relier toutes les paires source-destinations, ce qui correspond à une économie d'énergie colossale. Les algorithmes MIRW et k-means ont de loin les meilleures performances, et sont assez similaires sur leurs résultats, comme l'atteste la distribution de leurs valeurs (Figure 5.4b) : le coût de routage associé varie légèrement entre 500 et 600 transmissions, alors qu'il est compris entre 1000 à 1200 transmissions pour les algorithmes RND et FFD, soit deux fois plus élevé. Notons cependant que la densité n'a pas beaucoup d'impact sur le coût de routage.

Pour finir sur les métriques de robustesse, la Figure 5.5 présente les résultats des algorithmes quant à l'efficience réseau. Nous pouvons constater que k-means est l'algorithme permettant d'augmenter au maximum l'efficience réseau (+20% en moyenne), suivi par MIRW et FFD. RND, quant à lui, est très fidèle au graphe de référence, et n'impacte presque pas son efficience. Il est assez logique que k-means démontre les meilleures performances, puisque le principe du clustering et de regrouper les sommets proches : les routes entre ces sommets sont forcément plus courtes que ce qu'on obtiendrait avec un algorithme de division. Il est cependant intéressant de noter que l'augmentation de l'efficience est la plus forte lorsque le graphe est peu dense, impliquant que l'impact de la division (et du clustering) est le plus bénéfique à ce moment-là.



(a) Ecart moyen en fonction du temps (b) Distribution en fonction de la densité

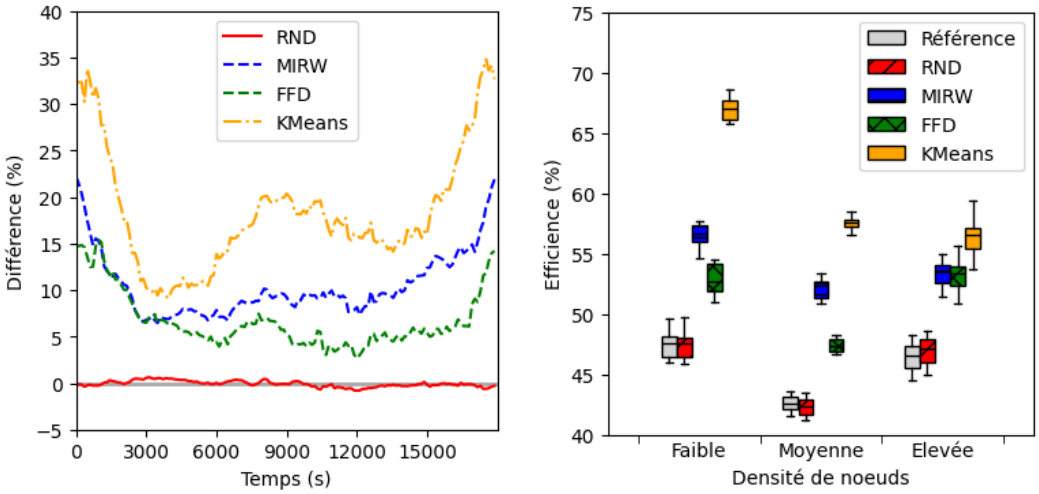
FIGURE 5.4 – Impact sur le coût de routage

Pour résumer, les quatre algorithmes présentés ont globalement un impact positif sur la robustesse. Le coût de routage, notamment, est fortement limité par les algorithmes MIRW et k-means, qui transmettent moins de 10% du trafic de base afin de joindre toutes les sources et destinations. Cette diminution a un impact majeur sur la prévention des pannes dues à la perte d'énergie. En revanche, il faut admettre qu'un algorithme de clustering comme k-means aura de meilleures performances sur l'efficience réseau, donc la longueur des routes, que n'importe quel algorithme de division (bien que MIRW ait, lui aussi, de bonnes performances). Mais contrairement à k-means, nos algorithmes permettent une amélioration du niveau de robustesse tout en ayant une faible complexité computationnelle : ce n'est pas à négliger dans le cadre d'un système distribué sur un essaim de nanosatellites, puisque la puissance de calcul des nanosatellites est limitée. Les algorithmes de division peuvent donc être déployés plus facilement en pratique, et assurer un niveau de robustesse correct pour l'observation de l'Espace.

Impact sur la résilience

La Figure 5.6 présente l'évolution de la redondance des chemins et l'écart obtenu par rapport à la référence, pour chacun des quatre algorithmes. Nous pouvons tout d'abord constater que la redondance augmente avec la densité du graphe (Figure 5.6b), ce qui est assez intuitif, puisque les arêtes sont plus nombreuses dans un graphe dense.

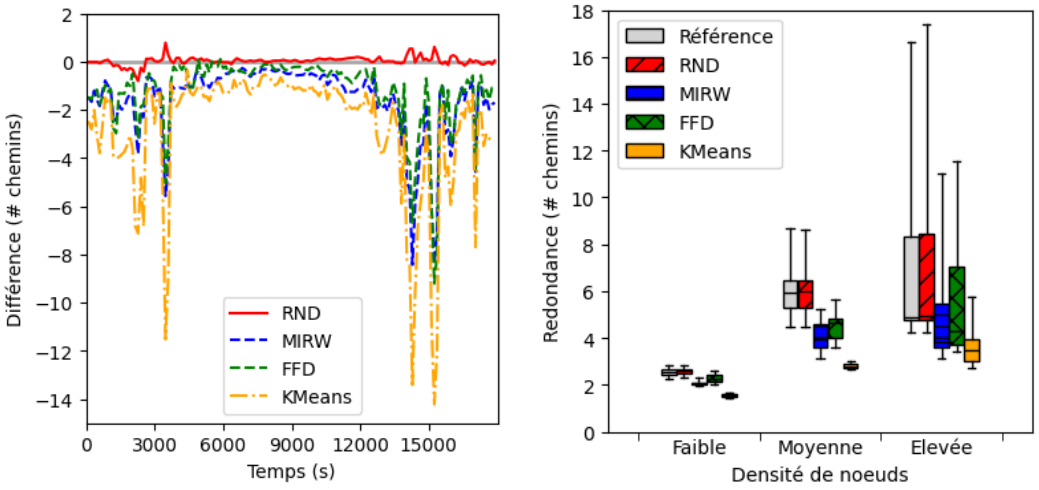
Ensuite, pour améliorer la résilience du système, il faudrait augmenter la redondance des chemins : or seul RND permet de maintenir le niveau de référence, les



(a) Ecart moyen en fonction du temps (b) Distribution en fonction de la densité

FIGURE 5.5 – Impact sur l'efficience réseau

trois autres algorithmes diminuant significativement cette redondance (Figure 5.6a). L'algorithme k-means est le pire des quatre en termes de performance, peu importe la densité du graphe. Cependant, il faut noter qu'avec les algorithmes de division, la redondance n'est jamais inférieure à 2 chemins : il y a donc toujours un chemin alternatif. Il faut également rappeler que notre définition des chemins alternatifs ne concerne que les *plus courts* chemins : si l'on souhaite faire remonter le niveau de redondance, on pourrait étendre cette définition à des chemins plus longs.

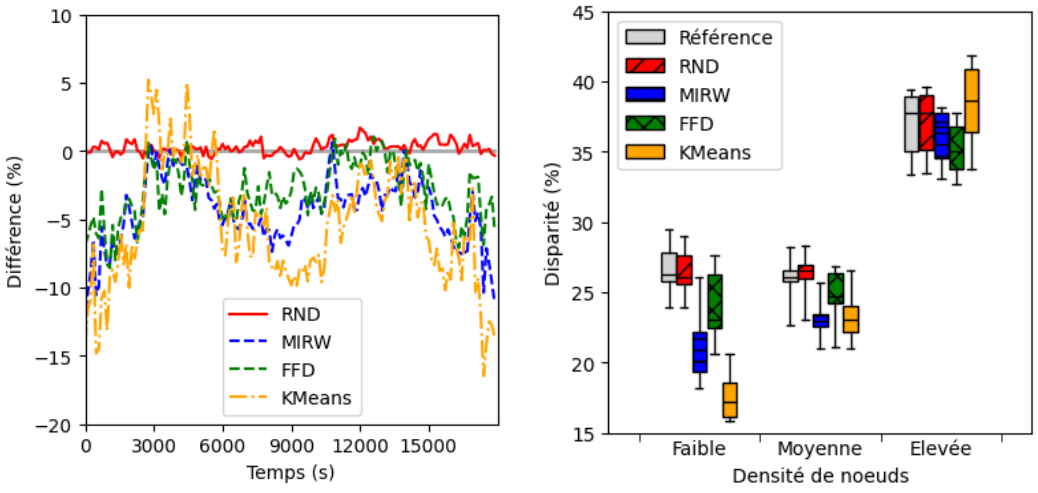


(a) Ecart moyen en fonction du temps (b) Distribution en fonction de la densité

FIGURE 5.6 – Impact sur la redondance des chemins

Intéressons-nous à présent à la diversité des chemins alternatifs disponibles : la Figure 5.7 présente l'impact des algorithmes sur la disparité des chemins. Similairement à la redondance, la disparité a tendance à augmenter avec la densité du graphe (Figure 5.7b). En revanche, il y a peu de corrélation entre le nombre de chemins alternatifs et leur disparité.

Comme pour la redondance, l'objectif est d'augmenter la disparité des chemins afin d'améliorer la résilience du système. Et comme pour la redondance, les algorithmes font globalement baisser le niveau de disparité, hormis RND qui arrive à maintenir le niveau de référence (Figure 5.7a). FFD fait assez peu baisser le niveau disparité (moins de 5%) ; MIRW provoque une diminution plus importante (de 5% à 10%) ; et enfin, k-means provoque la diminution la plus importante, sauf à certains instants, lorsque le graphe est dense. Cette diminution n'est pas sans conséquence, car cela signifie que les chemins alternatifs entre les paires source-destinations ont tendance à passer plus souvent par des sommets communs. Une panne sur un de ces sommets risque donc d'interrompre plusieurs chemins. Ajoutons à cela la réduction du nombre de chemins alternatifs (constatée sur la redondance) : on obtient donc moins de chemins, et ils passent plus souvent par des sommets communs. Par conséquent, les algorithmes FFD, MIRW et k-means ont tendance à avoir un impact négatif sur la résilience du système.



(a) Ecart moyen en fonction du temps

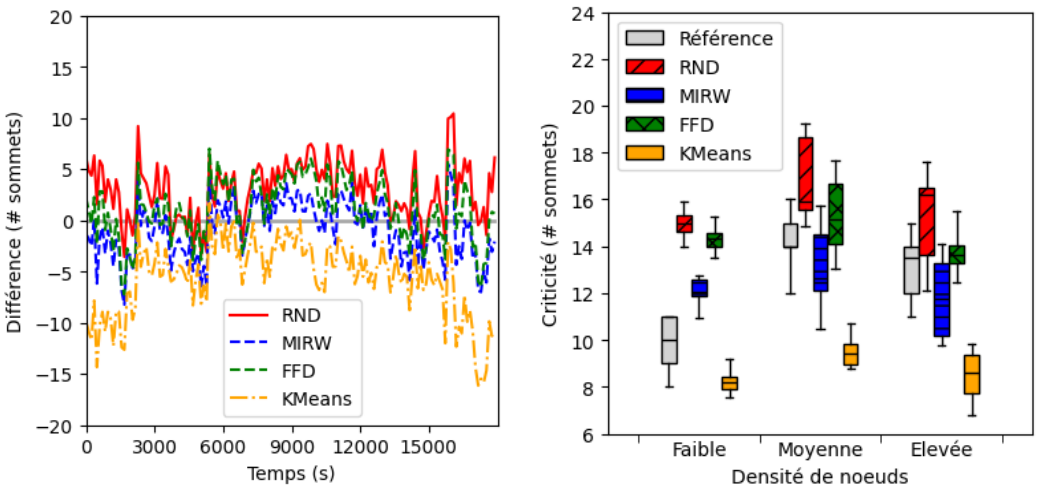
(b) Distribution en fonction de la densité

FIGURE 5.7 – Impact sur la disparité des chemins

Enfin, terminons en analysant la présence des sommets critiques dans le graphe. La criticité des noeuds est présentée sur la Figure 5.8, pour une valeur-seuil de centralité intermédiaire égale à $\epsilon = 5\%$ (rappelons que, pour considérer que le trafic

est bien réparti, la centralité intermédiaire des sommets devrait environ être égale à 2%, comme expliqué dans la Section 5.2.2).

Tout d'abord, nous pouvons constater que la présence et le nombre de sommets critiques ne dépend pas de la densité du graphe (Figure 5.8b). Par ailleurs, les performances des algorithmes de division sont plutôt mauvaises ; RND et FFD font même augmenter le nombre de sommets critiques dans le graphe (Figure 5.8a), en particulier lorsque le graphe est peu dense. MIRW arrive à légèrement diminuer la présence de sommets critiques lorsque le graphe est dense. L'algorithme de clustering k-means, quant à lui, démontre les meilleures performances en diminuant significativement le nombre de sommets critiques, et ce, peu importe la densité du graphe. Nous pouvons donc en conclure que le clustering reste la technique la plus efficace pour réduire la criticité des noeuds du système.



(a) Ecart moyen en fonction du temps

(b) Distribution en fonction de la densité

FIGURE 5.8 – Impact sur la criticité des noeuds

Pour résumer, les quatre algorithmes présentés ici ont chacun un impact mitigé sur la résilience du système. RND a finalement assez peu d'influence sur le niveau de résilience de référence, hormis l'augmentation du nombre de sommets critiques, ce qui est regrettable. FFD fait légèrement baisser la redondance et la disparité des chemins, et augmenter la présence de sommets critiques : c'est sûrement l'algorithme avec le plus mauvais impact sur la résilience. MIRW, quant à lui, fait également baisser les niveaux de redondance et de disparité des chemins, mais permet de réduire la présence des sommets critiques dans des topologies denses. Enfin, k-means fait largement baisser la redondance et la disparité, mais en contrepartie, il permet de réduire le nombre de sommets critiques dans le graphe, en particulier

lorsque ce dernier est dense. Ces deux derniers algorithmes peuvent donc avoir un impact intéressant sur la résilience, si l'on est prêt à faire des compromis.

5.3 Conclusion

La fiabilité d'un système distribué, tel qu'un essaim de nanosatellites, s'opère autour de deux axes : la capacité du système à résister aux pannes ou à les éviter, appelée *robustesse*, et sa capacité à maintenir un niveau de fonctionnement correct en présence de pannes, appelée *résilience*. Dans ce chapitre, nous avons présenté trois métriques de robustesse permettant d'évaluer la dynamique des transmissions de paquets dans le réseau :

— la robustesse de flux évalue la proportion de paires source-destinations joignables via un chemin à chaque instant ;
— le coût de routage quantifie le nombre total de paquets, retransmissions comprises, à transmettre dans le réseau entre toutes les sources et toutes les destinations ;
— l'efficience réseau caractérise la longueur des chemins empruntés, en informant sur le nombre de retransmissions nécessaires.

Nous avons ensuite présenté trois métriques de résilience, qui permettent de caractériser la fiabilité des routes empruntées par les paquets :

— la redondance des chemins quantifie le nombre de chemins alternatifs disponibles entre chaque paire source-destination, au cas où une panne ferait tomber le chemin principal ;
— la disparité des chemins évalue la diversité des sommets traversés par ces chemins alternatifs, et décrit le risque qu'une panne fasse tomber plusieurs routes simultanément ;
— la criticité des noeuds renseigne sur le nombre de sommets critiques dans le graphe, c'est-à-dire ceux qui font transiter une grande partie du trafic global, et qui risqueraient de tomber en panne prématurément.

La fiabilisation du système peut alors s'avérer complexe, car certaines métriques de robustesse sont antagonistes à d'autres métriques de résilience. Un compromis est donc nécessaire pour fiabiliser au mieux le système, comme expliqué dans la Section 2.3 : l'objectif est d'atteindre l'équilibre entre la robustesse et la résilience, car une solution miracle n'existe pas [1].

Dans ce but, nous avons étudié l'impact des algorithmes de division de graphe (RND, MIRW et FFD) et de clustering (k-means) sur les niveaux de robustesse et de résilience du système. Il en ressort que les quatre algorithmes permettent

1. ... encore ?

de drastiquement améliorer la robustesse du réseau, en faisant notamment baisser le coût de routage. Pour la résilience, en revanche, chaque algorithme a un impact spécifique : RND a très peu d'impact là où FFD fait globalement baisser la résilience du système. Les algorithmes MIRW et k-means sont plus nuancés, car ils diminuent les niveaux de redondance et de disparité, mais en contrepartie, ils diminuent la criticité des noeuds.

Quel est donc le meilleur algorithme pour fiabiliser l'essaim de nanosatellites ? Et bien, pour répondre à cette question, il faut prendre en compte la complexité de calcul des algorithmes proposés, et ne pas s'arrêter à la seule étude de leur impact. Ainsi, l'algorithme de clustering k-means est certes très performant, et permet d'améliorer au mieux la robustesse et la résilience du système. Cependant, comme expliqué dans la Section 4.2, il s'agit d'un algorithme de machine learning très complexe en temps de calcul : est-ce malin de le déployer sur un essaim de nanosatellites ayant de fortes contraintes énergétiques, dans l'espoir d'avoir une légère augmentation de la fiabilité du système ?

Finalement, les algorithmes de division RND, FFD et MIRW permettent d'atteindre de très bonnes performances, étant donné leur complexité. Si RND peut sembler peu intéressant de prime abord, il ne faut pas oublier que sa complexité est imbattable, puisqu'il s'exécute instantanément sur la totalité du réseau (il n'y a aucun problème de passage à l'échelle). Pour ce prix, il permet d'améliorer la robustesse du système sans trop impacter sa résilience. FFD est plus mitigé dans son impact global, mais reste peu complexe et peut être utile si l'on préfère favoriser la robustesse plutôt que la résilience. Et pour finir, MIRW est de loin le choix le plus judicieux, car il permet d'atteindre des performances similaires à k-means tout en étant très peu complexe en temps de calcul.

En conclusion : oui, il est possible de fiabiliser un essaim de nanosatellites fonctionnant comme un radiotélescope spatial distribué. Pour cela, il suffit de répartir les noeuds du réseau dans des groupes, grâce aux techniques de division de graphe présentées précédemment. Un algorithme de machine learning permettra d'obtenir la meilleure fiabilisation du système, mais risque d'être trop coûteux par rapport au bénéfice apporté. De simples algorithmes de sélection aléatoire (RND) ou d'exploration ordonnancée (MIRW) permettent de suffisamment fiabiliser le système, tout en étant très peu coûteux en temps de calcul. Ces algorithmes offrent donc la meilleure solution de fiabilisation pour un DSS ayant de fortes contraintes énergétiques.

Chapitre 6

Conclusion

Aussi loin que l'on puisse remonter dans l'Histoire de nos civilisations, l'Humanité a toujours eu un œil rivé sur le ciel : de la Préhistoire à la Renaissance, les hommes ont tenté de comprendre le monde dans lequel on évolue. Puis, en 1610, vint Galilée et sa première lunette d'observation : nous pouvions enfin observer les corps célestes qui nous entourent. A présent, nous disposons de technologies avancées qui nous permettent de cartographier le ciel, et d'observer les étoiles comme s'il s'agissait de notre Soleil. Aujourd'hui, la communauté scientifique redouble d'efforts pour trouver des solutions toujours plus innovantes pour repousser les limites de l'observation, bien au-delà de ce qu'il a été possible d'observer jusqu'à ce jour. C'est dans cette course effrénée à la découverte de l'Espace que s'inscrivent mes travaux de thèse, dans l'espoir de voir naître un jour des Systèmes Spatiaux Distribués (DSS) fiables et optimisés pour l'observation des signaux de basses fréquences.

Intérêt des essaims pour les missions spatiales

Afin d'observer l'Espace, les instruments sont spécifiquement fabriqués pour balayer certaines gammes de fréquences, comme la lumière visible ou les infrarouges, par exemple. La gamme des très basses fréquences (en-dessous de 100 MHz) suscite particulièrement l'intérêt des scientifiques, car ces signaux nous apportent des informations quant à la naissance et aux premiers jours de notre Univers : il s'agit de signaux de l'Univers Primordial. Malheureusement, ces signaux sont difficilement observables depuis la surface de la Terre, car ils sont particulièrement sensibles aux interférences radioélectriques (RFI) ainsi qu'à l'ionosphère. En d'autres termes, leur observation s'effectue en terrain très hostile.

La communauté scientifique propose donc de changer de terrain d'observation, et de déployer des radiotélescopes directement dans l'Espace. Cependant, les instruments et infrastructures conçues pour l'observation au sol ne peuvent pas être envoyés tels quels en orbite, ce serait bien trop facile. Ainsi, la solution viable est de déployer un DSS d'un nouveau genre, qui permette d'imiter le fonctionnement d'un interféromètre basse fréquence et de l'adapter à l'environnement spatial : un essaim de nanosatellites en orbite lunaire. Dans cette configuration, chaque nanosatellite

commence par observer les signaux basse fréquence, puis partage les signaux reçus avec les autres membres de l'essaim. Ces signaux sont ensuite combinés localement pour former une image interférométrique globale des signaux captés par l'essaim. Pour finir, l'image est envoyée à une station de base terrestre.

Cette thèse s'est concentrée sur la question de l'échange de données qui a lieu entre l'observation et le calcul de l'image interférométrique. En effet, des problèmes de congestion et de pertes de paquets peuvent survenir, dans le cas où les nanosatellites émettent leurs données simultanément. Par ailleurs, l'énergie est très limitée sur un nanosatellite : sans mécanisme de gestion de la consommation énergétique, des pannes franches peuvent apparaître et impacter l'efficacité de la mission scientifique. Pour éviter ces désagréments, nous avons établi les propriétés clé du réseau de l'essaim, et proposé un mécanisme permettant de fiabiliser le système en limitant la consommation énergétique et la charge du réseau.

Fiabilité de l'architecture réseau

Les essaims de nanosatellites sont peu étudiés dans la littérature existante, et en particulier du point de vue réseau. Les scientifiques s'accordent néanmoins sur le fait qu'un essaim corresponde à un réseau ad-hoc mobile (MANET), déployé dans l'Espace. Nous avons donc modélisé notre essaim de nanosatellites par la théorie des graphes, ce qui nous a permis d'en extraire des propriétés générales. Tout d'abord, un essaim de nanosatellites est un DSS qui suit un double modèle de mobilité, à savoir sa mobilité en orbite lunaire, et la mobilité relative de ses nanosatellites. Ensuite, la densité des noeuds et la disponibilité des liens inter-satellites (ISL) sont fortement hétérogènes : la topologie présente à la fois des zones de faible et forte densités, et la majorité des ISL sont peu stables dans le temps. Autrement dit, la charge du réseau est répartie de façon hétérogène, alors que la puissance de calcul et l'énergie sont réparties de façon homogène : cela peut engendrer des pannes d'énergie et des pertes de paquets. Nous avons ainsi conclu qu'un mécanisme de re-distribution de la charge réseau était nécessaire pour assurer le fonctionnement et la fiabilité du système.

De nombreux mécanismes de ré-équilibrage sont applicables aux MANET, et par extension, aux essaims de nanosatellites. Ces mécanismes reposent en général sur des techniques de clustering, à savoir le regroupement des noeuds en communautés. Cependant, ces mécanismes ont fait leurs preuves pour réduire la charge du réseau, sans pour autant la répartir équitablement entre les noeuds. Nous avons donc proposé un nouveau mécanisme, inspiré à la fois des techniques de clustering et d'échantillonnage, qui soit en mesure de réduire *et* de ré-équilibrer la charge

du réseau : il s'agit de la division de graphe. Cette technique consiste à regrouper les noeuds dans des groupes, et de faire en sorte que chaque groupe soit représentatif du graphe d'origine. Nous avons présenté trois algorithmes de division de graphe : l'algorithme de sélection aléatoire Random Node Division (RND), l'algorithme d'exploration ordonnancé Multiple Independent Random Walks (MIRW), et l'algorithme d'exploration épidémique Forest Fire Division (FFD). En comparant les performances de ces algorithmes avec un célèbre algorithme de clustering (k-means), nous en avons conclu que MIRW est l'algorithme le plus efficace énergétiquement, et assure une répartition équitable des ressources tout en ayant une faible complexité computationnelle.

Pour finir, nous avons mené une étude exhaustive de l'impact de la division de graphe sur la fiabilité du système en termes de robustesse et de résilience. La robustesse est la capacité du système à résister aux pannes, ou à les éviter ; la résilience est sa capacité à maintenir un niveau de fonctionnement correct ou dégradé en présence de pannes. Nos simulations ont montré que les algorithmes de division (RND, MIRW et FFD) et de clustering (k-means) permettent de drastiquement améliorer la robustesse du système, notamment en réduisant le nombre de transmissions de paquets, et donc la consommation énergétique. En termes de résilience, les algorithmes MIRW et k-means permettent de réduire la quantité de noeuds critiques, c'est-à-dire de noeuds centraux dans le réseau, par lesquels transite une part importante du trafic global. Cependant, les quatre algorithmes ont tendance à réduire le nombre de chemins disponibles dans le réseau. Dans tous les cas, il est nécessaire de trouver un compromis entre l'efficacité énergétique et la redondance du système : dans un système très redondant de base, tel que l'essaim, la baisse de la redondance des chemins n'a que peu d'impact sur la résilience du système.

Globalement, les algorithmes de division de graphe tels que MIRW permettent d'améliorer la fiabilité du système, notamment du point de vue de la consommation énergétique et de la répartition de la charge du réseau. Cette amélioration signifie que les essaims peuvent être utilisés de manière fiable pour des missions critiques, telles que l'interférométrie basse fréquence, en prolongeant la durée de vie du DSS et en préservant la qualité des images interférométriques générées. Cette fiabilité accrue permet aussi d'étendre l'utilisation des essaims de nanosatellites à d'autres applications spatiales, comme par exemple l'observation de la Terre ou d'autres corps célestes, qui ont des exigences similaires en termes d'efficacité et de fiabilité.

Perspectives et futurs travaux

Bien que l'algorithme MIRW améliore la robustesse, et réduit donc le risque de panne, il pourrait être optimisé pour augmenter davantage la résilience du système. En effet, nos simulations ont montré que son impact sur la résilience est assez variable en fonction de la topologie de l'essaim. Des recherches futures pourraient donc explorer de nouvelles versions de l'algorithme, en le combinant avec des techniques de machine learning par exemple, à l'image de k-means. Il ne faut cependant pas le complexifier à outrance pour un faible gain de performance.

De plus, notre analyse de l'impact de la division et du clustering sur la fiabilité du système est bien plus complète que de nombreuses études existantes, car elle prend en compte les effets conjoints sur les métriques de robustesse et de résilience, plutôt que de les analyser séparément. En revanche, cette étude pourrait être approfondie en analysant l'impact de la division sur le long terme. A ce jour, nous ne disposons que de données générées synthétiquement, puisqu'il n'existe pas d'essaim de nanosatellites en orbite (lunaire ou même terrestre). Il y a donc un intérêt majeur à mener cette étude sur des données expérimentales obtenues sur plusieurs semaines, voire plusieurs mois de déploiement en orbite. L'analyse sur un jeu de données long permettrait sûrement de mettre en lumière de potentielles détériorations, qui jouent un rôle central dans l'étude du comportement du système face aux pannes réelles.

Nous avons gardé la principale perspective pour la fin, et c'est de loin la plus importante. Après avoir financé de nombreuses thèses et travaux sur le sujet, il est maintenant primordial de mettre en commun la recherche autour du projet NOIRE, et de combiner les résultats sur la communication intra-essaim, la synchronisation des horloges, la localisation des nanosatellites, ou encore la formation des antennes. Cette mise en commun mettra sans doute en lumière des points blocants, ou des problèmes non escomptés. Mais elle sera également l'étape clé qui nous permettra, un jour, d'envoyer des essaims de nanosatellites en orbite, et de réaliser l'un des plus vieux rêves de l'Humanité : découvrir ce qui se cache au fin fond de l'Univers.

Bibliographie

[1] *Ad hoc mobile wireless networks : protocols and systems.* Pearson education, 2001. (Cité en pages 26 et 34.)

[2] *GPS et Galileo : Systèmes de navigation par satellites.* Editions Eyrolles, 2006. (Cité en pages v et 16.)

[3] *Global Positioning System : signals, measurements and performance (second revised edition).* Ganga-Jamusa Press, 2011. (Cité en page 15.)

[4] *Distributed space missions for earth system monitoring.* Springer Science and Business Media, 2012. (Cité en page 12.)

[5] *Springer Handbook of Global Navigation Satellite Systems*, chapter Galileo. Springer, Cham, 2017. (Cité en page 15.)

[6] *Fault-tolerant systems.* Morgan Kaufmann, 2020. (Cité en page 40.)

[7] *Guide to disaster-resilient communication networks.* Springer Nature, 2020. (Cité en page 44.)

[8] *Mobile Computing*, chapter Wireless communications challenges to flying ad hoc networks (FANET). InTechOpen, 2020. (Cité en page 29.)

[9] *Green Computing in Network Security*, chapter Systematic study of VANET : Applications, challenges, threats, attacks, schemes and issues in research. CRC Press, 2022. (Cité en page 27.)

[10] *Software Defined Networking for Ad Hoc Networks*, chapter Fuzzy approach-based stable energy-efficient AODV routing protocol in mobile ad hoc networks. Springer International Publishing, 2022. (Cité en page 34.)

[11] E. Akopyan, R. Dhaou, E. Lochin, B. Pontet, and J. Sombrin. Fair network division of nanosatellite swarms. *IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023. (Cité en pages 8 et 65.)

[12] E. Akopyan, R. Dhaou, E. Lochin, B. Pontet, and J. Sombrin. On the network characterization of nanosatellite swarms. *The 28th IEEE Symposium on Computers and Communications (ISCC)*, 2023. (Cité en pages 8 et 49.)

[13] E. Akopyan, R. Dhaou, E. Lochin, B. Pontet, and J. Sombrin. Division réseau équitable dans les essaims de nanosatellites. *Algotel-CoRes*, 2024. (Cité en pages 8 et 65.)

[14] E. Akopyan, R. Dhaou, E. Lochin, B. Pontet, and J. Sombrin. Resilient and robust strategies for swarming ad-hoc networks. Soumis à Computer Networks, 2024. (Cité en pages 8 et 79.)

[15] R. Al-Zaidi. *An IoT enabled system for marine data acquisition and cartography*. PhD thesis, University of Essex, 2018. (Cité en page 30.)

[16] M. A. Alandihallaj and M. R. Emami. Multiple-payload fractionated spacecraft for earth observation. *Acta Astronautica*, 2022. (Cité en page 20.)

[17] R. Alyassi, M. Khonji, A. Karapetyan, S. C. K. Chau, K. Elbassioni, and C. M. Tseng. Autonomous recharging and flight mission planning for battery-operated autonomous drones. *IEEE Transactions on Automation Science and Engineering*, 2023. (Cité en page 42.)

[18] M. Angurala, M. Bala, and V. Khullar. A survey on various congestion control techniques in wireless sensor networks. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2022. (Cité en page 42.)

[19] S. S. Anjum, R. M. Noor, and M. H. Anisi. Review on manet based communication for search and rescue operations. *Wireless personal communications*, 2017. (Cité en page 27.)

[20] S. Bandyopadhyay, G. P. Subramanian, R. Foust, D. Morgan, S. J. Chung, and F. Hadaegh. A review of impending small satellite formation flying missions. *53rd AIAA Aerospace Sciences Meeting*, 2015. (Cité en page 17.)

[21] P. Beigi, M. S. Rajabi, and S. Aghakhani. An overview of drone energy consumption factors and models. *Handbook of smart energy systems*, 2022. (Cité en page 42.)

[22] N. Beijar. Zone routing protocol (zrp). *Networking Laboratory, Helsinki University of Technology, Finland*, 2002. (Cité en page 35.)

[23] J.M. Bernabeu, L. Ortega, A. Blais, Y. Grégoire, and E. Chaumette. On the asymptotic performance of time-delay and doppler estimation with a carrier modulated by a band-limited signal. *EURASIP Journal on Advances in Signal Processing*, 47, 2024. (Cité en page 24.)

[24] R. Biggs, M. Schluter, and M. L. Schoon. *Principles for building resilience : sustaining ecosystem services in social-ecological systems*. Cambridge University Press, 2015. (Cité en page 46.)

[25] L. Brown and J. C. Novaco. Non thermal galactic emission below 10 mhz. *The Astrophysical Journal*, 1978. (Cité en page 1.)

[26] O. Brown, P. Eremenko, and P. Collopy. Value-centric design methodologies for fractionated spacecraft : Progress summary from phase i of the darpa system f6 program. *AIAA Space 2009 Conference and Exposition*, 2009. (Cité en page 19.)

[27] M. E. Bruni, S. Khodaparasti, and G. Perboli. Energy efficient uav-based last-mile delivery : A tactical-operational model with shared depots and non-linear energy consumption. *IEEE Access*, 2023. (Cité en page 42.)

[28] A. Budianu, A. Meijerink, and M. J. Bentum. Swarm-to-earth communication in olfar. *Acta astronautica*, 107 :14–19, 2015. (Cité en pages v et 22.)

[29] A. Budianu, R. T. Rajan, S. Engelen, A. Meijerink, C. J. M. Verhoeven, and M. J. Bentum. Olfar : Adaptive topology for satellite swarms. *International Astronautical Congress*, pages 3–7, 2011. (Cité en pages 22, 43 et 66.)

[30] B. Cecconi, M. Dekkali, and C. Briand. Noire study report : towards a low frequency radio interferometer in space. *2018 IEEE Aerospace Conference*, pages 1–19, 2018. (Cité en pages 3, 23 et 26.)

[31] E. Chauhan, M. Sirswal, D. Gupta, and A. Khanna. A critical review : Sanet and other variants of ad hoc networks. *International Conference on Innovative Computing and Communications : Proceedings of ICICC 2020*, 2021. (Cité en page 30.)

[32] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun. Fanet : Communication, mobility models and security issues. *Computer Networks*, 2019. (Cité en page 28.)

[33] J. Chu, J. Guo, and E. Gill. Functional, physical, and organizational architectures of a fractionated space infrastructure for long term earth observation missions. *IEEE Aerospace and Electronic Systems Magazine*, 2014. (Cité en page 19.)

[34] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). Technical report, RFC 3626, 2003. (Cité en page 35.)

[35] G. Curzi, D. Modenini, and P. Tortora. Large constellations of small satellites : a survey of near future challenges and missions. *Aerospace*, 2020. (Cité en page 13.)

[36] I. Daanoune, B. Abdennaceur, and A. Ballouk. A comprehensive survey on leach-based clustering routing protocols in wireless sensor networks. *Ad Hoc Networks*, 114, 2021. (Cité en pages 43 et 66.)

[37] G. Di Caro, F. Ducatelle, and L. M. Gambardella. Anthocnet : an ant-based hybrid routing algorithm for mobile ad hoc networks. *International Conference on Parallel Problem Solving from Nature*, 2004. (Cité en page 35.)

[38] S. Engelen, E. K. A. Gill, and C. J. M. Verhoeven. On the reliability of spacecraft swarms. *4S Symposium : Small Satellites, Services and Systems*, 2012. (Cité en pages 22 et 39.)

[39] S. Engelen, C. J. Verhoeven, and M. J. Bentum. Olfar, a radio telescope based on nano-satellites in moon orbit. *24th annual AIAA/USU conference on small satellites*, 2010. (Cité en page 21.)

[40] C. E. Fossa, R. A. Raines, G. H. Gunsch, and M. A. Temple. An overview of the iridium (r) low earth orbit (leo) satellite system. *Proceedings of the IEEE 1998 National Aerospace and Electronics Conference (NAECON 1998)*, 1998. (Cité en page 14.)

[41] N. Ganeshkumar and S. Kumar. Obu (on-board unit) wireless devices in vanet(s) for effective communication — a review. *Computational Methods and Data Engineering : Proceedings of ICMDE 2020*, 2020. (Cité en page 27.)

[42] J. Gangestad, B. Hardy, and D. Hinkley. Operations, orbit determination, and formation control of the aerocube-4 cubesats. *27th Annual AIAA/USU Conference on Small Satellites*, 2013. (Cité en page 18.)

[43] M. A. Gleyzes, L. Perret, and P. Kubik. Pleiades system architecture and main performances. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012. (Cité en page 16.)

[44] Y. Gu, X. Fu, Z. Liu, X. Xu, and A. Chen. Performance of transportation network under perturbations : Reliability, vulnerability, and resilience. *Transportation Research Part E : Logistics and Transportation Review*, 2020. (Cité en page 45.)

[45] G. He. Destination-sequenced distance vector (dsdv) protocol. *Networking Laboratory, Helsinki University of Technology*, 2002. (Cité en page 35.)

[46] P. Hu and W. C. Lau. A survey and taxonomy of graph sampling. *arXiv preprint arXiv :1308.5865*, 2013. (Cité en pages 68, 69 et 70.)

[47] D. Hutchison and J. P. Sterbenz. Architecture and design for resilient networked systems. *Computer Communications*, 2018. (Cité en page 44.)

[48] W. Jing, X. Xu, and Y. Pu. Route redundancy-based network topology measure of metro networks. *Journal of Advanced Transportation*, 2019. (Cité en pages 45 et 83.)

[49] M. Joa-Ng and I. T. Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE Journal on selected areas in communications*, 1999. (Cité en page 35.)

[50] D. B. Johnson, D. A. Maltz, and J. Broch. Dsr : The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 2001. (Cité en page 34.)

[51] G. Kaur and P. Thakur. Routing protocols in manet : An overview. *IEEE 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2019. (Cité en page 34.)

[52] H. S. Z. Kazmi, N. Javaid, M. Awais, M. Tahir, S. O. Shim, and Y. B. Zikria. Congestion avoidance and fault detection in wsns using data science techniques. *Transactions on Emerging Telecommunications Technologies*, 2022. (Cité en page 42.)

[53] G. Kbidy, G. Adamski, and N. May. Design concepts and challenges for the iridium next command and control system. *2018 SpaceOps Conference*, 2018. (Cité en page 13.)

[54] J. Kim, D. Lee, J. Hwang, S. Hong, D. Shin, and D. Shin. Wireless sensor network (wsn) configuration method to increase node energy efficiency through clustering and location information. *Symmetry*, 2021. (Cité en pages 43 et 66.)

[55] S. Koutra, M. B. Mondejar, and V. Becue. The nexus of urban resilience and energy efficiency in cities. *Current Research in Environmental Sustainability*, 2022. (Cité en page 46.)

[56] H. J. Kramer and A. P. Cracknell. An overview of small satellites in remote sensing. *International journal of remote sensing*, 2008. (Cité en page 12.)

[57] G. Krieger, M. Zink, and A. Moreira. Tandem-x : a radar interferometer with two formation flying satellites. *Proceedings International Astronautical Congress*, 2012. (Cité en page 17.)

[58] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 2003. (Cité en page 71.)

[59] W. Liu, X. Li, T. Liu, and B. Liu. Approximating betweenness centrality to identify key nodes in a weighted urban complex transportation network. *Journal of Advanced Transportation*, 2019. (Cité en page 45.)

[60] M. J. N. Mahi, S. Chaki, S. Ahmed, M. Biswas, M. S. Kaiser, M. S. Islam, and M. Whaiduzzaman. A review on vanet research : Perspective of recent emerging technologies. *IEEE Access*, 2022. (Cité en page 27.)

[61] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, and J. M. Alcaraz-Calero. Search and rescue operation using uavs : A case study. *Expert Systems with Applications*, 2021. (Cité en page 28.)

[62] C. Mathieu and A. Weigel. Assessing the fractionated spacecraft concept. *Space 2006*, 2006. (Cité en page 19.)

[63] H. McPhee, J.Y. Tourneret, D. Valat, P. Paimblanc, J. Delporte, and Y. Grégoire. A robust time scale based on maximum likelihood estimation. *54th*

Annual Precise Time and Time Interval Systems and Applications Meeting, 2023. (Cité en page 23.)

[64] L. Merino, F. Caballero, J. R. Martínez-de Dios, I. Maza, and A. Ollero. An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent and Robotic Systems*, 2012. (Cité en page 28.)

[65] D. K. Mishra, M. J. Ghadi, A. Azizivahed, L. Li, and J. Zhang. A review on resilience studies in active distribution systems. *Renewable and Sustainable Energy Reviews*, 2021. (Cité en page 44.)

[66] D. Moad, S. Djahel, and F. Nait-Abdesselam. Improving the quality of service routing in olsr protocol. *IEEE International Conference on Communications and Information Technology (ICCIT)*, 2012. (Cité en page 35.)

[67] H. E. D. Mohamed, H. Azmi, and S. Taha. Using manet in iot healthcare applications : a survey. *International Journal of Computing and Digital Systems*, 2021. (Cité en page 27.)

[68] S. Mohseni, R. Hassan, A. Patel, and R. Razali. Comparative review study of reactive and proactive routing protocols in manets. *4th IEEE International Conference on Digital ecosystems and technologies*, 2010. (Cité en page 34.)

[69] D. M. Molla, H. Badis, A. A. Desta, L. George, and M. Berbineau. Sdr-based reliable and resilient wireless network for disaster rescue operations. *International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–7, 2019. (Cité en page 45.)

[70] J. P. Onnela, J. Saramäki, J. Kertész, and K. Kaski. Intensity and coherence of motifs in weighted complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 2005. (Cité en page 59.)

[71] P. Paimblanc and C. Mailhes. Définition d'un référentiel spatio-temporel autonome dans une constellation de satellites. Technical report, CNES - TéSA, 2018. (Cité en page 53.)

[72] H. Pallubinsky, R. P. Kramer, and W. D. van Marken Lichtenbelt. Establishing resilience in times of climate change — a perspective on humans and buildings. *Climatic Change*, 2023. (Cité en page 46.)

[73] J. Parham, M. Kromis, P. Teng, A. Zosuls, B. Walsh, and J. Semeter. Andesite : A student built swarm from concept to launch and beyond. *32nd Annual AIAA/USU Conference on Small Satellites*, 2018. (Cité en page 20.)

[74] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *IEEE Proceedings of INFOCOM'97*, 1997. (Cité en page 35.)

[75] D. N. Patel, S. B. Patel, H. R. Kothadiya, P. D. Jethwa, and R. H. Jhaveri. A survey of reactive routing protocols in manet. *IEEE International Conference on Information Communication and Embedded Systems (ICICES2014)*, 2014. (Cité en page 34.)

[76] G. Pei, M. Gerla, and T. W. Chen. Fisheye state routing : A routing scheme for ad hoc wireless networks. *2000 IEEE International Conference on Communications (ICC 2000) Global Convergence Through Communications*, 2000. (Cité en page 35.)

[77] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, RFC 3561, 2003. (Cité en page 34.)

[78] V. Perrier, E. Lochin, J. Y. Tourneret, and P. Gélard. Attention networks for time series regression and application to congestion control. *IEEE IFIP Networking Conference*, 2022. (Cité en page 43.)

[79] S. Persson, P. Bodin, E. Gill, J. Harr, and J. Jörgensen. Prisma : an autonomous formation flying mission. *ESA Small Satellite Systems and Services Symposium (4S)*, 2006. (Cité en page 17.)

[80] M. A. Rajan, M. G. Chandra, L. C. Reddy, and P. Hiremath. Concepts of graph theory relevant to ad-hoc networks. *International Journal of Computers, Communications and Control*, 3 :465–469, 2008. (Cité en page 50.)

[81] G. Santhanamari, J. Premi, S. Rahavi, and S. Vijay. Health monitoring of soldiers using efficient manet protocol. *IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2020. (Cité en page 27.)

[82] J. Saramäki, M. Kivelä, J. P. Onnela, K. Kaski, and J. Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 2007. (Cité en page 59.)

[83] J. Sarkis and Q. Zhu. Environmental sustainability and production : taking the road less travelled. *International Journal of Production research*, pages 743–759, 2018. (Cité en page 46.)

[84] K. Schilling. Perspectives for miniaturized, distributed, networked cooperating systems for space exploration. *Robotics and Autonomous Systems*, 2017. (Cité en page 12.)

[85] M. Z. Serdar, M. Koç, and S. G. Al-Ghamdi. Urban transportation networks resilience : indicators, disturbances, and assessment methods. *Sustainable Cities and Society*, 2022. (Cité en page 45.)

[86] F. J. Simons, G. Nolet, P. Georgief, J. M. Babcock, L. A. Regier, and R. E. Davis. On the potential of recording earthquakes for global seismic tomography by low-cost autonomous instruments in the oceans. *Journal of Geophysical Research : Solid Earth*, 2009. (Cité en page 30.)

[87] S. K. Singh and J. Prakash. Energy efficiency and load balancing in manet : a survey. *IEEE 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020. (Cité en page 43.)

[88] D. N. Sivapriya and D. R. Mohandas. Multi constraint multicasting analysis with fault tolerance routing mechanism. *Telematique*, 2022. (Cité en page 45.)

[89] H. Song, J. Lee, and Y. Yi. Feasibility study of communication access via iridium constellation for small-scale magnetospheric ionospheric plasma experiment mission. *Journal of Astronomy and Space Sciences*, 2022. (Cité en pages v et 15.)

[90] W. Sun, P. Bocchini, and B. D. Davidson. Resilience metrics and measurement methods for transportation infrastructure : the state of the art. *Sustainable and Resilient Infrastructure*, pages 168–199, 2020. (Cité en page 45.)

[91] C. K. Toh. Associativity-based routing for ad hoc mobile networks. *Wireless personal communications*, 1997. (Cité en page 35.)

[92] R. E. Ulanowicz. Quantifying sustainable balance in ecosystems configurations. *Current Research in Environmental Sustainability*, pages 1–6, 2020. (Cité en page 47.)

[93] Z. Ullah. A survey on hybrid, energy efficient and distributed (heed) based energy efficient clustering protocols for wireless sensor networks. *Wireless personal communications*, 2020. (Cité en page 43.)

[94] K. Upreti, N. Kumar, M. S. Alam, A. Verma, M. Nandan, and A. K. Gupta. Machine learning-based congestion control routing strategy for healthcare iot enabled wireless sensor networks. *IEEE Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021. (Cité en page 42.)

[95] R. Usha, B. S. Premananda, and K. V. Reddy. Performance analysis of manet routing protocols for military applications. *IEEE International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017. (Cité en page 27.)

[96] S. Wang, W. Lv, J. Zhang, S. Luan, C. Chen, and X. Gu. Method of power network critical nodes identification and robustness enhancement based on a cooperative framework. *Reliability Engineering and System Safety*, 2021. (Cité en page 45.)

[97] T. J. Willink Castro. Antenna system design for olfar's inter-satellite link. Master's thesis, University of Twente, 2012. (Cité en page 21.)

[98] H. Xiang and L. Tian. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (uav). *Biosystems engineering*, 2011. (Cité en page 28.)

[99] S. Yogarayan. Wireless ad hoc network of manet, vanet, fanet and sanet : a review. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 2021. (Cité en page 31.)

[100] W. Zhong, C. Yang, W. Liang, J. Cai, L. Chen, J. Liao, and N. Xiong. Byzantine fault-tolerant consensus algorithms : A survey. *Electronics*, 2023. (Cité en page 41.)

Résumé :

Le domaine de l'observation de l'Espace s'intéresse de près aux signaux très basse fréquence, car ils fournissent d'importantes informations quant à la naissance et aux premiers jours de l'Univers. A ce jour, les interféromètres observant ces signaux se situent à la surface de la Terre, dans des zones arides. Malheureusement, ces signaux sont très sensibles aux interférences terrestres ainsi qu'à l'ionosphère, et sont donc difficilement observables. Une solution à ce problème serait d'observer les signaux directement depuis l'Espace, en déployant un essaim de nanosatellites en orbite autour de la Lune. Cet essaim constitue un Système Spatial Distribué (DSS), fonctionnant en tant qu'interféromètre, dont l'orbite lunaire le préserve des interférences terrestres et de l'ionosphère. Cependant, la configuration de l'essaim de nanosatellites en interféromètre spatial est un défi de taille en termes de communication, notamment en raison de l'absence d'infrastructure dans l'Espace et du volume de données d'observation à propager au sein du système. L'objectif de la thèse est donc de définir une configuration d'architecture fiable, répondant aux contraintes conjointes d'un réseau MANET et d'un système distribué. La thèse commence par caractériser le réseau de l'essaim de nanosatellites et met en avant sa forte hétérogénéité. Ensuite, elle propose des algorithmes permettant de répartir équitablement la charge du réseau, en se basant sur des techniques de division de graphe, et compare les performances de ces algorithmes sur des critères d'équité. Enfin, elle évalue la sensibilité du système aux pannes en termes de robustesse (résistance aux pannes) et de résilience (maintien du fonctionnement en présence de pannes) et étudie l'impact de la division de graphe sur la fiabilité globale de l'essaim. Les algorithmes de division développés dans cette thèse devront garantir la Qualité de Service (QoS) essentielle au bon fonctionnement d'un interféromètre spatial. Pour atteindre cet objectif, des solutions de routage pertinentes devront être minutieusement étudiées et intégrées, afin de s'assurer qu'elles répondent aux exigences strictes de performance et de fiabilité de cette application avancée.

Mots clés : Architecture, Essaim, Système Spatial Distribué, Fiabilité, Robustesse, Résilience

Abstract :

The study of the low-frequency range is essential for Deep Space observation, as it extracts precious information from Dark Ages signals, which are signatures of the very early Universe. To this day, the majority of low-frequency radio interferometers are deployed in desertic regions on the surface of the Earth. However, these signals are easily distorted by radio-frequency interferences as well as the ionosphere, making them hardly observable when they are not completely masked. One solution to this problem would be to observe the low-frequency signals directly from Space, by deploying a nanosatellite swarm in orbit around the Moon. This swarm is defined as a Distributed Space System (DSS) operating as an interferometer, while being shielded by the Moon from terrestrial interferences and ionospheric distortions. However, the configuration of a nanosatellite swarm as a space observatory proves to be a challenging problem in terms of communication, mostly because of the lack of external infrastructure in Space, and the amount of observation data to propagate within the swarm. Thus, the objective of the thesis is to define a reliable network architecture that would comply with the requirements of a MANET and a distributed system, simultaneously. This thesis starts by characterizing the network of the nanosatellite swarm and highlights its strong heterogeneity. Then, it introduces a set of algorithms, based on graph division, to fairly distribute the network load among the swarm, and compares their performance in terms of fairness. Finally, it assesses the fault tolerance of the system in terms of robustness (capacity to resist faults) and resilience (capacity to maintain functionality when faults occur) and evaluates the impact of graph division on the overall reliability of the swarm. The division algorithms developped in this thesis should ensure the Quality of Service (QoS) necessary to the proper functioning of a Space interferometer. To this end, relevant routing protocols should be thoroughly studied and integrated, in order to meet the strict requirements of this advanced application in terms of performance and reliability.

Keywords :

Architecture, Swarm, Distributed Space System, Reliability, Robustness, Resilience