



**HAL**  
open science

# Vers une dissémination efficace de données volumineuses sur des réseaux Wi-Fi denses

Lyes Hamidouche

► **To cite this version:**

Lyes Hamidouche. Vers une dissémination efficace de données volumineuses sur des réseaux Wi-Fi denses. Informatique. Sorbonne Université, Faculté des Sciences et Ingénierie, 2018. Français. NNT : . tel-01953300v1

**HAL Id: tel-01953300**

**<https://inria.hal.science/tel-01953300v1>**

Submitted on 12 Dec 2018 (v1), last revised 23 Jan 2020 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ÉCOLE DOCTORALE EDITE DE PARIS (ED130)  
INFORMATIQUE, TÉLÉCOMMUNICATION ET ÉLECTRONIQUE

THÈSE DE DOCTORAT DE  
L'UNIVERSITÉ SORBONNE

SPÉCIALITÉ : **INGÉNIERIE / SYSTÈMES INFORMATIQUES**

PRÉSENTÉE PAR : **LYES HAMIDOUCHE**

POUR OBTENIR LE GRADE DE :  
**DOCTEUR DE L'UNIVERSITÉ SORBONNE**

SUJET DE LA THÈSE :  
**VERS UNE DISSÉMINATION EFFICACE DE DONNÉES  
VOLUMINEUSES SUR DES RÉSEAUX WI-FI DENSES**

SOUTENUE LE : 21 JUIN 2018

DEVANT LE JURY COMPOSÉ DE :

<i>Rapporteurs :</i>	GABRIEL ANTONIU	Directeur de recherche, INRIA Rennes – Bretagne Atlantique
	VIVIEN QUÉMA	Professeur, Université Grenoble INP
<i>Examineurs :</i>	MARIA POTOP-BUTUCARU	Professeur, Sorbonne Université
	CHRISTOPHE CÉRIN	Professeur, Université Paris 13 - Sorbonne Paris Cité
<i>Directeurs de thèse :</i>	SÉBASTIEN MONNET	Professeur, Université Savoie Mont Blanc
	PIERRE SENS	Professeur, Sorbonne Université
<i>Invité :</i>	DIMITRI REFAUVELET	Directeur technique, Magency





*À mes parents*

# Remerciements

Ma thèse est une expérience scientifique et humaine, pour moi, unique. Fort de tous ceux qui m'ont entouré et donné la force d'aller de l'avant, et fort de tout ce que j'ai appris. Aujourd'hui, j'arrive au bout de mon aventure et tiens à rendre hommage à ceux là qui m'ont accompagné tout ce temps.

Je tiens tout d'abord à remercier mes directeurs de thèse et encadrants Sébastien Monnet, Pierre Sens ainsi que Dimitri Refauvelet. C'est grâce à eux que mon projet de thèse s'est concrétisé.

Sébastien, je te remercie pour ton soutien, pour tous ces échanges que l'on a eu et tout le temps que tu m'as consacré, à Paris ou à Annecy. Ton énergie et ton enthousiasme m'ont été bénéfiques plus d'une fois. Je te remercie de m'avoir fait bénéficier généreusement de ton expérience, de ton écoute, tes conseils et tous les encouragements que tu m'as témoignés.

Pierre, je te remercie de m'avoir suivi et soutenu tout le long de mon parcours de thésard et d'avoir été très disponible pour moi. Merci de tout le temps me rappeler qu'il vaut toujours mieux se concentrer sur l'essentiel :-).

Dimitri, je te remercie pour m'avoir fait confiance dès mes premiers pas de stagiaire chez Magency, ce qui m'a permis de gagner vite en autonomie. Merci de m'avoir soutenu pour la construction de mon projet de thèse et d'avoir respecté mes choix et soutenu tout le long.

À tous les trois, avoir évolué auprès de vous est un plaisir et un grand honneur pour moi, j'ai beaucoup appris à vos côtés, je vous en remercie infiniment.

Je tiens à remercier Gabriel Antoniu et Vivien Quéma d'avoir pris sur leur temps de recherche pour examiner ce manuscrit. Je remercie également les autres membres du jury Maria Potop-Butucaru et Christophe Cérin pour le temps qu'ils acceptent de m'accorder pour examiner mes travaux.

Je remercie également Vincent Bruneau et Johan Mazeyrat de m'avoir accueilli au sein de Magency et de m'avoir permis d'effectuer ma thèse. Je remercie aussi mes anciens collègues de Magency avec une pensée particulière à Jeffrey Thiot, Amine Nabil et Frédéric Bardolle. Je n'oublie pas les nombreuses aventures que l'on a partagées ... De toutes manières, Amine saura me les rappeler si je les oublie ;-).

Je tiens à remercier également tous les membres de l'équipe DELYS, au sein de laquelle j'ai évolué, notamment Julien Sopena, Luciana Arantes, Marc Shapiro, Jonathan Lejeune et Franck Petit. Je remercie aussi Thi-Mai-Trang Nguyen et Syphax Iskounene de l'équipe Phare avec lesquels j'ai été très honoré de collaborer.

Je pense à tous les docteurs, doctorants et ingénieurs avec lesquels j'ai partagé cette aventure au LIP6. Merci à Damien Carver, Gauthier Voron, Ilyas Toumlilt, Maxime Bittan, Redha Gouicem, Florent Coriat, Francis Laniel, Cédric Courtaud, Antoine Blin et Hakan Métin pour les bons moments que l'on a partagés ensemble, pour les nombreux débats (généralement sensés :-)) qu'on a tenus. Je n'oublie pas Marjorie Bournat, Denis Jeanneau, Guillaume Farysse, Ludovic Lefrioux, Sébastien Bouchard, Joao Paulo De Araujo, Alejandro Tomsic, Rudyar Cortes, Maxime Lorrillère et Mahsa Najafzadeh.

Chez Interactive Mobility, j'ai aujourd'hui la chance d'aller dans la continuité de mon domaine de recherche ; merci à Tanguy Morel, Michael Serres et Julien Desenne pour l'opportunité qu'ils m'offrent et pour la confiance qu'ils m'accordent.

Je n'oublie pas mes amis qui m'entourent au quotidien : Émilia, Lakhdar, Nabil, Mohand, Aris, Massine, Hend, Sabrina, Djabir, Yacine et Massinissa. Merci pour le soutien que vous m'apportez, votre présence et pour tous les beaux moments que l'on partage ensemble.

Enfin et pas des moindres, je remercie ma famille au sein de laquelle je puise ma force. Merci à mes deux grandes sœurs Amina et Katia, d'avoir été des modèles à suivre, de m'avoir encouragé et soutenu inconditionnellement, d'avoir partagé mes succès comme mes moins bons moments. Je remercie mon frère Aghilas-Billy, soutien incontestable et compagnon de chambre de longue date, dont l'énergie et l'humour ravageur me mettent toujours de bonne humeur.

Je remercie mon père et ma mère, Mouloud et Zahia. Mes chers parents, je vous remercie pour tout ce que vous faites pour nous, de faire de notre bien-être votre priorité. Merci pour tous les grands sacrifices que vous avez faits afin que l'on puisse avoir la chance de faire nos études dans le meilleur cadre possible, pour nous avoir encouragé à aller le plus loin possible dans nos études et dans notre vie. Merci pour votre soutien inconditionnel et les encouragements que vous nous témoignez tous les jours. Toute ma vie, je ne vous remercierai jamais assez.



# Table des matières

<b>I</b>	<b>Introduction</b>	<b>1</b>
I.1	Contexte . . . . .	2
I.2	Contributions . . . . .	4
I.3	Organisation du manuscrit . . . . .	5
I.4	Liste des publications . . . . .	6
<b>II</b>	<b>Approches existantes pour le partage et la diffusion de données</b>	<b>9</b>
	Introduction . . . . .	10
II.1	Partage et diffusion de données sur les réseaux pair-à-pair (P2P) . . .	10
II.1.1	Partage de fichiers . . . . .	11
II.1.2	Tables de hachage distribuées (DHT) . . . . .	15
II.1.3	Systèmes de fichiers distribués . . . . .	17
II.1.4	Synthèse . . . . .	18
II.2	Partage et diffusion de données sur les réseaux mobiles . . . . .	18
II.2.1	Réseaux mobiles adhoc (MANETs) . . . . .	19
II.2.2	Diffusion de données sur les réseaux D2D . . . . .	21
II.2.3	Consommation d'énergie sur les réseaux D2D . . . . .	26
II.2.4	Synthèse . . . . .	27
II.3	Zoom sur les réseaux Wi-Fi . . . . .	27
II.3.1	Modes de diffusion de données en Wi-Fi . . . . .	28
II.3.2	Réseaux Wi-Fi denses . . . . .	33
II.3.3	Synthèse . . . . .	36
	Conclusion . . . . .	37
<b>III</b>	<b>EDWiN : dissémination D2D sur les réseaux Wi-Fi denses</b>	<b>39</b>
	Introduction . . . . .	40
III.1	Architecture et hypothèses . . . . .	41
III.1.1	Architecture du système . . . . .	41
III.1.2	Représentation du réseau . . . . .	43
III.1.3	Représentation d'un périphérique mobile . . . . .	43
III.2	Algorithme de dissémination . . . . .	45
III.2.1	État global et types de messages . . . . .	45
III.2.2	Serveur / ordonnanceur : . . . . .	46
III.2.3	Clients . . . . .	50
	Conclusion . . . . .	51



<b>IV EDWiN' : prise en compte de la consommation d'énergie de l'hétérogénéité des périphériques mobiles</b>	<b>55</b>
Introduction . . . . .	56
IV.1 Réduction de la consommation énergétique par regroupement (clustering) . . . . .	56
IV.2 Hétérogénéité des batteries . . . . .	57
IV.2.1 Modèle de batterie . . . . .	57
IV.2.2 Description de l'algorithme . . . . .	58
IV.3 Hétérogénéité des batteries et des bandes passantes . . . . .	61
IV.3.1 Modèle de bande passante . . . . .	61
IV.3.2 Description de l'algorithme . . . . .	63
Conclusion . . . . .	66
<b>V Évaluation d'EDWiN</b>	<b>69</b>
Introduction . . . . .	70
V.1 Simulateur et scénario d'expérimentation . . . . .	70
V.1.1 Description du simulateur . . . . .	70
V.1.2 Métriques . . . . .	72
V.1.3 Configurations de la simulation . . . . .	73
V.2 Évaluation de l'algorithme de dissémination D2D . . . . .	76
V.2.1 Degré de parallélisme . . . . .	76
V.2.2 Passage à l'échelle . . . . .	77
V.2.3 Volumes de données échangés . . . . .	78
V.2.4 Énergie et contrôle de l'activité des périphériques mobiles . . . . .	81
V.2.5 Prise en compte de l'hétérogénéité des batteries . . . . .	83
V.2.6 Prise en compte de l'hétérogénéité des batteries et des bandes passantes . . . . .	88
Conclusion . . . . .	90
<b>VI Topologies hiérarchiques pour la dissémination de données : regroupement (clustering) sur les réseaux Wi-Fi Direct</b>	<b>93</b>
VI.1 Clustering basé sur Wi-Fi Direct : approche par SDN . . . . .	94
VI.1.1 Architecture du réseau . . . . .	94
VI.1.2 Algorithme de clustering . . . . .	96
VI.1.3 Gestion des interférences et planification de la transmission . . . . .	99
VI.1.4 Calcul du nombre optimal de clusters . . . . .	101
VI.2 Évaluation des performances . . . . .	102
VI.2.1 Performance du clustering . . . . .	102
VI.2.2 Simulation avec OMNet++ . . . . .	104
Conclusion . . . . .	108
<b>VII Conclusion et perspectives</b>	<b>111</b>
VII.1 Conclusion . . . . .	112
VII.2 Perspectives . . . . .	114

---

Bibliographie	117
Liste des figures	127
Liste des tableaux	129
Liste des algorithmes	130
Résumé	131



# CHAPITRE



## Introduction

---

### Sommaire

---

I.1	Contexte . . . . .	2
I.2	Contributions . . . . .	4
I.3	Organisation du manuscrit . . . . .	5
I.4	Liste des publications . . . . .	6

---

## I.1 Contexte

Nous sommes à l'ère des technologies mobiles, leur utilisation se répand rapidement dans la société moderne. L'utilisation des smartphones et tablettes tactiles s'étend à de plus en plus d'industries. L'évolution des performances de ces périphériques s'accompagne d'une croissance du volume des contenus utilisés par les applications. En général, pour télécharger ces données, les périphériques mobiles utilisent leur connexion Wi-Fi ou leur connexion cellulaire. L'utilisation classique de ces technologies de communication, c'est-à-dire basé sur le mode infrastructure<sup>1</sup>, ne permet pas un passage à l'échelle lorsque la charge sur le réseau augmente. De plus, en vue de la croissance du trafic Internet attendue pour les prochaines années [1], il devient important de s'intéresser à des mécanismes de diffusion de données plus efficaces sur les réseaux sans fil, notamment en ce qui concerne les réseaux Wi-Fi qui occupent une place importante.

Les séminaires et conférences sont des exemples d'événements pendant lesquels se rassemble un grand nombre d'individus autour d'un intérêt qu'ils partagent (formation, exposition, etc.). L'entreprise Magency<sup>2</sup> est spécialisée dans l'animation de ce type d'événement à grande échelle. Le rôle des équipes de Magency est de fournir aux participants les périphériques et les applications mobiles qu'ils vont utiliser pour participer aux différentes activités et accéder aux contenus qui sont mis à leur disposition. Il existe plusieurs types d'application :

- **Les lecteurs multimédia** : ces applications permettent aux participants d'accéder aux documents mis à leur disposition pour l'événement (par exemple : le plan du site, des vidéos de présentation, les documents utilisés dans les ateliers, etc.). Les données utilisées par ces applications sont partagées entre tous les utilisateurs et nécessitent une bande passante importante en raison de leur grande taille ;
- **Les applications collaboratives** : ces applications ont pour objectif d'animer des activités participatives (par exemple : une séance de questions/réponse, vote instantané, questions à réponse rapide, etc.). Ces applications utilisent une bande passante réduite en raison du petit volume de données qu'elles font transiter sur le réseau. Cependant, elles requièrent des temps de communication courts ;
- **Les applications de type quizz** : ces applications ont pour objectif de permettre à l'utilisateur de répondre à des questions ou de traiter des exercices proposés dans le cadre de l'événement. Les données utilisées par ce type d'application occupent un espace disque de l'ordre de quelques centaines de kilo-octets voir quelques méga octets ;
- **Les applications utilitaires** : ces applications ont pour but d'aider les participants à avoir les informations nécessaires au bon déroulement de leur évé-

---

1. les différents modes de communication sont détaillés au chapitreII

2. [www.magency.me](http://www.magency.me)

ment afin de simplifier leurs échanges (par exemple : plan de table, messagerie, e-carte de visite, agenda, etc.);

Ainsi, chaque événement dispose de son propre lot d'applications et de données (fichiers vidéos, documents PDF, etc), ces données peuvent être volumineuses. Les périphériques mobiles communiquent via un réseau Wi-Fi déployé sur le site. Le besoin d'offrir une bonne qualité d'expérience (QoE) perçue par l'utilisateur implique d'avoir l'accès le plus rapide possible aux données utilisées par les applications sachant que le trafic des différentes applications doit cohabiter sur le même réseau.

Cette thèse est motivée par la problématique de montée en charge du trafic sur les réseaux Wi-Fi utilisés lors de ces événements à grande échelle (séminaires, conférences, salons d'exposition, etc.). L'accès simultané de plusieurs périphériques mobiles à une même donnée pouvant être volumineuse provoque une contention et une baisse des performances du réseau, augmentant ainsi le temps de dissémination de la donnée sur le réseau. En pratique, pour palier ce manque d'efficacité des infrastructures Wi-Fi existantes, celles-ci doivent être surdimensionnées par rapport au besoin réel du trafic. De plus, afin de réduire ce problème, en pratique, les données accédées par les utilisateurs sont, en général, manuellement pré-chargées sur les périphériques mobiles. Ce choix a un coût humain important et le surdimensionnement ne répond que partiellement au problème car cela fait augmenter la densité d'antennes Wi-Fi, ce qui accroît les risques d'interférences et de contention sur les différents canaux utilisés.

De nombreuses approches reposant sur le paradigme pair-à-pair (P2P) [2, 3, 4, 5] ont été proposées et permettent de concevoir des systèmes de distribution de données à grande échelle afin de garantir la robustesse, la fiabilité et le passage à l'échelle. Ce type de solution peut être mis en place au niveau des serveurs d'applications afin d'offrir des services passant à l'échelle. Cependant, ces solutions conçues initialement pour des réseaux filaires ne résolvent que partiellement le problème de passage à l'échelle sur des réseaux dont les clients sont exécutés sur des périphériques mobiles. En effet, le coup pour maintenir l'information sur le réseau de recouvrement (*overlay*) est non négligeable. De plus, cela implique l'échange d'un nombre important de messages ainsi que l'établissement de multiples connexions autonomes dont le coût, sur une technologie sans fils est, là encore, non négligeable.

Les technologies de communication de mobile à mobile (M2M), aussi appelées *device-to-device* en anglais (D2D), telles que Bluetooth ou Wi-Fi Direct permettent aux périphériques de créer des communications directes sans utiliser de points d'accès. Elles ont suscité l'intérêt de nombreux travaux dans les réseaux cellulaires [6] dont l'objectif est de permettre de meilleures performances des réseaux grâce à ces technologies. Ces communications permettent de traiter des problèmes tels que le déchargement des infrastructures [7], l'amélioration de la qualité de service [8] et de la couverture du réseau [9]. Les communications D2D ont vocation à être utilisées

de manière contrôlée. Leur mise en œuvre dans des architectures avec un contrôle centralisé et des communications parallèles, a l'avantage de simplifier l'orchestration et offre une meilleure visibilité de l'état global du réseau par rapport à l'utilisation d'approches décentralisées pures.

Nous présentons dans cette thèse deux types d'approches pour faire face à ce problème de dissémination : (i) une approche basée sur des connexions D2D en point-à-point entre les périphériques mobiles, supervisées par un serveur, et (ii) une approche reposant sur des topologies hiérarchiques. Nous considérons, de manière pragmatique, une architecture semi-centralisée pour la dissémination de données D2D sur les réseaux Wi-Fi. En effet, la simplicité de mise en œuvre, l'applicabilité directe dans un contexte industriel et l'adaptabilité de ce type de solution sont des avantages non négligeables. De plus, l'information sur l'état du réseau centralisée permet de s'adapter plus aisément aux changements de conditions.

Cette thèse s'inscrit dans le cadre d'une convention CIFRE entre Magency et le LIP6<sup>3</sup> (Laboratoire d'Informatique de Paris 6). Elle a été préparée au sein l'équipe REGAL<sup>4</sup> sous la direction de Sébastien MONNET et de Pierre SENS.

## I.2 Contributions

Dans cette thèse, nous abordons deux manières d'appréhender la problématique de dissémination de données sur les réseaux Wi-Fi en utilisant les communications D2D. Le mode de déploiement pour ces deux approches est le même : il repose sur une supervision centralisée au niveau du serveur et une dissémination (parallèle) distribuée des données. Nous avons une première approche en point-à-point plus facilement adaptable à des réseaux dans lesquels les périphériques mobiles peuvent avoir un voisinage dynamique. La deuxième approche est adaptée à des configurations plus statiques des périphériques mobiles. Cette thèse se découpe en trois principales contributions :

**Un protocole de dissémination D2D en point-à-point :** notre première contribution, EDWiN<sup>5</sup>, propose une implémentation d'un protocole de dissémination de données en utilisant des connexions D2D de point-à-point entre les périphériques mobiles. Les périphériques établissent des connexions D2D avec leurs voisins au cours de la dissémination pour échanger des données selon un modèle proche de BitTorrent [2]. Elle prend en compte la nécessité de fournir un service de dissémination efficace qui ne crée pas de contention sur les points d'accès. Cette approche peut être adaptée en fonction de l'état courant du réseau afin de limiter le nombre d'échanges D2D en parallèle, ce qui dégraderait les performances du réseau (à cause des inter-

---

3. [www.lip6.fr](http://www.lip6.fr)

4. [team.inria.fr/regal](http://team.inria.fr/regal)

5. Efficient data Dissemination over Wi-Fi Networks

férences). Les travaux liés à cette contribution ont été publiés dans la conférence nationale [Compas'16], ainsi que dans la conférence internationale [AINA'17].

**La prise en compte de la consommation énergétique et de l'hétérogénéité des périphériques mobiles :** les périphériques mobiles bénéficient en permanence de nouvelles évolutions de leurs composants. Par conséquent, les périphériques plus anciens deviennent moins performants en comparaison avec les plus récents. Ils disposent de batteries avec une faible autonomie et de bandes passantes moins élevées. Dans ce contexte, il est important de maintenir une efficacité énergétique globale qui répartit équitablement le coût énergétique sur les périphériques. Cela permet de préserver les périphériques dont les batteries sont les plus faibles en augmentant la charge sur les périphériques les plus performants. Nous avons donc étendu EDWiN pour prendre en compte l'hétérogénéité des périphériques mobiles sur le réseau afin d'offrir une dissémination qui est efficace du point de vue de la consommation énergétique et de l'utilisation de la bande passante. Les résultats de cette contribution ont été publiés dans la conférence nationale [Compas'17], ainsi que dans la conférence internationale [ICPADS'17].

**La dissémination sur des topologies hiérarchiques en Wi-Fi Direct :** au cours de ma thèse, j'ai eu l'opportunité de participer à une collaboration inter-équipes avec l'équipe réseau Phare<sup>6</sup> du LIP6. Dans ce cadre, j'ai pu participer à l'encadrement du stage de Master 2 de Syphax Iskounen "Transmission de données par communications device-to-device(D2D)". Durant son stage, il a développé un simulateur Wi-Fi Direct sur OMNet++. Nous nous intéressons ici à des topologies statiques qui permettent de mettre en place des méthodes de regroupement (clustering) pour créer des groupes de communication Wi-Fi Direct afin de construire une topologie de dissémination hiérarchique qui améliore les performances. Ces travaux ont été publiés dans une conférence internationale [NoF'16]. Par la suite, ces travaux ont été étendus pour prendre en considération les interférences pouvant survenir avec ce type de topologie, et étudier l'importance du choix de la taille des clusters. Ces résultats ont été publiés dans le journal "*Annals of Telecommunications (AoT)*"[AoT'17].

### I.3 Organisation du manuscrit

La suite de cette thèse est organisée de la façon suivante :

- **Chapitre II : Approches existantes pour le partage et la diffusion de données.** Dans ce chapitre, nous présentons un état de l'art des réseaux pair-à-pair et des différentes approches de partage et de diffusion de données sur les réseaux mobiles. Nous présentons des travaux existants pour les réseaux mobiles adhoc et D2D. Nous décrivons également les mécanismes de transmission

---

6. [www.lip6.fr/phare](http://www.lip6.fr/phare)



de données sur les réseaux Wi-Fi ainsi que des travaux menés dans le but d'améliorer les performances de ces réseaux.

- **Chapitre III : EDWiN : dissémination D2D sur les réseaux Wi-Fi denses.** Dans ce chapitre, nous présentons notre première contribution ; le modèle et les hypothèses d'EDWiN ainsi que l'algorithme de dissémination de base. Nous décrivons l'architecture de notre système et les différents mécanismes mis en place pour améliorer la dissémination de données sur des réseaux Wi-Fi.
- **Chapitre IV : EDWiN' : prise en compte de la consommation d'énergie de l'hétérogénéité des périphériques mobiles.** Dans ce chapitre, nous présentons notre deuxième contribution ; les différentes approches que nous avons utilisées pour la prise en compte de la consommation énergétique et de l'hétérogénéité des périphériques mobiles.
- **Chapitre V : Évaluation d'EDWiN.** Dans ce chapitre, nous présentons une évaluation d'EDWiN et discutons de l'impact des différents paramètres sur les performances de la dissémination.
- **Chapitre VI : Topologies hiérarchiques pour la dissémination de données : regroupement (clustering) sur les réseaux Wi-Fi Direct.** Dans ce chapitre, nous présentons notre dernière contribution ; les mécanismes utilisés pour une dissémination de données sur des topologies hiérarchiques sur les réseaux Wi-Fi. Nous discutons de la formation contrôlée de clusters Wi-Fi Direct et étudions aussi l'impact du nombre et de la taille des clusters pour optimiser les performances de la dissémination.
- **Chapitre VII : Conclusion et perspectives.** Enfin, dans ce chapitre, nous concluons cette thèse, nous discutons des avantages et des limites de notre approche et présentons les perspectives de recherche qu'il serait intéressant d'explorer par la suite.

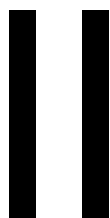
## I.4 Liste des publications

- [ICPADS'17] L. Hamidouche, S. Monnet, P. Sens, and D. Refauvelet, "Toward heterogeneity-aware device-to-device data dissemination over Wi-Fi networks," in *ICPADS 2017 - International Conference on Parallel and Distributed Systems*, (Shenzhen, China), Dec. 2017
- [AoT'17] T.-M.-T. Nguyen, L. Hamidouche, F. Mathieu, S. Monnet, and S. Iskounen, "SDN-based Wi-Fi Direct Clustering for Cloud Access in Campus Networks," *Annals of Telecommunications*, Springer, 2017
- [Compas'17] L. Hamidouche, S. Monnet, P. Sens, and D. Refauvelet, "Prise en compte de l'autonomie des batteries des périphériques mobiles pour la diffusion efficace de données en D2D.," in *Compas'2017*, (Sophia Antipolis, France), 2017
- [AINA'17] L. Hamidouche, S. Monnet, F. Bardolle, P. Sens, and D. Refauvelet, "Edwin: Leveraging device-to-device communications for efficient data dissemination over wi-fi networks," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, pp. 228–235, March 2017

- 
- [NoF'16] S. Iskounen, T. M. T. Nguyen, S. Monnet, and L. Hamidouche, "Device-to-device communications using wi-fi direct for dense wireless networks," in *2016 7th International Conference on the Network of the Future (NOF)*, pp. 1–3, Nov 2016
- [Compas'16] L. Hamidouche, S. Monnet, and P. Sens, "EDWiN : Vers une diffusion efficace de données sur les réseaux Wi-Fi," in *Compas'2016*, (Lorient, France), 2016



# CHAPITRE



## Approches existantes pour le partage et la diffusion de données

---

### Sommaire

---

<b>Introduction</b> . . . . .	<b>10</b>
<b>II.1 Partage et diffusion de données sur les réseaux pair-à-pair (P2P)</b>	<b>10</b>
II.1.1 Partage de fichiers . . . . .	11
II.1.2 Tables de hachage distribuées (DHT) . . . . .	15
II.1.3 Systèmes de fichiers distribués . . . . .	17
II.1.4 Synthèse . . . . .	18
<b>II.2 Partage et diffusion de données sur les réseaux mobiles</b> . . . . .	<b>18</b>
II.2.1 Réseaux mobiles adhoc (MANETs) . . . . .	19
II.2.2 Diffusion de données sur les réseaux D2D . . . . .	21
II.2.3 Consommation d'énergie sur les réseaux D2D . . . . .	26
II.2.4 Synthèse . . . . .	27
<b>II.3 Zoom sur les réseaux Wi-Fi</b> . . . . .	<b>27</b>
II.3.1 Modes de diffusion de données en Wi-Fi . . . . .	28
II.3.2 Réseaux Wi-Fi denses . . . . .	33
II.3.3 Synthèse . . . . .	36
<b>Conclusion</b> . . . . .	<b>37</b>

---

## Introduction

La performance d'accès aux données a toujours été au cœur des préoccupations des réseaux informatiques. De nombreuses contributions dans la littérature ont eu pour but d'améliorer les performances du partage et de la diffusion de données entre les différents nœuds des réseaux. L'augmentation des performances et des capacités de stockage des réseaux peut avoir un coût non négligeable. Les systèmes pair-à-pair (P2P) permettent de mutualiser les ressources des nœuds du réseau (bande passante, espace de stockage, puissance de calcul). Cette mutualisation permet d'utiliser efficacement les ressources disponibles sur le réseau et d'avoir des systèmes capables de monter en charge sans nécessairement nécessiter un coût supplémentaire. Les réseaux mobiles n'échappent pas à cette règle. Cependant, la manière dont les technologies fonctionnent dans ce type de réseau ne permettent pas de gérer le partage de ressources réseau comme le permettent les réseaux filaires. En effet, les réseaux mobiles peuvent être limités par la portée du signal. De plus, les communications peuvent subir des interférences.

Dans ce chapitre, nous présentons les principaux travaux autour du partage et de la diffusion de données sur les réseaux informatiques et notamment sur les réseaux mobiles. Nous prêtons une attention particulière aux réseaux mobiles et plus particulièrement aux réseaux Wi-Fi qui sont au cœur de la problématique de cette thèse. Dans la section II.1, nous présentons des travaux de partage et de diffusion de données sur les réseaux P2P. Ensuite, nous présentons les approches existantes pour la diffusion et le partage de données sur les réseaux mobiles dans la section II.2. Dans la section II.3, nous faisons un zoom sur les réseaux Wi-Fi en présentant les mécanismes existants pour la communication avec cette technologie.

## II.1 Partage et diffusion de données sur les réseaux pair-à-pair (P2P)

Les réseaux pair-à-pair (P2P) ont été introduits afin reconsidérer l'architecture des systèmes distribués pour concevoir des systèmes plus performants, plus robustes et qui passent à l'échelle. Ils permettent de mettre en commun les ressources disponibles sur le réseau (puissance de calcul, stockage, bande passante, etc.) afin d'éviter les contentions que l'on peut avoir avec des architectures client-serveur centralisées. La littérature fait état de nombreux travaux autour de ce paradigme. On le retrouve dans de nombreux systèmes comme des systèmes de fichiers distribués [16, 17, 18], des structures de données distribuées [19, 20, 4], des protocoles de partages de fichiers [2, 21, 22] ou des systèmes de calcul distribué [23, 24].

Un réseau P2P peut avoir une architecture centralisée ou décentralisée. En fonction des interconnexions entre les nœuds du réseau, celui-ci peut avoir une structure plate (mono-tier) ou hiérarchique (multi-tiers) et une topologie structurée ou non-structurée. Les nœuds sont connectés grâce à un réseau logique (overlay) qui

définit l'ensemble de mécanismes de communication entre les nœuds, les stratégies de routage des messages ainsi que la répartition des ressources. En plus de ces deux catégories, nous pouvons avoir des réseaux hybrides qui permettent de bénéficier des avantages des deux architectures. Nous nous intéressons ici à ces familles de contribution : le partage de fichiers, les structures de données distribuées et systèmes de fichiers distribués.

### II.1.1 Partage de fichiers

Les protocoles de partage de fichiers sont des cas d'utilisation très populaires de l'utilisation des réseaux P2P. Dans cette sous-section, nous présentons Napster [22], BitTorrent [2], KaZaa [3] et Gnutella [21].

#### II.1.1.1 Napster

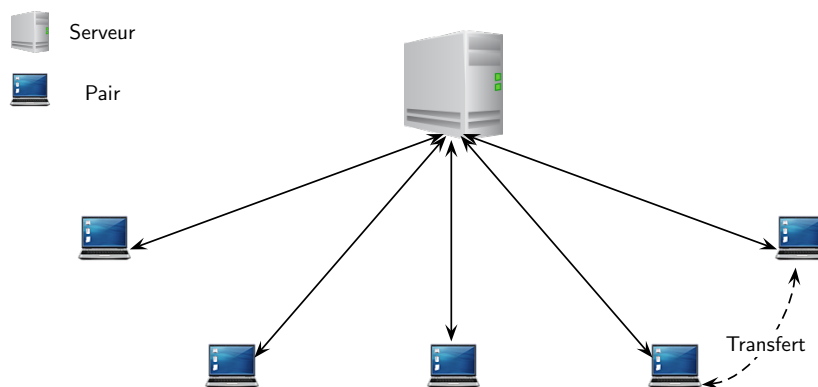


FIGURE II.1: Architecture de Napster

Napster [22] est le premier logiciel libre de partage de fichiers en P2P dont la recherche de contenus se base sur une architecture centralisée. Un serveur central (voir figure II.1) permet l'indexation et la localisation des données contenues sur chacun des nœuds du réseau et les échanges P2P ont directement lieu entre les nœuds concernés par le partage. Le serveur remplit aussi la fonction de point de rencontre pour les nœuds (clients) qui utilisent son service.

Lorsqu'un nœud rejoint le réseau, il envoie au serveur la liste des fichiers qu'il souhaite partager. La recherche d'un fichier se fait par l'envoi d'une requête au serveur, qui lui retourne la liste des nœuds qui le possèdent. Ainsi, un échange P2P entre les nœuds concernés permet le partage du fichier. Cette architecture a l'avantage d'être simple à mettre en œuvre et à administrer. Cependant, le serveur constitue un goulot d'étranglement pour le passage à l'échelle du système. En effet,

lorsqu'il subit un trafic de requêtes important, cela engendre des temps traitement longs. De plus, si le serveur est défaillant, tout le système s'effondre.

### II.1.1.2 Gnutella

Gnutella [21], est un système de partage de fichiers P2P complètement distribué dont la topologie est non-structurée. La structure du réseau étant plate, chaque nœud remplit la même fonction que ses voisins.

Lorsqu'un nœud se connecte au réseau Gnutella, il diffuse un message de type "PING" pour indiquer sa présence. Lorsque les autres nœuds du réseau reçoivent ce message, ils répondent par un message "PONG" pour l'informer qu'ils ont pris connaissance du nœud qui vient de s'insérer sur le réseau. Ainsi, le nœud qui vient d'arriver établit son voisinage et peut participer au partage de fichiers. La recherche de contenus sur le réseau se base sur la diffusion, par inondation, de requêtes à travers le réseau. Un nœud n'a pas nécessairement besoin d'informer ses voisins de lorsqu'il quitte le réseau. Cependant, chaque nœud maintient son voisinage en diffusant des requêtes "PING" qui lui permettent de vérifier si ses voisins sont toujours connectés.

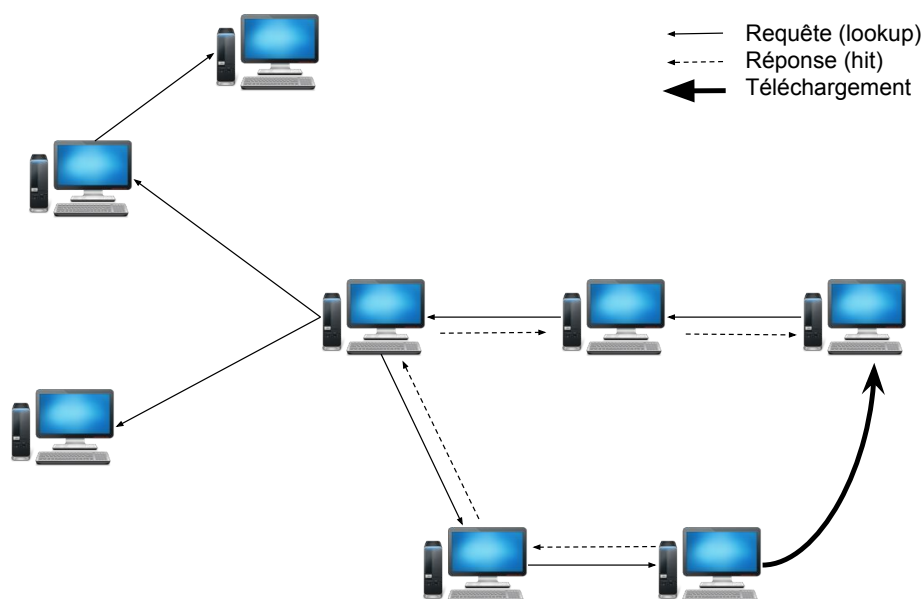


FIGURE II.2: Illustration d'un partage de fichier dans un réseau Gnutella.

Comme l'illustre la figure II.2, la recherche de fichiers se fait par un mécanisme d'inondation. Lorsqu'un nœud recherche un fichier sur le réseau, il diffuse une requête de recherche (lookup) à tous ses voisins. Ces derniers le transmettent à leur voisins de la même manière qu'ils l'ont reçu. Un nœud contenant la donnée

peut répondre par un message de type “hit” s’il contient le fichier. Ce message de réponse emprunte le même chemin que le message de “lookup” dans le sens inverse. Lorsque la donnée est localisée, le transfert est effectué directement depuis le nœud source. Sinon, le message de recherche est diffusé jusqu’à ce que tout le réseau soit parcouru ou que la durée de vie (TTL ou *time-to-live* en anglais) du message ait expiré.

La diffusion par inondation des requêtes a l’avantage de permettre des réponses rapides aux requêtes et de résister aux pannes. Cependant, plus le réseau est grand plus le nombre de messages diffusés pour la recherche de donnée est grand. Cela peut d’ailleurs engendrer une congestion du réseau. De plus, l’usage de cette méthode de communication est inefficace, surtout lorsque la donnée recherchée sur le réseau est rare. En effet, la recherche engendre un trafic important à cause du nombre de messages diffusés et la recherche peut prendre un temps considérable à cause du nombre de pairs interrogés. De plus, la requête de recherche peut ne pas être satisfaite si le TTL expire avant que le nœud contenant la donnée ne soit atteint.

### II.1.1.3 KaZaa

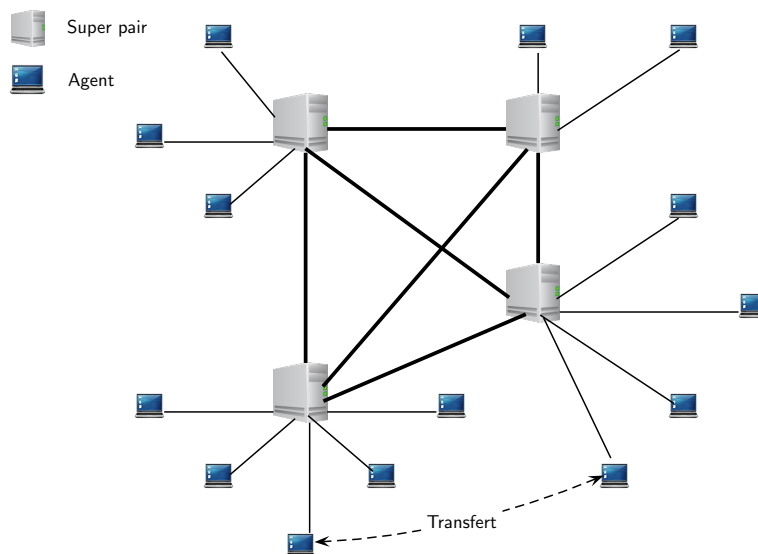


FIGURE II.3: Architecture de KaZaa/FastTrack

KaZaa [3] est un protocole de partage de fichier P2P qui est basé sur le protocole FastTrack [25]. Il repose sur une architecture hybride (semi-centralisée) à 2 tiers : (i) des super-nœuds et (ii) des nœuds agents (clients). Les super-nœuds sont interconnectés entre eux et se chargent d’indexer les méta-données sur les contenus de leurs essaims d’agents respectifs. Ainsi, Lorsqu’un agent cherche une donnée, il interroge plusieurs super-nœuds pour trouver la liste des agents depuis lesquels



il peut télécharger la donnée. Avec plusieurs super-nœuds, KaZaa peut monter en charge et résister aux pannes.

#### II.1.1.4 BitTorrent

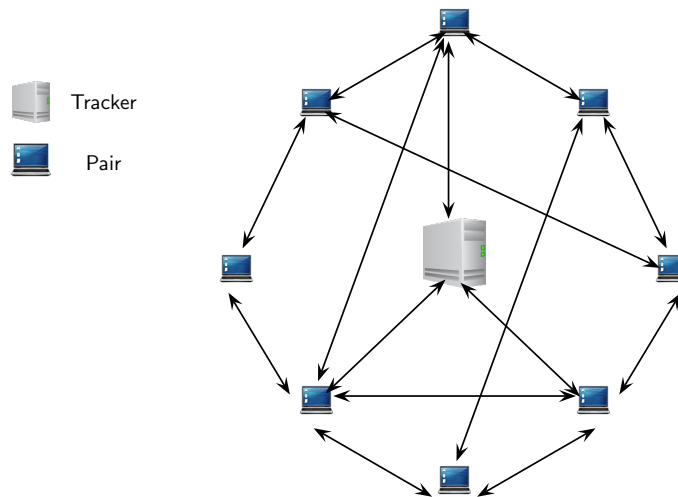


FIGURE II.4: Architecture de BitTorrent

BitTorrent [2] est un système de partage de fichiers P2P qui repose sur l'utilisation simultanée de plusieurs sources de données pour le téléchargement d'un fichier. La figure II.4 représente l'architecture de la version initiale de BitTorrent. Elle compte trois types de nœuds sur le réseau :

- *Tracker* : c'est un serveur centralisé qui se charge d'orchestrer le partage de données entre les différents pairs ;
- *Seeder* : il s'agit d'un nœud du réseau ayant téléchargé le fichier partagé en premier. Ce nœud est utilisé comme source de données ;
- *Leecher* : c'est le nœud qui télécharge le fichier partagé ;

Afin de pouvoir augmenter rapidement le nombre de sources sur le réseau, chaque fichier partagé est découpé en morceaux (chunks). Chaque chunk est aussi découpé en sous-chunks. Pour qu'un *leecher* puisse télécharger des fichiers sur le réseau, il utilise des fichiers ".torrent" qui sont accessibles sur des réseaux autres que le réseau BitTorrent (ex : site web dédié, dépôt de fichiers FTP, etc.). Ce fichier ".torrent" contient les métadonnées nécessaires au téléchargement d'un fichier (ex : adresse du tracker, nom, taille du fichier, adresses de seeders, taille des chunks, etc).

Pour accélérer le téléchargement des fichiers, BitTorrent utilise un ensemble de politiques qui lui permettent de rapidement créer des copies des chunks et d'augmenter le nombre de sources disponibles sur le réseau :

- Un chunk doit être téléchargé dans sa totalité avant de passer au chunk suivant ;
- Utilisation d'une politique "*rarest first*", qui favorise le téléchargement des chunks les plus rares sur le réseau. Cela permet d'augmenter le nombre de sources à partir duquel ils peuvent être téléchargés ;
- Le premier chunk téléchargé est choisi aléatoirement, l'utilisation de la politique "*rarest first*" n'est pas efficace car le nombre de sources des pièces les plus rares est réduit, ce qui peut engendrer des contentions sur le peu de sources (voir l'unique source) du chunk le plus rare ;
- La fin du téléchargement d'un fichier peut être ralentie si les derniers morceaux sont téléchargés depuis une source offrant un débit faible. Le mode "endgame" (fin de jeu) permet de diffuser les requêtes de téléchargement des derniers chunks sur tout le réseau afin d'augmenter le nombre de source et rapidement finir le téléchargement.

Bittorrent utilise un ensemble de mécanismes qui encouragent le partage entre les nœuds du réseau et qui défavorisent les nœuds les moins coopératifs en permettant leur blocage jusqu'à ce qu'ils coopèrent.

Depuis sa version 4.1, BitTorrent utilise un *tracker* distribué qui consiste en la contribution de chaque nœud en tant que *tracker* léger. Ce protocole repose sur l'utilisation des tables de hachage distribuées (DHT) qui lui permettent d'être plus robuste, de s'auto-configurer plus aisément et de mieux passer à l'échelle.

### II.1.2 Tables de hachage distribuées (DHT)

Afin d'avoir un fonctionnement performant, un système P2P doit être capable de fournir un ensemble de mécanismes garantissant des délais de communication et d'accès aux ressources bornés. Les tables de hachage distribuées (DHT) [4, 20, 26, 5] sont des systèmes P2P dont l'architecture a une structure. Dans ce type de réseau, les nœuds construisent un réseau d'interconnexions logiques qui permet de structurer la topologie des connexions entre les différents nœuds et de garantir ainsi un chemin déterministe que suivent les messages et ce afin de pouvoir rapidement localiser l'emplacement des données stockées sur le réseau. L'avantage que peuvent avoir ces systèmes est leur capacité à s'adapter aux départs et arrivées des nœuds (churn) sur le réseau, en permettant la reconstruction des données d'un nœud partant grâce aux copies stockées par ses voisins.

Ces systèmes sont conçus pour passer à l'échelle et sont capables de s'auto-configurer. En répartissant l'espace de nommage sur le réseau, chaque nœud devient responsable du stockage d'un certain nombre d'adresses. La lecture et l'écriture des données sont respectivement assurées par des opérations de type *Get(key)* et *Put(key, val)*.

Afin d'identifier les données et les différents pairs de manière unique, une fonction de hash (par exemple : SHA-1) peut être utilisée à cet effet. L'unicité des identifiants permet de répartir l'espace d'adresses de stockage sur les différents nœuds du réseau et de trouver ainsi, pour chaque donnée, le nœud responsable de son stockage. A l'arrivée d'un pair sur le réseau, il reçoit un identifiant et devient à

responsable d'un espace d'adresses qui définit les adresses des données qu'il peut stocker.

Les DHT peuvent être implémentées dans différentes topologies, nous citons ici quelques exemples de DHT connus. Chord [4] et Pastry [20] sont des DHT qui utilisent un overlay en forme d'un anneau dans lequel les pairs sont organisés en fonction de leurs identifiants. Pour chaque donnée, le système calcule son identifiant grâce à une fonction de hachage afin de déterminer l'adresse (identifiant) du pair dans lequel elle sera stockée. Chaque nœud établit et maintient une table de routage qui lui permet de router les requête qu'il ne peut pas traiter en les redirigeant vers le nœud de son voisinage qui est le plus proche du nœud responsable de son stockage.

Prenons l'exemple de Chord. Dans Chord, par exemple, chaque pair connaît l'identifiant de son successeur et de son prédécesseur et maintient aussi une table de routage appelée "*finger table*". Cette table a au plus  $m$  entrées ( $m$  étant un paramètre du système). Dans la "*finger table*" d'un pair  $p$ , l' $i^{\text{me}}$  entrée correspond au pair dont l'identifiant est la plus petite valeur supérieure ou égale à  $p + 2^{i-1} \bmod(2^m)$ . Ainsi lorsque la recherche d'une donnée n'est pas satisfaite par un nœud ou son successeur, la requête est relayée au nœud de la *finger table* dont l'identifiant est le plus proche.

L'intérêt d'une structure comme la *finger table* est de réduire le nombre d'étapes de recherche dans le réseau. En effet, au lieu d'explorer la totalité des nœuds du réseau, pour la recherche, le relayage de la requête au pair dont l'identifiant est le plus proche de la clé permet de réduire le nombre d'itération et d'obtenir une recherche en  $\mathcal{O}(\log N)$ , où  $N$  est la taille du réseau.

CAN [5] est un système de stockage distribué qui repose sur l'utilisation d'un espace de nommage à  $d$  dimensions. De manière analogique avec les approches présentées précédemment, chaque nœud est identifié par des coordonnées qui permettent de définir la partition de l'espace de nommage dont il est responsable. Chaque pair maintient ainsi une table de routage qui contient les coordonnées de chacun de ses voisins. Lors de l'arrivée d'un nœud sur le réseau, il sélectionne un point dans l'espace de nommage, découvre le pair qui en est responsable. Les deux pairs se partagent alors cet espace et chacun devient responsable de l'une des divisions.

Il existe également un autre type de DHT avec une topologie en arbre binaire tel que c'est le cas avec Kademia [26]. Cette DHT se base sur la notion de distance pour le routage de requêtes sur le réseau. Chaque nœud dispose d'un identifiant unique sur 160 bits. L'utilisation d'une fonction XOR pour le calcul de la distance entre deux nœuds permet d'ordonner le réseau sous la forme d'un arbre binaire. Lorsque l'on insère ou qu'on cherche une donnée dans Kademia, le protocole choisit le nœud dont l'identifiant a le préfixe commun le plus long avec l'identifiant de la donnée. Pour le routage des messages, chaque nœud maintient une liste de voisins nommée  $k$ -bucket. La propriété de cette liste est que l'identifiant stocké dans

l'entrée  $i$  du  $k$ -buket, partage un préfixe de taille  $i - 1$  avec l'identifiant du nœud local.

Ces systèmes P2P permettent gestion performante du stockage de grands volumes de données. Ils permettent, en un nombre réduit de messages de recherche, de localiser les données et les nœuds du système qui en sont responsables. Ces mécanismes ont d'ailleurs été adoptés par des systèmes de partage de fichiers [2], des systèmes de fichiers distribués [27], des systèmes de stockage distribués [28, 29], diffusion de données [30], etc.

### II.1.3 Systèmes de fichiers distribués

Des systèmes de fichiers distribués comme *Google File System* (GFS) [16], *Hadoop Distributed File System* HDFS [17] ou *OceanStore* [18] ont pour but d'offrir un service de stockage de fichiers performant, pérenne et disponible. La réplication des données sur plusieurs nœuds du réseau augmente le nombre de sources permettant un accès performant et une reconstruction fiable lorsqu'un nœud du réseau est défaillant. GFS [16], est le système de fichier développé et utilisé par Google pour les différents services qu'ils proposent notamment Google Drive. Comme l'illustre la figure II.5, l'architecture de GFS consiste en 3 tiers : les clients, le serveur principal (*master*) et les serveurs de chunks (*chunkserver*).

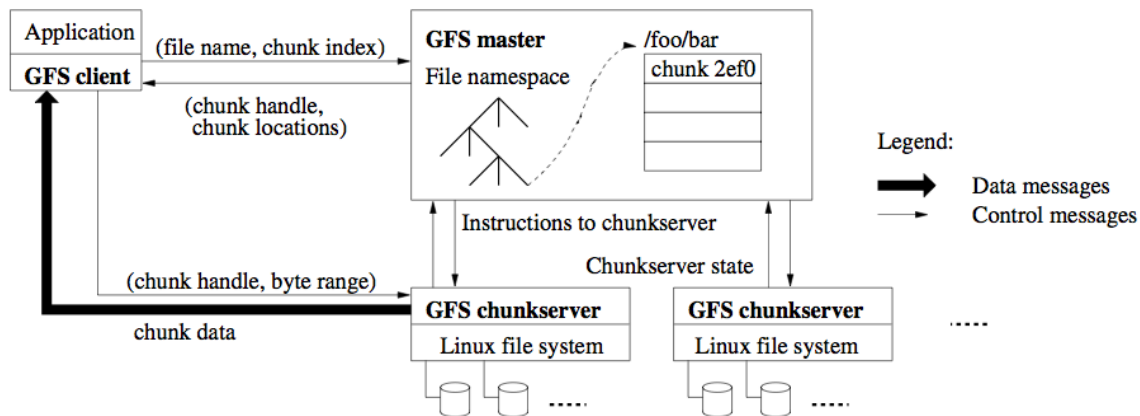


FIGURE II.5: Architecture de GFS (crédits Ghemawat et al. [16]).

Le serveur principal a pour rôle de stocker les méta-données qui permettent de localiser les données stockées sur les serveurs de chunks. Les serveurs de chunks stockent les copies des différents chunks sur le réseau. Dans ce système de fichiers prévu pour le stockage de fichiers volumineux (jusqu'à plusieurs To), la taille des chunks est de 64 Mo, elle permet de réduire la sollicitation du serveur principal pour le téléchargement d'un fichier. Pour permettre la reconstruction de la donnée sur le réseau, chaque chunk a trois copies sur 3 serveurs de chunks différents. Ainsi, lorsque le client accède à un fichier, il interroge le serveur principal qui lui retourne

l'adresse du chunk à télécharger. Le client peut alors contacter directement le serveur de chunks en charge de l'accès à ce dernier. HDFS [17] est utilisé par le moteur Hadoop [31] pour la répllication et le placement efficace des données utilisées dans les applications de calcul distribué. L'architecture de ce système de fichiers a une analogie avec GFS. Cependant, l'implémentation de HDFS est plus orienté pour permettre des calculs performants sur des grands volumes de données que sur du stockage distribué.

D'autres systèmes de fichiers P2P comme PAST [27] ou IPFS [29] reposent respectivement sur Pastry [20] et Kademlia [26] afin de proposer des systèmes de fichiers performants, robustes et capables de résister au churn sur le réseau (départs et arrivés de nœuds).

### II.1.4 Synthèse

Les approches de distribution et de partage de données que nous venons de présenter sont prévues pour être implémentées sur des serveurs qui communiquent via des réseaux filaires. Bien que ces solutions soient performantes et que leurs avantages soient considérables, elles ne sont pas adaptées aux réseaux sans fil. En effet, si l'on les applique telles quelles, elles ne passeraient pas à l'échelle. Pour maintenir l'état du réseau sur ce type de solution, les périphériques mobiles échangeraient une grande quantité de messages de contrôle. Cela provoquerait une sollicitation importante des infrastructures de communication et donc plus de chances de créer des interférences entre les périphériques mobiles, ce qui pourrait conduire à une dégradation des performances du réseau. Ces interférences peuvent être provoqués par la présence d'autres réseaux adjacents qui utilisent les mêmes fréquences radio pour communiquer ou par des périphériques qui partagent la même ressource réseau (fréquences, point d'accès, etc.).

De nombreuses approches existent pour traiter les problèmes de passage à l'échelle au niveau des serveurs applicatifs (duplication de serveurs, répartition de charge, réseaux P2P, ...). La mise en œuvre de mécanismes et de systèmes pour le passage à l'échelle des réseaux mobiles nécessitent, en plus, de prendre en compte plusieurs facteurs liés au médium de communication (interférences, allocation de fréquences, gestion de la puissance du signal, etc.).

## II.2 Partage et diffusion de données sur les réseaux mobiles

Les réseaux mobiles occupent une place importante dans le réseau Internet actuel, le trafic qui y transite est en croissance permanente. De plus en plus de contenus multimédias sont accédés par des périphériques mobiles. Même si ces périphériques sont capables de stocker des volumes de données de plus en plus importants et de les acheminer avec des débits en constante croissance, la performance de leur accès aux

données reste conditionné par la qualité de service qu'offre l'infrastructure qu'ils utilisent.

Les réseaux mobiles adhoc (MANETs ou *Mobile Adhoc Networks*) et les réseaux mobile-à-mobile (D2D ou *device-to-device*) sont deux types de réseaux P2P mobiles qui exploitent les capacités de communication directe des périphériques mobiles pour créer des réseaux P2P. D'un point de vue architectural, ces deux types de réseaux mobiles peuvent sembler identiques. Pourtant, ces réseaux ne bénéficient pas des mêmes mécanismes de fonctionnement et n'ont ni les mêmes contraintes, ni les mêmes cadres applicatifs. Les MANETs [32] ont pour objectif d'assurer la diffusion d'information dans un contexte dynamique (apparitions et disparitions de nœuds), où les ressources sont très limitées (bande passante et autonomie de batteries) pour les périphériques (stockage et communication) rendant la qualité de service (QoS) difficile à garantir, en particulier à cause du fonctionnement autonome des nœuds. Les technologies D2D ont vocation à être contrôlées par l'infrastructure. Cela permet de mieux visualiser l'état du réseau. Ce qui peut permettre une utilisation plus efficace des ressources et donc des communications plus performantes.

### II.2.1 Réseaux mobiles adhoc (MANETs)

Avec l'apparition des premiers périphériques mobiles, de nombreux travaux se sont intéressés à l'implémentation des paradigmes P2P sur des réseaux sans fil afin de mettre en place des systèmes mobiles autonomes capables de communiquer en réseau sans utiliser d'infrastructure. Les réseaux mobiles adhoc (MANET) se définissent par leur caractère très dynamique (beaucoup de départs/arrivés de nœuds), leurs ressources limitées (stockage, énergie) ainsi que leurs communications en multi-saut (portée réduite). Comme pour les réseaux de capteurs, la topologie du réseau est maillée, les périphériques se transmettent des messages les uns aux autres en passant par leur voisins.

Le partage de fichiers, comme pour les réseaux P2P, a été traité dans de nombreux travaux. Liu et al. ont publié une étude [33] dans laquelle ils ont classé différentes approches de partage de fichier sur les MANET. De la même manière qu'un réseau P2P de partage de fichier classique, le partage de fichiers a besoin de la localisation des données sur le réseau. Parmi les nombreuses familles d'approches, nous citons les approches suivantes :

- **Approches par DHT** : DPSR [34], Ekta [35] et LocP2P [36] sont des approches de partage de fichiers qui reposent sur des DHTs pour l'indexage et le stockage des données sur les différents nœuds du réseau. Leur utilisation peut s'avérer efficace. Cependant, la maintenance de l'état global du réseau sur ce type d'infrastructure est coûteux à cause d'une couche protocolaire lourde, ce qui provoque un trafic de messages important. De plus, lorsqu'un nœud se déplace, son voisinage peut changer. Dans ce cas, les déplacements des nœuds peuvent rendre le réseau logique inadapté à la nouvelle topologie physique du réseau.

- **Approches par inondation (*flooding*)** : 7DS [37], PDI (*Passive Distributed Indexing*) [38] et ORION [39] reposent sur des mécanismes d'inondation pour la communication et la recherche d'informations sur le réseau. Lorsqu'un nœud recherche une information sur le réseau, il diffuse la demande à tous ses voisins directs. Si l'un de ces nœuds contient la donnée recherchée, il lui répond directement. Sinon, le réseau continue à être inondé de requêtes tant qu'un nœuds contenant l'information n'a pas été trouvé. Une fois la donnée trouvée, le périphérique à l'origine de la recherche récupère l'information lui indiquant le chemin (les sauts sur le réseau) à suivre pour acheminer la donnée. Au niveau applicatif, ces approches maintiennent des tables de routage et d'indexage sur les différents nœuds du réseau.
- **Approches par connexions sociales** : Les réseaux sociaux ont la caractéristique de rassembler un ensemble d'individus autour d'un même intérêt. Cet intérêt peut être exploité par certaines approches de partage de fichiers comme MOPS [40], SocialCast [41], ContentPlace [42] ou SPOON [43] qui exploitent les relations sociales entre les utilisateurs de périphériques mobiles pour adapter la diffusion et la récupération de contenu en fonction de leurs intérêts et les communautés auxquelles ils appartiennent.

Les approches de partage de fichiers que nous venons de décrire peuvent s'avérer utiles pour la diffusion et le partage de données dans des réseaux MANET. Lorsque la mobilité des nœuds sur le réseau est forte, l'utilisation d'approches nécessitant moins de contrôle pour la diffusion de données peut permettre de disséminer plus rapidement grâce à un nombre réduit de messages de contrôle. Nous citons parmi les approches existantes les familles suivantes :

- **Approches par diffusion épidémique** : Un nœud réplique ses données de manière épidémique en envoyant une copie à tous ses voisins. Plusieurs approches utilisent ce mécanisme de réplication de données dans les réseaux mobiles. *Epidemic Routing* [44] utilise ce modèle de transmission de données sur le réseau. Lorsque le réseau n'est pas congestionné, l'utilisation de ce type d'approches permet de diffuser des données sur le réseau de manière efficace et rapide. Cependant, lorsque le nombre de nœuds est trop important, le nombre de messages émis sur le engendre une perte de performances due à la saturation de la bande passante. D'autres mécanismes, comme RAPID [45] ou PREP [46], adaptent la propagation de la donnée en fonction de leur priorité ou de leur utilité sur le réseau. Cela permet de réduire le nombre de messages diffusés. Des approches à base de compteurs comme *Spray and Wait* [47] et *Spray and Focus* [48] permettent aussi de réduire le nombre de messages diffusés sur le réseau en utilisant des compteurs qui limitent le nombre de nœuds qui gardent une copie de la donnée.
- **Approches opportunistes** : Lorsqu'un utilisateur de périphérique mobile se déplace, le voisinage de son périphérique mobile évolue. C'est cet aspect là qu'exploitent les réseaux mobiles opportunistes pour diffuser l'information. OpenGarden (FireChat) [49], Scampi [50], Haggle [51], N4C [52], ou C3PO [53] proposent des mécanismes de communication opportunistes prévus pour des

réseaux tolérants aux délais (DTN ou Delay Tolerant Networks). L'utilisation de cette famille d'approche peut s'avérer très utile pour faire communiquer des périphériques mobiles lorsque les infrastructures de communication classiques s'effondrent. Un exemple connu d'application pour grand public est FireChat [49], elle permet, par le biais de connexions P2P entre des périphériques géographiquement proches, des communications entre les utilisateurs en utilisant le moins d'infrastructure réseau possible. C3PO [53] exploite la capacité des périphériques à communiquer en adhoc afin de propager rapidement des informations dans des réseaux à connexions intermittentes. Le but de C3PO est d'exploiter de manière opportunistes tous les moyens de communication possibles dans un réseau de type MANET qui se forme de manière spontanée et opportuniste comme cela peut être le cas lors d'un marathon par exemple. La performance de ces réseaux repose sur leur capacité à faire transmettre des données à travers le réseau avec le moins de ressources possibles.

### Limites des réseaux MANETs pour la dissémination de données

Les approches que nous venons de présenter proposent différentes manières de partager et de diffuser des données dans des réseaux mobiles dynamiques tels que les MANETs. Leur principal avantage est de permettre à un réseau de communiquer sans nécessiter d'infrastructure. L'utilisation de structures de données pour distribuées pour localiser les données disponibles sur le réseau telles que les DHTs peut être utile pour la dissémination de données sur les réseaux Wi-Fi, en particulier lorsque certains périphériques ne sont pas à la portée d'un point d'accès. Elles nécessiteraient de mettre en œuvre des politiques de propagation d'informations sur le réseau en implémentant des protocoles de routage propres ce type de réseau (par exemple : AODV [54], OLSR [55]).

Cependant, maintenir une couche de contrôle permettant d'assurer la propagation d'informations de bout en bout et de maintenir l'état global du réseau nécessite l'échange d'un nombre important de messages de contrôle. Cela peut provoquer des interférences entre les différents périphériques mobiles qui constituent le réseau. De plus, comme les déplacements de périphériques sont imprévisibles, la transmission des données est incertaine. Cela représente un frein à la performance de la dissémination de données. Un changement de topologie physique peut rendre le réseau logique (overlay) inefficace. De plus, l'implémentation de mécanismes de sécurité de communication peut s'avérer complexe à mettre en œuvre à cause de ce mode fonctionnement des périphériques. L'utilisation ou l'adaptation d'approches basées sur des réseaux MANET pour la dissémination de données sur des réseaux Wi-Fi est donc peu adaptée.

### II.2.2 Diffusion de données sur les réseaux D2D

De plus en plus d'équipements réseaux (points d'accès, routeurs, etc.) intègrent des couches applicatives rendant leur utilisation et leur configuration plus simples



pour les utilisateurs ainsi que des API ouvertes pour les développeurs. Cette évolution permet de mettre en place des architectures hybrides (semi-centralisées) qui utilisent l'infrastructure des réseaux sans-fil afin de d'en mesurer l'état et de mieux contrôler les communications D2D. Les systèmes qui fonctionnent de cette manière peuvent ainsi de mieux bénéficier des capacités des communications qu'ils utilisent.

D'après un rapport publié par Cisco [1], le trafic internet mondial, hors objets connectés, avoisinera les 50 Exaoctets<sup>1</sup> par mois en 2021, dont 86% de trafic mobile. Avec la croissance que l'on observe avec les technologies mobiles, les réseaux cellulaires actuels ne pourront absorber la charge de ce trafic, et ce même si les couches physique des réseaux actuels sont de plus en plus efficaces. L'attention de la recherche se tourne depuis quelques années vers les réseaux de télécommunications pour l'après 4G : la 5G. La connectivité, la qualité d'expérience (QoE) perçue ainsi que l'efficacité énergétique représentent les challenges auxquels les communications sans fil font face.

Aujourd'hui, de nombreux services sont basés sur la localisation et la proximité des utilisateurs, la sollicitation de la collaboration entre dispositifs mobiles se retrouvant à proximité les uns des autres est un des leviers d'amélioration des performances des réseaux de télécommunication. En effet, l'émergence de technologies de communication D2D telles que Bluetooth, Wi-Fi Direct, Zigbee et LTE-D2D offrant la possibilité de créer, de manière contrôlée ou autonome, des réseaux indépendants des infrastructures classiques (stations cellulaires, points d'accès, etc.) permettent de nouveaux usages des communications D2D. Cela a fait l'objet de nombreux travaux sur des problématiques telles que le déchargement d'infrastructures [56, 7, 57, 58], la dissémination de données [59, 60, 61, 62, 63], la gestion des ressources radio [64] ou les caches distribués [65, 66].

Les technologies D2D peuvent être mises en œuvre de deux manières différentes. La première, dans laquelle les communications D2D et les communication d'infrastructure partagent le même medium de communication, vise à améliorer l'utilisation du support de communication des périphériques. La deuxième, dans laquelle les communications D2D ne partagent pas les mêmes bandes de fréquence, ont pour but de tirer profit de l'utilisation simultanée de plusieurs technologies de communication sur le réseau pour offrir la meilleure QoS possible aux utilisateurs. La littérature sur les réseaux D2D traite principalement des réseaux cellulaires. Asadi et al [6] ont d'ailleurs publié un rapport dans lequel ils ont établi une taxonomie et un classement de plusieurs contributions qui utilisent les communications D2D dans les réseaux cellulaires. Les bénéfices pouvant être tirés de l'utilisation de cette technologie sont nombreux : amélioration de la couverture, optimisation du signal, réduction du coût énergétique, contrôle des interférences, etc.

---

1. 1 Exaoctet =  $2^{20}$  Téraoctets

### II.2.2.1 Communications D2D en utilisant la même technologie

Lorsque le support de communication est partagé par le trafic D2D et le celui de l'infrastructure, ce partage peut se faire de deux manières différentes : les communications des deux modes de communication partagent les mêmes fréquences, ou alors chaque type de communication dispose de ses fréquences dédiées. La littérature utilisant ce mode de fonctionnement traite de problématiques liées à l'utilisation des mêmes bandes de fréquence notamment l'efficacité du spectre [67, 68, 69, 70, 71, 72], la consommation énergétique [73, 74, 75], l'optimisation par contraintes (énergie ou QoS) [76, 77], couverture [78], fiabilité [79, 80].

L'objectif de ces contributions est d'améliorer l'efficacité des infrastructures, de réduire les effets des interférences ou d'adapter le fonctionnement du système à l'état du réseau. Avec un contrôle des communications par l'infrastructure, les réseaux pourront bénéficier de meilleurs débits, d'une plus grande capacité de prise en charge de clients, de plus de performances de communications D2D, etc. Ce mode de fonctionnement a pour avantage d'avoir un meilleur contrôle sur les communications D2D et cellulaires.

L'étude et l'utilisation de ces mécanismes est utile pour mieux utiliser les ressources radio des réseaux mobiles (cellulaires et autres). Cependant, la contribution principale de cette thèse ayant une approche de haut niveau, nous ne détaillerons pas plus cette partie. Elle est à mettre en relation avec le chapitre VI qui traite cas de l'utilisation de topologies hiérarchiques pour la dissémination de données en Wi-Fi Direct. A travers une approche basée sur le regroupement (clustering) de périphériques, nous y abordons l'importance de la partition du réseau et de l'impact qu'elle peut avoir sur les interférences et sur les performances de la dissémination de données.

### II.2.2.2 Communications D2D utilisant plusieurs technologies D2D

Avec l'émergence d'applications gourmandes en bande passante, l'utilisation simultanée de plusieurs technologies D2D offre la possibilité de bénéficier d'une plus grande capacité pour le transfert des données. Ce mode de fonctionnement permet de ne pas avoir de problèmes d'interférences entre le trafic cellulaire et le trafic D2D. Cependant, cela nécessite d'utiliser deux interfaces de communication, utilisant souvent d'autres technologies. D'ailleurs, le débit et la portée qu'offrent les réseaux Wi-Fi en ont fait un candidat intéressant pour le déchargement d'infrastructures et l'amélioration de performances du réseau [81, 82, 83].

Afin de tirer profit de meilleures performances possibles des communications cellulaires et Wi-Fi, Asadi et al. [81] et [82] se basent sur la formation de clusters D2D dans la portée Wi-Fi d'un périphérique pour améliorer le débit cellulaire et l'efficacité énergétique. Leur approche consiste à favoriser les périphériques dont le débit cellulaire est le plus performant pour qu'ils servent de relai aux autres périphériques du réseau. Dans leur analyse, les auteurs ont démontré que cette

approche permettait d'améliorer l'utilisation des fréquences cellulaires et d'avoir une meilleure qualité du signal, d'améliorer l'usage de l'énergie des périphériques clients qui mettaient leur interface cellulaire dans le mode "sauvegarde d'énergie" et de répartir équitablement le signal sur les différents clusters, permettant aux équipements avec une faible qualité de signal d'accéder au réseau. Leurs études ont montré que le débit du réseau pouvait être amélioré respectivement jusqu'à 30% et 50%. Dans la même optique, pour faire face aux problèmes liés aux interférences causées par la contention d'un canal Wi-Fi. Spinella et al. [83] proposent de répartir les utilisateurs D2D d'un réseau en groupes en fonction de leurs besoins en qualité de services et n'autorisent qu'un seul utilisateur à utiliser le canal de communication à la fois. Cette approche leur a permis d'augmenter le débit des connexions D2D de 25%.

Toujours dans l'optique de réduire la charge sur l'infrastructure et d'accélérer l'accès aux données, les réseaux D2D peuvent être utilisés pour mettre en place des caches distribués [65, 84, 62]. Pan et al. [65] ont pointé l'inefficacité de la diffusion (broadcast) de données sur les réseaux Wi-Fi Direct et ont proposé des architectures basées sur des structures en anneaux ou arborescentes qui pouvaient permettre une diffusion plus efficace des données dans le cas de réseaux Wi-Fi Direct multi-groupes. Golrezaei et al. [84], [85] se concentrent sur le cas de contenus vidéos populaires dans les réseaux cellulaires. Leur approche consiste à découper une cellule<sup>2</sup> en plusieurs sous-cellules (clusters) et de mettre en cache au niveau des clusters pour récupérer les données. Lorsque le contenu est déjà présent dans le cluster d'un périphérique qui le demande, le serveur lui envoie l'instruction de le télécharger depuis un voisin avec lequel il établit une connexion D2D pour récupérer le contenu auquel il essaie d'accéder en utilisant une bande de fréquence différente de celle du réseau cellulaire. Sinon, le périphérique télécharge à partir du point d'accès et devient détenteur du contenu sur le réseau. Il peut ainsi délivrer cette donnée au prochain périphérique du même cluster qui la demande. Les technologies utilisant les communications D2D facilitent la mise en place de disséminations de données distribuées avec un contrôle centralisé au niveau de l'infrastructure comme c'est le cas de Thava et al. [86], qui proposent un système de partage D2D basé sur l'établissement d'une cartographie des connexions et des contenus de chaque périphérique ainsi que le type d'interface réseau pouvant être utilisé pour communiquer avec ses voisins. Ces approches ont des points de similitude avec notre problématique. Notre cas a la particularité de n'utiliser qu'une seule interface réseau (Wi-Fi) et les périphériques mobiles sont localisés au même endroit et à portée de signal de leurs voisins.

Plus proches de nos travaux, MicroCast [61] et MoVi/MoVi+ [59, 60] utilisent simultanément le lien avec le serveur et des communications directes avec les périphériques mobiles pour augmenter le débit du réseau et améliorer l'efficacité de dissémination de données pour le streaming vidéo.

---

2. Une cellule ici correspond à une antenne et aux périphériques mobiles situés dans sa zone de couverture.

Keller et al. [61] proposent un système de streaming vidéo collaboratif “*Micro-Cast*”. Ce système vise à accélérer la récupération d’un même contenu vidéo par un ensemble de périphériques mobiles à proximité les uns des autres. Le contenu en question est initialement hébergé sur le serveur, qui est l’unique source de données du réseau. Afin de favoriser la collaboration des périphériques mobiles, la donnée est découpée en morceaux de petite taille (chunks).

Pour ne pas créer de contention sur l’infrastructure réseau, les périphériques se coordonnent entre eux pour le choix des morceaux à télécharger et du périphérique qui télécharge. Chaque périphérique se fait attribuer une liste de chunks à télécharger et c’est celui qui a le moins d’éléments dans sa liste qui est favorisé.

Le partage des données entre les périphériques du réseau se fait avec une pseudo-diffusion (“pseudo-broadcast”) via Wi-Fi. Lorsqu’un émetteur envoie des données vers un périphérique avec lequel il a établi une connexion, les périphériques voisins écoutent les paquets transmis pour reconstituer la donnée disséminée.

Dans un réseau de petite taille, cette approche peut s’avérer efficace. En plus de réduire le temps de téléchargement de contenus depuis le serveur, le trafic du aux messages de contrôle est réduit sur le réseau. Lorsque le réseau est dense, l’implémentation d’une telle approche reviendrait à avoir une multitude de groupes de périphériques qui reproduiraient le même comportement. Cela peut engendrer rapidement des interférences entre les différentes périphériques mobiles et causer la détérioration des performances du téléchargement à cause les interférences Wi-Fi.

Toujours dans le cadre du streaming collaboratif sur périphériques mobiles, MoVi/MoVi+ [59, 60] utilise un mécanisme de communication permettant aux périphériques de communiquer en utilisant des connexions adhoc opportunistes déclenchées dès que les conditions le permettent (lorsque la qualité du signal permet d’acheminer à un débit acceptable) Les périphériques mobiles sont capables de détecter leurs voisins lorsqu’ils se déplacent et d’évaluer la qualité du signal environnant qui fluctue tout le long. En mesurant la qualité du signal et le niveau d’interférences avec ses voisins, un périphérique est capable d’évaluer le débit avec lequel il peut échanger avec lui. Leurs simulations ont permis de montrer que l’utilisation d’une telle solution permet d’améliorer la performance de la dissémination et de réduire le temps de début de lecture par rapport à une approche traditionnelle de streaming. L’exploitation opportuniste du niveau d’interférences peut permettre de créer des connexions D2D entre des périphériques permettant ainsi de réduire la charge sur le serveur et les infrastructures réseaux. L’exploitation opportuniste des fluctuations des interférences entre les périphériques peut être utile pour la dissémination de données sur les réseaux Wi-Fi. Ce type d’approche pourrait s’appliquer à notre cas. Cependant, la réplique dans un service de streaming collaboratif impose un ordre sur l’accès aux chunks de la donnée elle peut être limitante en terme de parallélisme et donc de débit pour la diffusion des données. D’ailleurs, nous n’avons pas cette contrainte dans notre cas.

Il est nécessaire dans un réseau Wi-Fi de pouvoir permettre aux périphériques de

se connecter rapidement et qu'ils puissent bénéficier de canaux qui offrent de bonnes performances. La connaissance de l'instant d'arrivée d'un périphérique sur le réseau peut être utile pour cela. Lee et al. [64] essaient de permettre aux périphériques qui arrivent sur un réseaux D2D d'obtenir les canaux de communication offrant les meilleurs performances possible. Ils proposent pour cela un algorithme d'affectation dynamique de canaux. Ils arrivent à obtenir, avec leur approche, des performances proches des cas optimaux que l'on peut avoir avec des algorithmes ayant une connaissance préalable des moments d'arrivée des périphériques.

### II.2.3 Consommation d'énergie sur les réseaux D2D

Avec la richesse de l'écosystème mobile actuel, il y a une grande hétérogénéité des périphériques, notamment sur les aspects performance réseau, calcul et autonomie des batteries. La consommation d'énergie est un aspect qui est au cœur des réseaux mobiles et les réseaux D2D ne font pas exception. Exploiter les connexions D2D permet d'améliorer les performances du réseau mais leur coût énergétique ne doit pas être négligé pour autant. Par exemple, lorsque certains périphériques peuvent être choisis comme relais ou sélectionnés plusieurs fois pour transmettre des données, ils consomment plus d'énergie que leurs voisins car ils sont plus sollicités.

À notre connaissance, les contributions autour des technologies D2D traitant de l'aspect énergétique [87, 88, 89, 90] ont pour but de réduire la consommation globale et ne prennent pas en compte l'hétérogénéité des batteries dans un réseau mobile. Nobach et al. [90] ont traité le problème de sécurité et d'énergie que l'on peut avoir avec la communication D2D lorsque les périphériques scannent activement le réseau à la découverte d'autres périphériques. Wang et al. [88] proposent une solution pour l'allocation de ressources (canaux de communication) prenant en compte l'énergie des périphériques. Ils modélisent l'allocation de ressource comme un jeu de compétition entre des périphériques demandant la même ressource. Sung et al. [87] traitent ce problème comme un équilibre à maintenir entre une autonomie plus longue pour la diffusion des donnée et la performance de diffusion de données. Ils ont proposé deux algorithmes de routage de requêtes et d'ordonnancement de paquets afin de maintenir cet équilibre.

De manière générale, les travaux que nous avons étudié visent une économie d'énergie globale. Nous pensons qu'il est important de prendre en compte les différents niveaux de batterie, qu'il est préférable de consommer globalement plus, voir de perdre en performance, si cela permet d'épargner la consommation sur les périphériques ayant une batterie faible. Ainsi, tous les utilisateurs pourront bénéficier des données reçues. Dans un réseau hétérogène, traiter tous les périphériques de manière équivalente peut conduire à une sur-consommation d'énergie sur certains périphériques.

## II.2.4 Synthèse

Les contributions autour des technologies de communication D2D ont pour but d'améliorer les performances des réseaux notamment les réseaux cellulaires (amélioration de la couverture, consommation d'énergie, équité de l'accès aux ressources, le déchargement des infrastructures, etc.), et ce en exploitant la capacité des périphériques mobiles à établir des communications directes avec leurs voisins. Il existe deux manières d'utiliser les communications D2D : la première a pour but permettre une meilleure utilisation des ressources réseaux (infrastructure et D2D). La deuxième vise à utiliser plusieurs technologies de communication D2D pour offrir la meilleure qualité de service possible aux utilisateurs. L'utilisation de connexions D2D a pour avantage d'améliorer les performances d'un réseau cellulaire en utilisant d'autres interfaces de communication. Les réseaux Wi-Fi ont été principalement utilisés en parallèle de connexions cellulaires afin d'améliorer les débits et les performances globales d'un ensemble de périphérique à proximité les uns des autres.

L'écosystème mobile se caractérise par une considérable hétérogénéité des périphériques. Leurs caractéristiques varient en fonction de leur marque et leur modèle. La consommation d'énergie dans les réseaux D2D est souvent traitée dans le but de réduire la consommation d'énergie globale d'un réseau. À ce jour, nous n'avons pas pris connaissance d'une approche prenant en compte l'hétérogénéité des périphériques mobiles en termes de batteries. Avec l'hétérogénéité des périphériques mobiles sur le marché, considérer les périphériques de manière semblable sur le réseau peut conduire à la réduction accélérée de l'autonomie de la batterie de certains périphérique ayant une batterie faible.

Il y a des cadres applicatifs pour lesquels il est important d'avoir l'autonomie la plus grande possible afin que chaque périphérique puisse être utilisé le plus longtemps possible comme cela peut être le cas dans le cadre de conférences ou dans des cours magistraux. Dans ces cas, une consommation d'énergie efficace réside dans une répartition équitable<sup>3</sup> de la consommation énergétique, quitte à ce que certains périphériques consomment plus que d'autres. Bien entendu, ce type de démarche peut se faire plus facilement dans un cadre où les périphériques sont maîtrisés. Dans le cas contraire, l'acceptation d'une consommation d'énergie supplémentaire pour préserver des périphériques à batteries faibles relève plus de l'altruisme des utilisateurs.

## II.3 Zoom sur les réseaux Wi-Fi

Wi-Fi (*Wireless Fidelity*) est une technologie de communication sans fil régie par la norme IEEE 802.11. Depuis son apparition en 1997, elle permet la création de réseaux locaux sans fil (WLAN). Aujourd'hui, elle fait partie des technologies de communication sans fil les plus répandues. Les évolutions de cette technologie sont

---

3. On demande plus aux périphériques qui ont le plus d'autonomie sur le réseau.

sur plusieurs aspects : mécanismes de communication radio, sécurité, fonctionnalités et débits. En parallèle des standards du groupe de travail IEEE 802.11, le groupe Wi-Fi alliance œuvre activement pour la développer et l'ouvrir à de nouveaux usages comme les réseaux D2D et l'internet des objets (IoT). Dans cette section, nous présentons les différents modes de communication en Wi-Fi. Puis, nous discutons des problèmes de performances ainsi que de contributions qui visent à optimiser les performances.

### II.3.1 Modes de diffusion de données en Wi-Fi

Les périphériques mobiles actuels sont équipés d'antennes Wi-Fi [91] et Bluetooth [92] ces deux technologies offrent la possibilité de communiquer en D2D, sans utiliser une infrastructure. La bande passante élevée du Wi-Fi et sa portée en ont fait le candidat idéal dans plusieurs études pour le délestage de données sur les réseaux cellulaires. Nous décrivons ici les différents modes de communication en Wi-Fi.

**Wi-Fi infrastructure (ou managed Wi-Fi) :** il s'agit de l'usage classique que l'on connaît du Wi-Fi. Un point d'accès émet un signal Wi-Fi que les périphériques sont capables de capter. Ce dernier émet des trames balises (*beacons*) permettant de l'identifier le réseau. Cette trame véhicule des informations sur le réseau, notamment l'ESSID (SSID) qui est son nom. Lorsqu'un périphérique se connecte à ce point d'accès, ils forment un *ensemble de services de base* (**Basic Service Set** ou BSS) dont l'identifiant *BSSID* est l'adresse MAC du point d'accès.

Lorsque plusieurs points d'accès sont utilisés dans un réseau Wi-Fi, ces derniers sont inter-connectés grâce à un réseau de distribution et dépendent généralement d'un routeur/contrôleur Wi-Fi, la figure II.6 illustre la topologie d'un tel réseau. Ils forment alors un *ensemble de services étendu* (**Extended Service Set** ou ESS) qui est identifié par l'ESSID. Cette information permet aux périphériques mobiles qui se déplacent dans le même ESS de se déplacer entre les points d'accès de manière transparente. Certains points d'accès sont capables de former un réseau maillé (Mesh) et de ne pas utiliser un réseau de distribution. Ce mode de fonctionnement est plus tolérant aux pannes. Dans ce cas d'utilisation, une partie de la bande passante se charge de maintenir les connections entre les différents points d'accès, ce qui réduit la bande passante utilisée pour l'acheminement des données.

Le mode infrastructure est aujourd'hui le plus utilisé dans les réseaux Wi-Fi. La bande passante dont peuvent bénéficier les périphériques pour l'accès aux données est définie par plusieurs facteurs dont :

- La bande passante du réseau de distribution : qui est définie par la bande passante que les routeurs et les câbles utilisés ;
- La capacité du point d'accès : en fonction de facteurs liés à la fabrication du point d'accès (débit maximal, norme de fabrication, nombre d'antennes, etc.);
- Le débit maximal du périphérique ;

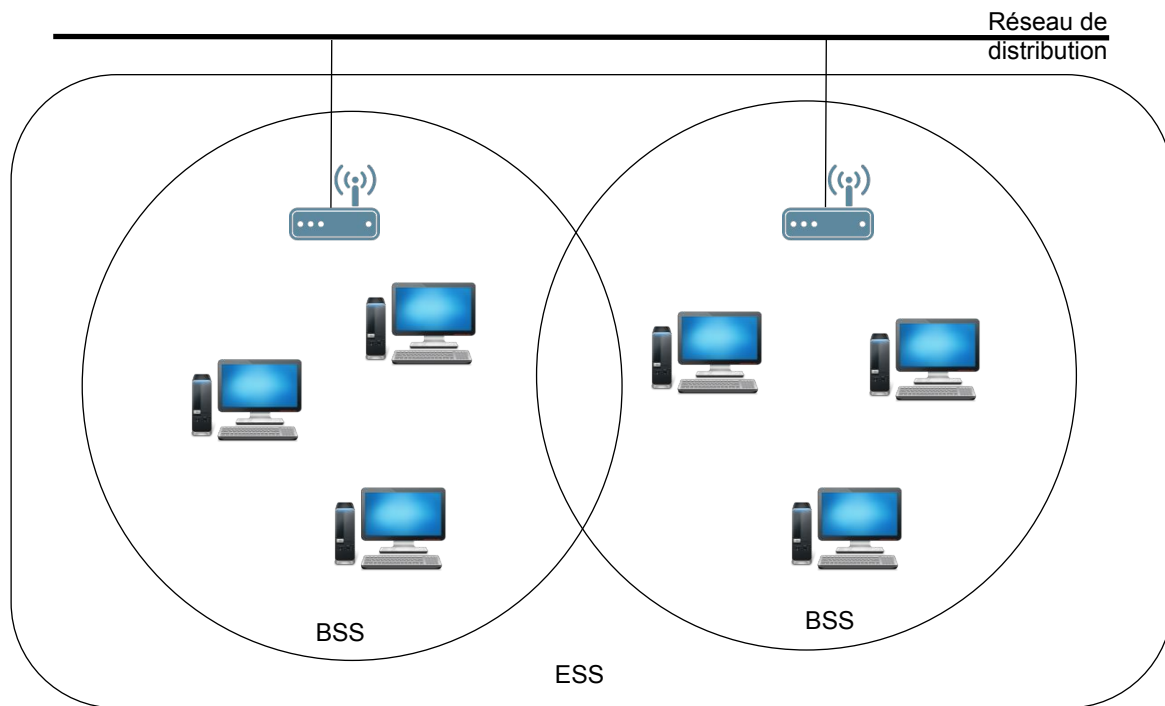


FIGURE II.6: Topologie d'un réseau Wi-Fi à plusieurs points d'accès.

- L'emplacement des périphériques par rapport au point d'accès;
- L'orientation des antennes des points d'accès;
- L'environnement (réseaux adjacents, qualité du signal, etc.);

Lorsque l'on implémente des protocoles P2P (voir section II.1) tels quels sur des réseaux Wi-Fi, une quantité importante de messages de contrôle, en plus des flux de données échangées, transiterait par le point d'accès. Cela peut engendrer un trafic important et réduire le débit utilisé pour la transmission de données vers les périphériques en plus d'accroître les effets des interférences sur le réseau notamment la dégradation des performances.

**Wi-Fi broadcast :** en Wi-Fi, on peut transmettre des données en mode radio. Ce mode de fonctionnement consiste à utiliser le point d'accès (ou l'émetteur) pour émettre constamment des données à diffuser et les récepteurs, en mode "*promiscuité*" (promiscuous) ne font qu'écouter sur leur interface Wi-Fi pour recevoir des données. Ils reçoivent ainsi de manière non filtrée tout le trafic Wi-Fi qui transite sur le réseau, même s'il ne leur est pas adressé. Sans aucune interférence ou avec peu d'interférences, ce mode de communication peut s'avérer efficace pour la transmission de données sur plusieurs périphériques en simultané. Cependant, la reconstitution des paquets perdus nécessiterait une couche protocolaire supplémentaire pour garantir la transmission des données. Keller et al. [61] utilisent en partie le mode



broadcast du Wi-Fi. Ils implémentent une couche protocolaire pour permettre aux périphériques en mode promiscuité de récupérer les paquets émis par leur voisin émetteur. De plus, l'utilisation du mode broadcast sur certains périphériques s'est avérée coûteuse en énergie (leur approche consommait plus de batterie en comparaison du mode infrastructure ou de Bittorrent).

**Wi-Fi Adhoc :** aussi appelé *Independent Basic Service Set (IBSS)* est un mode de fonctionnement du Wi-Fi qui ne nécessite pas d'utiliser des points d'accès pour communiquer. En Wi-Fi adhoc les périphériques communiquent avec les périphériques qui sont à portée de signal. Comme l'illustre la figure II.7, pour communiquer avec un périphérique qui n'est pas à proximité directe, un voisin commun sert de relais pour transmettre des données. D'un point de vue pratique, ce mode peut s'avérer pertinent pour la diffusion de données sur le réseau. Cependant, les problèmes de

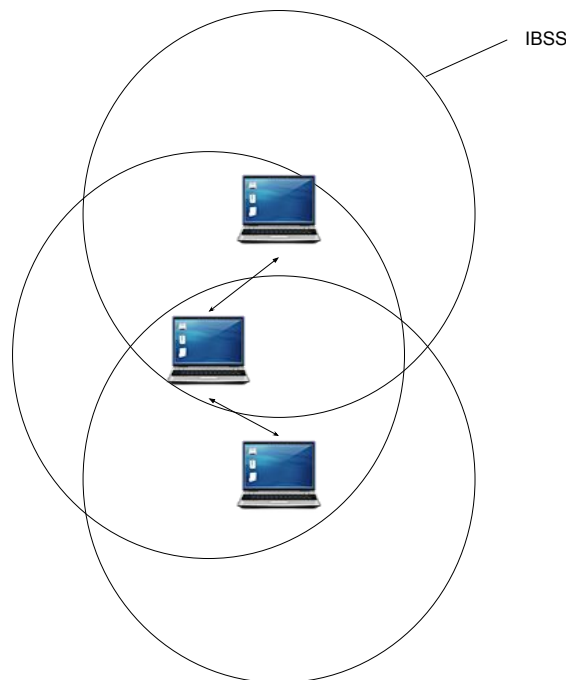


FIGURE II.7: Topologie d'un groupe de communication Wi-Fi adhoc

sécurité et de maîtrise de la puissance du signal ainsi que le débit réduit par rapport à un au mode infrastructure sont des facteurs limitants car ils enlèvent toute maîtrise sur l'usage du réseau. Dans les périphériques présents sur le marché mobile, seul ceux qui ont pour système d'exploitation Android<sup>4</sup> peuvent avoir ce fonctionnement et uniquement avec un accès administrateur (root).

4. [www.android.com](http://www.android.com)

**Wi-Fi Direct ou Wi-Fi P2P :** aussi appelé Wi-Fi P2P, est un standard proposé par Wi-Fi Alliance<sup>5</sup> dédié à la communication P2P en Wi-Fi. Comme le mode adhoc, Wi-Fi Direct ne nécessite pas d'utiliser une infrastructure pour fonctionner.

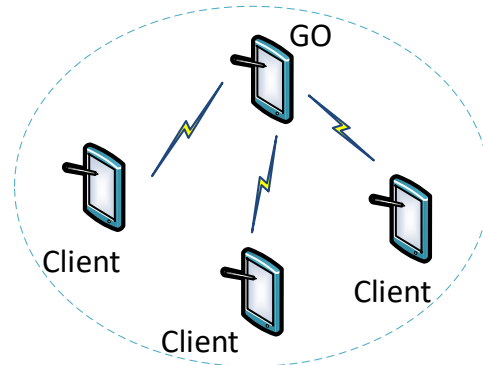


FIGURE II.8: Topologie d'un groupe de communication Wi-Fi Direct.

La différence entre ce mode et le mode infrastructure est que les périphériques établissent une connexion Wi-Fi Direct avant de pouvoir communiquer. Ce mécanisme est appelée formation de groupe. À l'issue de celle-ci, un périphérique est choisi comme *Group Owner* (GO) et joue le rôle de point d'accès. Un périphérique n'ayant pas été élu à l'issue de cette négociation devient alors client du GO qui vient d'être sélectionné. La figure II.8 illustre la topologie que l'on peut avoir dans un réseau Wi-Fi Direct.

Il existe trois manières d'établir une connexion Wi-Fi Direct, elles sont illustrées dans la figure II.9 :

- **Formation de groupe standard :** ce mode de formation de groupe est le mode par défaut. Cet établissement commence par une phase de *découverte*. Les périphériques alternent ici entre des phases d'annonce et de recherche de voisins. Les durées de ces phases sont choisies aléatoirement. Une fois que l'annonce (*probe request*) d'un périphérique est reçue par son interlocuteur, celui-ci lui l'informe en lui envoyant une *Probe response* et les deux périphériques passent ainsi à la phase de *négociation*. Lors de cette phase, le GO est choisi. S'en suivent les phases d'*initialisation sécurisée de connexion* (WPS : Wi-Fi Protected Setup) et de *configuration d'adresses* (IP).
- **Formation de groupe persistante :** lorsque une connexion Wi-Fi Direct entre les périphériques concernés a eu lieu, celle-ci peut être déclarée comme persistante. Les deux périphériques gardent en mémoire cette formation de groupe. À la prochaine connexion entre ces deux périphériques, la phase de *négociation* est remplacée par une *invitation* faite par le GO et la phase d'initialisation de la connexion Wi-Fi est réduite à l'*authentification* du client.
- **Formation autonome :** dans ce cas, un périphérique crée un groupe de communication Wi-Fi Direct dont il est le GO. Comparée aux autres modes de

5. [www.wi-fi.org](http://www.wi-fi.org)

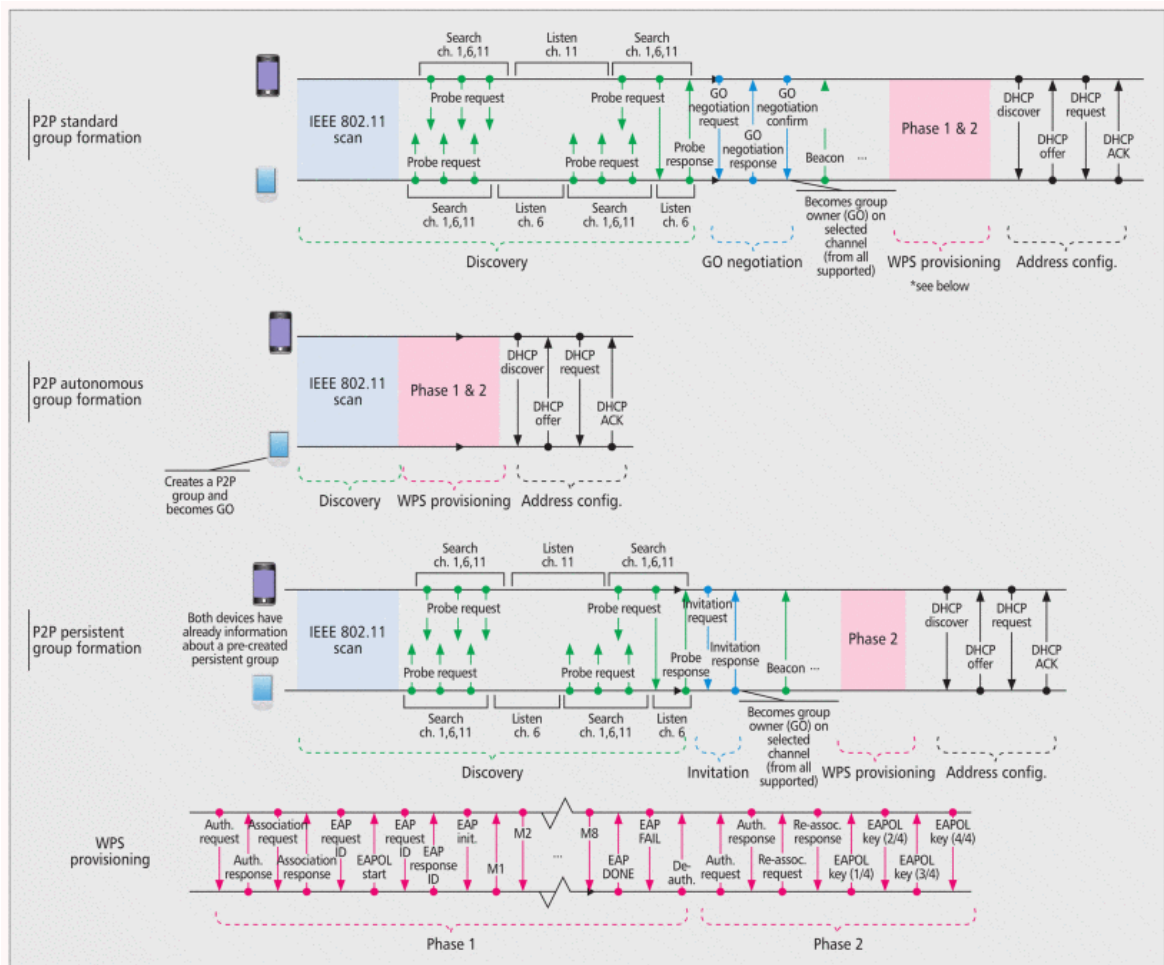


FIGURE II.9: Modes de formation de groupes Wi-Fi Direct (crédits Camps-Mur et al. [93])

formation de groupes, le périphérique qui se connecte à ce groupe ne passe pas par les phase de recherche et de négociation, la découverte se faisant de la même manière qu'une recherche de réseau Wi-Fi classique.

Comparée aux réseaux adhoc, Wi-Fi Direct permet de contrôler la création et la suppression de groupes de communication, ainsi que les mécanismes de sécurisation des communication et d'économie d'énergie. Le temps de formation d'un groupe est généralement en dessous de 10 secondes, les formations autonomes sont les plus rapides (environ 5 secondes en moyenne) grâce à leur nombre réduit d'étapes.

Récemment, de nombreuses recherches se sont intéressées à la technologie Wi-Fi Direct. Khan et al. [94] ont récemment publié un état de l'art complet des travaux de recherches autour du Wi-Fi Direct. Elle a principalement fait l'objet d'études pour l'amélioration de formation de groupes [95, 96], l'économie d'énergie [97, 98] ou la performance [99, 100].

En pratique, certaines contributions peuvent être mises en œuvre afin de simpli-

fier les formations de connexions Wi-Fi Direct ou de permettre à groupe Wi-Fi Direct de bien réagir au départ du GO. Cherif et al. [101] essaient d'améliorer le temps de connexion D2D. Leur approche consiste à élire, dans un groupe Wi-Fi Direct, un GO de secours dont ses voisins vont prendre connaissance afin d'aider à réduire le temps de reformation de groupes lorsque le GO se déconnecte. Dans la même optique, Li et al. [102] proposent d'utiliser des canaux communs pour simplifier la découverte et la création de groupes D2D. Ces mécanismes peuvent s'avérer bénéfiques pour allouer efficacement les ressources réseaux (canaux et fréquences) et pour raccourcir la durée des formations de groupes Wi-Fi Direct.

**Wi-Fi Aware :** Aussi connue sous le nom *Neighbour Awareness Networking* [103], cette technologie a été spécifiée en 2015 par Wi-Fi Alliance et devenue disponible sur les systèmes d'exploitation mobiles Android récents (version 8 et plus). Elle permet de connaître les services proposés sur un réseau ou un mobile voisin sans nécessiter l'établissement d'une connexion. Les délais de formation de groupe Wi-Fi Direct étant longs, l'usage cette de technologie pour découvrir les services proposés sur un réseau en amont d'un éventuel établissement de connexion peut permettre de gagner du temps en économisant des établissements de connexions inutiles.

**MultipeerConnectivity :** MultipeerConnectivity [104] (MC) est la bibliothèque de communication P2P entre les périphériques sous les systèmes d'exploitation d'Apple. Elle permet le développement d'applications qui permettent aux périphériques mobiles géographiquement proches de communiquer en P2P. En fonction des réseaux et interfaces disponibles, MC peut avoir différents modes de fonctionnement. Elle s'appuie sur l'utilisation des interfaces Wi-Fi et Bluetooth pour communiquer. Les trois cas possibles sont :

- Wi-Fi P2P et Bluetooth dans le cas où les deux interfaces sont allumées et que les périphériques ne sont pas connectés au même réseau Wi-Fi.
- Bluetooth ou Wi-Fi P2P uniquement si seulement une des interfaces est allumée.
- L'infrastructure Wi-Fi dans le cas où les périphériques sont connectés au même réseau.

### II.3.2 Réseaux Wi-Fi denses

Les réseaux Wi-Fi sont devenus omniprésents au cours de la dernière décennie, ils sont largement déployés dans les campus, commerces, centres de congrès, etc. La densité des points d'accès a d'ailleurs considérablement augmenté durant ces dernières années. En plus des smartphones et tablettes qui utilisent le Wi-Fi comme moyen de communication, de nouveaux objets connectés utilisent également cette technologie ce qui va accroître la densité des périphériques sur ces réseaux. D'ailleurs, dans les années à venir, le trafic estimé des réseaux Wi-Fi représentera 49% du trafic

internet mondial [1].

### II.3.2.1 Difficulté de passage à l'échelle

La croissance et la densification des réseaux Wi-Fi provoque une baisse de leurs performances. En effet, la qualité de service peut être très rapidement dégradée dans les réseaux Wi-Fi denses. Maity et al. [105] ont montré que les performances des téléchargements TCP étaient dégradées dans des réseaux Wi-Fi denses, même avec seulement 20 à 30 clients. Magency[106] déploie lors d'événements à grande échelle des infrastructures qui permettent aux utilisateurs d'utiliser les applications liées à l'événement auquel ils participent. L'architecture déployée est composée de serveurs, routeurs et points d'accès pour diffuser le réseau Wi-Fi qui sera utilisé pour communiquer. Afin de mieux comprendre comment ce type d'infrastructure réagit à une charge due qui est engendrée par plusieurs clients qui accèdent à une donnée volumineuse en parallèle. Nous avons donc effectué une campagne de mesures de temps de dissémination d'une donnée volumineuse en fonction du nombre de périphériques téléchargeant en parallèle cette donnée. La figure II.10 illustre le temps de dissémination d'un fichier de 100 Mo en fonction du nombre de téléchargements parallèles sur le réseau. Ces mesures ont été réalisées en utilisant des périphériques Apple iPad Mini et un serveur serveur doté d'un processeur 4 cœurs/ 8 threads Intel Core i7 cadencé à 2.2 GHz et d'une mémoire de 8Go. Les données sont accessible via un serveur web nginx [107]. Ce serveur web a été configuré pour utiliser la totalité des ressources CPU dont la machine dispose. Les données en questions ont été chargées sur un RAMDISK afin d'éviter d'avoir les latences liées aux accès au disque. Le point d'accès utilisé est **UniFi UAP AC PRO** de première génération, offrant un débit maximal de 500Mbps en 2.4GHz et 1300Mbps en 5GHz.

La courbe montre que le temps de dissémination total croit linéairement en fonction du nombre de téléchargements. Cette architecture ne passe pas à l'échelle. Nous avons aussi remarqué qu'un ensemble des périphériques réussissaient à récupérer l'information très tôt en comparaison de leurs voisins. Il s'agissait des périphériques placés proches du point d'accès. Kemerlis et al. [108] ont d'ailleurs étudié l'équité du débit de connections TCP en Wi-Fi. Leur étude a montré que lorsque des périphériques avec la même puissance de signal essayaient d'accéder à une donnée via le Wi-Fi, le débit de téléchargement était équitablement répartis. Dans le cas contraire, les périphériques les plus éloignés étaient défavorisés. À cause de leur puissance plus faible, les temps de transferts étaient plus longs sur ces périphériques.

Les équipes d'Alethea [109] ont récemment conduit des tests de performances sur 4 modèles de points d'accès prévus pour des réseaux Wi-Fi denses : UniFi **UAP-AC-HD**[110], **Ruckus R710**[111], Aruba **IAP-325-US** [112] et Meraki **MR52** [113]. Ces points d'accès sont tous compatibles avec la norme IEEE 802.11ac de deuxième génération, ils sont tous capables de supporter plus de 500 utilisateurs en parallèle et fournissent un débit maximal pouvant atteindre 800Mbps sur la

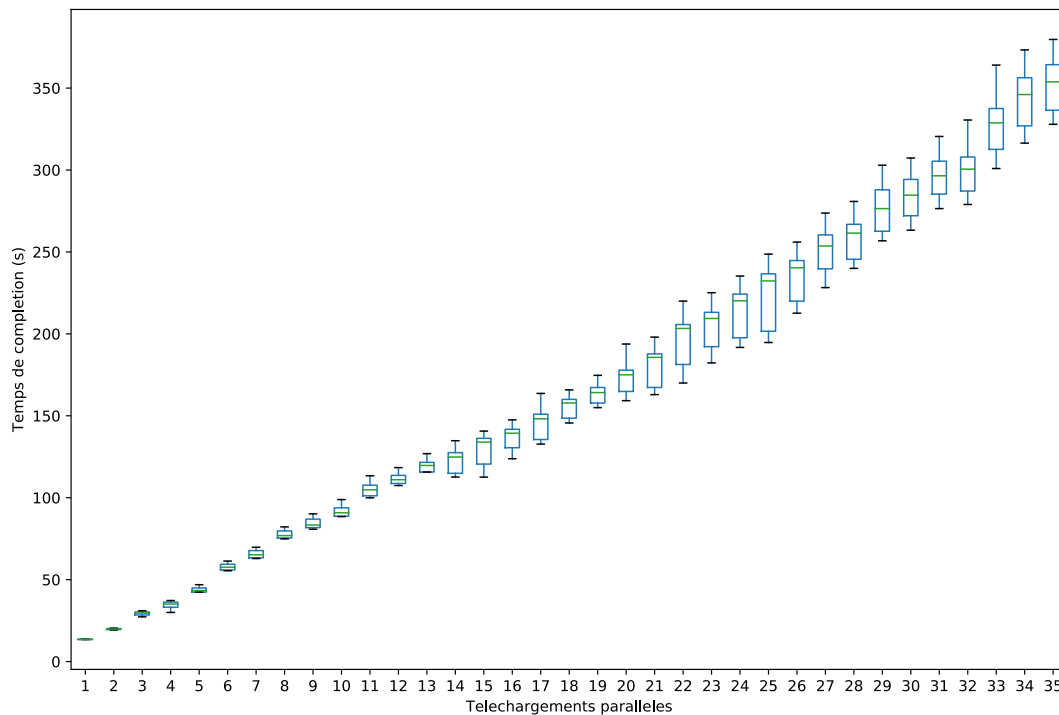


FIGURE II.10: Temps de complétion pour un fichier de 100 Mo en fonction du nombre de téléchargements parallèles sur une infrastructure Wi-Fi.

bande de fréquence 2.4GHz et 1733Mbps sur la bande de fréquence 5GHz. Les tests ont été conduits dans une salle de cours dans laquelle un seul point d'accès est utilisé.

Pour les clients, 100 ordinateurs portables ont été utilisés pour faire ces tests. Les points d'accès ont été configurés en choisissant les canaux ayant la meilleure qualité du signal. Les points d'accès ont été évalués sur les débits maximaux qu'ils étaient capables d'atteindre ainsi que l'expérience de l'utilisateur pendant la lecture de vidéos de haute définition diffusée en streaming sur cette infrastructure. Le débit total du réseau en téléchargement dépasse à peine 300 Mbps pour les meilleurs points d'accès, et ce en utilisant simultanément les deux bandes de fréquence 2.4GHz et 5GHz. En ce qui concerne la qualité d'expérience pour la diffusion vidéo, plus le nombre de clients est grands, plus la qualité d'expérience globale est réduite, de qualité moyenne voir faible pour certains (exemple : perte de qualité d'image en cours de lecture), mauvaise pour d'autres (ex : qualité vidéo réduite, lecture saccadée).

### II.3.2.2 Optimisation des performances des réseaux Wi-Fi

Pour palier les problèmes de performances des réseaux Wi-Fi, trois directions ont été principalement proposées dans la littérature : communications D2D, contrôle

de la puissance et sélection de canaux de communication. En effet, les connexions D2D peuvent permettre de réduire le trafic des points d'accès, et une gestion efficace de la puissance du signal peut permettre une meilleure gestion des interférences.

Le paradigme SDN (Software Defined Networking) permet d'ouvrir des fonctions réseaux aux applications, ce afin de permettre à ces applications de mieux exploiter leur réseau. Dans les réseaux Wi-Fi, la vue centralisée maintenue par le contrôleur peut optimiser de manière efficace la complexité de l'environnement sans-fils. D'ailleurs Riggio et al. [114] utilisent le paradigme SDN a été utilisé pour la gestion d'interférences. Les auteurs de [115] ont proposé une architecture permettant de gérer les flux de données transitant par les points d'accès. L'utilisation des SDN permet ici de maintenir les états de points d'accès virtuels qui sont recalculés lorsque les utilisateurs se déplacent dans le réseau, ce dans le but d'avoir des changements de points d'accès de manière fluide. Seyedebrahimi et al.[116], utilisent le paradigme SDN pour l'affectation de canaux dans les réseaux Wi-Fi denses. L'affectation de canaux Wi-Fi a été modélisé comme un problème d'optimisation, cela a été implémenté dans le contrôleur SDN afin de pouvoir réagir lorsqu'un certain niveau d'interférences a été atteint sur le réseau. L'utilisation de ce type de solution a permis d'augmenter le débit de données grâce à une meilleure gestion du support de communication.

D'autres solutions [117, 118] essaient d'améliorer la qualité des réseaux Wi-Fi en mettant en place des systèmes d'ordonnancement des paquets, ou d'adaptation de flux afin d'améliorer l'utilisation de la bande passante des réseaux utilisés. La norme **802.11ax** [119], qui devrait commencer à être implémentée sur des cartes Wi-Fi à partir de la fin de l'année 2018 ou au début de l'année 2019, pourrait offrir des débits de l'ordre de ceux de la norme **802.11ad** [120] (jusqu'à 6.75Gbps) sans utiliser la bande de fréquence 60GHz. Ces améliorations se feront grâce à une meilleure utilisation du spectre de communication et d'une gestion plus efficace de la couche physique par rapport aux normes précédentes.

### II.3.3 Synthèse

La technologie Wi-Fi facilite aujourd'hui la communication de périphériques dans des réseaux sans fil. Ces problèmes de performances peuvent être dus à la contention qu'il peut y avoir sur les différents canaux (interférences). Fort heureusement, cette technologie évolue pour supporter une plus grande densité de périphériques [119] et pour offrir de meilleurs débits [120]. De plus, de nombreuses solutions peuvent être mises en œuvre au niveau de l'infrastructure [117, 118, 116, 115] pour optimiser son utilisation.

Il existe de nombreuses manières pour communiquer en P2P en Wi-Fi dont les protocoles Wi-Fi Direct [93] et Wi-Fi Aware [103] qui permettent de communiquer en D2D de manière contrôlée. Cependant, il n'y a pas de standard défini pour cela. Par exemple, Wi-Fi Direct et Wi-Fi Aware n'ont pas été intégrés aux standard

IEEE 802.11. De plus, les deux systèmes d'exploitation majoritairement utilisés sur les périphériques mobiles (iOS et Android) ne tendent pas vers un mécanisme de communication D2D commun en Wi-Fi.

## **Conclusion**

De nombreux travaux de l'état de l'art ont pour objectif une diffusion et un partage de données qui passent à l'échelle. Différentes approches basées sur le paradigme P2P [2, 3, 26, 4, 16] permettent aux périphériques de communiquer en P2P et de diffuser des données sur le réseau de manière efficace. Cependant, étant donné que la bande passante ne peut pas être partagée dans les réseaux sans fil comme elle l'est sur les réseaux filaires, les approches P2P classiques ne sont pas directement applicables.

Bien que la littérature dans les technologies D2D fasse état de nombreux travaux, ces travaux se concentrent principalement sur les réseaux cellulaires [6, 7]. En ce qui concerne la technologie Wi-Fi, elle a été utilisée dans de nombreux travaux comme moyen de déchargement d'infrastructures cellulaires [81, 82, 85, 84]. Les approches de dissémination de données qui utilisent le Wi-Fi se focalisent principalement sur les contenus vidéo sur des petits groupes de périphériques [61, 86, 59, 60]. De plus, la contrainte d'ordre d'accès aux données peut limiter l'exploitation complète des connexions D2D pour la dissémination de données.

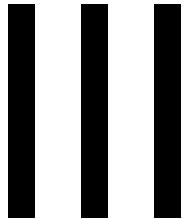
Nous visons, par le biais de connexions D2D créées de manière supervisée, entre les périphériques mobiles d'un réseau Wi-Fi, une dissémination efficace de données volumineuses sur les réseaux Wi-Fi. Notre objectif est de fournir un protocole de niveau applicatif afin qu'il soit adaptable à toute technologie capable de communiquer en D2D.

Pour l'implémentation de nos différentes approches, nous faisons le choix d'une orchestration centralisée. Elle permet de réduire le nombre de messages de contrôle. De plus, elle permet d'avoir une gestion plus efficace du réseau. Enfin, le découpage d'un fichier en plusieurs morceaux (chunks) de grande taille peut permettre de réduire le nombre de messages échangés pour le contrôle de la dissémination d'une donnée.





# CHAPITRE



## EDWiN : dissémination D2D sur les réseaux Wi-Fi denses

---

### Sommaire

---

<b>Introduction</b> . . . . .	<b>40</b>
<b>III.1 Architecture et hypothèses</b> . . . . .	<b>41</b>
III.1.1 Architecture du système . . . . .	41
III.1.2 Représentation du réseau . . . . .	43
III.1.3 Représentation d'un périphérique mobile . . . . .	43
<b>III.2 Algorithme de dissémination</b> . . . . .	<b>45</b>
III.2.1 État global et types de messages . . . . .	45
III.2.2 Serveur / ordonnanceur : . . . . .	46
III.2.3 Clients . . . . .	50
<b>Conclusion</b> . . . . .	<b>51</b>

---

## Introduction

Dans ce chapitre, nous présentons "EDWiN<sup>1</sup>" notre approche de dissémination de données sur les réseaux Wi-Fi. Nous nous intéressons au cas des événements à grande échelle pour lesquels les participants partagent un intérêt (séminaire, conférence, ...). Les applications prévues pour ces événements sont mises à la disposition des participants pour animer leurs activités, servent d'outil d'accompagnement. Ces applications utilisent un ensemble commun de données (fichiers vidéos, documents, ...). De nos jours, les technologies mobiles bénéficient de plus en plus de puissance de calcul et de stockage ainsi que de nombreuses fonctionnalités. L'utilisation de contenus numériques de très haute définition engendre une consommation accrue de la bande passante. Ces contenus sont hébergés dans un serveur avec lequel les périphériques communiquent via Wi-Fi.

Lorsque la taille des données utilisées est de l'ordre de plusieurs centaines de mégaoctets voir de quelques gigaoctets, l'utilisation d'une architecture centralisée pour les disséminer sur de nombreux périphériques ne permet pas de passer à l'échelle. La performance d'accès est limitée par la bande passante du réseau, dont la capacité est partagée par l'ensemble des périphériques. Lorsque les périphériques mobiles essaient d'accéder simultanément à ces données, le trafic généré par leur téléchargement provoque alors une congestion du réseau, ce qui peut entraîner des temps de dissémination très longs.

Notre but est d'améliorer les performances de la dissémination sur un tel type de réseau. Lorsque des périphériques partagent le même intérêt pour une donnée, ils peuvent être des sources potentielles pour celle-ci et peuvent aussi être utilisés pour la disséminer en parallèle. On peut ainsi augmenter le débit global du réseau et réduire le temps de dissémination par rapport à une approche centralisée. Dans le cas des réseaux Wi-Fi, on peut, en outre, bénéficier d'une vision globale du réseau à faible coût et de connexions Wi-Fi Direct parallèles qui permettront de délester l'infrastructure et d'augmenter le parallélisme lors de la dissémination. Notre objectif est donc de réduire ce temps de propagation afin rendre la donnée disponible sur tous les périphériques le plus rapidement possible. L'effet de bord positif que l'on peut avoir est de permettre le déchargement de l'infrastructure réseau afin que la bande passante ne soit pas monopolisée par la dissémination et qu'elle puisse aussi servir à d'autres usages sur le réseau. Nous utilisons une approche semi centralisée où les points d'accès peuvent être utilisés pour mesurer l'état du réseau (activité des clients, canaux disponibles, ...), et de permettre au serveur de maintenir des informations sur l'état des différents périphériques (et des échanges D2D en cours) par l'intermédiaire de messages de contrôle de petite taille. Grâce à cette architecture, les périphériques ne communiqueront que pour échanger des données. Ce qui permet d'économiser des messages de contrôle pour la recherche de données comme cela peut être le cas avec une approche décentralisée.

---

1. Acronyme de : Efficient data Dissemination over Wi-Fi Networks

La suite de ce chapitre se découpe en 2 sections : dans la section III.1, nous présentons l'architecture de notre système ainsi nos hypothèses. Dans la section III.2, nous décrivons notre algorithme de dissémination D2D de base. Enfin, nous concluons ce chapitre.

## III.1 Architecture et hypothèses

### III.1.1 Architecture du système

La figure III.1 représente l'architecture d'un réseau Wi-Fi tel qu'il est déployé lors d'un événement à grande échelle par Magency. Plusieurs points d'accès sont reliés à un serveur (ou plusieurs) par le biais d'un réseau de distribution (Ethernet). Chaque point d'accès permet de couvrir les périphériques situés dans sa zone ; l'ensemble qu'ils forment est appelé *ensemble de services de base* (BSS ou **basic service set**, voir la sous section II.3.1). Notre objectif est d'améliorer la dissémination de données sur chaque BSS (partie grisée sur la figure III.1). En reproduisant le même comportement sur les BSS, on peut réduire le temps de dissémination considérablement sur le réseau. On dénombre trois entités dans notre système :

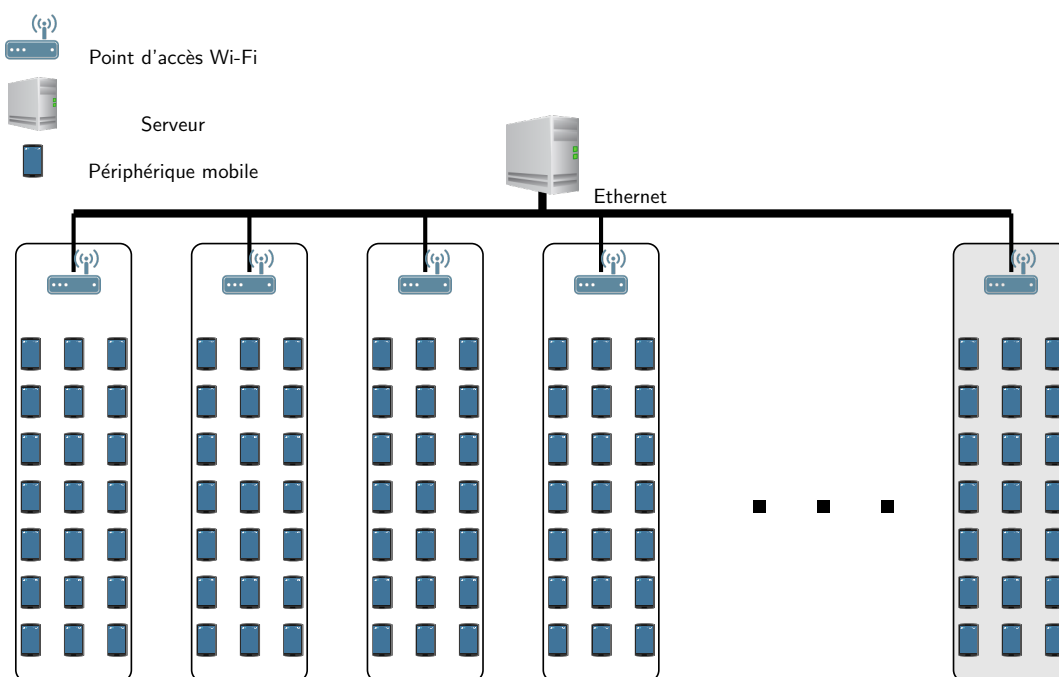


FIGURE III.1: Représentation d'un réseau Wi-Fi à grande échelle

**Le serveur :** son rôle est d'héberger la donnée accédée par les périphériques mobiles mais surtout d'orchestrer les échanges D2D sur le réseau ;

**le point d'accès Wi-Fi :** utilisé pour communiquer avec les périphériques, il peut permettre aussi d'avoir une vision globale et précise de l'état du réseau. Le point d'accès et le serveur étant indissociables dans notre système, ils seront confondus par la suite ;

**Les périphériques mobiles :** ces derniers accèdent aux données hébergées sur le serveur. Dans le cadre de nos travaux, ces périphériques accèdent à une même donnée volumineuse qui peut être sous n'importe quelle forme (vidéo, fichier audio, archives regroupant plusieurs fichiers, ...). Chaque périphérique a la capacité de communiquer en mode infrastructure via le point d'accès auquel il est connecté et en D2D grâce au Wi-Fi Direct. De nos jours, les périphériques mobiles disposent de plusieurs gigaoctets d'espace de stockage, ce qui permet de stocker des quantités importantes de données utilisées par les applications. Ces espaces de stockage peuvent être étendus par des cartes mémoires dont les capacités peuvent atteindre quelques centaines de gigaoctets. On suppose donc que chaque périphérique dispose de l'espace nécessaire pour stocker les données qu'il télécharge.

Notre approche consiste à exploiter les communications D2D entre les différents périphériques afin de réduire le temps de diffusion des données. En favorisant l'apparition de plusieurs sources de données parmi les périphériques, nous pouvons accroître le nombre d'échanges parallèles sur le réseau.

L'utilisation d'une technologie de communication D2D telle que Wi-Fi Direct implique la nécessité d'un établissement d'une connexion avant chaque échange de données. Ce coût est non négligeable et peut être en raison de la diminution des performances de la dissémination de données volumineuses, notamment si les périphériques qui échangent en D2D ont peu de données à partager. En utilisant l'architecture centralisée des réseaux Wi-Fi infrastructure, il est possible de maintenir un état global du réseau et d'orchestrer les échanges D2D à un faible coût. La connexion entre le serveur et les périphériques étant déjà existante, envoyer des messages de contrôle de très petite taille au serveur a un coût négligeable.

Ainsi, nous pouvons avoir une approche semi-centralisée pour la dissémination : le serveur joue le rôle d'orchestrateur centralisé et les périphériques sont capables d'échanger efficacement et indépendamment de l'infrastructure. En parallélisant les échanges D2D dans le réseau, nous augmentons le débit global sur le réseau. Cependant, plus on augmente le nombre de connexions Wi-Fi Direct, plus l'occupation des fréquences Wi-Fi est importante. Les canaux deviennent congestionnés et les périphériques plus exposés aux effets des interférences. Il en résulte une baisse importante des débits de transmission de données voir des échecs de transferts. Les interférences dans les réseaux sans fils sont directement liées à l'environnement dans lequel évoluent les périphériques (densité, matériaux des murs, emplacement, ...). Avec la technologie Wi-Fi, en fonction de la bande de fréquences sur laquelle les périphériques communiquent et de l'environnement, il existe des canaux

orthogonaux qui ne s'interfèrent pas entre eux que l'on peut d'ailleurs exploiter pour répartir les fréquences utilisées. Ainsi, les échanges D2D parallèles ne s'interfèrent pas.

Cet aspect est important dans notre cas, car il permet de considérer un nombre de connexions D2D simultanées qui impactent peu les performances de la dissémination. Dans le cas du Wi-Fi, il peut s'agir d'un nombre de canaux orthogonaux disponibles, ou de canaux offrant les meilleures conditions pour du transfert de données sans provoquer une baisse de performance globale. Des mesures que nous avons pu mener en amont de notre étude (figure II.10, tableau V.1) nous ont permis d'avoir une idée sur la différence de débit qu'il peut y avoir entre l'utilisation d'une connexion D2D et de l'infrastructure. Sur un échantillon de périphériques utilisés par Magency, nous avons pu observer un facteur 10 entre les débits de l'infrastructure (entre AP et périphériques) et des connexions D2D. En D2D, nous avons observé qu'un échange bidirectionnel était globalement plus rapide que deux échanges unidirectionnels successifs.

### III.1.2 Représentation du réseau

Dans un réseau Wi-Fi à grande échelle, plusieurs points d'accès sont utilisés afin d'améliorer la couverture et de répartir la charge sur les différents points d'accès. Les algorithmes que nous allons présenter se focalisent sur une sous-partie du réseau, c'est-à-dire les périphériques connectés à un même point d'accès. En effet, nous faisons l'hypothèse que les périphériques connectés à ce dernier sont capables d'établir un lien avec n'importe quel autre périphérique connecté à ce même point d'accès. De plus, les périphériques connectés au même point d'accès, se retrouvent assez éloignés des autres points d'accès et périphériques mobiles. Ce cas est observable lors d'événements à grande échelle répartis dans plusieurs salles. Diffuser sur un réseau composé de plusieurs points d'accès revient simplement à reproduire le même comportement sur chaque point d'accès et ses clients.

### III.1.3 Représentation d'un périphérique mobile

Pour orchestrer la dissémination de données, le serveur dispose d'informations sur les périphériques qui lui permettent d'orchestrer la dissémination. Ces informations sont :

**L'identifiant :** est utilisé par le serveur pour identifier (*id*) le périphérique et accéder à ses informations ;

**L'état :** Un périphérique peut être dans deux états : (*i*) IDLE et (*ii*) EXCH(Exchanging). Cette information est maintenue par le serveur dans un champs *state* pour chaque périphérique. Initialement, tous les périphériques sont dans l'état IDLE. L'état EXCH correspond à trois sous-états différents : (*i*) réception, (*ii*) envoi ou (*iii*) échange de données. Lors d'un échange bidirectionnel (*iii*), dès que l'un des périphériques a

fini son téléchargement, il envoie une notification au serveur. Si le périphérique avec lequel il échange n'a pas encore fini de télécharger ses données, ils restent dans l'état *EXCH* jusqu'à la fin du transfert. Dans le cas d'un échange unidirectionnel (*i* et *ii*), dès que le transfert se termine, les deux périphériques repassent à l'état *IDLE*. En résumé, quand un périphérique est en échange de données (unidirectionnel ou bidirectionnel), il est dans l'état *EXCH*. Sinon, il est dans l'état *IDLE*.

**Liste des chunks :** Afin de répliquer rapidement la donnée à disséminer, celle-ci est découpée en plusieurs morceaux (chunks). Le serveur maintient à jour la liste des chunks qui ont été téléchargés par chaque périphérique. Il utilise cette information par la suite pour sélectionner le(s) prochain(s) chunk(s) à diffuser sur les périphériques suivant la politique de réplification choisie; ainsi que les paires de périphériques qui vont effectuer un échange D2D. Lors de la dissémination le serveur maintient pour chaque périphérique la liste des chunks présents (*chunks*) ainsi que la liste des chunks en cours de transfert (*currentTransfer*).

**Autonomie de batterie :** Elle représente la durée restante estimée avant le déchargement complet du périphérique mobile. En plus de leur autonomie limitée, la vitesse de déchargement de chaque batterie lui est propre et varie d'un périphérique à l'autre.

**Débit :** Le débit réel mesuré (*bandwidth*) est toujours inférieur au débit maximal théorique. Nous considérons ici le débit réel mesuré dans la représentation des périphériques mobiles pour nos algorithmes de dissémination de données ainsi que pour l'évaluation.

**Établissement de connexions D2D :** L'utilisation de communications directes entre les périphériques implique d'établir une connexion D2D indépendante de l'infrastructure. Suivant le type de technologie utilisée, les procédures de connexions entre les deux périphériques peuvent varier. Dans notre cas, nous supposons l'utilisation du Wi-Fi Direct (WD ou Wi-Fi P2P).

En Wi-Fi Direct (voir sous-section II.3.1, paragraphe Wi-Fi Direct), les périphériques forment un groupe P2P pour communiquer. Afin que ce groupe soit créé, un périphérique est choisi pour permettre aux autres périphériques du groupe de communiquer. Ce périphérique est appelée le "Group Owner" (GO). Après la création du groupe, les autres périphériques (les clients) se connectent au GO de la même manière qu'avec un réseau Wi-Fi classique.

En ayant une connaissance globale du réseau et des informations sur les périphériques, il est préférable d'utiliser des formations de groupes autonomes, car les informations sur les différents périphériques le permettent. Cela nous permet d'éviter les phases de négociations et de sélection du GO et d'avoir des temps d'établissement de connexion plus courts [93].

## III.2 Algorithme de dissémination

Dans cette section, nous détaillons notre algorithme de dissémination de base publié à la conférence AINA 2017. L'objectif premier de cet algorithme est d'augmenter le nombre de sources d'une donnée sur le réseau afin de réduire le temps de sa dissémination. Pour cela, nous découpons la donnée en chunks. Nous utilisons la politique "*Rarest first*" BitTorrent [2] pour le choix des chunks à envoyer vers les périphériques. Nous nous inspirons de cette politique de choix de chunks à télécharger car elle permet d'avoir rapidement des copies de tous les chunks sur le réseau. Ainsi, cela favorise les échanges entre les différents périphériques mobiles et les chunks sont répartis sur le réseau de manière uniforme.

### III.2.1 État global et types de messages

Le serveur héberge les chunks de la donnée à disséminer et orchestre les échanges D2D entre les périphériques mobiles. Les informations relatives aux différents périphériques sont stockées dans une structure que nous appelons tableau d'état du réseau (*NST* ou *Network State Table*). La *NST* permet de stocker et maintenir l'état des différents périphériques et les échanges en cours. Cependant, cette information à elle seule ne suffit pas pour orchestrer la dissémination. En effet, le serveur maintient et utilise les variables suivantes pour l'orchestration :

- $Max_{para}$  : est le nombre maximal de périphériques autorisés à échanger sur le réseau en parallèle. Cette variable permet de contrôler le degré de parallélisme des échanges D2D. Elle est dépendante de la technologie utilisée et des conditions de l'environnement <sup>2</sup>.
- $ChunkSize$  : est la taille des chunks de la donnée;
- $Max_{AP}$  : est nombre maximal de chunks téléchargés depuis le serveur en un transfert;
- $Max_{D2D}$  : est le nombre maximal de données échangées entre les périphériques en un échange D2D;

Nous utiliserons par la suite les deux variables *activeNodes* et *ChunksCount* qui peuvent être déduites de la *NST* et du paramétrage du réseau. *activeNodes* représente le nombre de périphériques entrain de communiquer sur le réseau, sa valeur ne doit pas dépasser  $Max_{para}$ . *ChunksCount* représente le nombre de chunks, il peut être déduit en fonction de la taille de la donnée à disséminer et la taille des chunks *ChunkSize*.

#### III.2.1.1 Types et structure des messages

La structure générale d'un message est  $\langle type, Cs_1, Cs_2, src, dest \rangle$  où :

- *type* indique le type du message envoyé;

2. Ex : Réseaux adjacents, niveau d'occupations des canaux Wi-Fi, interférences,...



- *src* est l'identifiant du périphérique choisi comme source de données dans les échanges unidirectionnels et hôte de la connexion D2D lors d'un échange bidirectionnel ;
- *dest* est l'identifiant du périphérique sélectionné pour se connecter à une source avant d'échanger des données ;
- $Cs_1$  et  $Cs_2$  sont les ensembles de chunks qui sont respectivement envoyés par *src* et par *dest* pour *dest* et *src*.

Les deux périphériques impliqués lors de la dissémination de données utilisent les types de messages suivants :

**EXCHANGE:** Ce type de message est envoyé à deux périphériques qui ont été sélectionnés pour un échange de chunks. Lors d'un échange unidirectionnel, le périphérique dont l'identifiant est *src* crée une connexion D2D afin de permettre à *dest* de se connecter pour recevoir les chunks indiqués par  $Cs_1$  ( $Cs_2 = \emptyset$ ). Lors d'un échange bidirectionnel ( $Cs_2 \neq \emptyset$ ), le périphérique dont l'identifiant est *src* reçoit les chunks indiqués par  $Cs_2$  en même temps qu'il envoie des chunks à *dest*. Nous nous restreignons à ce type de message afin de simplifier le protocole de déclenchement des échanges au niveau du serveur. Ainsi chaque client interprète les instructions qui lui sont transmises en fonction de la structure et du contenu du message qu'il reçoit.

**FINISHED:** Ce type de message est envoyé au serveur par un périphérique mobile pour l'informer qu'il vient de finir un téléchargement ou un échange D2D. Ce message de notification permet au serveur de mettre à jour l'état global du réseau et de relancer de nouveaux transferts. Il véhicule les informations nécessaires pour identifier les deux périphériques impliqués dans l'échange D2D ainsi que les chunks qui viennent d'être disséminés.

### III.2.2 Serveur / ordonnanceur :

Les algorithmes 1 et 2 décrivent les procédures exécutées au niveau du serveur (ordonnanceur) qui sont *INIT* et *RECEIVE\_SRV*. La procédure *INIT* a pour rôle de lancer la dissémination. La procédure *RECEIVE\_SRV* est déclenchée par la réception d'un message de type *FINISHED*, et son rôle consiste à mettre à jour l'état de la NST et à déclencher de nouveaux échanges D2D sur le réseau lorsque c'est possible.

**Initialisation de la dissémination :** La procédure *INIT* décrite dans l'algorithme 1 initialise la NST et déclenche la dissémination. Les entrées de tous les périphériques sont initialisées avec une liste de chunks vide à l'exception de celle du serveur qui contient les identifiants de tous les chunks de la donnée (algorithme 1, lignes 2 à 4).

Chaque périphérique dispose d'un identifiant unique et connaît celui du serveur (qui est 0). Une fois la structure initialisée, le serveur déclenche le premier téléchar-

gement sur le réseau en faisant appel à la procédure  $scheduleAPDownload(Max_{AP})$  (ligne 5).

---

**Algorithme 1** Initialisation et lancement de la dissémination
 

---

```

1: procedure INIT( $NST, Cs$ )
2:    $NST[0].chunks \leftarrow Cs$ 
3:    $NST[i].chunks \leftarrow \emptyset, \forall i \in [1, N]$ 
4:    $NST[i].state \leftarrow IDLE, \forall i \in [1, N]$ 
5:    $ScheduleAPDownload(NST, Max_{AP})$ 
6: end procedure

7: procedure SCHEDULEAPDOWNLOAD( $T, MAX_{AP}$ )
8:   if  $\exists d \in T, |d.chunks| < ChunksCount$  then
9:      $\triangleright$  S'il existe des périphériques n'ayant pas reçu toute la donnée
10:     $node \leftarrow getIncompleteNode(T)$ 
11:     $Cs \leftarrow rarestChunks(node, Max_{AP})$ 
12:     $send(< EXCHANGE, Cs, \emptyset, node, 0 >, node)$ 
13:     $NST[0].state \leftarrow EXCH$ 
14:     $NST[node].state \leftarrow EXCH$ 
15:     $NST[node].currentTransfer \leftarrow Cs$ 
16:     $activeNodes ++$ 
17:   end if
18: end procedure

```

---

De la ligne 8 à 16, un périphérique n'ayant pas encore reçu la donnée dans son intégralité est sélectionné aléatoirement pour recevoir les premiers  $Max_{AP}$  chunks à disséminer. Le serveur sélectionne les chunks en suivant une politique "*Rarest First*".

L'entrée correspondante au périphérique choisi pour le transfert est mise à jour en lui affectant l'état EXCH, et en affectant les identifiants des chunks  $Cs$  au champs  $currentTransfer$ . Le serveur ayant été choisi comme source de données, son état est aussi changé en EXCH. À la fin de son téléchargement, le périphérique envoie un message de notification au serveur. Ainsi, la dissémination commence par le lancement de ce premier téléchargement, la figure III.2 illustre son déroulement.

**Ordonnanceur** : Lorsqu'un périphérique  $dest$  termine un téléchargement sur le réseau, il notifie le serveur avec un message d'acquiescement  $m = \langle FINISHED, Cs, src, dest \rangle$ .  $Cs$  représente ici la liste des identifiants des chunks qui viennent d'être reçus par  $dest$  depuis le périphérique  $src$ . Ce message véhicule les informations qui sont utilisées pour mettre à jour la NST en faisant appel à la fonction  $updateNST(m)$  (algorithme 2, ligne 2). Les instructions de cette fonction sont décrites dans l'algorithme 3.

Le serveur commence par mettre à jour la liste des chunks disponibles sur  $dest$ , en y ajoutant les identifiants contenus dans  $Cs$  et réinitialise le champs  $currentTransfer$ . Si le périphérique  $dest$  est toujours en cours d'envoi de données à

**Algorithme 2** Comportement du serveur

---

```

1: procedure RECEIVE_SRV( $m = \langle FINISHED, Cs, src, dst \rangle$ )
2:   updateNST( $m$ )
3:    $\mathcal{I} \leftarrow \{ \forall d \in NST ; d.state = IDLE \}$ 
4:   if  $NST[0].state = IDLE$  and  $\mathcal{I} \neq \emptyset$  then
5:      $cl \leftarrow \text{ScheduleAPDownload}(\mathcal{I}, Max_{AP})$ 
6:      $I \leftarrow \mathcal{I} \setminus \{cl\}$ 
7:   end if
8:    $p \leftarrow \text{selectExchange}(\mathcal{I}, Max_{D2D})$ 
9:   while  $p \neq Nil$  and  $activeNodes < Max_{Para}$  do
10:    triggerExchange( $p$ )
11:     $\mathcal{I} \leftarrow \mathcal{I} \setminus \{p.n1, p.n2\}$ 
12:     $p \leftarrow \text{selectExchange}(\mathcal{I}, Max_{D2D})$ 
13:   end while
14: end procedure

```

---

$src$ , alors les états des périphériques ne sont pas mis à jour (échange en cours). Sinon, les deux périphériques redeviennent disponibles et leurs états respectifs passent à IDLE.

Le nombre de périphériques actifs est alors réduit de 2 dans le cas d'un échange D2D et de 1 dans le cas où les chunks ont été téléchargés depuis le serveur (algorithme 2 lignes 7 à 12). L'utilisation du serveur comme source de données est plus avantageuse pour les périphériques (économie d'établissement de connexions et débit élevé). Si le serveur n'est pas en train de transférer des données par le point d'accès, il déclenche alors un téléchargement en appelant *scheduleAPDownload()* (algorithme 1).

Ensuite, parmi les périphériques disponibles du réseau, des paires de périphériques sont sélectionnées pour des échanges de données (algorithme 2 lignes 8 à 13). Un lien D2D met plusieurs secondes avant d'être établi, ce qui est non négligeable. Afin d'exploiter cette connexion de la manière la plus efficace possible, il est nécessaire d'échanger un maximum de données. Cependant, si la durée estimée d'un échange peut provoquer des pertes de performances (trop longue), l'utilisation de  $MAX_{D2D}$  permet d'en réduire la durée en cas de besoin. Des échanges D2D longs peuvent ralentir la dissémination vers la fin. Le serveur peut rester inactif et donc sous-utilisé, la définition d'une limite de blocs échangés entre les périphériques peut permettre de réduire la durée des échanges D2D et de ce fait privilégier l'utilisation du serveur. Une autre solution serait d'interrompre les téléchargements D2D et de finir la dissémination en utilisant le serveur.

Pour maximiser les échanges D2D, l'algorithme calcule la distance de Hamming [121] sur la liste des identifiants des chunks entre toutes les paires possibles parmi les périphériques disponibles sur le réseau. Ensuite, le serveur déclenche l'échange pour la paire ayant la distance maximale, c'est-à-dire les périphériques ayant le moins de

**Algorithme 3** Fonction *updateNST* du serveur

---

```

1: procedure UPDATENST( $m = \langle FINISHED, Cs, src, dest \rangle$ )
2:    $NST[dest].chunks \cup Cs$ 
3:    $NST[dest].currentTransfer \leftarrow \emptyset$ 
4:   if  $NST[src].currentTransfer = \emptyset$  then
5:      $NST[dst].state \leftarrow IDLE$ 
6:      $NST[src].state \leftarrow IDLE$ 
7:     if  $src = 0$  then
8:        $activeNodes --$ 
9:     else
10:       $activeNodes -= 2$ 
11:    end if
12:  end if
13: end procedure

```

---

données en commun. Ce calcul est fait par la fonction *selectExchange*( $Max_{D2D}$ ) (algorithme 2, ligne 8). Cette fonction retourne  $p$  qui contient les informations nécessaires pour un échange entre deux périphériques :

$n1$  et  $n2$  : les deux périphériques impliqués dans l'échange ;

$Cs_1$  (resp.  $Cs_2$ ) : les chunks que  $n1$  doit télécharger depuis  $n2$  (resp. que  $n2$  doit télécharger depuis  $n1$ ) ;

Ainsi, la procédure *triggerExchange*( $p$ ) (algorithme 4) déclenche l'échange entre les deux périphériques  $p.n1$  et  $p.n2$  qui ont été sélectionnés, met à jour les entrées de la NST de  $n1$  et  $n2$  et incrémente de 2 le nombre de périphériques actifs (*activeNodes*).

**Algorithme 4** Déclenchement d'un échange D2D

---

```

1: procedure TRIGGEREXCHANGE( $p$ )
2:    $send(\langle EXCHANGE, p.Cs1, p.Cs2, p.n2, p.n1 \rangle, p.n1)$ 
3:    $send(\langle EXCHANGE, p.Cs1, p.Cs2, p.n2, p.n1 \rangle, p.n2)$ 
4:    $NST[p.n1].currentTransfer = p.Cs1$ 
5:    $NST[p.n2].currentTransfer = p.Cs2$ 
6:    $NST[p.n1].state \leftarrow EXCH$ 
7:    $NST[p.n2].state \leftarrow EXCH$ 
8:    $activeNodes += 2$ 
9: end procedure

```

---

Tant qu'il y a des périphériques disponibles sur le réseau qui peuvent être utilisés pour des échanges D2D ( $p \neq Nil$ ) et que le nombre de périphériques actifs *activeNodes* n'a pas atteint le degré de parallélisme maximum  $Max_{Para}$ , le serveur calcule et déclenche des échanges D2D. Lorsqu'un transfert unidirectionnel est

lancé, le champs  $p.Cs_2$  est vide, le périphérique  $n_2$  ne fait alors que télécharger des chunks depuis  $p.n_1$ .

### III.2.3 Clients

Les périphériques mobiles (clients) traitent les instructions qui leurs sont envoyées par le serveur dans leur procédure  $RECV\_CLT\_EXCH$  qui est décrite dans l'algorithme 5. Un périphérique mobile est toujours en attente d'instructions de-

---

#### Algorithme 5 Procédure des clients (périphériques mobiles)

---

```

1: procedure RECV_CLT_EXCH(< EXCH, Cs1, Cs2, src, dest >)
2:   if id = src then
3:     D2DAccept(dest)
4:     if Cs2 ≠ 0 then
5:       download(Cs2, dst)           ▶ Téléchargement de Cs2 depuis dst
6:     end if
7:   else
8:     if src ≠ 0 then                 ▶ Si la source est un périphérique mobile
9:       D2DConnect(src)
10:    end if
11:    download(Cs1, dest)
12:  end if
13: end procedure

14: Upon download(Cs, src) finished :   ▶ À la fin de chaque téléchargement
15:  send ( < FINISHED, Cs, src, id >, 0)

```

---

puis le serveur. Lorsqu'il reçoit une instruction, suivant l'identifiant de la source de donnée (le serveur ou un autre périphérique mobile) et le type de message, deux comportements sont possibles :

- Un échange D2D (unidirectionnel ou bi-directionnel);
- Un téléchargement depuis le serveur (ressemble à un échange unidirectionnel à l'exception qu'il ne nécessite pas d'établissement de connexion D2D).

Les fonctions de bases que nous allons utiliser par la suite sont :

- $D2DConnect(host)$  : le périphérique courant établit une connexion D2D avec le périphérique  $host$ ;
- $D2DAccept(cl)$  : le périphérique courant crée une connexion D2D, puis attend que  $cl$  se connecte;
- $download(Cs, s)$  : lance le téléchargement des chunks  $Cs$  depuis  $s$ .

**Téléchargement depuis le serveur :** lorsqu'un téléchargement depuis le serveur est déclenché, le périphérique sélectionné reçoit un message de type **EXCHANGE** dont l'identifiant de la source est égal à celui du serveur (0). Dans ce cas, la connexion Wi-Fi avec le point d'accès étant déjà existante, le périphérique télécharge les chunks  $Cs_1$  directement en faisant appel à la fonction *download()*. (algorithme 5, ligne 11). La figure III.2 illustre un exemple de ce type d'échange.

**Échange D2D :** Comparé à un téléchargement depuis le serveur, les échanges D2D nécessitent un établissement de connexion. Le périphérique choisi comme hôte de la connexion D2D (GO Wi-Fi Direct) est celui dont l'identifiant est affecté au champs *src* des messages envoyés pour déclencher les échanges. Dès réception d'un message de type **EXCHANGE**, il crée une connexion D2D et se met à attendre que le périphérique *dest* se connecte en faisant appel à la fonction *D2DAccept()*. Une fois la connexion établie, il déclenche un transfert en appelant la fonction *upload()*. S'il s'agit d'un échange bidirectionnel ( $Cs_2 \neq \emptyset$ ), ce même émetteur lance alors un téléchargement en simultané depuis *dest*.

La figure III.3 illustre une séquence durant laquelle un échange bidirectionnel est déclenché. Lorsqu'un périphérique est choisi pour recevoir des données, il se connecte au périphérique qui lui a été désigné comme source en utilisant la fonction *D2DConnect(src)*, avant de lancer le téléchargement depuis *src* (*download(Cs<sub>1</sub>, src)*).

À la fin de chaque téléchargement de données (*download()*) depuis le serveur ou depuis le client, le périphérique ayant reçu des données  $Cs$  depuis une source *src* envoie le message de notification  $\langle \text{FINISHED}, Cs, src, id \rangle$  au serveur (algorithme 5, ligne 15).

## Conclusion

Dans ce chapitre, nous avons présenté une approche de dissémination de données sur les réseaux Wi-Fi qui utilise les connexions D2D que les périphériques mobiles peuvent établir entre eux afin d'améliorer le temps de dissémination comparé à l'utilisation de l'infrastructure seule. La solution que nous proposons est au niveau applicatif, elle est indépendante de la technologie de communication utilisée.

Cette solution est aussi bien adaptée pour des configurations dans lesquelles le voisinage d'un périphérique mobile ne change pas, qu'à une configuration dans laquelle le voisinage change. Pour des raisons de clarté, nous faisons ici l'hypothèse implicite que tous les périphériques peuvent communiquer entre eux. Cependant, il est envisageable que les périphériques communiquent leur voisinage au serveur afin que celui-ci le prenne en compte lors de l'établissement des paires. L'algorithme bénéficie des échanges D2D en favorisant à la fois les échanges entre les périphériques qui ont le moins de données en commun et la dissémination des chunks les plus rares sur le réseau. Dans le chapitre suivant,

---

nous présentons l'extension de notre approche pour la prise en compte de la consommation énergétique et de l'hétérogénéité des périphériques mobiles.

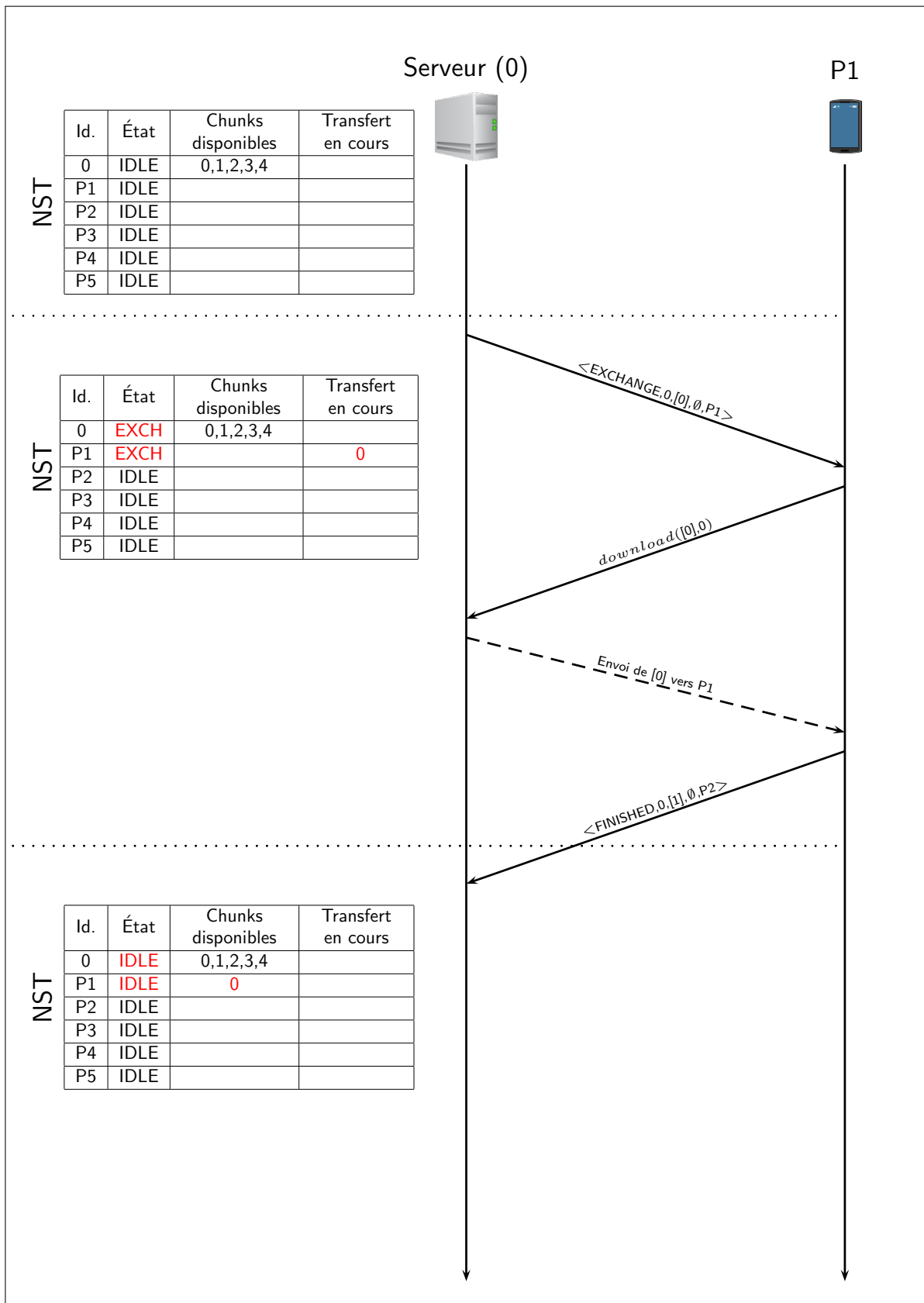


FIGURE III.2: Diagramme de séquence d'un téléchargement depuis le serveur.



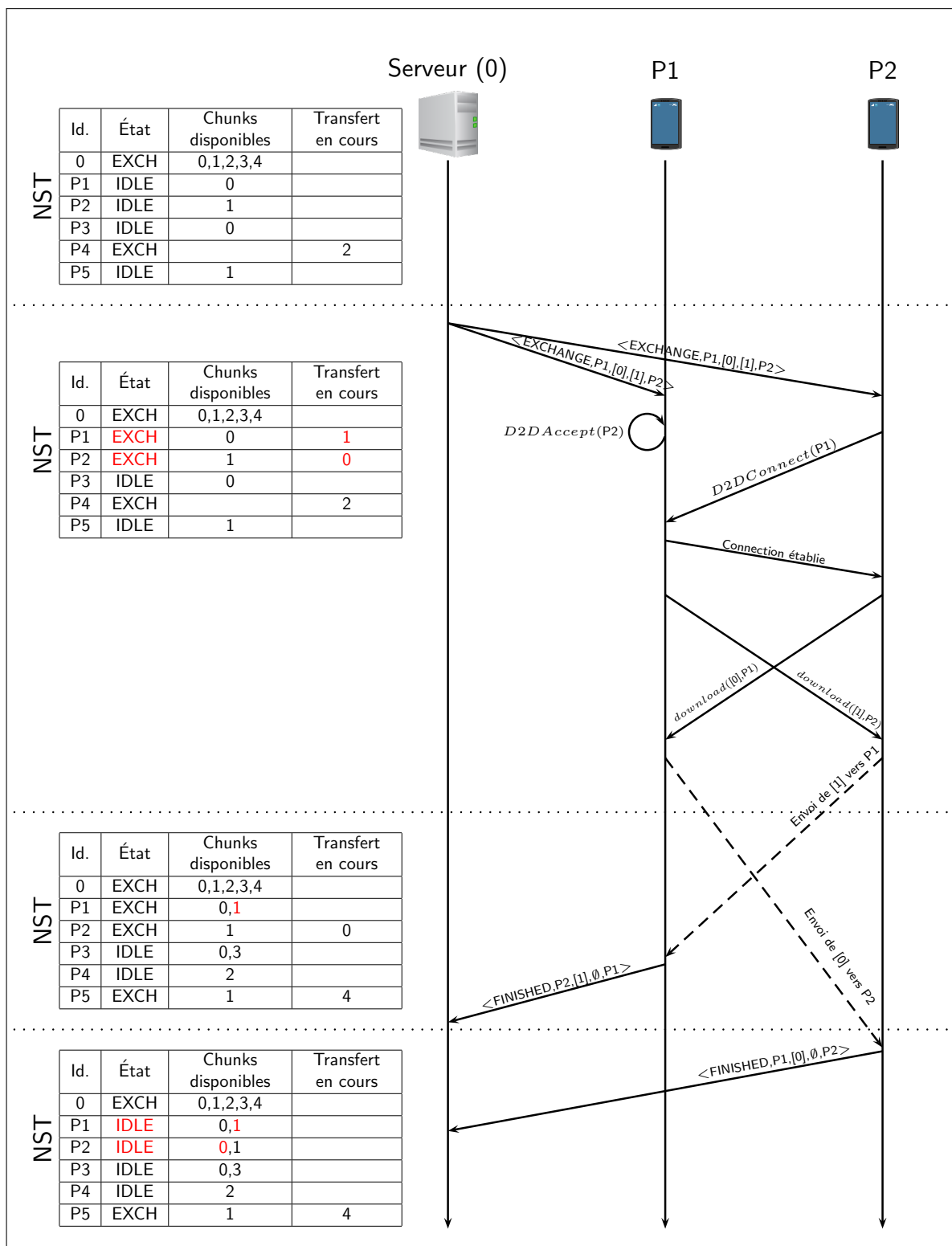


FIGURE III.3: Diagramme de séquence d'un échange D2D bidirectionnel.

## CHAPITRE

## IV

EDWiN' : prise en compte de la  
consommation d'énergie de  
l'hétérogénéité des périphériques  
mobiles

---

## Sommaire

---

<b>Introduction</b> . . . . .	<b>56</b>
<b>IV.1 Réduction de la consommation énergétique par regroupement (clustering)</b> . . . . .	<b>56</b>
<b>IV.2 Hétérogénéité des batteries</b> . . . . .	<b>57</b>
IV.2.1 Modèle de batterie . . . . .	57
IV.2.2 Description de l'algorithme . . . . .	58
<b>IV.3 Hétérogénéité des batteries et des bandes passantes</b> . . . . .	<b>61</b>
IV.3.1 Modèle de bande passante . . . . .	61
IV.3.2 Description de l'algorithme . . . . .	63
<b>Conclusion</b> . . . . .	<b>66</b>

---

## Introduction

Les périphériques mobiles ont une autonomie limitée, c'est un facteur important à prendre en compte. En plus de leur hétérogénéité en termes de caractéristiques, la manière dont ils sont utilisés l'est aussi. Chaque batterie a une durée de vie définie et vieillit selon la manière dont le périphérique est utilisé.

Limiter l'activité des périphériques sur le réseau permet de réduire la consommation globale d'énergie, mais ne considère pas l'hétérogénéité des périphériques. Dans notre cas, il est également important de répartir la consommation énergétique sur les différents périphériques afin de protéger ceux qui ont une autonomie restante en dessous d'un certain seuil, et ce afin que tous les utilisateurs puissent utiliser leurs périphériques et accéder à la donnée à la fin de la dissémination.

Il existe aujourd'hui de nombreux modèles de périphériques mobiles, leur hétérogénéité en termes de batteries, performance et bande passante est non négligeable. Dans ce chapitre, nous présentons une extension d'EDWiN pour la prise en compte de la consommation des batteries ainsi que l'hétérogénéité des périphériques mobiles. Les résultats de cette contribution ont été publiés dans deux conférences [Compas'2017] et [ICPADS'2017].

La suite de ce chapitre se déroule comme suit : la section IV.1 présente une première approche qui consiste à limiter l'activité des périphériques pour économiser de l'énergie. La section IV.2 présente l'extension d'EDWiN qui prend en compte l'hétérogénéité des batteries des périphériques mobiles. Dans section IV.3, nous présentons une autre extension qui prend en compte l'hétérogénéité des batteries et des bandes passantes. Enfin, nous concluons.

## IV.1 Réduction de la consommation énergétique par regroupement (clustering)

Plus un périphérique mobile passe du temps à échanger sur le réseau, plus sa batterie se décharge rapidement. Ainsi, les premiers périphériques qui acquièrent des données sur le réseau sont les plus utilisés lors de la dissémination pour fournir des chunks à d'autres périphériques. Par exemple, lorsque ces premiers auront reçu l'intégralité de la donnée, ils peuvent l'acheminer dans son intégralité en un seul envoi. Lorsque cette tâche est répétée plusieurs fois, leurs batteries se déchargent plus rapidement car elles sont plus sollicitées.

Une première approche que nous avons mise en œuvre pour éviter la surexploitation des périphériques mobiles sur le réseau consiste à limiter le temps d'activité de chaque périphériques pour des échanges D2D. En limitant ce temps, on peut réduire la consommation globale d'énergie en gardant de bons temps de complétion

qu'une dissémination centralisée. Un regroupement des périphériques mobiles du réseau par clusters nous permet d'accorder à chaque périphérique participant à la dissémination un temps défini pendant lequel il est autorisé à communiquer en D2D. Cette méthode consiste à :

- (i) Diviser le réseau en clusters  $\mathcal{C} = c_1, c_2, \dots, c_N$  de tailles égales ;
- (ii) Envoyer les données depuis le serveur vers un seul cluster  $c_k$  à la fois ;
- (iii) Lorsque le serveur envoie la donnée vers les périphériques du groupe  $c_k$ , les périphériques du groupe  $c_{k-1}$  sont utilisés pour commencer à envoyer des données ceux du groupe  $c_{k+1}$  afin d'aider le serveur à disséminer plus rapidement celui ci, lorsqu'il aura fini sur le cluster  $c_k$ .

Dans ce cas, les périphériques mobiles ne communiquent que lorsqu'ils envoient des données aux périphériques aux périphériques d'un autre cluster ou lorsqu'ils reçoivent des données. Dans le chapitre V, nous présentons une évaluation de cette approche qui met en évidence l'impact que peut avoir le contrôle de l'activité des périphériques sur les performances de la dissémination et sur la batterie. Notre approche permet de réduire le temps d'activité moyen des périphériques mobiles sur le réseau. Cependant, elle peut être limitante en terme de performance, car les connexions D2D ne sont pas exploitées de manière efficace.

## IV.2 Hétérogénéité des batteries

Nous nous concentrons ici sur l'hétérogénéité de l'autonomie des batteries des périphériques pour le choix de la politique d'ordonnancement. Si on se place dans un contexte où les périphériques doivent tenir toute la durée d'un événement ; la surexploitation de ces derniers peut entraîner des pertes importantes de batterie et peut réduire leur autonomie de manière considérable. De plus, étant donné que les débits des connexions D2D sont inférieurs à ceux que l'on peut avoir avec un point d'accès, ces derniers induisent des communications plus longues et plus coûteuses. Pour gérer efficacement la consommation énergétique sur le réseau, nous visons à la répartir de manière équitable (et non "égale") entre les différents périphériques. Nous exploitons les périphériques disposant d'une grande autonomie afin de protéger ceux ayant une autonomie plus courte (en dessous d'un certain seuil).

### IV.2.1 Modèle de batterie

Nous étendons notre modèle afin de considérer les batteries des périphériques mobiles. Les périphériques peuvent avoir des autonomies et des vitesses de perte différentes lors de l'utilisation du réseau ou d'autres fonctionnalités énergivores de l'appareil. Nous ajoutons ces deux aspects afin de prendre en considération les pertes d'énergie que l'on peut avoir sur les périphériques mobiles.

### Représentation des périphériques

En plus des informations décrites en section III.1 le serveur stocke maintenant une information supplémentaire pour chaque périphérique :

**La durée de batterie restante** : elle représente, en secondes, l'autonomie restante estimée avant la décharge complète du périphérique.

Cette autonomie peut varier d'un périphérique à l'autre. À l'état neuf, la batterie a une autonomie maximale. Les cycles répétés de déchargement et de chargement usent la batterie et réduisent sa durée de vie et son autonomie. De plus, chaque utilisateur a un mode de consommation différent. C'est cette hétérogénéité que nous considérons dans notre étude afin de prendre en compte la batterie de chaque périphérique de manière individuelle et de répartir équitablement la consommation énergétique sur le réseau.

Dans un contexte industriel, Magency [106] dispose d'un parc de tablettes tactiles qu'elle prête aux participants des événements qu'elle organise. Les périphériques utilisés sont tous du même modèle (iPad mini ou iPad 2) mais sont de générations différentes et ont tous été utilisés différemment. On a pu observer lors des événements<sup>1</sup>, que certaines tablettes étaient retournées par les participants au bout d'une demi journée avec une batterie presque vide ( 20%), alors que d'autres revenaient en fin de journée avec une autonomie aux alentours de 50%.

La définition d'une durée minimale *minBattery* à garantir pour un événement pour un événement permet de définir les périphériques à solliciter pour des échanges D2D et à les protéger d'une sur-activité réseau. Par abus de langage, les périphériques qui peuvent utiliser les connexions D2D sont dits "*bons*"; les périphériques à préserver sont dits "*mauvais*". Un bon périphérique peut devenir mauvais si son autonomie descend en dessous de *minBattery* lors de la dissémination.

Ainsi, notre objectif est de répartir la consommation énergétique de manière équitable de sorte que les périphériques à protéger (les mauvais) ne soient pas trop sollicités dans le réseau et que les *bons* périphériques soient plus utilisés pour des échanges D2D.

#### IV.2.2 Description de l'algorithme

L'objectif de cette approche est de maintenir un niveau de parallélisme maximum sur les *bons* périphériques afin de permettre au serveur de disséminer séquentiellement la donnée entière vers les *mauvais* de manière séquentielle. Cela permet de minimiser le nombre de communications D2D des *mauvais* périphériques sur le réseau. De plus, cela leur permet de bénéficier du débit plus élevé du point d'accès et d'avoir la possibilité de se mettre en veille une fois la donnée reçue. Le serveur, tel que l'illustre le schéma de la figure IV.1, alterne entre deux phases :

---

1. Témoignage et observations recueillies auprès des équipes opérationnelles de Magency.

- A. **Dissémination vers les *bons* périphériques** : cette phase consiste à envoyer des données et déclencher des échanges D2D sur l'ensemble des *bons* périphériques de la même manière qu'avec l'approche classique (voir section III.2).
- B. **Dissémination vers les *mauvais* périphériques** : Lorsque le nombre de *bons* périphériques qui échangent sur le réseau atteint le degré de parallélisme ( $activeNodes \geq MAX_{para} - 1$ ), ou s'ils ont tous reçu la donnée dans sa totalité et qu'il reste des mauvais périphériques en attente de la donnée, le serveur dissémine alors séquentiellement la donnée entière sur l'ensemble des *mauvais* périphériques;

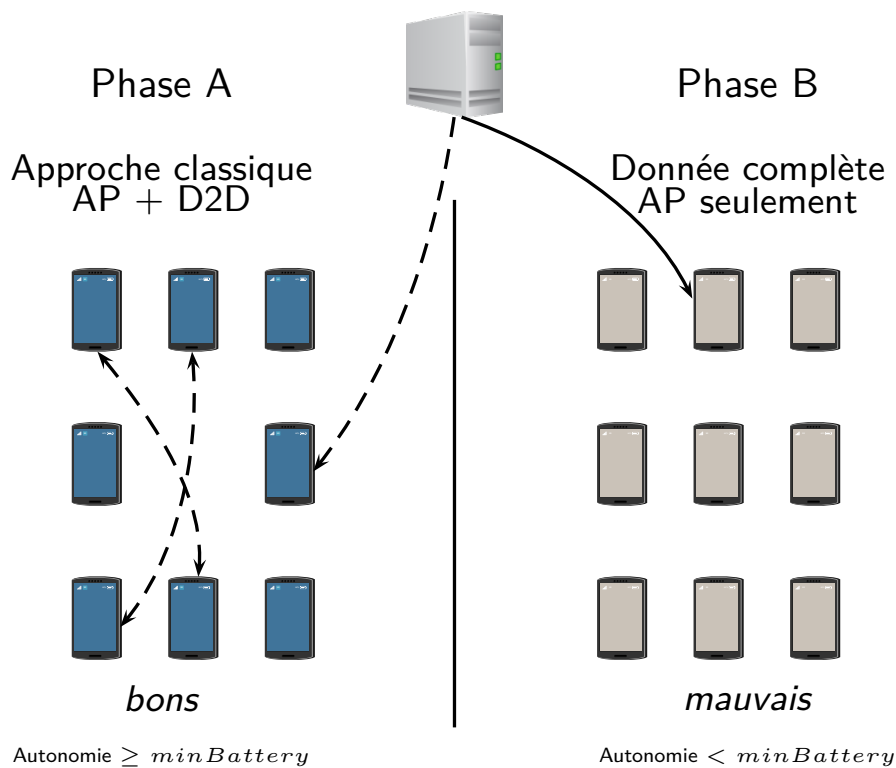


FIGURE IV.1: Description des différentes phases de dissémination.

L'algorithme 6 représente cette politique de dissémination. Lorsqu'il n'y a aucun *mauvais* périphérique sur le réseau, le serveur fonctionne de la même manière que l'approche initiale. Cependant, lorsqu'une partie du réseau est composée de mauvais périphériques, le serveur commence à lancer des envois de données depuis le serveur vers les *bons* périphériques et à déclencher des transferts D2D entre eux, et ce jusqu'à ce que le nombre de périphériques actifs en parallèle sur le réseau atteigne le degré maximal de parallélisme (algorithme 6 ligne 5).

En effet, cela permet à un nombre maximal de *bons* périphériques de s'échanger des données entre eux en parallèle pendant que le serveur dissémine sur les mauvais

périphériques. Pour chaque *mauvais* périphérique, le serveur envoie la donnée entière en un seul transfert. Les échanges D2D parallèles sont ici sélectionnés et déclenchés de la même manière qu'avec l'approche de base présentée en section III.2 sur les bons périphériques en utilisant le calcul basé sur la distance de Hamming (algorithme 6, ligne 8 et lignes 11 à 16).

---

**Algorithme 6** Dissémination avec prise en compte de la batterie
 

---

```

1: procedure SCHEDULEAPBATT()
2:    $\mathcal{I} \leftarrow \{ \forall d \in NST ; d.state = Idle \}$ 
3:    $good \leftarrow \{ \forall d \in \mathcal{I} ; d.battery \geq minBattery \}$ 
4:    $bad \leftarrow \{ \forall d \in \mathcal{I} ; d.battery < minBattery \}$ 

5:   if ( $activeNodes < MAX_{Para} - 1$  and  $good \neq \emptyset$ ) or  $bad = \emptyset$  then
6:      $scheduleAPDownload(good, Max_{AP})$   $\triangleright$  Seulement  $MAX_{AP}$  chunks
7:   else if ( $bad \neq \emptyset$  and  $activeNodes = MAX_{Para} - 1$ ) then
8:      $scheduleAPDownload(bad, ChunksCount)$   $\triangleright$  Donnée complète
9:   end if
10:   $updateLists()$   $\triangleright$  Met à jour les listes :  $\mathcal{I}, good$  et  $bad$ 

11:   $p \leftarrow selectExchange(good, Max_{D2D})$ 
12:  while  $p \neq Nil$  and  $nbTransfers < Max_{Para}$  do
13:     $triggerExchange(p)$ 
14:     $good \leftarrow good \setminus \{p.n1, p.n2\}$ 
15:     $p \leftarrow selectExchange(good, Max_{D2D})$ 
16:  end while
17: end procedure

18: procedure RECEIVE_BATT_AWARE( $m = \langle FINISHED, Cs, src, dst \rangle$ )
19:   $updateNST(m)$ 
20:   $ScheduleAPBatt()$ 
21: end procedure

```

---

Les périphériques perdent tous de la batterie en cours de dissémination. Un *bon* périphérique peut perdre de sa batterie et devenir *mauvais* lorsque son autonomie restante descend en dessous du seuil minimal du système *minBattery*. Dans ce cas, le serveur arrête de l'utiliser pour les échanges D2D sur le réseau et lui envoie par la suite les chunks qui lui restent en un seul envoi (cf. ligne 8).

L'alternance entre les deux phases décrites plus haut permet :

- Une bonne exploitation des connexions D2D en préservant un parallélisme à son maximum (proche à  $Max_{Para}$ ) sur les *bons périphériques* ;
- Une économie d'établissements de connexions, étant donné que les *mauvais* périphériques récupèrent leurs données depuis le serveur uniquement, ils peuvent bénéficier d'un débit plus élevé ;

- D'économiser de l'énergie des périphériques en leur permettant de se mettre en veille pendant une durée (estimée) pendant laquelle ils ne seront pas sollicités. Lorsque les *mauvais* périphériques sont en veille, ils n'émettent pas d'ondes Wi-Fi, ce qui peut réduire le taux d'interférences sur le réseau.

Lorsque le nombre de *mauvais* périphériques est trop élevé sur le réseau, la dissémination peut perdre en performance (voir V.2.5.2). Si on les utilise les *mauvais* périphériques comme sources, cela peut être très désavantageux pour leurs batteries car ils sont exposés à être surexploités pour la dissémination de données. Cependant, lorsque le nombre de mauvais périphériques dépasse un certain seuil, on peut envisager d'utiliser les communications D2D sur les *mauvais* périphériques, ce uniquement en réception. Cela peut permettre de mieux bénéficier plus du parallélisme sur le réseau et d'avoir de meilleures performances. Dans ce cas, même si les batteries des mauvais périphériques sont plus exploitées, leur temps de communication D2D reste borné car les *mauvais* périphériques ne seront pas sélectionnés pour envoyer ou échanger des données avec les autres périphériques. L'utilisation des *mauvais* périphériques pour récupérer la donnée en totalité ou en partie via un échange D2D peut alors présenter un bon compromis énergie/performance pour la dissémination.

Dans la section suivante, nous abordons la prise en compte d'un autre aspect de l'hétérogénéité des périphériques mobiles qui est celui de la bande passante.

### IV.3 Hétérogénéité des batteries et des bandes passantes

Les technologies de communication évoluent et leurs performances aussi. Les fabricants d'appareils mobiles intègrent les composants les plus récents à leurs différents produits. Les puces Wi-Fi n'échappent pas à cette règle. De la même manière que les batteries des périphériques, leurs bandes passantes sont aussi hétérogènes. Leurs performances et débits dépendent non seulement de la norme avec laquelle elles ont été fabriquées, mais aussi d'autres facteurs liés à la manière dont elles sont implémentées. Il est important d'exploiter cette hétérogénéité lors de la dissémination en utilisant au mieux les "bonnes" bandes passantes. De plus, nous avons observé une forte corrélation entre les périphériques qui ont une bande passante élevée et ceux avec une bonne batterie (i.e, avec un nombre de cycles de chargement/déchargement faible) car les périphériques les plus récents bénéficient souvent de cartes réseaux Wi-Fi plus efficaces. Ce type de périphérique peut être utilisé afin d'améliorer les performances de la dissémination. En effet, les échanges D2D entre les périphériques rapides peuvent permettre d'obtenir rapidement des sources efficaces qui contiennent la totalité des chunks.

#### IV.3.1 Modèle de bande passante

Pour prendre en compte l'hétérogénéité, nous ajoutons un critère de sélection des périphériques concernant la vitesse de leur bande passante. Nous considérons



maintenant deux types de périphériques en termes de bande passante : les "rapides" et les "lents". Un périphérique rapide se caractérise par une bande passante supérieure à un seuil minimal  $minFastBW$  (paramètre du système). Ainsi, nous considérons les quatre types de périphériques suivants :

- **Bons et rapides** : ce sont des périphériques disposant d'une batterie dont l'autonomie est supérieure au seuil minimal  $minBattery$  et d'une bande passante supérieure à  $minFastBW$ .
- **Bons et lents** : ce sont des périphériques disposant d'une batterie dont l'autonomie est supérieure au seuil minimal  $minBattery$  et d'une bande passante inférieure à  $minFastBW$ .
- **Mauvais et rapides** : ce sont des périphériques disposant d'une batterie dont l'autonomie est inférieure au seuil minimal  $minBattery$  et d'une bande passante supérieure à  $minFastBW$ .
- **Mauvais et lents** : ce sont des périphériques disposant d'une batterie dont l'autonomie est inférieure au seuil minimal  $minBattery$  et d'une bande passante inférieure à  $minFastBW$ .

Pour la suite de cette section, nous appellerons "mauvais" tout périphérique appartenant aux deux dernières catégories que nous venons de citer. En effet, le critère batterie est plus important : on ne veut donc pas surexploiter un périphérique ayant une mauvaise batterie. Donc, pour les périphériques avec une mauvaise batterie, on ne considère pas le critère "bande passante".

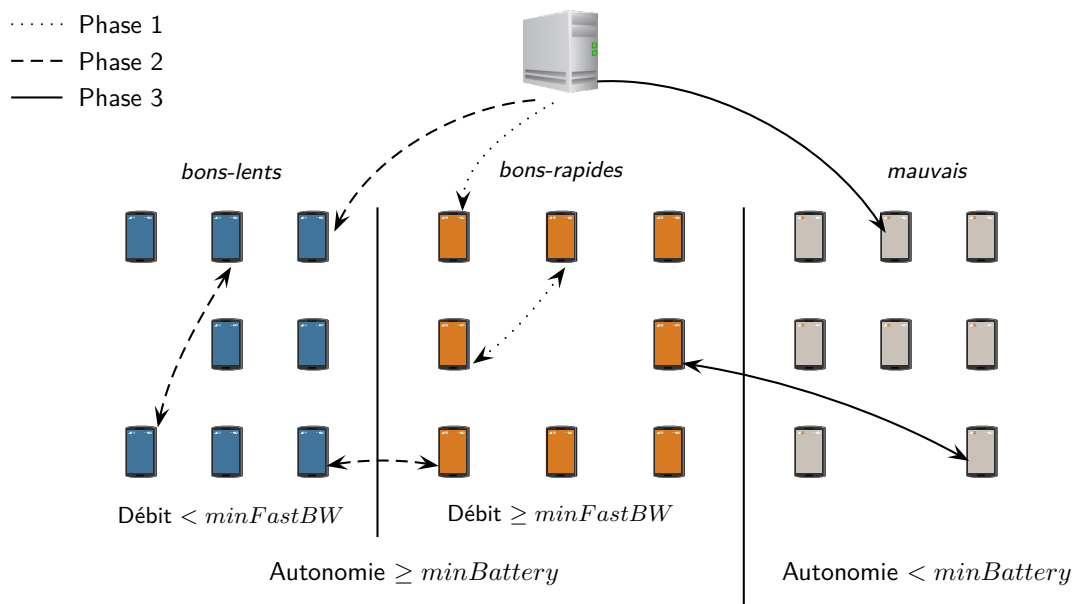


FIGURE IV.2: Prise en compte de la batterie et de la bande passante des périphériques : les phases de dissémination.

Comme le montre la figure IV.2, la dissémination se déroule en trois phases :

1. **Remplissage des *bons-rapides*** : le serveur va se concentrer uniquement sur l'ensemble des périphériques *bons-rapides* au début, en leur envoyant des chunks et en déclenchant des échanges D2D sur ce groupe de périphériques en priorité;
2. **Remplissage des *bons-lents* et des *bons-rapides*** : une fois que les *bons-rapides* ont eu toute la donnée ou qu'il n'y a plus de périphériques *bons-rapides* disponibles sur le réseau ; si le degré de parallélisme maximum n'a pas été atteint, alors le serveur commence à envoyer des données aux *bons-lents* et à déclencher des échanges D2D entre eux ainsi qu'avec les *bons-rapides* disponibles ;
3. **Remplissage des *mauvais*** : lorsque tous les *bons* périphériques ont eu la donnée complète ou sont en train de finir de la télécharger. Les *bons-rapides* sont sélectionnés pour transférer des chunks aux *mauvais* périphériques.

L'objectif de cette approche est de donner une plus grande priorité aux *bons-rapides* afin qu'ils deviennent rapidement des sources complètes de données. En effet, leur bande passante plus importante leur permet d'échanger des données en D2D plus rapidement et d'être des sources complètes de données par la suite (nombre d'établissements de connexions D2D réduit). Ici, Lorsque les *mauvais* périphériques récupéreront la donnée en D2D, ils le feront en un seul échange. Les mauvais périphériques étant laissés en dernier pour la récupération des données, ils peuvent se mettre en veille pendant que le serveur communique avec les *bons-\** périphériques afin d'économiser de l'énergie et de sortir de veille au moment de télécharger les données. Il est en effet aisé d'estimer le temps que prendront les deux premières phrases.

### IV.3.2 Description de l'algorithme

L'algorithme 7 représente la politique d'ordonnancement des échanges sur le réseau. L'initialisation et le début de la dissémination commencent de la même manière que l'approche avec prise en compte de la batterie des périphériques.

La procédure *RECEIVE\_BW\_AWARE* décrite dans cet algorithme commence par mettre à jour l'état de la *NST* à chaque fin de transfert, puis appelle la procédure *ScheduleAPBW()* (algorithme 7 lignes 5 à 21). La différence réside dans la politique de sélection de données sur les différentes familles de périphériques. De la ligne 6 à 10, le serveur calcule les ensembles de périphériques des différentes catégories *good*, *bad*, *fast* et *fastCompleted* qui sont compris dans la liste  $\mathcal{I}$  des périphériques disponibles dans la *NST*.

Ensuite, de la ligne 11 à la ligne 14, si le point d'accès est disponible, le serveur déclenche un téléchargement via le point d'accès sur un périphérique mobile. La priorité de choix de périphériques pour lesquels envoyer des données suit l'ordre suivant : les *bons-rapides*, ensuite les *bons-lents* et enfin les *mauvais*.

Cette tâche est exécutée en appelant *scheduleAPTransfer()* (algorithme 8) qui permet de lancer un transfert depuis le serveur en suivant l'ordre de priorité :

**Algorithme 7** Prise en compte de la batterie et de la bande passante

---

```

1: procedure RECEIVE_BW_AWARE( $m = \langle FINISHED, Cs, src, dst \rangle$ )
2:   updateNST( $m$ )
3:   ScheduleAPBW()
4: end procedure

5: procedure ScheduleAPBW()()
6:    $\mathcal{I} \leftarrow \{ \forall d \in \mathcal{N} ; d.state = Idle \}$            ▷ Périphériques disponibles
7:    $good \leftarrow \{ \forall d \in \mathcal{I} ; d.battery \geq minBattery \}$    ▷ bons-lents et bons-rapides
8:    $bad \leftarrow \{ \forall d \in \mathcal{I} ; d.battery < minBattery \}$      ▷ mauvais
9:    $fast \leftarrow \{ \forall d \in good ; d.bandwidth \geq minFastBW \}$    ▷ bons-rapides
10:   $fastCompleted \leftarrow \{ \forall d \in fast ; |d.chunks| = ChunkSize \}$  ▷ bons-rapides
    avec tous les chunks

11:  if  $NST[0].state = Idle$  then
12:    scheduleAPTtransfer( $Max_{AP}$ )
13:    updateLists() ▷ Met à jour les listes :  $\mathcal{I}, good, bad, fast$  et  $fastCompleted$ 
14:  end if

15:   $p \leftarrow selectExchange$ ()
16:  while  $p \neq Nil$  and  $activeNodes < Max_{Para} - 1$  do
17:    triggerExchange( $p$ )
18:    updateLists()
19:     $p \leftarrow selectExchange$ ()
20:  end while
21: end procedure

```

---

*bons-rapides*, *bons-lents* puis *mauvais*. Tant que la totalité des *bons-rapides* n'ont pas reçu la totalité de la donnée, ( $|fastCompleted| < N_{fast}$ , où  $N_{fast}$  est le nombre total de périphériques *bons-rapides*), le serveur choisit un périphérique *bon-rapide*. Sinon il choisit un périphérique *bon-lent* ou un *mauvais* (ce qui arrive en fin de dissémination). Lorsque le périphérique choisi est un *mauvais*, alors la donnée lui est envoyée dans son intégralité. Sinon, alors seulement  $Max_{AP}$  chunks lui sont envoyés. Ensuite, l'ordonnanceur déclenche des échanges D2D entre les périphériques tant que cela est possible (algorithme 7, lignes 15 à 20). Cela se fait lorsqu'il appelle la fonction *selectExchange\_BW*() qui retourne une paire de périphériques  $p$  pour déclencher un échange D2D.

Cette fonction est décrite dans l'algorithme 9. Elle permet le choix de paires de périphériques pour des échanges D2D dans l'ordre de priorité suivant :

1. *bon-rapide* avec *bon-rapide*;
2. *bon-rapide* avec *bon-lent*;
3. *bon-lent* avec *bon-lent*;

**Algorithme 8** Algorithme de sélection des transferts depuis le point d'accès

---

```

procedure scheduleAPTtransfer()
  node ← Nil
  if |fastCompleted| < Nfast then
    node ← selectIncompleteNode(fast)
  else if good ≠ ∅ then
    node ← selectIncompleteNode(good)
  else
    node ← selectIncompleteNode(bad)
  end if
  if node ≠ Nil then
    Cs ← Nil
    if node ∈ bad then
      Cs ← unavailableChunks(node)
    else
      Cs ← rarestChunks(node, MaxAP)
    end if
    send(< EXCHANGE, 0, Cs, ∅, node >, node)
    NST[0].state ← EXCHANGING
    NST[node].state ← EXCHANGING
    NST[node].currentTransfer ← Cs
    activeNodes ++
  end if
end procedure

```

---

4. *bon-rapide* avec *mauvais*;

Ce choix est fait en fonction des périphériques disponible sur le réseau. La fonction *selectIncompleteNode(L)* choisit dans la liste *L* passée en paramètre le périphérique qui a le moins de chunks (*Rarest First*). La fonction *selectExchangeFromList(L)* (algorithme 7, ligne 16), permet de sélectionner une paire de périphériques de la liste *L* qui ont une distance de Hamming maximale sur leurs ensembles de chunks. Cela est le cas lorsque la liste *fastCompleted* est vide (algorithme 9, lignes 6 et 16).

En permettant aux *mauvais* périphériques d'acquérir des données en D2D, ces derniers passent plus de temps à communiquer sur le réseau. Le but de cette approche est de maintenir un équilibre entre performances et consommation d'énergie. Dans l'évaluation de cette stratégie, nous verrons l'impact positif que cette approche peut avoir sur les performances du réseau ainsi que l'impact que cela pourrait avoir sur la consommation d'énergie.

De plus, il est possible d'estimer une durée pendant laquelle les mauvais périphériques doivent se mettre en veille avant qu'ils ne se réveillent et rejoignent le réseau pour récupérer leurs données. Cela permettrait d'économiser encore plus

**Algorithme 9** Sélection des paires d'échanges D2D

---

```

1: function selectExchange_BW()
2:   if  $|fastCompleted| > 0$  then
3:      $fastDevice \leftarrow selectNode(fastCompleted)$ 
4:      $d \leftarrow Nil$ 
5:     if  $|fast| > 0$  then
6:       return selectExchangeFromList(fast)
7:     else if  $|good| > 0$  then
8:        $d \leftarrow selectIncompleteNode(good)$ 
9:     else if  $|bad| > 0$  then
10:       $d \leftarrow selectIncompleteNode(bad)$ 
11:    end if
12:    if  $d \neq Nil$  then
13:      return createExchange(fastDevice, d)
14:    end if
15:  end if
16:  return selectExchangeFromList(good) ▷ voir algorithme 2
17: end function

```

---

d'énergie car au lieu de rester dans une attente active qui leur est coûteuse en énergie, les *mauvais* périphériques seraient en veille. De plus, en désactivant leurs communications Wi-Fi pendant cette durée, les périphériques mobiles n'émettent aucun signal pouvant interférer avec leurs voisins et dégrader leurs performances. Enfin, même si les *mauvais* périphériques peuvent être plus utilisés, ils peuvent quand même gagner en énergie.

## Conclusion

Nous avons proposé une approche qui permet d'économiser de l'énergie des périphériques ayant de mauvaises batteries. Ces algorithmes favorisent les échanges D2D entre les périphériques ayant une plus grande autonomie. Cependant, privilégier l'utilisation des bons périphériques peut mener à une décharge plus importante. La question de l'utilisation plus intensive des périphériques munis d'une bonne batterie lorsque les périphériques sont maîtrisés ne représente pas un problème. Sa mise en œuvre peut s'avérer plus complexe dans le cas de périphériques appartenant aux participants, ces derniers peuvent choisir s'ils partagent leur charge avec d'autres participants.

Lorsqu'on considère l'hétérogénéité des bandes passantes, la dissémination peut bénéficier des périphériques les plus rapides pour faire émerger des sources complètes de données rapidement grâce aux échanges D2D qui se déroulent plus rapidement que sur les autres périphériques, plus lents.

Ainsi, la prise en compte de l'hétérogénéité des périphériques peut permettre une dissémination qui protège les batteries les plus faibles d'une décharge due à une surexploitation sur le réseau sans forcément détériorer les performances de la dissémination.



## CHAPITRE

## V

## Évaluation d'EDWiN

## Sommaire

---

<b>Introduction</b> . . . . .	<b>70</b>
<b>V.1 Simulateur et scénario d'expérimentation</b> . . . . .	<b>70</b>
V.1.1 Description du simulateur . . . . .	70
V.1.2 Métriques . . . . .	72
V.1.3 Configurations de la simulation . . . . .	73
<b>V.2 Évaluation de l'algorithme de dissémination D2D</b> . . . . .	<b>76</b>
V.2.1 Degré de parallélisme . . . . .	76
V.2.2 Passage à l'échelle . . . . .	77
V.2.3 Volumes de données échangés . . . . .	78
V.2.4 Énergie et contrôle de l'activité des périphériques mobiles . . . . .	81
V.2.5 Prise en compte de l'hétérogénéité des batteries . . . . .	83
V.2.6 Prise en compte de l'hétérogénéité des batteries et des bandes passantes . . . . .	88
<b>Conclusion</b> . . . . .	<b>90</b>

---



## Introduction

Dans ce chapitre nous présentons une évaluation des gains apportés par notre approche pour la dissémination d'une donnée volumineuse. L'évaluation de systèmes basés sur des réseaux sans fils est une tâche difficile dont la reproductibilité est complexe, voir impossible. Cela est dû à plusieurs facteurs dont les interférences entre les signaux des périphériques mobiles et les conséquences qu'elles engendrent (taux d'erreurs élevés sur la transmission des paquets, latence, perte de connexions, etc.). Nous utilisons la simulation à événements discrets pour évaluer notre approche car elle permet d'obtenir, en ordre de grandeur, les gains potentiels que l'on pourrait avoir sur les performances de la dissémination à grande échelle et de tester rapidement différentes configurations.

La suite de ce chapitre se structure comme suit : la section V.1 décrit l'architecture du simulateur ainsi que le protocole d'expérimentation. La section V.2 présente l'évaluation de l'algorithme de dissémination de ainsi que celle de l'évaluation de la prise en compte de l'hétérogénéité des périphériques mobiles sur la dissémination. Enfin, nous concluons.

## V.1 Simulateur et scénario d'expérimentation

### V.1.1 Description du simulateur

Pour évaluer notre approche, nous utilisons le moteur de simulation à événements discrets Peersim [122]. Ce simulateur offre deux modèles de simulation : un modèle cyclique (cycle-based) et un modèle basé sur des événements discrets (event-based). Pour notre cas d'étude, nous utilisons le moteur à événements discrets car ce dernier permet d'introduire les délais liés à la communication entre les différents périphériques du réseau ainsi que les temps de téléchargement des données plus précisément. Chaque événement (message) est identifié par sa source, son destinataire, son contenu et son délai d'acheminement. Ce simulateur permet de représenter le réseau de structurer les nœuds en différentes couches comme on pourrait l'avoir dans le monde réel. Pour fonctionner, le simulateur récupère à partir d'un fichier les informations nécessaires à la construction du réseau et à sa simulation (taille du réseau, topologie, délais de communication, etc.).

#### V.1.1.1 Modélisation du réseau

Pour simuler notre réseau, nous implémentons chaque nœud en deux couches :

1. **La couche applicative** : elle représente le comportement du nœud sur le réseau. C'est dans cette couche que nous avons implémenté les différentes approches de dissémination détaillées dans les chapitres III et IV. Son fonctionnement consiste à traiter les messages qu'elle reçoit sous forme d'événements via la couche "Transport".

2. **La couche transport** : elle est utilisée par la couche applicative pour assurer l'acheminement de messages entre les différents nœuds du réseau. Lorsqu'un nœud veut transmettre un message, sa couche applicative fait appel à la couche transport qui va, en fonction du message et des nœuds source et destination, calculer un délai de transmission pour planifier la réception de celui-ci par le destinataire. Ce message déclenchera un événement au niveau du récepteur une fois que le délai calculé est passé.

**Calcul des délais d'acheminement des données** : Dans Peersim, les communications se font de point à point. Afin de planifier un événement sur un nœud, la couche transport calcule le délai de transmission de celui-ci en fonction du type du message et du volume de données qu'il contient et des nœuds source et destination. Le temps de communication étant discrétisé, le simulateur est configuré afin que les délais de communications correspondent en ordre de grandeur à ceux que l'on pourrait avoir sur un réseau réel. Dans notre protocole, le serveur envoie des messages de contrôle lorsqu'il déclenche des échanges sur le réseau et lorsque ces derniers se terminent, l'acheminement de ces messages se faisant par le biais du point d'accès, leur temps de transmission est négligeable. Cependant, pour l'échange de données, le délai de transmission est plus long et dépend de la taille de la donnée, de la bande passante de l'émetteur et du récepteur et du mode de communication.

Le temps de téléchargement d'une donnée depuis le serveur  $t_{server}$  est égal à :

$$t_{server} = \frac{S}{BW_S}$$

où  $S$  est la taille de la donnée en octets et  $BW_S$  est le débit d'émission du serveur. Lorsque deux périphériques mobiles  $m_1$  et  $m_2$  échangent des données, le temps de d'envoi  $t_{D2D}$  est égal à

$$t_{D2D} = t_{connect} + \frac{S}{\min(BW_{m_1}, BW_{m_2})}$$

où  $t_{connect}$  représente le délai d'établissement de la connexion  $D2D$  et  $\min(BW_{m_1}, BW_{m_2})$  la bande passante du transfert, le débit de transfert est limité par la bande passante du périphérique ayant le débit le plus réduit sur les deux.

Pour paramétrer notre simulateur, nous avons préalablement réalisé des mesures sur des périphériques utilisés en production. Le tableau V.1 illustre les débits de transferts de données et les délais d'établissements de connexion qui ont été utilisés pour paramétrer le simulateur.

### V.1.1.2 Autonomie de la batterie

Un périphérique dispose d'une autonomie initiale en veille. Au fur et à mesure de la dissémination, le périphérique perd en autonomie. La consommation d'énergie

que peut provoquer une application sur un périphérique varie en fonction de la manière dont les composants (GPS, réseau, affichage, etc.) sont sollicités. Plusieurs modèles et méthodes de mesure et de surveillance de consommation d'énergie ont été proposés dans la littérature [123, 124, 125]. Hoque et al. [126] ont d'ailleurs présenté une étude complète des différents outils et approches de mesure et de profilage de consommation d'énergie existant dans la littérature. Les modèles et outils proposés dans ces différentes approches permettent de mesurer la consommation d'une application par les composants qu'elles utilisent. Dans notre cas, nous considérons que les périphériques doivent disposer d'une autonomie suffisante à la fin de la dissémination. L'autonomie que nous modélisons est celle qu'un périphérique peut avoir dans un état de veille. Les échanges qu'il a avec d'autres périphériques ou le serveur provoquent un coût énergétique supplémentaire dans le déchargement de la batterie du périphérique.

### V.1.2 Métriques

Dans la section III.1, nous avons décrit un ensemble de paramètres liés aux périphériques et au système et qui influencent ses performances. Afin d'évaluer notre approche, nous définissons les métriques suivantes :

1. **Temps de complétion** : représente le temps pris pour disséminer la donnée sur tous les périphériques ;
2. **Périphériques actifs** : représente le nombre de périphériques qui sont en train de recevoir ou d'échanger des données sur le réseau ;
3. **Périphériques complétés** : représente le nombre de périphériques qui ont reçu l'intégralité de la donnée sur le réseau ;
4. **Autonomie restante estimée** : représente la durée d'autonomie restante estimée du périphérique dans un état de veille ;
5. **Temps d'activité sur le réseau** : représente la durée cumulée des échanges d'un périphérique.

Ici, la métrique principale en termes de performance est le temps de complétion. Notre but est d'obtenir le temps le plus petit possible, en bénéficiant au maximum du parallélisme des échanges sur le réseau. Le nombre de périphériques qui échangent en parallèle doit donc vite atteindre le degré de parallélisme voulu (voir sous-section III.1) et doit rester dans cet état jusqu'à la fin de la dissémination. Dans un premier temps, nous étudions l'impact de paramètres liés à la topologie du réseau et au découpage de la donnée disséminée. Ces paramètres sont :

- Le degré de parallélisme ;
- La taille du réseau ;
- La taille des chunks ;
- Le temps d'établissement de connexion D2D ;
- Le nombre maximal de chunks par transfert depuis le serveur ;

- Le nombre maximal de chunks par échange D2D.

Dans un second temps, nous définissons les différents types de périphérique suivant les modèles décrits dans les sections IV.2 et IV.3, et nous évaluons l'impact des proportions des différents types sur les performances de la dissémination.

### V.1.3 Configurations de la simulation

Afin que nos simulations soient proches de cas réels, nous avons fait le choix de calibrer notre simulateur en utilisant des valeurs mesurées sur des périphériques utilisés par Magency. Pour les mesures du débit de téléchargement depuis le point d'accès, nous avons utilisé un serveur de production utilisé par Magency avec un point d'accès Wi-Fi Unifi UAC AP Pro (voir section II.3.1). Le serveur utilisé disposait d'un processeur Intel Core i7 à 2 cœurs cadencé à 2.7Ghz et de 8GO de Mémoire. Les fichiers utilisés pour effectuer les mesures ont été placés dans un RAMDISK.

Nous considérons un réseau tel que nous l'avons décrit dans la section III.1. Nous étudions ici l'impact des différents paramètres du système sur les performances de la dissémination. Chaque expérimentation que nous présenterons par la suite est simulée 20 fois afin de prendre en compte les variations de débit que nous avons mesurées. En effet, à chaque transmission de données, le temps calculé inclut une partie aléatoire correspondant aux variations des mesures de débit.

#### V.1.3.1 Calibration des échanges D2D

Notre évaluation a pour but d'évaluer les gains de l'utilisation de notre approche pour la dissémination de données volumineuses.

Nous simulons ici un réseau "sans interférences" afin de mesurer les performances du système en fonction de ses paramètres. Les interférences sont omniprésentes dans les réseaux sans fil. Elles dépendent de plusieurs facteurs, le mode d'utilisation du medium. En pratique, les débits et les interférences fluctuent dans le temps ainsi que les temps d'établissement de connexions. Ces derniers peuvent varier en fonction des conditions environnementales et des interférences entre les différents réseaux et périphériques à proximité.

Pour mesurer les débits des échanges D2D, nous avons utilisé des tablettes Apple iPad Air 2. Les mesures ont été réalisées pendant des journées de travail. Elles ont eu lieu à l'intérieur d'un immeuble de bureaux dans lequel plusieurs réseaux Wi-Fi cohabitent.

Afin de réaliser ces mesures, nous avons utilisé la bibliothèque MultipeerConnectivity [104] qui permet de faire des communications P2P sur les périphériques de la marque Apple. Les périphériques utilisés ont été réinitialisés et n'ont été connectés à aucun réseau Wi-Fi, seulement les puces Wi-Fi ont été activées sur ces périphériques. Cela nous a permis de forcer la création de communications D2D en Wi-Fi entre les périphériques utilisés. Le tableau V.1 représente les débits et temps mesurés sur les

tablettes utilisées.

Débit serveur	Transfert D2D	Échange D2D	Établissement de connexion
10 Mo/s	900 Ko/s	540 Ko/s	entre 4 et 7 s

Tableau V.1: Délais et débits moyens sur des tablettes Apple iPad Air 2 (20 mesures)

Le but de ces mesures était d'obtenir l'ordre de grandeur des débits que l'on peut avoir dans un réseau Wi-Fi. Le calcul des débits des échanges D2D sont la moyenne de 20 échanges de chaque type :

- Pour les transferts, le débit moyen mesuré était de 909Ko/s avec un écart type de 239 Ko/s. Le débit maximal mesuré est de 1344 Ko/s et le débit minimal 502Ko/s. Le débit médian mesuré est 850 Ko/s et l'écart type est de 244.
- Pour les échanges bidirectionnels, le débit moyen mesuré est de 540 Ko/s les débits observés sur les périphériques variaient entre 416 Ko/s et 796 Ko/s. Le débit médian mesuré est de 513 Ko/s et l'écart type est de 124.

Ces mesures nous ont permis de réaliser que la bande passante dans les échanges bidirectionnels n'était pas tout le temps divisée par 2 comparée à un transfert unidirectionnel. Cependant, les débits observés sur les communications D2D ont un débit bas par rapport au débit théorique de leurs cartes réseau Wi-Fi et au débit qu'ils permettaient d'obtenir s'ils utilisaient le mode infrastructure. Ces valeurs que nous utilisons pour calibrer notre simulateur nous permettent d'obtenir des gains réalistes sur ce type de périphérique.

D'autres mesures ont été effectuées sur des périphériques sous le système d'exploitation Android. Pour cela, nous avons utilisé l'API Wi-Fi P2P d'Android. Sur les périphériques "Samsung Galaxy S7 edge" et "iRULU eXpro 2 Plus", nous avons observé des débits entre 6200Ko/s et de 3200Ko/o. Ces débits sont d'ailleurs plus élevés que sur les tablettes Apple. En pratique, il est impossible de faire communiquer des périphériques mobiles utilisant différents systèmes (iOS et Android principalement) en D2D en Wi-Fi de manière automatisée car l'implémentation est différente sur chaque famille de périphériques. Nous pouvons cependant envisager d'avoir dans un futur proche la possibilité de faire communiquer des périphériques de ces deux familles.

Dans cette thèse, nous nous intéressons au cas d'étude de Magency. Pour cela, nous simulons donc une dissémination représentative des cas que Magency pourrait avoir. Les réglages par défaut du simulateur sont illustrés dans le tableau V.2. Cette simulation consiste en la dissémination d'une donnée (un fichier) de 100 Mo sur un réseau de 100 périphérique. La donnée est découpée en 20 chunks (taille d'un chunk = 5 Mo), à chaque téléchargement depuis le serveur, un périphérique ne récupère qu'un chunk à la fois. Le degré de parallélisme par défaut est de 33 périphériques, ce qui représente 16 échanges D2D avec un transfert depuis le serveur.

Dans un premier temps, nous simulons notre réseau avec les débits que nous avons mesurés avec les tablettes Apple iPad Air2 afin de connaître les gains que l'on pourrait avoir avec des périphériques offrant des débits similaires. Nous nous intéressons uniquement au gain que l'on peut avoir en termes de temps de complétion. Par la suite, nous simulerons des réseaux hétérogènes dans lesquels nous évaluerons l'impact de l'utilisation de périphériques avec des débits et des batteries différents.

Paramètre	Valeur
Taille du réseau	100 périphériques
Taille de la donnée	100 Mo
Nombre de chunks	20
Degré maximal de parallélisme	33 périphériques
Nombre maximal de chunks par envoi du serveur	1

Tableau V.2: Configuration par défaut du simulateur

### V.1.3.2 Hétérogénéité des périphériques

Pour l'évaluation de nos différents algorithmes, nous utilisons le même simulateur que nous avons étendu pour prendre en compte l'hétérogénéité des périphériques. Le tableau V.3 représente la configuration par défaut de la simulation que nous allons avoir.

Afin de voir le comportement de nos différents algorithmes, nous simulons des disséminations plus longues avec un réseau de 200 périphériques qui récupèrent une donnée de 500Mo avec un degré de parallélisme permettant à 33 périphériques d'échanger en parallèle.

En nous basant sur des observations faites sur des périphériques mobiles utilisés en production par Magency. Les *bons* périphériques que nous simulons ici disposent d'une autonomie est de 12 heures ; les *mauvais* périphériques, ces derniers disposent d'une autonomie de 8 heures qui est aussi le seuil minimal *minBattery* en dessous duquel une batterie doit être préservée d'une surexploitation en échanges D2D. Cela nous permet de découper le réseau en deux groupes (*bons* et *mauvais*) en fonction de leur batterie.

En ce qui concerne la bande passante. Nous considérons que les périphériques *rapides* que nous simulons offrent un débit de 4000Ko/s, qui est la bande passante minimale *minFastBW*. Nous divisons ainsi chaque groupe en deux sous-groupes (*rapides* et *lents*) en fonction de leurs bande passante. Avec les périphériques récents, il y a une corrélation entre la qualité des batteries et les bandes passantes. Ils sont équipés de cartes réseau Wi-Fi compatibles avec les dernières normes et offrant des débits élevés, ainsi que de batteries dont l'autonomie est en constante progression. Nous avons moins de chances de trouver des périphériques avec de mauvaises batteries et des bandes passantes élevées.

Paramètre	Valeur
Taille du réseau	200
Taille de la donnée	500 Mo
Nombre de chunks	20
Degré maximal de parallélisme	33 périphériques
Autonomie d'une bonne batterie	12 h
Autonomie d'une mauvaise batterie	8 h
Seuil minimal de batterie <i>minBattery</i>	8 h
Bande passante rapides	4000 Ko/s
Bande passante lents	8 Mbps
Débit minimal pour les périphériques rapides	4000 Ko/s

Tableau V.3: Configuration du simulateur pour l'évaluation de la prise en compte de l'hétérogénéité des périphériques mobiles.

## V.2 Évaluation de l'algorithme de dissémination D2D

### V.2.1 Degré de parallélisme

Le degré de parallélisme constitue un facteur important de performance pour notre approche, car plus le parallélisme permis sur le réseau est élevé, plus le nombre d'échanges D2D pouvant être déclenchés l'est aussi. On augmente ainsi la bande passante agrégée du réseau, ce qui permet de réduire de manière considérable le temps de complétion de la dissémination.

Ce paramètre représente la capacité du système à paralléliser les échanges D2D. Il dépend fortement de la qualité du signal sur le réseau et du niveau d'occupation des différents canaux Wi-Fi. Les points d'accès peuvent en outre être utilisés pour scanner les différents canaux afin de choisir ce niveau de parallélisme. Par exemple : Lee et al. [59, 60] (MoVi) utilisent les points d'accès pour évaluer l'état du réseau et d'adapter leur système aux fluctuations des interférences Wi-Fi afin de déclencher des échanges entre les périphériques mobiles pour lorsque la qualité du signal Wi-Fi permet un échange à un débit acceptable. Ce type de solution peut s'adapter à notre cas pour définir les canaux les moins encombrés qui pourront être affectés pour des échanges D2D.

Afin d'évaluer l'impact du degré de parallélisme sur les performances de la dissémination, nous avons mesuré le temps de complétion et l'évolution du nombre de périphériques complets en faisant varier le degré de parallélisme. La donnée que l'on cherche à diffuser ici est découpée en 20 chunks de 5Mo.

Sur la figure V.1a, on peut observer que le temps de complétion diminue lorsqu'on augmente le degré de parallélisme et permet jusqu'à 60% de gain si l'on

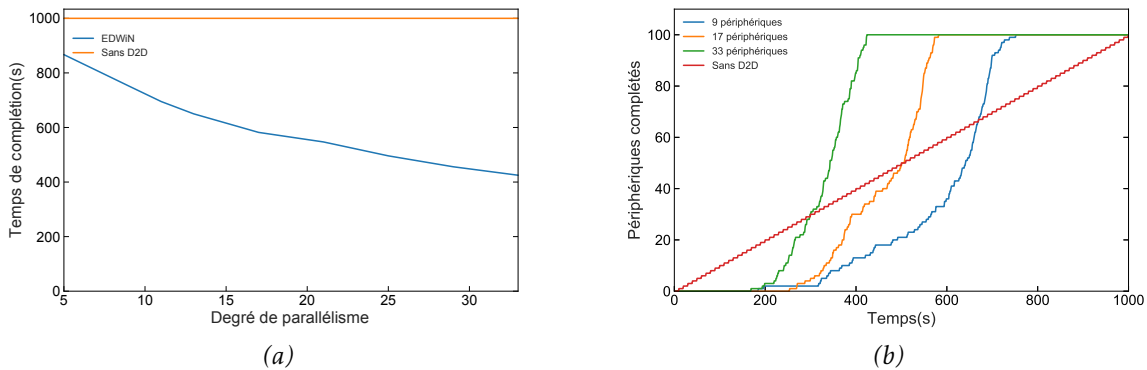


FIGURE V.1: (a) Impact du degré de parallélisme sur le temps de complétion (b) Évolution du remplissage des chunks

autorise 33 périphériques à communiquer, ce qui représente 16 échanges simultanés. La figure V.1b illustre l'évolution du pourcentage de périphériques complétés dans le temps pour différents degrés de parallélisme. On observe que plus le degré de parallélisme est élevé, plus vite des périphériques obtiennent la totalité de la donnée car on dispose de plus de sources de données. Rappelons ici que l'on considère le cas idéal d'absence d'interférences. En pratique, il existe peut y avoir une limite au degré de parallélisme que l'on peut atteindre même en ayant peu d'interférences.

On peut remarquer aussi qu'au début de la dissémination, pendant un certain temps, aucun périphérique n'a reçu la totalité de la donnée lorsqu'on utilise notre approche alors que si on charge séquentiellement la donnée sur les périphériques, ces derniers se remplissent au fur et à mesure. Cet effet est dû à la politique de réplication "*Rarest first*" qui, à chaque transfert depuis le serveur, favorise la dissémination des chunks les moins diffusés sur les périphériques ayant le moins de chunks sur le réseau. Ainsi, notre approche va favoriser par la suite la sélection de paires de périphériques qui enverront la donnée dans son intégralité à un périphérique qui n'a encore rien reçu ou bien des échanges avec le plus grand nombre de chunks possibles. C'est cela permet de disséminer plus efficacement sur le réseau car on échange un maximum de données en un minimum de connexions D2D.

## V.2.2 Passage à l'échelle

Dans un réseau Wi-Fi à grande échelle, la couverture est assurée par plusieurs points d'accès auxquels sont connectés les périphériques mobiles. Nous rappelons, que l'échelle maximale que nous visons est de l'ordre de quelques centaines de périphériques, ce qui représente la capacité moyenne d'un point d'accès. Ainsi, chaque point d'accès assure les connexions d'un ensemble de périphériques mobiles.



Les autres étant moins à sa portée, sont liés à d'autres points d'accès du réseau. Ainsi, la dissémination sur tous les périphériques du réseau se déroule, parallèlement, de la même manière pour tous les sous-réseaux couverts par les différents points d'accès. Nous nous concentrons donc sur ce qu'il se passe au niveau d'un seul point d'accès.

Pour étudier le passage à l'échelle de notre approche pour des périphériques connectés à un point d'accès, nous mesurons le temps de complétion en fonction du nombre de périphériques comme illustré dans la figure V.2.

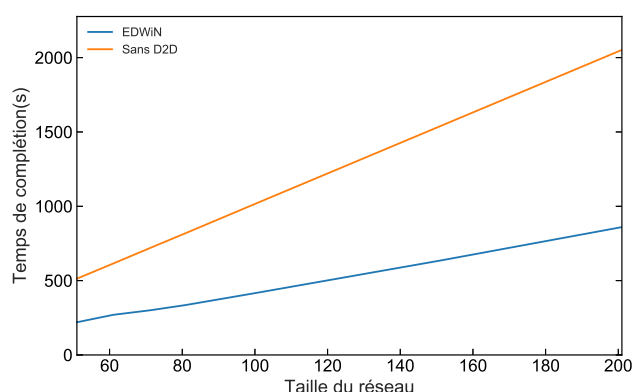


FIGURE V.2: Impact de la taille du réseau sur le temps de complétion.

La courbe de la figure V.2 montre que plus on augmente la taille du réseau, plus le gain en temps de complétion est important. Notre approche passe donc mieux à l'échelle.

Notons toutefois que la courbe du temps de complétion reste croissante pour EDWiN. En effet, le temps de complétion a triplé en agrandissant la taille du réseau. Nous rappelons ici que nous simulons un réseau dont le degré de parallélisme permet à 16 paires de périphériques d'échanger en D2D en parallèle en plus du serveur qui diffuse depuis le point d'accès (33 périphériques). Ce paramètre limite le nombre d'échanges parallèles de données.

Cependant, si l'on a un grand nombre de périphériques couverts par le point d'accès, la surface dans laquelle évoluent les périphériques serait plus grande et on pourrait avoir un degré de parallélisme plus élevé. Notre approche peut être adaptée dans ce cas pour en bénéficier et d'obtenir des gains plus importants.

## V.2.3 Volumes de données échangés

### V.2.3.1 Taille des chunks

La taille des chunks joue un rôle clé dans la vitesse d'apparition de nouvelles sources de données. Il faut qu'elle permette d'avoir à la fois une utilisation efficace des connexions D2D et qu'elle ne ralentisse pas le serveur dans la parallélisation des échanges D2D.

Afin d'étudier l'impact de la taille des chunks sur les performances de la dissémination, nous mesurons le temps de complétion en fonction de la taille des chunks que l'on fait varier de 1Mo (100 chunks) à 100Mo (1 chunk) pour un fichier de 100Mo, la figure V.3 illustre les temps de complétion.

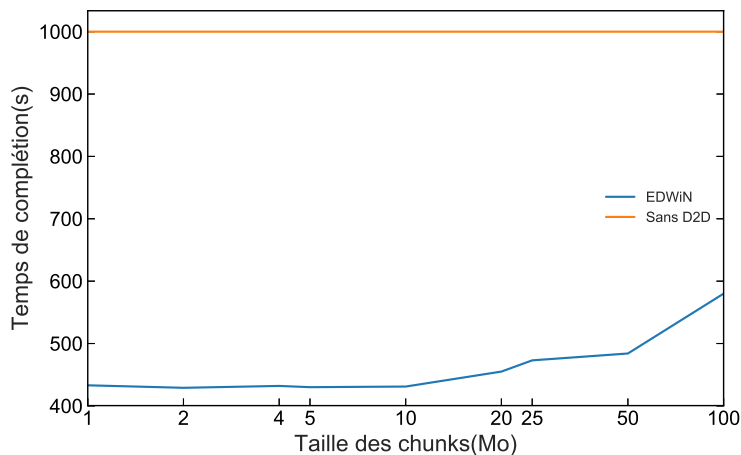


FIGURE V.3: Impact de la taille des chunks sur le temps de complétion

On peut observer que lorsque les chunks sont de grande taille, le temps de dissémination croît. En revanche, plus la taille des chunks diminue, plus le temps de dissémination diminue.

Lorsque la taille des chunks est grande, le réseau met plus de temps pour atteindre le degré de parallélisme. Dans le cas contraire, plus la taille de la donnée est petite, plus le temps de dissémination est court. De nouvelles sources de données (incomplètes) apparaissent plus vite et permettent au réseau de rapidement atteindre son degré de parallélisme.

On observe également que pour notre configuration, en dessous de 10Mo par chunk (nombre chunks égal à 10), le temps de complétion se stabilise à la même valeur. Cependant, cela nécessitera l'établissement de plus de connexions D2D et d'avoir un nombre accru d'échanges de petits volumes de données, et ce particulièrement au début de la dissémination.

### V.2.3.2 Délais d'établissement de connexions D2D

Chaque échange entre deux périphériques mobiles implique l'établissement d'une communication D2D. Plus son établissement est long, plus les périphériques mettent du temps avant d'échanger, ce qui ralentit globalement la dissémination sur le réseau. Nous observons sur la courbe de la figure V.4 que plus ce temps se prolonge, plus on se rapproche des performances d'une approche centralisée.

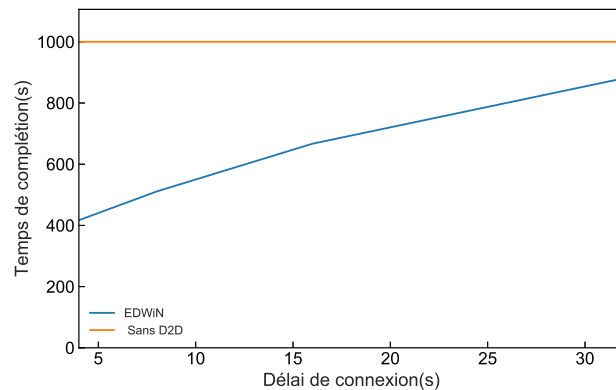


FIGURE V.4: Impact du temps d'établissement de connexion sur les performances de la dissémination.

En pratique, la présence d'interférences rend difficile pour les périphériques mobiles d'établir des connexions D2D, ce qui peut prolonger leur délais d'établissement ou même nécessiter plusieurs tentatives avant d'établir un échange D2D. La dissémination en D2D dans ces cas peut s'avérer moins efficace car le temps de connexion est trop grand pour être utilisé efficacement et les transmissions de données D2D pourraient subir des latences trop importantes.

### V.2.3.3 Nombre maximal de chunks par échange

Comme la taille des chunks, le nombre de chunks par envoi a également un impact sur le rapport entre le temps de connexion et temps de transfert. Il peut aussi impacter le temps d'apparition des sources de données. Nous mesurons l'impact de la variation de deux facteurs qui sont  $Max_{AP}$ , qui est le nombre maximal de chunks envoyés dans un transfert depuis le serveur, et  $Max_{D2D}$  qui est le nombre maximal de chunks échangés entre les différents périphériques.

Dans la figure V.5, nous mesurons le temps de complétion en faisant varier  $Max_{AP}$ , le nombre maximal de chunks par envoi depuis le serveur. On observe que plus ce nombre est grand plus le temps de complétion augmente. Le serveur, en envoyant un volume de données plus important aux clients, met plus de temps à répliquer des chunks sur les différents périphériques. Cela ralentit l'arrivée du réseau à son degré de parallélisme maximal.

Comme nous l'avons mentionné plus haut, il est important d'exploiter les connexions D2D de manière efficace. Pour cela, définir le nombre de chunks échangés entre les périphériques est aussi important pour la dissémination. La figure V.6 illustre l'impact de la valeur de  $Max_{D2D}$ . On observe que la configuration optimale pour notre cadre de simulation est de 5 chunks. Lorsqu'on augmente ce nombre, on gagne toujours en temps de complétion comparé à une approche centralisée mais celui-ci augmente légèrement. En effet, si on augmente la taille des données échangées entre les périphériques, on gagne en efficacité d'utilisation des

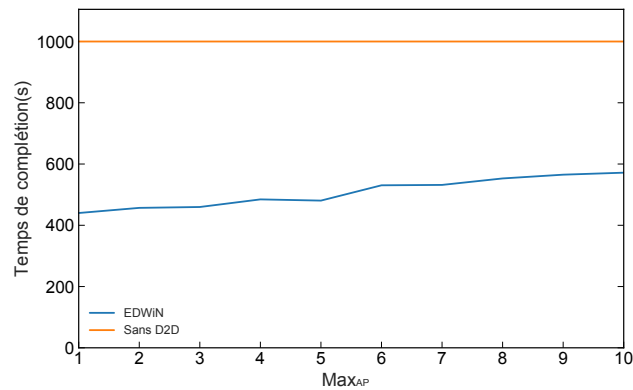


FIGURE V.5: Impact du nombre de chunks téléchargés depuis le serveur  $Max_{AP}$ .

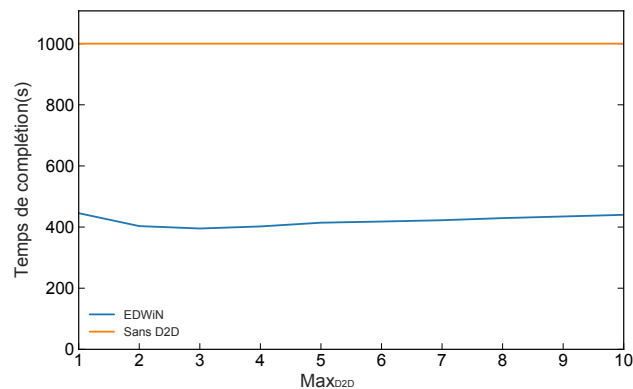


FIGURE V.6: Impact du nombre maximum de chunks par échange D2D  $Max_{D2D}$ .

connexions D2D car le temps d'établissement de connexion est rentabilisé. Dans le cas où les périphériques mobiles ont un débit beaucoup plus bas que celui du serveur, il serait intéressant de réduire le nombre maximal de chunks échangés entre les périphériques et afin de favoriser la connexion avec le serveur.

Le serveur, peut dans certains cas se retrouver dans un état où il est inactif sur le réseau en attendant que des échanges D2D se terminent. C'est par exemple le cas vers la fin d'une dissémination lorsqu'il reste encore quelques périphériques qui finissent de s'échanger leurs données. Si le débit du serveur permet d'acheminer plus mes chunks restants, il peut interrompre les échanges D2D afin de finir envoyer les chunks restants directement aux périphériques concernés.

## V.2.4 Énergie et contrôle de l'activité des périphériques mobiles

En utilisant les communications D2D pour accélérer la dissémination, le temps d'activité sur le réseau d'un périphérique mobile peut varier en fonction du moment

à partir duquel il peut être utilisé comme source de données et du nombre de fois qu'il est sollicité pour des échanges.

Lorsque les débits des périphériques mobiles ne sont pas du même ordre de grandeur que le débit assuré par le serveur (tableau V.1), la transmission de données par les périphériques mobiles est plus longue et donc plus coûteuse en énergie. Si on ne contrôle pas le niveau de participation des périphériques, cela peut provoquer une importante consommation d'énergie liée à la communication sur le réseau.

Pour mieux comprendre cela, nous mesurons les temps d'activités de chaque périphérique et étudions l'impact de la limitation du temps d'activité sur les performances de la dissémination ainsi que pour la répartition des temps d'activités des périphériques sur le réseau. La figure V.7 illustre les temps d'activités mesurés pour une dissémination d'un fichier de 100Mo sur 100 périphériques mobiles avec un degré de parallélisme de 33. Comme nous pouvons le voir, en ne limitant pas l'activité des périphériques sur le réseau, 80% d'entre eux participent au moins pendant la moitié de la durée de la dissémination.

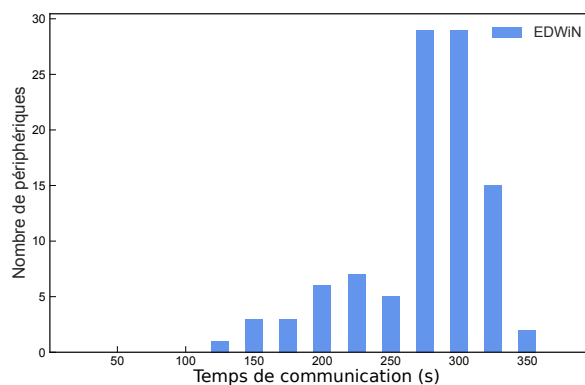


FIGURE V.7: Comparaison des temps d'activité sur le réseau sans clustering.

Une stratégie de regroupement (clustering) permet de restreindre la dissémination depuis le serveur à un seul groupe et donc de réduire la durée d'activité des périphériques mobiles (voir section IV.1). La limitation de cette durée n'est pas sans impact sur les performances du réseau. En effet dans ce cas, la dissémination des données peut ne pas bénéficier complètement du parallélisme sur le réseau. La figure V.8a illustre une comparaison de l'évolution de la complétion des périphériques dans le temps pour trois différentes approches de dissémination :

- Une approche séquentielle centralisée : envoi depuis le serveur uniquement, périphérique par périphérique ;
- Notre approche D2D sans clustering ;
- Notre approche D2D avec clustering ;

On observe que, pour des clusters de 20 périphériques, la restriction de la dissémination à un seul groupe rallonge la durée de la dissémination sur le réseau. En effet, lorsque l'on réduit le parallélisme sur le réseau, on diminue les performances

de la dissémination car les communications D2D ne sont pas exploitées efficacement.

Cependant, la limitation de l'activité des périphériques permet d'avoir des temps de complétion en dessous de la moitié de la durée de la dissémination. Comme nous pouvons le constater sur la figure V.8b, 85% des périphériques passent moins de 50% de la durée de la dissémination lorsqu'on limite la durée d'activité des périphériques mobiles.

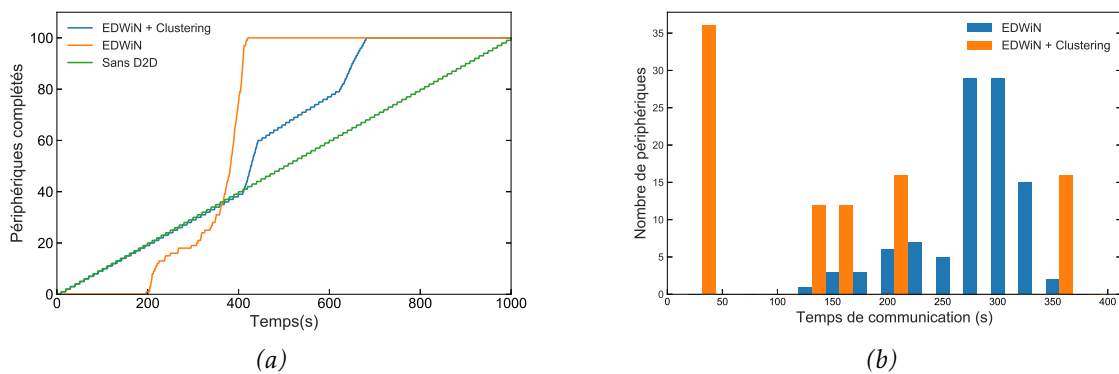


FIGURE V.8: (V.8a) Comparaison de l'évolution de la complétion des périphériques; (V.8b) Comparaison des temps d'activité sur le réseau des différentes approches.

Il y a donc un compromis entre consommation énergétique et performance. Dans ce cas, on peut l'ajuster en changeant la taille des clusters. Rappelons ici que le degré de parallélisme est limité afin de limiter les interférences. Le clustering permet de sélectionner les périphériques autorisés à communiquer sur le réseau et en D2D.

## V.2.5 Prise en compte de l'hétérogénéité des batteries

Nous avons étendu notre simulateur présenté dans la section V.1.1 pour prendre en compte l'hétérogénéité des batteries et des bandes passantes des périphériques.

L'approche présentée dans la section IV.2 considère que les périphériques dont l'autonomie de batterie est inférieure à un seuil minimal ne doivent pas être utilisés pour les échanges D2D entre les périphériques mobiles du réseau. La métrique "*temps de complétion*" peut alors être impactée par la proportion de mauvais périphériques sur le réseau. Du point de vue de la consommation d'énergie, l'objectif principal est de diminuer la consommation d'énergie sur les "*mauvais*" périphériques en réduisant leur niveau de participation à la dissémination, et ce quitte à avoir une consommation accrue sur l'ensemble du réseau.

Notre objectif dans ce cas est d'évaluer le gain que peut apporter la considération de l'hétérogénéité des périphériques mobiles en termes de performance et de consommation d'énergie. Nous comparons ici les deux approches présentées dans la section IV.2 avec l'algorithme de dissémination de base (section III.2). Nous étudions d'abord uniquement des réseaux hétérogènes au niveau des batteries

des périphériques. Ensuite, nous évaluons l'approche qui prend en compte l'hétérogénéité des périphériques en termes de batterie et de bande passante afin de montrer l'impact que peut avoir l'exploitation de périphériques rapides pour accélérer la dissémination de données, en particulier au début pour rapidement obtenir des sources complètes de la donnée disséminée.

### V.2.5.1 Activité des périphériques sur le réseau

Afin d'évaluer l'impact de la protection des batteries faibles sur l'activité des périphériques, nous mesurons la différence des niveaux de batterie à la fin de la dissémination.

Pour cela, nous comparons l'activité engendrée par la dissémination sur tous les périphériques du réseau. Puis, nous calculons ensuite la moyenne globale et la moyenne pour les *mauvais* périphériques. Nous montrons également la batterie la plus faible sur le réseau.

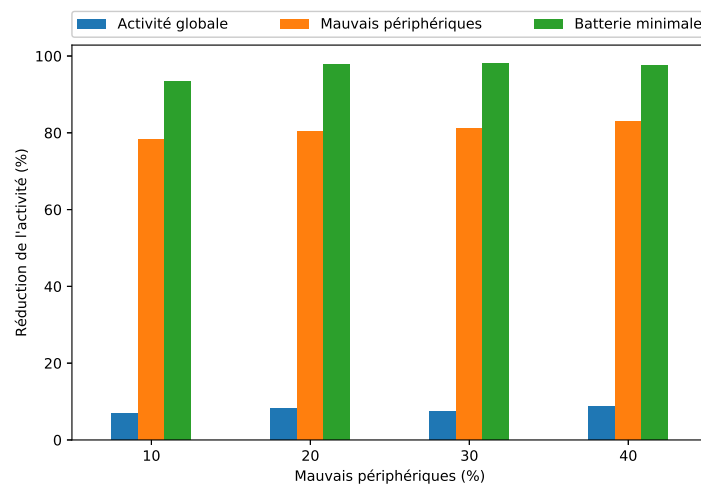


FIGURE V.9: Réduction de l'activité des périphériques sur le réseau avec la prise en compte de l'hétérogénéité des batteries.

Comme l'illustre la figure V.9, nous observons que l'activité des *mauvais* périphériques peut être réduite de l'ordre de 80% en comparaison avec la non prise en compte de l'hétérogénéité d'énergie. En ce qui concerne la batterie la plus faible du réseau, la réduction du temps d'activité, plus importante, est de l'ordre de 97%.

Lorsque l'on protège les *mauvais* périphériques d'échanges D2D potentiellement longs et coûteux, on peut s'attendre à avoir une augmentation de l'activité moyenne sur le réseau. On peut cependant voir que dans notre cas, on peut observer gain sur la globalité du réseau. Nous verrons que cela est due une meilleure utilisation du point d'accès. Nous avons simulé l'évolution de la batterie sur le réseau et nous avons comparé l'autonomie que l'on pourrait avoir en fin de dissémination en fonction de

l'approche utilisée. La figure V.10a illustre la différence d'autonomie estimée en fin de dissémination de données lorsque l'hétérogénéité des périphériques est prise en compte. Nous comparons l'autonomie finale estimée pour (i) tout le réseau, (ii) les *mauvais* périphériques et (iii) la batterie minimale sur le réseau. Nous observons que l'on gagne considérablement en autonomie de batterie sur le réseau pour les périphériques dont les batteries sont les plus faibles (entre 1.5 et 2 heures pour notre simulation). En effet, en préservant ceux-ci des échanges D2D, ils ne subissent pas les coûts d'une surexploitation sur le réseau. On peut d'ailleurs le voir sur la figure V.10b, dans laquelle on compare l'évolution de l'autonomie minimale sur le réseau en fonction de la prise en compte ou non de la batterie. On observe que lorsqu'on ne prend pas en compte la batterie des périphériques, la batterie se décharge plus vite. Les paliers de baisse de batterie sur EDWiN sont dûs aux échanges D2D.

Les *mauvais* gagnent en moyenne quelque dizaines de minutes d'autonomie pour notre simulation. En ce qui concerne la consommation globale du réseau, on observe une baisse globale de l'autonomie lorsque le taux de mauvais périphériques augmente au delà de 30% (figure V.10a). Celle-ci est due à une sollicitation plus importante des *bons* périphériques qui passent plus de temps à communiquer sur le réseau et donc consomment plus d'énergie.

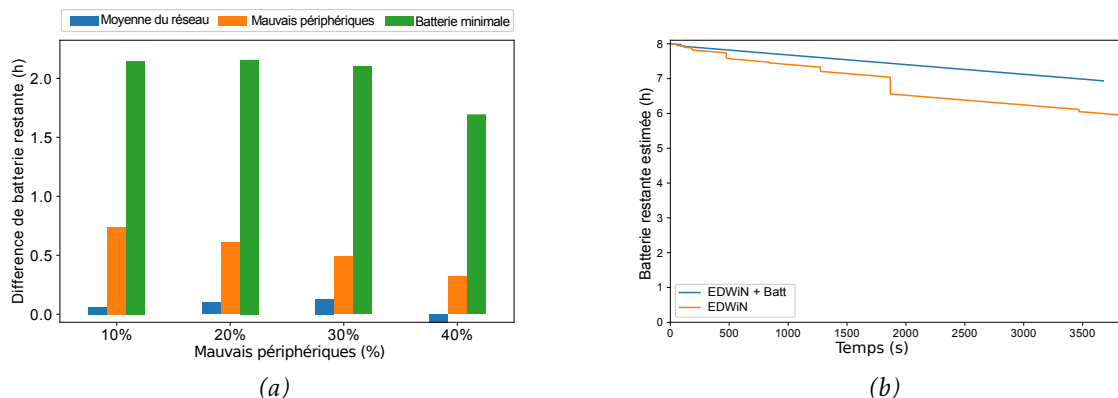


FIGURE V.10: (V.10a) Différence de batterie finale en fin de dissémination avec la prise en compte l'hétérogénéité des batteries; (V.10b) Évolution de la batterie minimale sur le réseau pour un réseau dont 20% des périphériques sont *mauvais*.

### V.2.5.2 Impact sur les performances de la dissémination

On peut s'attendre, en disséminant la donnée séquentiellement sur un ensemble des périphériques du réseau, à ce que la dissémination perde en performance. Or, ce n'est pas forcément le cas tout le temps.

En effet, grâce à la séquentialisation des téléchargements sur les *mauvais* périphériques depuis le serveur, ces derniers sont non seulement protégés d'échanges D2D plus longs et plus coûteux en consommation énergétique, mais bénéficient aussi



d'une source de données avec un débit plus élevé.

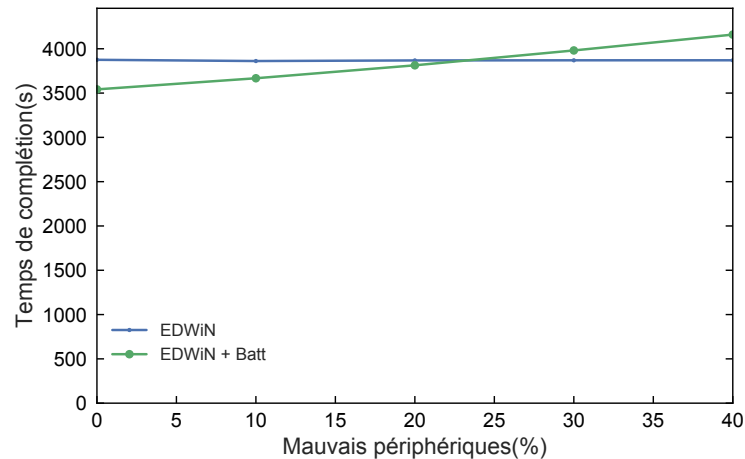


FIGURE V.11: Comparaison des temps de complétion.

En transférant la donnée en un seul transfert vers un *mauvais* périphérique, on peut par la suite le mettre en veille jusqu'à la fin de la dissémination. De plus, le serveur change ainsi moins d'un périphérique à l'autre, ce qui peut offrir un gain en temps de complétion. Cela est dû à la réduction du nombre de connexions D2D pour les échanges de données et à une meilleure exploitation de la bande passante du serveur.

La courbe de la figure V.11 illustre l'impact de la prise en compte de l'hétérogénéité des batteries sur le temps de complétion. Nous comparons les temps de complétion obtenus avec les deux approches (avec et sans prise de l'hétérogénéité des batteries) en fonction du pourcentage de *mauvais* périphériques sur le réseau.

On observe qu'en dessous de 30% nous avons des temps de complétion légèrement meilleurs que ceux que l'on obtient sans considération des *mauvaises* batteries. La prise en compte de l'hétérogénéité des batteries permet dans ce cas de gagner en performance. Cependant, pour une proportion de *mauvais* périphériques plus grande (40%), nous perdons en performance. Cela est dû à une monopolisation du serveur par les *mauvais* périphériques qui l'empêchent de contribuer à la dissémination sur les bons périphériques. La taille de la donnée disséminée et le nombre de *mauvais* périphériques rallongent ici la durée de dissémination, ce qui engendre une baisse de performance.

### V.2.5.3 Distribution des temps de dissémination

Nous avons décrit dans la section IV.2 que le serveur alternait entre deux phases dans lesquelles il dissémine respectivement sur les *bons* et les *mauvais* périphériques.

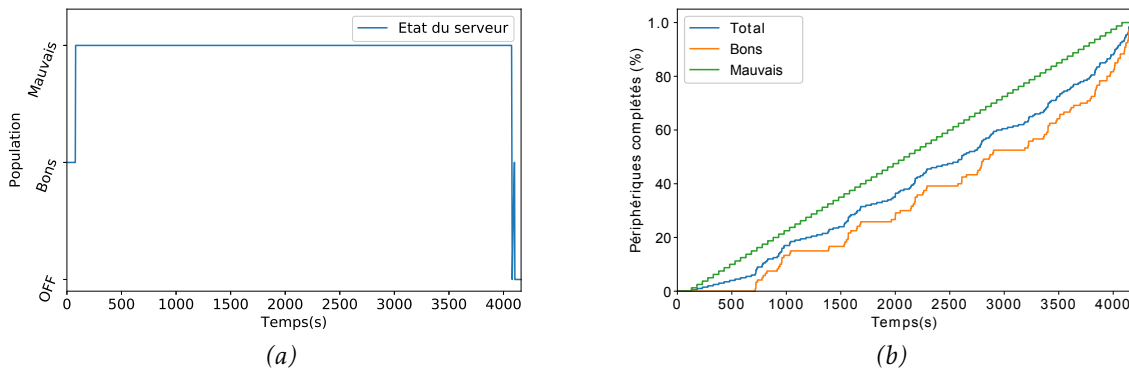


FIGURE V.12: (V.12a) Phases de dissémination du serveur avec la prise en compte de l'hétérogénéité des batteries pour un réseau avec 40% de mauvais périphériques et (V.12b) évolution de la complétion des périphériques.

La figure V.12a illustre le temps passé par le serveur sur chaque phase durant la dissémination pour un réseau dont 40% des périphériques ont des batteries à protéger. On observe que les *mauvais* périphériques monopolisent ici le serveur qui est leur unique source de données. En effet, celui-ci passe presque toute la durée de la dissémination à transférer la donnée sur ces périphériques. Les *bons* périphériques, privés de la bande passante plus élevée du serveur, mettent plus de temps à la récupérer (figure V.12b).

Lorsque la proportion des mauvais périphériques n'est pas trop importante, on gagne en performance car on parallélise plus les transferts D2D, et on exploite plus efficacement les deux types de source de données possibles sur le réseau. Nous illustrons dans ce cas un réseau dans lequel les *mauvais* périphériques représentent 20% des périphériques du réseau (figures V.13a et V.13b).

La courbe V.13a illustre l'alternance des phases pour un réseau comptant 20% de *mauvais* périphériques. Au début de la dissémination, le serveur envoie uniquement vers les *bons* périphériques. Dès que le degré maximal de parallélisme est atteint, ce dernier passe à la deuxième phase durant laquelle il envoie des données aux *mauvais* périphériques. Pendant ce temps, les périphériques se remplissent au fur et à mesure. Une fois que tous les *mauvais* périphériques ont récupéré la donnée, le serveur repasse sur le groupe des *bons* périphériques afin de les aider à récupérer la donnée plus rapidement. Sur la courbe V.13b, on observe que la totalité des *mauvais* périphérique finissent de récupérer la donnée avant la fin de la dissémination. Lorsque le serveur repasse sur les *bons* périphériques, ces derniers récupèrent la donnée plus rapidement. L'avantage de cette approche, est que les *mauvais* périphériques peuvent avoir une économie d'énergie conséquente s'ils sont mis en veille dès qu'ils ont reçu la donnée dans son intégralité. Il est également assez simple d'estimer l'heure à laquelle un périphérique peut être utilisé. Grâce à cette

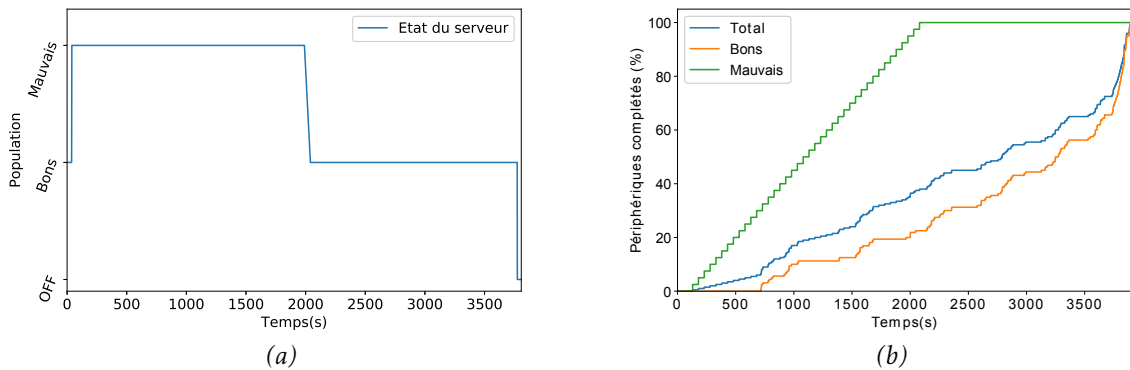


FIGURE V.13: (V.13a) Phases de dissémination par serveur avec prise en compte de l'hétérogénéité des batteries pour un réseau avec 20% de *mauvais* périphériques et (V.13b) évolution de la complétion des périphériques.

connaissance, certains périphériques peuvent être mis en veille pendant toute la durée pendant laquelle ils ne participent pas à la dissémination.

Nous venons de voir que d'empêcher les *mauvais* périphériques d'échanger en D2D n'est pas la meilleure solution dans certains cas. Lorsque le taux de *mauvais* périphériques est élevé (autour de 40%), les performances de la dissémination se dégradent (voir figure V.11). Toutefois, en dessous d'une certaine proportion, l'impact de cette approche est positif, car on exploite les deux types de source de données plus efficacement.

## V.2.6 Prise en compte de l'hétérogénéité des batteries et des bandes passantes

Comme nous l'avons vu, en plus d'être hétérogènes en termes de batteries, les périphériques mobiles le sont aussi au niveau de leur bande passante (ainsi que pour d'autres caractéristiques comme l'espace de stockage, fréquence du processeur, ...). En effet, plus les périphériques sont de fabrication récente, plus ils ont de chance d'être équipés de puces Wi-Fi aux normes les plus récentes et donc bénéficier de débits élevés. Cette caractéristique peut être utilisée pour améliorer les performances de la dissémination.

Nous étudions ici l'impact de la prise en compte de périphériques *rapides* sur lesquels on peut s'appuyer pour accélérer la dissémination et réduire la consommation d'énergie sur le réseau. Nous évaluons ici l'algorithme présenté dans la section IV.3. Dans ce cas, les *mauvais* périphériques peuvent utiliser les communications D2D mais uniquement pour télécharger les données.

### V.2.6.1 Performance de la dissémination

Dans un premier temps, nous souhaitons étudier l'impact des bandes passantes élevées sur le temps de complétion de la dissémination. Nous partons ici de notre pire cas de la section précédente, celui avec 40% de *mauvais* périphériques sur le réseau. Rappelons qu'un périphérique *bon-rapide* peut devenir un *mauvais* périphérique si l'autonomie de sa batterie se décharge et passe en dessous du seuil minimal *minBattery*.

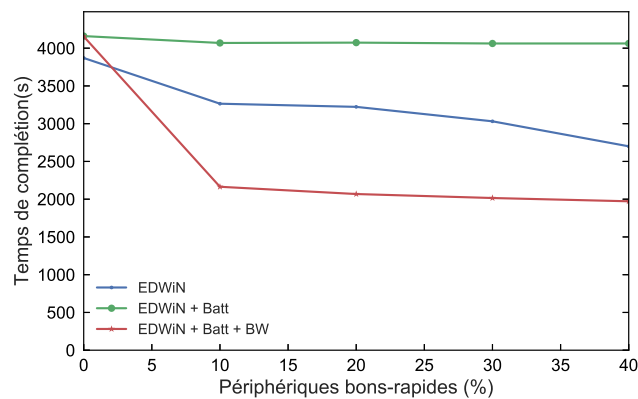


FIGURE V.14: Impact de la prise en compte de la bande passante sur le temps de complétion.

La figure V.14 illustre une comparaison du temps de complétion en fonction du pourcentage de périphériques *bons-rapides* sur le réseau pour les trois approches que nous avons présentées dans le chapitre III. On observe qu'à partir d'un petit pourcentage de ces périphériques sur le réseau (10%) améliore considérablement les performances de la dissémination. En effet, favoriser les périphériques *bons-rapides* permet d'avoir rapidement des sources de données complètes avec une bonne bande passante. De plus, l'utilisation des connexions D2D des *mauvais* périphériques uniquement pour recevoir les données permet de réduire le temps passé par le serveur à leur envoyer les données et à maintenir le parallélisme des échanges sur le réseau.

### V.2.6.2 Énergie

La figure V.15 montre le gain en énergie que l'on peut avoir sur le réseau en exploitant les périphériques rapides. Étant donné le fait que l'on utilise les *mauvais* périphériques pour recevoir la donnée en D2D, on peut s'attendre à une augmentation de la consommation d'énergie sur le réseau. On observe ici, que l'on gagne en moyenne en consommation d'énergie sur le réseau grâce au gain de temps de dissémination apporté par les périphériques *bons-rapides*. Pour notre configuration, on a des gains moindres en dessous de 30% de périphériques *bons-rapides*. Avec 40% de mauvais périphériques, on gagne en moyenne 8.4% de batterie sur le réseau. Pour les mauvais périphériques, on gagne en moyenne 11,3%

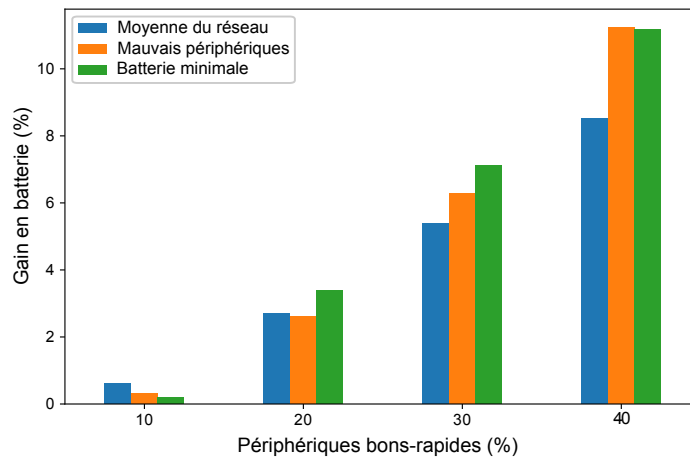


FIGURE V.15: Gain énergétique de la prise en compte de la batterie et de la bande passante comparée à la prise en compte de la batterie (40% de *mauvais* périphériques).

comparé à si on considérait seulement la batterie des périphériques.

Comme pour l'approche qui prend uniquement compte la batterie, on peut aisément connaître le temps à partir duquel les mauvais périphériques commencent à télécharger la donnée. Cette connaissance, peut, en pratique, permettre d'économiser de l'énergie en mettant en veille les *mauvais* périphériques en attendant l'instant à partir duquel il commenceront à télécharger la donnée.

## Conclusion

En utilisant des connexions D2D, on peut améliorer les temps de dissémination de données sur les périphériques d'un réseau Wi-Fi. L'ajustement du degré de parallélisme a un impact sur les performances de dissémination car il définit la bande passante globale du réseau. En favorisant les appariements dans lesquels on échange le plus d'informations entre les périphériques, on augmente l'efficacité d'utilisation des connexions D2D.

Nous venons de voir l'impact positif que peut avoir l'utilisation de connexions D2D entre les périphériques mobiles d'un réseau Wi-Fi sur la dissémination d'une donnée volumineuse. Nos évaluations reposent sur des mesures réelles, elles montrent des résultats intéressants pour les configurations que nous avons simulées. Nous avons pu voir que notre approche passe mieux à l'échelle qu'une dissémination par Wi-Fi seulement. Les gains que l'on peut obtenir sont plus importants lorsque le degré maximal de parallélisme sur le réseau est grand.

Le découpage en chunks permet d'augmenter le nombre de sources sur le réseau et donc de paralléliser d'avantage. Cependant, une taille de chunks trop grande

peut dégrader les performances du réseau car on exploite moins le parallélisme sur le réseau et les transferts peuvent être longs. Nous avons aussi pu voir que les performances se stabilisent autour d'une certaine valeur à partir d'une certaine taille (10Mo par chunk pour un fichier de 100Mo), il n'est donc pas nécessaire d'avoir des chunks de trop petite taille. En effet, en plus d'augmenter le nombre d'établissements de connexions entre les périphériques mobiles, elle augmente aussi le nombre d'appariements nécessaires à la dissémination.

En pratique, les périphériques mobiles sont hétérogènes en batterie et en bande passante. L'exploitation de cette caractéristique sur le réseau permet à la fois d'avoir une utilisation plus efficace des connexions D2D et du point d'accès. Les périphériques échangent plus de données en moins de connexions avec leurs voisins. Une protection stricte de périphériques dont les batteries sont faibles (*mauvais*) peut cependant engendrer des pertes de performance. Dans ce cas, le serveur est monopolisé par une dissémination séquentielle qu'il effectue sur nombre important de périphériques qui l'ont pour seule source de données. Il passe donc moins de temps à participer à la dissémination sur tous les périphériques du réseau, ce qui peut entraîner son ralentissement.

Dans le cas où le réseau contient des périphérique mobiles capables de fournir un débit de communication D2D plus élevé. Il peut être intéressant d'exploiter ces derniers pour accélérer la dissémination. En mettant la priorité sur ces bons périphériques pour qu'ils deviennent des sources complètes de données, il y aura plus d'échanges D2D efficaces qui envoient la donnée en un seul transfert.



## CHAPITRE

## VI

Topologies hiérarchiques pour la  
dissémination de données :  
regroupement (clustering) sur les  
réseaux Wi-Fi Direct

---

**Sommaire**

---

<b>VI.1 Clustering basé sur Wi-Fi Direct : approche par SDN . . . . .</b>	<b>94</b>
VI.1.1 Architecture du réseau . . . . .	94
VI.1.2 Algorithme de clustering . . . . .	96
VI.1.3 Gestion des interférences et planification de la transmission	99
VI.1.4 Calcul du nombre optimal de clusters . . . . .	101
<b>VI.2 Évaluation des performances . . . . .</b>	<b>102</b>
VI.2.1 Performance du clustering . . . . .	102
VI.2.2 Simulation avec OMNet++ . . . . .	104
<b>Conclusion . . . . .</b>	<b>108</b>

---



Ce chapitre présente une contribution qui a été réalisée dans le cadre d'une collaboration entre mon équipe (DELYS<sup>1</sup>.) et l'équipe Réseau (*Phare*) du LIP6, pour le stage de Master 2 de Syphax Iskounen [127] que j'ai participé à encadrer. À la suite de son stage, ce travail a été étendu et publié dans le journal "*Annals of Telecommunications*" (AoT) [11]. Nous présentons ici une approche qui répond à une problématique proche de celles des chapitres III et IV. Cette contribution se concentre sur le cas particulier que l'on peut trouver pendant des conférences ou lors de cours magistraux en amphithéâtre où les périphériques ne se déplacent pas voir très peu, les participants étant généralement assis. Nous exploitons ce côté statique de la localisation des périphériques afin de construire une topologie hiérarchique pour la diffusion de données et regrouper les périphériques par petits groupes (clusters) géographiquement proches qui communiqueront via une connexion Wi-Fi Direct. Ce regroupement permettra ainsi d'ajuster la puissance du signal au sein des groupes afin de réduire les interférences entre les groupes.

La suite de ce chapitre se déroule comme suit : dans la section VI.1, nous présentons l'architecture du réseau et discutons de la mise en œuvre d'une approche pour partitionner le réseau ainsi qu'une méthode pour le calcul du nombre optimal de clusters. La section VI.2 présente une évaluation de la performance du clustering ainsi qu'une simulation d'un réseau typique avec OMNet++. Enfin, nous concluons ce chapitre.

## VI.1 Clustering basé sur Wi-Fi Direct : approche par SDN

SDN (Software-Defined Networking), d'après la définition de [128], est une approche de conception, d'implémentation et de gestion des réseaux qui sépare le contrôle du réseau de la transmission de données afin d'offrir à l'utilisateur la meilleure expérience possible. Cette segmentation du réseau offre de nombreux avantages en terme de flexibilité et de contrôlabilité du réseau. En effet, l'ouverture du contrôle du réseau aux applications permet entre autres la possibilité d'avoir une gestion plus efficace du réseau. L'approche que nous décrivons ici suppose l'utilisation des SDN pour la gestion et le contrôle de la dissémination de données.

### VI.1.1 Architecture du réseau

La figure VI.1 illustre une architecture basée sur l'utilisation du paradigme SDN pour une dissémination de données sur des réseaux Wi-Fi Direct. Elle est constituée de deux couches : la couche de contrôle (SDN) et la couche d'accès aux données, qui est constituée des points d'accès et des périphériques mobiles qui y sont connectés.

---

1. Anciennement *Regal*

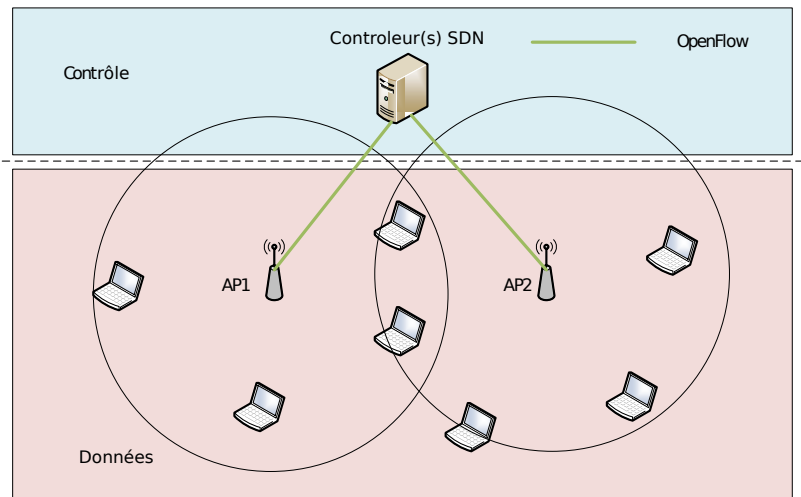


FIGURE VI.1: Architecture du réseau.

Pour contrôler la dissémination de données, un ou plusieurs contrôleurs SDN peuvent être connectés aux points d'accès Wi-Fi (AP, pour *Access Point* en anglais) du réseau. Lorsqu'un utilisateur rejoint un réseau Wi-Fi, le contrôleur SDN récupère, en plus des informations typiques d'authentification, d'autres informations utiles à la dissémination. Par exemple, la connaissance des positions des utilisateurs lui permet de donner des informations au contrôleur réseau. Avec ces informations, le contrôleur peut gérer de manière efficace le choix des canaux, fréquences ainsi que la puissance du signal émis depuis chaque périphérique. Dans notre cas, cela est également utile pour affecter des rôles aux périphériques du réseau : certains sont choisis pour devenir clients de la dissémination, d'autres sont utilisés comme *Group Owners* (GO) et servent également de relais pour les points d'accès du réseau.

Grâce à ce type d'administration, des mécanismes et algorithmes de contrôle avancés du réseau peuvent être mis en place de manière simplifiée grâce aux fonctionnalités offertes par le SDN. La maintenance d'un état global de la topologie réseau dans un point centralisé permet de sélectionner efficacement les clusters et de mettre en place des algorithmes de gestion des interférences de manière simple. Dans ce travail, nous faisons l'hypothèse que les positions des périphériques sont connues. Cela est réaliste car des techniques de géo-localisation en intérieur [129] ou un protocole spécifique peuvent être utilisés pour récupérer la position géographique des utilisateurs. Par exemple, un utilisateur peut saisir le numéro de son siège dans l'amphi dans lequel il se trouve, ce qui est simple à mettre en place.

Comme la figure VI.2 l'illustre, le contrôleur collecte constamment des informations des périphériques qu'il utilise pour le partitionnement du réseau et le contrôle de l'utilisation des ressources de communication (choix de fréquences, puissance du signal, etc.).

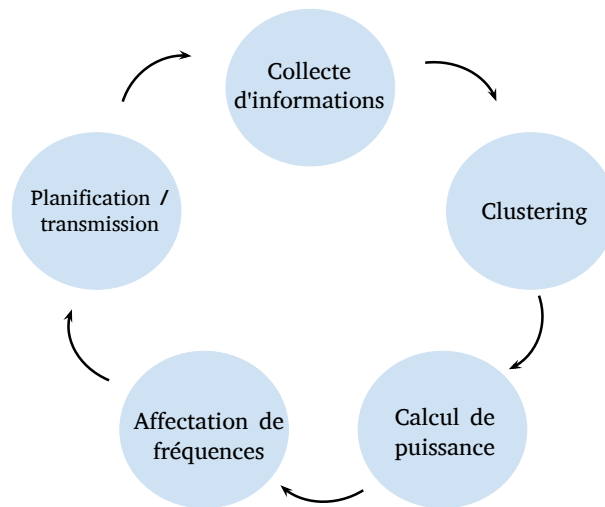


FIGURE VI.2: Boucle de contrôle SDN.

De la même manière qu'en mode infrastructure, chaque GO émet des beacons<sup>2</sup> qui permettent à ses clients de l'identifier sur le réseau. Lorsqu'un périphérique se déplace ou lorsque le GO se déconnecte du réseau, il peut détecter d'autres GO et points d'accès grâce aux beacons qu'ils émettent. Dans ces cas, les clients non connectés à un GO essaieront de se connecter à un point d'accès proche, comme le feraient de nouveaux périphériques qui arrivent sur le réseau. Ces événements sont détectés par le contrôleur qui réassigne de nouveaux GO pour les différents groupes. Les détails de ces phases de contrôle du réseau sont détaillés dans les sous-sections suivantes.

### VI.1.2 Algorithme de clustering

Le regroupement (clustering) consiste à partitionner un ensemble donné en plusieurs sous-ensembles (clusters) qui ont des caractéristiques en commun. Nous utilisons cette approche dans notre cas afin de découper le réseau en plusieurs clusters de périphériques mobiles qui formeront des groupes de communication Wi-Fi Direct pour la dissémination de données en fonction de leur localisation. Nous nous intéressons ici à l'utilisation de l'algorithme des  $K$ -moyennes [130]. Cet algorithme suppose la connaissance de l'espace euclidien à partitionner. Il prend en entrée un ensemble de positions (les emplacements des périphériques dans notre cas) et retourne une  $K$  partition (un ensemble de clusters avec leurs barycentres). L'objectif de cet algorithme est de regrouper les points d'un ensemble en offrant une

---

2. Trames balises qui véhiculent des informations permettant d'identifier un réseau (voir sous-section II.3.1).

distance minimale entre les points de chaque partition.

Appliqué à notre architecture présentée dans la figure VI.1, le périphérique le plus proche du barycentre (centroïde) est sélectionné comme GO pour son cluster. Chaque cluster représente un groupe de communication Wi-Fi Direct. Nous considérons un réseau  $\mathcal{N}$  composé d'un ensemble de  $N = |\mathcal{N}|$  périphériques. Par simplicité de notation, chaque périphérique du réseau  $\mathcal{N}$  est identifié par un entier  $i \in [1, N]$ . Les positions des périphériques sont stockées dans un ensemble  $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$ , où  $p_j = (x_j, y_j)$  est la position du périphérique  $j$ .  $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$  est la partition de  $\mathcal{N}$  en  $K$  clusters calculés avec l'algorithmes des  $K$ -moyennes. Enfin,  $\mathcal{O} = \{o_1, o_2, \dots, o_K\}$  est l'ensemble des centroïdes associés ( $o_k = (x_{o_k}, y_{o_k})$  est le centroïde du cluster  $C_k$ ). Ces informations vont nous permettre d'obtenir une topologie similaire à celle présentée dans la figure VI.3, où chaque point d'accès est connecté à ses GO qui sont responsables chacun un cluster auquel il retransmettent les données à disséminer.

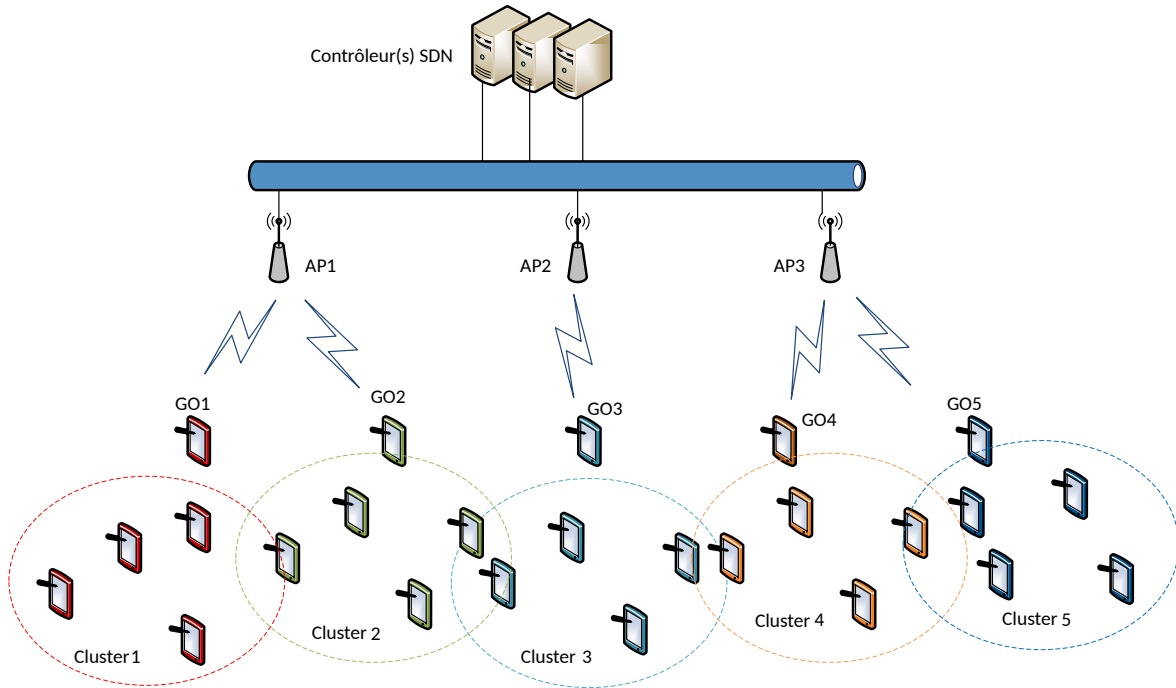


FIGURE VI.3: Topologie du réseau obtenue après le clustering

Pour calculer les clusters, nous choisissons les  $K$  centroïdes initiaux aléatoirement dans l'ensemble  $\mathcal{P}$ . Après plusieurs itérations, l'algorithme retourne un ensemble  $\mathcal{C}$  de  $K$  clusters auxquels sont associés leurs centroïdes  $\mathcal{O}$ . Il en résulte un diagramme de Voronoi dans lequel pour chaque point appartenant à un cluster, sa distance au centroïde est minimale. Cette propriété est représentée dans l'équation (VI.1) :

$$\forall C_k, C_l \in \mathcal{C}, C_k \neq C_l, \forall j \in C_k, \text{dist}(j, o_k) < \text{dist}(j, o_l), \quad (\text{VI.1})$$

où  $dist(i, j)$  est la distance euclidienne entre deux points  $i$  et  $j$  :

$$dist(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

---

**Algorithme 10** Algorithme des  $K$ -moyennes
 

---

```

1: function K-MEANS( $\mathcal{N}, K$ )
2:    $finished \leftarrow false$ 
3:    $\mathcal{O} \leftarrow randomCentroids(\mathcal{N}, K)$    ▶ Sélectionner des centroïdes aléatoirement
4:   while ! $finished$  do
5:      $\mathcal{O}' \leftarrow \mathcal{O}$    ▶ Mémoriser les centroïdes utilisés précédemment
6:      $\mathcal{C} \leftarrow assignClusters(\mathcal{O}, \mathcal{N})$    ▶ Calculer les nouveaux clusters
7:      $\mathcal{O} \leftarrow calculateCentroids(\mathcal{C})$    ▶ Calculer les nouveaux centroïdes
8:     if  $\mathcal{O} == \mathcal{O}'$  then
9:       return  $\langle \mathcal{C}, \mathcal{O} \rangle$ 
10:    end if
11:  end while
12: end function

```

---

L'algorithme 10 décrit cette procédure. Il prend en entrée un réseau  $\mathcal{N}$  (y compris l'ensemble  $\mathcal{P}$  des positions des périphériques) et  $K$  le nombre de clusters à calculer. Il retourne  $\mathcal{C}$ , la  $K$ -partition optimale ainsi que les centroïdes associés. Comme mentionné plus haut, les premiers centroïdes sont sélectionnés aléatoirement parmi les positions des périphériques (ligne 3). Ensuite, la fonction itère (lignes 4 à 11) jusqu'à converger vers un ensemble de centroïdes qui ne changent plus d'une itération à l'autre (lignes 8 à 10). À chaque itération, les centroïdes précédemment calculés (itération  $i - 1$ ) sont stockés pour une comparaison ultérieure avec les nouveaux centroïdes. Ensuite, la fonction  $assignClusters(\mathcal{O}, \mathcal{N})$  permet d'affecter chaque point  $j \in \mathcal{N}$  à un cluster  $C_k \in \mathcal{C}$ , dont le centroïde  $o_k \in \mathcal{O}$  est le plus proche géographiquement de  $j$ . L'algorithme 11 détaille les instructions effectuées par cette fonction. La fonction crée initialement un ensemble de clusters vides (ligne 2). Ensuite, pour chaque périphérique  $i$ , l'algorithme trouve le centroïde  $o \in \mathcal{O}$  le plus proche afin de déterminer le cluster auquel affecter le périphérique.

Revenons à l'algorithme 10. L'ensemble des clusters calculés en faisant appel à la fonction  $assignClusters(\mathcal{O}, \mathcal{N})$  (algorithme 10, ligne 6) est utilisé pour calculer les nouveaux centroïdes (algorithme 10, ligne 7). Ainsi, pour chaque cluster  $C_k$ , la distance entre son centroïde et ses points (positions des périphériques) est minimale en comparaison avec les autres centroïdes de l'ensemble  $\mathcal{O}$ . Ensuite, la fonction  $calculateCentroids(\mathcal{C})$  calcule les nouveaux centroïdes  $\mathcal{O}'$  dont les coordonnées sont la moyenne des coordonnées des points des nouveaux clusters de  $\mathcal{C}$  (équation (VI.2)).

**Algorithme 11** Affectation de clusters

```

1: function ASSIGNCLUSTERS( $\mathcal{O}, \mathcal{N}$ )
2:    $\mathcal{C} \leftarrow \{C_1 = \emptyset, C_2 = \emptyset, \dots, C_{|\mathcal{O}|} = \emptyset\}$ 
3:   for  $i \in \mathcal{N}$  do
4:      $k = \min_{l \in [1, K]} \text{dist}(i, o_l)$            ▶ Trouver le centroïde le plus proche
5:      $C_k = C_k \cup \{i\}$                          ▶ Ajouter au cluster associé
6:   end for
7:   return  $\mathcal{C}$ 
8: end function
    
```

$$x_{o_k} = \frac{\sum_{i \in C_k} x_i}{|C_k|}, y_{o_k} = \frac{\sum_{i \in C_k} y_i}{|C_k|} \quad (\text{VI.2})$$

Si les centroïdes ne changent plus d'une itération à l'autre ( $\mathcal{O} = \mathcal{O}'$ ), l'algorithme arrête d'itérer et retourne alors la  $K$ -partition optimale pour le réseau. Ainsi, pour chaque cluster, le périphérique le plus proche du centroïde est sélectionné comme *Group Owner* (GO) pour la dissémination de données. Les autres périphériques du cluster s'y connecteront pour récupérer des données.

### VI.1.3 Gestion des interférences et planification de la transmission

Après la sélection et la formation des groupes Wi-Fi Direct, le SDN calcule les puissances de signal pour chaque périphérique afin de réduire les interférences sur le réseau. En se basant sur la position des périphériques, deux puissances de transmission sont choisies telles qu'illustrées dans la figure VI.4.

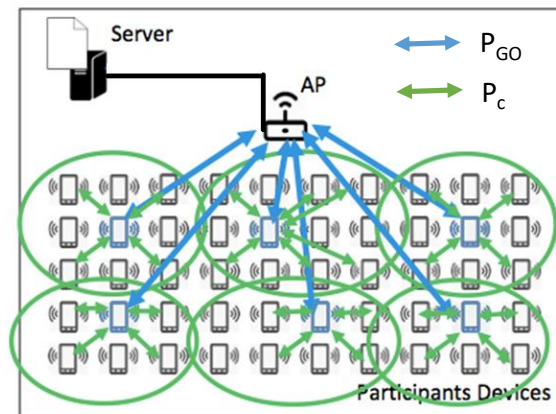


FIGURE VI.4: Contrôle de la puissance du signal avec le clustering

Une puissance de transmission  $P_{GO}$  est affectée pour chaque client afin qu'il communique avec le point d'accès auquel il est rattaché. Les périphériques qui ne sont pas des GO émettent avec une puissance de signal réduite qui leur suffit pour atteindre leur GO et communiquer avec lui. La diminution de la puissance permet de réduire la portée du signal. Elle permet ainsi à deux périphériques éloignés de ne pas interférer. Après la phase de contrôle de puissance de transmission (voir la figure VI.2), le contrôleur SDN vérifie s'il peut y avoir des interférences entre les groupes. Deux clients en bordure de clusters peuvent interférer l'un avec l'autre s'ils utilisent la même fréquence.

Afin de limiter cela, le SDN construit un graphe d'interférences dans lequel les clusters sont représentés sous forme de nœuds et chaque arête reliant deux nœuds représente la probabilité d'interférence entre les nœuds des groupes. Ainsi, l'affectation des canaux Wi-Fi peut se faire en implémentant une solution inspirée des algorithmes de coloration de graphes [131]. Rappelons qu'en pratique il est possible de choisir des canaux Wi-Fi orthogonaux (Par exemple : les canaux 1, 6 et 11 de la bande de fréquences 2.4GHz). L'algorithme de coloration de graphes permet d'affecter des canaux aux différents clusters dans notre cas.

En Wi-Fi Direct, la notification d'absence (NoA : Notification of absence [93]) permet au GO d'informer ses clients lorsqu'il se déconnecte du groupe de manière temporaire pour repasser dans le mode infrastructure. Cela lui permet de maintenir la communication dans le groupe en son absence (temporaire). Cette fonctionnalité permet alors d'organiser la transmission de données en deux phases : la transmission entre le point d'accès et les GO, et la transmission entre les GOs et leurs clients. Le GO peut utiliser ce type de message pour informer d'une absence temporaire qui va lui permettre de récupérer depuis le point d'accès les prochaines données à leur transmettre.

Le problème de la gestion des interférences dans les réseaux Wi-Fi Direct est similaire à celui des réseaux cellulaires. Pour améliorer la couverture d'un réseau cellulaire, des points d'accès *Femtocell* [9] peuvent être déployés dans une la zone de couverture d'un point d'accès cellulaire (*macrocell*). Une femtocell peut être associée à une autre femtocell ou une macrocell pour agrandir sa couverture. Elles ont la même fonction que peut avoir un GO dans notre architecture. En effet, les mécanismes de gestion du signal et de l'affectation de ressources utilisés dans les réseaux femtocell peuvent être appliqués avec les réseaux Wi-Fi.

La différence principale entre les réseaux *Femtocell* et les réseaux Wi-Fi réside dans l'utilisation du médium de communication et l'orthogonalité des canaux de communication. Les réseaux Wi-Fi se basent sur un accès aléatoire (CSMA/CA) alors que les réseaux cellulaires se basent sur l'allocation de ressources radio orthogonales. Les canaux Wi-Fi se superposent partiellement les uns sur les autres alors que dans les réseaux *Femtocell* les canaux sont tous orthogonaux. Ces différences rendent les mécanismes de gestion d'interférence complexes et difficiles sur les réseaux Wi-Fi.

L'utilisation de mécanismes existants dans les réseaux cellulaires n'est pas triviale car ces derniers ne sont pas directement applicables aux réseaux Wi-Fi.

En divisant le réseau en groupes de petites tailles et en réduisant la puissance de transmission dans un groupe, notre approche peut réduire les interférences sur le réseau et augmenter les performances (débit croissant et taux d'erreurs plus bas). Cependant la puissance de transmission des GO doit être plus importante que celle du reste des périphériques. Pour prendre en compte cet aspect, d'autres informations liées aux signal peuvent être ajoutées afin d'avoir une sélection plus efficace des clusters et des GO. D'un point de vue énergétique, il est envisageable de changer de GO dans un cluster afin d'économiser l'énergie d'un GO qui aurait été très exploité.

#### VI.1.4 Calcul du nombre optimal de clusters

L'algorithme de clustering que nous utilisons ( $K$ -moyennes) peut seulement être utilisé lorsque l'on connaît le nombre de clusters à former. Il est nécessaire pour le SDN d'être capable de déterminer la valeur de  $K$  avant de calculer les différents clusters. La valeur de  $K$  doit être choisie afin d'avoir de bonnes performances sur le réseau. Nous proposons de choisir en règle générale un nombre de clusters proche de  $\sqrt{N}$ . Ce choix est motivé par la considération et la résolution du modèle dont les hypothèses sont les suivantes :

- Les périphériques partagent tous la même fréquence ;
- Les communications peuvent être décrites par un graphe de disques où chaque périphérique  $p$  a une portée de communication  $\rho_p$ .  $p$  peut envoyer des informations à n'importe quel périphérique dans le disque dont il est le centre et dont le rayon est  $\rho_p$ , mais cela créera des interférences avec les autres périphériques à l'intérieur du disque.  $p$  n'a pas d'impact en dehors de sa portée de communication ;
- La performance globale du système n'est pas linéairement croissante avec le nombre d'interférences.

Ainsi, pour maximiser les performance du réseau, nous avons besoin de minimiser le nombre d'interférences. Chaque périphérique a besoin de communiquer avec le point d'accès. Sachant que la distance entre chaque GO et le point d'accès est du même ordre de grandeur que les dimensions de la salle dans laquelle se situent les périphériques. Un périphérique peut alors subir des interférences d'une fraction non négligeable de  $\alpha K$  périphériques, où  $K$  est le nombre de clusters.

De plus, chaque périphérique doit pouvoir communiquer avec son GO. Il a donc besoin d'avoir une portée de signal du même ordre de grandeur que le rayon du cluster. Sous l'hypothèse que les périphériques sont uniformément répartis dans la salle dans laquelle ils se trouvent, la portée d'interférence d'un périphérique non-GO (client) devrait être du même ordre de taille qu'un cluster typique, c'est-à-dire  $\beta \frac{N}{K}$  pour une constante  $\beta > 0$ .



Le nombre total d'interférences d'un périphérique peut alors être calculé avec  $\alpha K + \beta \frac{N}{K}$  où  $\alpha K$  est le nombre de GO interférents et  $\beta \frac{N}{K}$  représente le nombre de périphériques clients interférents. Ce nombre peut être minimisé lorsque  $K = \sqrt{\frac{\beta}{\alpha} N}$ . Intuitivement,  $\alpha$  et  $\beta$  devraient être similaires étant donné qu'ils mesurent tous les deux le même type de ratio géométrique sur différentes échelles car ils permettent de représenter pour chaque périphérique la fraction des périphériques avec lesquels il interfère respectivement au niveau du réseau et du cluster. Ce qui conduit donc à  $K \approx \sqrt{N}$ .

Notons que ce même type de raisonnement devrait être appliqué si un périphérique utilisait de multiples fréquences :  $\alpha$  et  $\beta$  devraient être divisés par le nombre de canaux orthogonaux disponibles, le résultat restant inchangé. De manière similaire, le modèle de transmission pourrait être affiné, par exemple en considérant quelques canaux au lieu d'une simple portée de signal, mais cela ne devrait pas changer la règle de  $\sqrt{N}$ . Ce qui veut dire que l'interférence peut venir :

- depuis une communication ayant lieu entre l'un des  $K$  GOs et le point d'accès ;
- depuis la communication entre des périphériques et leur GO, ce qui impacte un certain nombre de périphériques proportionnel à la taille d'un cluster,  $\frac{N}{K}$ .

Dans un contexte dynamique, la valeur de  $N$  peut fluctuer avec le temps. Le changement du nombre optimal de clusters, comme l'architecture interne des clusters peut être coûteuse en temps de calcul. Dans ces cas, l'approche à utiliser qui permette de ne pas avoir un coût important serait de retarder les modifications de clusters par un mécanisme d'hystérèse. Par exemple, le recalcul du nombre de clusters seulement lorsque la différence entre le nombre optimal et le nombre courant de clusters  $K$  dépasse un certain seuil. Toutefois, si les périphériques sont constamment en mouvement, il serait plus approprié d'utiliser une approche semblable à celles décrites dans les chapitres III et IV qui sont plus adaptées dans ce cas.

## VI.2 Évaluation des performances

### VI.2.1 Performance du clustering

Dans cette section, nous évaluons le débit de données qu'un client reçoit en fonction du nombre de clusters sur le réseau. Considérons un réseau de  $N$  périphériques, divisé en  $K$  clusters comme présenté ci-dessus. Sous l'hypothèse que le mécanisme d'accès au médium de communication est équitable entre les périphériques (c'est-à-dire que chaque périphérique aura la même probabilité d'accéder à un canal et sa capacité sera partagée de manière équitable).

Nous calculons ici le débit dont peut bénéficier un périphérique en fonction du nombre de clusters sur le réseau en prenant en considération les hypothèses de la

section VI.1.4. Le débit dont peut bénéficier un périphérique est calculé en fonction du nombre de voisins qui interfèrent avec lui, c'est à dire qui partagent le même médium de communication que lui. La figure VI.5 illustre le débit de données moyen qu'un périphérique non-GO. De plus, la capacité du canal est partagée entre les  $\frac{N-K}{K}$  périphériques dans un cluster et les  $K$  GOs. Le débit moyen  $d$  par périphérique est mesuré en divisant le débit maximal  $D_{max}$  que l'on peut avoir par le nombre de périphériques partageant le même canal  $K + \frac{N-K}{K}$  (voir équation VI.3).

$$d = \frac{D_{max}}{K + \frac{N-K}{K}} \quad (VI.3)$$

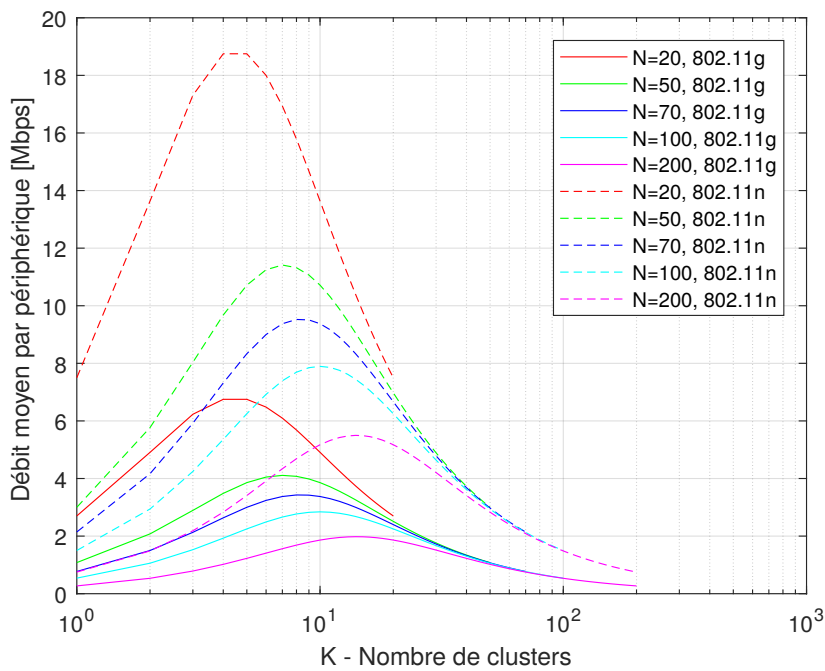


FIGURE VI.5: Évaluation du débit pour 802.11g(54Mbps) et 802.11n(150Mbps).

Comme nous pouvons le voir sur la figure VI.5, dans un réseau 802.11g opérant à 54Mbps (avec un débit effectif de 22.5Mbps), le meilleur débit peut être obtenu lorsque le nombre de clusters  $K \approx \sqrt{N}$ . Pour une classe de 20 étudiants, le bon nombre de clusters est de 4 ou 5, offrant un débit de données moyen de 6.75 Mbps par périphérique. Pour un amphithéâtre de 200 étudiants le bon nombre optimal de clusters est 11 ou 12 avec un débit de données moyen de 1.86Mbps et 1.91Mbps respectivement par périphérique. Ces débits sont théoriques et ne prennent pas en compte le coût introduit par les entêtes des protocoles utilisés à différentes couches et la réduction du débit due au mécanisme de back-off dans le mécanisme d'accès au médium dans les réseaux Wi-Fi.

Les normes 802.11 récentes offrent des débits plus importants tels que 802.11n et 802.11ac, le débit de donnée optimisé qu'un utilisateur peut obtenir est plus élevé que celui illustré précédemment dans la figure VI.5. Par exemple, sur un réseau 802.11n à 150Mbps, chaque utilisateur dans une salle de 20 étudiants peut bénéficier d'un débit de 18.75 Mo et chaque utilisateur dans un amphithéâtre de 200 étudiants peut bénéficier d'un débit de 5.32 Mbps. Le standard utilisé par un périphérique peut aussi être utilisé comme paramètre pour le choix des GO, permettant ainsi de favoriser les périphériques aux plus hauts débits pour être choisis comme GOs.

Le temps de calcul des clusters dépend du nombre de périphériques dans le réseau  $N$  et du nombre de clusters  $K$ . La figure VI.6 représente les mesures de temps de calcul de l'algorithme des  $K$ -moyennes en fonction du nombre de clusters à créer sur le réseau. Ces calculs ont été effectués avec un ordinateur équipé d'un processeur Intel Core i5 de 2 cœurs cadencé à 2.7 GHz avec 8Go de mémoire vive sous le système d'exploitation Linux Ubuntu (version 16.10). Nous pouvons voir que ce temps croît linéairement avec le nombre de clusters  $K$ . Cependant les temps de calculs restent acceptables dans la mesure où cette opération ne se produit pas en permanence sur le réseau.

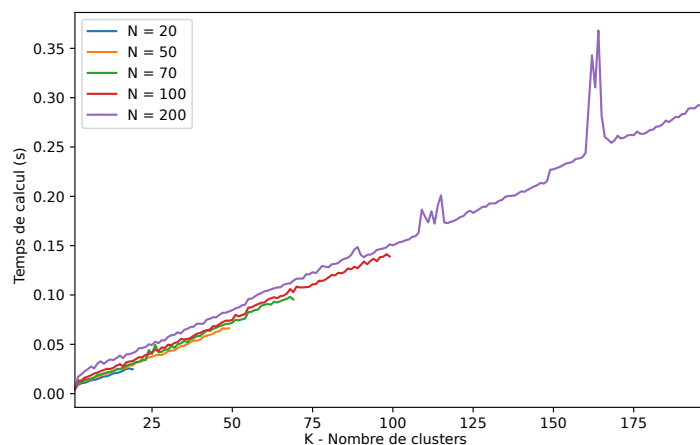


FIGURE VI.6: Mesure de temps de calcul de clusters.

## VI.2.2 Simulation avec OMNet++

Afin d'évaluer les performances de transmission de manière plus réaliste, le protocole Wi-Fi Direct a été implémenté dans OMNet++ [127]. Le modèle permettant de simuler des réseaux Wi-Fi est fourni dans OMNet++<sup>3</sup> à travers la bi-

3. [www.omnetpp.org](http://www.omnetpp.org) [version utilisée : 4.2]

bibliothèque INET<sup>4</sup> qui inclut les modèles permettant la simulation de réseaux filaires et mobiles. Le module Wi-Fi Direct qui a été développé durant le stage de Sypfax Iskounen est accessible en ligne [132]. Le modèle 802.11 a été surchargé afin de prendre en charge le protocole Wi-Fi Direct et de permettre de réaliser les différentes tâches qu'effectue un périphérique en Wi-Fi Direct : détection de voisins, négociation pour l'élection des GO, établissement de connexions, configuration d'adresses.

Avec EDWiN (chapitres III et IV), nous nous étions intéressés aux gains que pouvaient offrir les communications D2D sur de grands volumes de données. Notre but était d'évaluer l'impact de différents paramètres du système sur les performances de la dissémination. Nous avons utilisé Peersim [122], car il permet d'évaluer plus rapidement plusieurs paramètres et sur de grandes échelles. Dans cette contribution, nous nous intéressons à de plus petites échelles et à un nombre réduit de périphériques mobiles afin de prendre en compte plus finement l'impact de la technologie sur le débit et sur le taux d'erreur sur le réseau (formation des groupes Wi-Fi Direct, les interférences, ...). C'est pour cela que notre choix s'est tourné vers OMNet++.

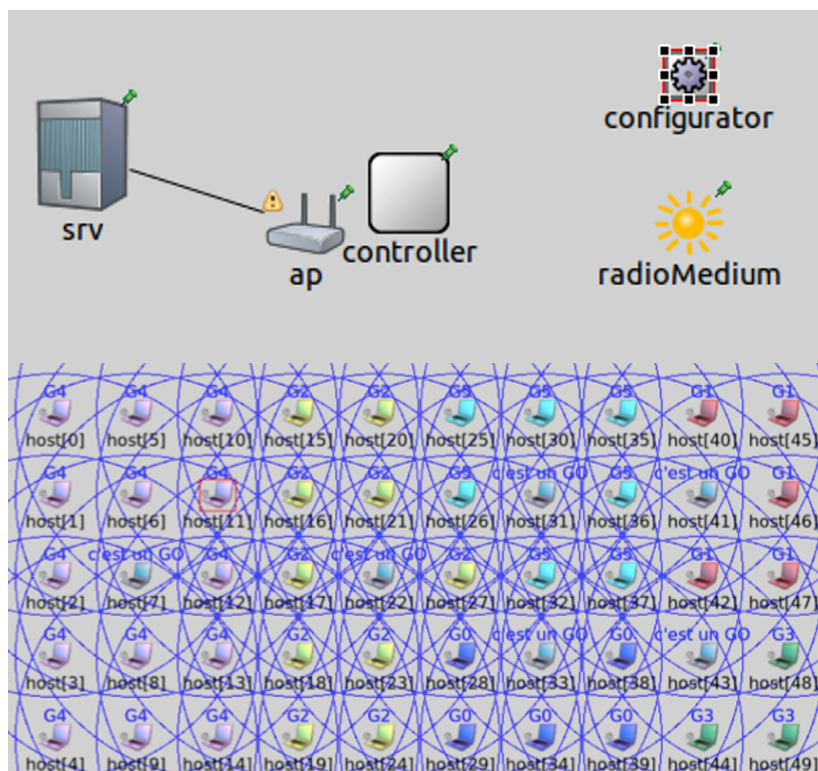


FIGURE VI.7: Topologie de simulation sur OMNET++ (credits S. Iskounen [127])

4. [inet.omnetpp.org](http://inet.omnetpp.org) [version utilisée : 2.0.0]

### VI.2.2.1 Topologie

La topologie simulée est représentée dans la figure VI.7. Dans ce réseau, 50 périphériques sont disposés sur une topologie en grille composée de 5 lignes et 10 colonnes. Les distances horizontale et verticale entre deux périphériques sont d'un mètre, ce qui représente approximativement la distance entre deux personnes dans un amphithéâtre. Le débit de données de la connexion Ethernet entre le point d'accès Wi-Fi et le serveur est de 100Mbps. Le contrôleur SDN est utilisé pour contrôler la formation des groupes Wi-Fi Direct (clustering). Les paramètres radio de la simulation sont présentés dans le tableau VI.1.

Tableau VI.1: Paramètres de simulation

Paramètre	Valeur
Norme Wi-Fi	802.11g
Fréquence	2.4 GHz
Largeur de bande	22 MHz
Débit de transmission radio	48 Mbps
Sensibilité du récepteur	-85 dBm
Seuil SINR	4 dB
Puissance du bruit	-90 dBm
Puissance de transmission du GO	1.47 dBm
Puissance de transmission d'un client	-23 dBm

### VI.2.2.2 Scénarios simulés

Afin d'évaluer les performances du réseau, nous avons défini deux scénarios. Dans un premier temps, les 50 périphériques téléchargent un fichier directement depuis le point d'accès (ce qui correspond au cas particulier  $K=1$  de notre approche). Dans un second temps, les périphériques sont découpés en  $K$  groupes Wi-Fi Direct. La puissance de transmission des périphériques non-GO est réglée afin d'atteindre uniquement leur GO. Le GO est utilisé comme relais entre ses clients et le serveur ().

La figure VI.8 illustre le temps moyen de téléchargement en fonction de la taille des fichiers. Les 50 périphériques sont répartis en 4 groupes Wi-Fi Direct (clusters). La taille des fichiers varie entre 50Ko (qui pourrait correspondre au planning des cours dans un tableur excel) et 2Mo (qui pourrait correspondre à un fichier PDF contenant les notes du cours). Cette simulation montre que notre utilisation du Wi-Fi Direct peut permettre de gagner jusqu'à 34% de temps de téléchargement en comparaison avec l'utilisation du mode Infrastructure.

Afin de comprendre l'origine de ce gain de performances en terme de temps de téléchargement, nous considérons le rapport entre le taux d'erreurs et le temps de téléchargement. La figure VI.9 montre le temps de téléchargement d'un

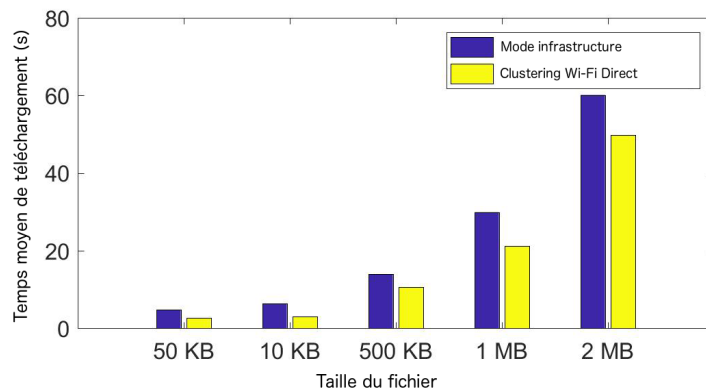


FIGURE VI.8: Temps moyen de téléchargement avec  $K = 4$

fichier de 50 Ko et le taux d'erreur pour 3 différents cas de clustering :  $K = 1$  (infrastructure),  $K = 4$ ,  $K = 8$ . Le taux d'erreur explique le gain obtenu dans le temps de téléchargement. Lorsque tous les utilisateurs utilisent une puissance de signal élevée afin de communiquer avec le point d'accès et télécharger le fichier simultanément, le niveau d'interférence entre les périphériques provoque une augmentation du taux d'erreur de 21% et un temps de téléchargement élevé de 4.68 secondes. Avec notre méthode, nous réduisons la puissance de transmissions des périphériques non-GO, le niveau d'interférences sur le réseau est de ce fait réduit, il en résulte un taux d'erreur plus bas (14.1%) dans le cas de 4 clusters et de 13.08 % dans le cas de 8 clusters.

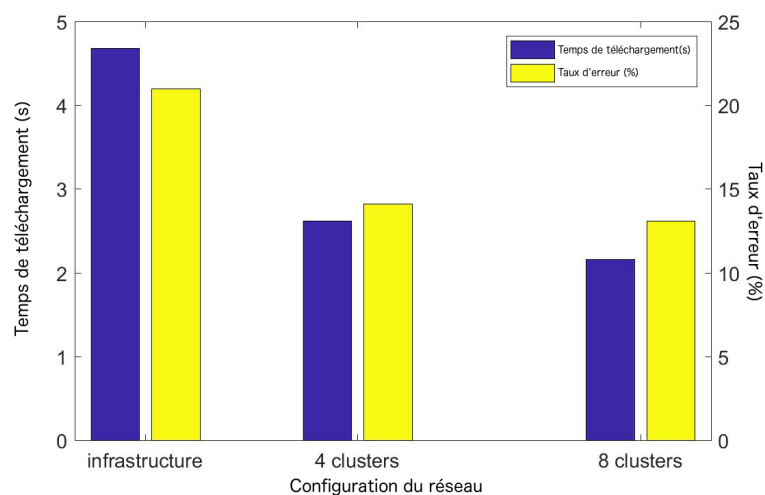


FIGURE VI.9: Impact de l'utilisation de clusters sur le taux d'erreurs et sur le temps de téléchargement.

Afin d'analyser l'impact du nombre de clusters sur la performance globale du

réseau nous mesurons le temps de téléchargement (temps de formation des groupes Wi-Fi Direct compris) et le taux d'erreur pour les cas de 4, 6, 8 et 10 clusters. La figure VI.10 montre qu'avec une configuration de 8 clusters, le réseau offre de meilleures performances en terme de temps de téléchargement et en taux d'erreurs que les autres configurations. Cette observation confirme que notre calcul du nombre optimal de clusters devrait être  $K \approx \sqrt{N}$ .

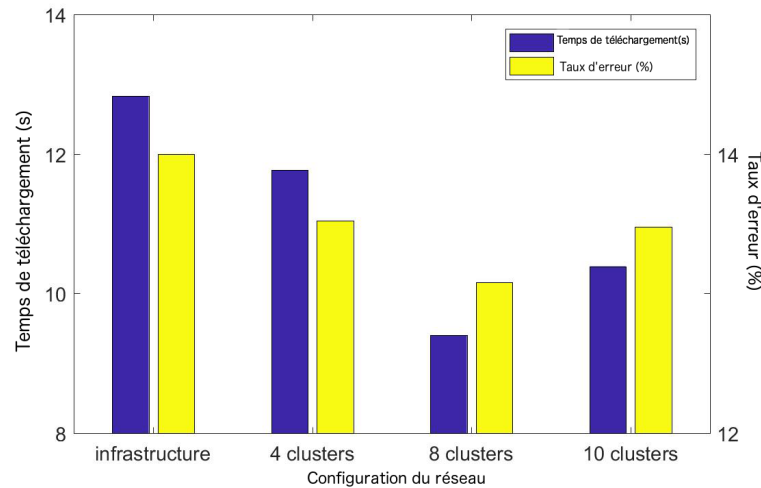


FIGURE VI.10: Nombre optimal de clusters.

## Discussion

Nous venons de présenter une nouvelle approche pour la dissémination de données reposant cette fois sur l'utilisation des SDN afin de diffuser sur des topologies réseaux hiérarchiques. L'utilisation du clustering permet de diviser le réseau en petits groupes Wi-Fi Direct (clusters) de petite taille. Réduire la puissance de transmission des clients pour chaque groupe permet de réduire les interférences globales sur le réseau. Nous proposons une méthode permettant de calculer un nombre de clusters en fonction de la taille du réseau. Les résultats de la simulation montrent que cette architecture permet d'obtenir de meilleures performances en terme de temps de téléchargement et en taux d'erreur comparée à l'utilisation de l'infrastructure Wi-Fi.

Comparée à une approche en point à point, l'utilisation de topologies hiérarchiques sur le réseau permet de gérer plus aisément la puissance du signal sur chaque périphérique afin de provoquer le moins d'interférences possibles. Ceci nous permet de tirer avantage de la faible mobilité des périphériques sur le réseau afin de répartir efficacement le débit de données sur les différents sous-réseaux (groupes de communication Wi-Fi Direct). Cependant, l'approche hiérarchique présentée dans

ce chapitre suppose que le trafic du réseau est dédié à la dissémination de données, alors que dans le cas d'EDWiN (chapitres III et IV), nous considérons que la bande passante est partagée par la dissémination ainsi que d'autres usages (internet, messageries, etc.), d'où la limitation l'utilisation du point d'accès à un périphérique à la fois.





CHAPITRE

**VII**Conclusion et perspectives

---

**Sommaire**

---

VII.1 Conclusion . . . . .	112
VII.2 Perspectives . . . . .	114

---

## VII.1 Conclusion

Cette thèse s'intéresse à la dissémination de données volumineuses sur des réseaux mobiles avec des périphériques qui ont la capacité de communiquer en D2D. Elle a été motivée par le cas d'étude concret auquel fait face l'industriel Magency : des événements à grande échelle (ex : conférences, séminaires) durant lesquels de nombreux périphériques mobiles accèdent à des volumes de données importants. Cette dissémination doit partager la bande passante avec d'autres applications qui utilisent ce même réseau (messagerie, vote instantané, ...). L'utilisation d'un medium sans fil pour communiquer pose un problème de passage à l'échelle. Dans des systèmes basés sur des réseaux filaires, le passage à l'échelle peut être assuré en agrandissant l'infrastructure. Dans les réseaux mobiles, agrandir une infrastructure implique le déploiement d'un nombre accru de points d'accès pour prendre en charge plus de clients, ce qui augmente la densité dans ces réseaux ainsi que leur risque de subir les conséquences des interférences. Les technologies D2D permettent d'utiliser la capacité des périphériques à communiquer de manière autonome avec leurs voisins afin d'améliorer les performances du réseau et de délester les nœuds d'infrastructure. Dans cette thèse nous nous sommes intéressés au cas des réseaux Wi-Fi et proposons une mise en œuvre pragmatique des connexions D2D pour améliorer les performances du réseau lors de la dissémination de données volumineuses.

Dans le chapitre III, nous avons présenté EDWiN, notre approche de dissémination D2D en point-à-point qui a pour but disséminer de manière efficace des données volumineuses sur des réseaux Wi-Fi. Elle s'appuie sur une politique de dissémination inspirée de BitTorrent en découpant la donnée en chunks, et en utilisant la politique de sélection de chunks "*Rarest first*". En privilégiant les appariements D2D permettant d'échanger le plus de données possible, cette approche vise à avoir l'utilisation la plus efficace possible des connexions D2D. Cette approche fonctionne sous l'hypothèse que le réseau Wi-Fi permet à un certain nombre de périphériques d'échanger en parallèle sans trop générer d'interférences. Ce paramètre que l'on appelle degré de parallélisme est, en pratique, lié à l'état d'occupation des canaux Wi-Fi sur le réseau.

Disséminer des données sans considération de la consommation que peuvent avoir les périphériques mobiles peut amener à un déséquilibre dans la participation de chaque périphérique à la dissémination. En effet, certains d'entre eux peuvent se retrouver sollicités plus fréquemment que d'autres. C'est d'ailleurs souvent le cas des périphériques qui récupèrent des données en premier. Une utilisation intensive des communications réseaux consomme de l'énergie, cela peut conduire à un déchargement de batterie plus important sur ces périphériques. De plus, avec l'hétérogénéité de l'écosystème mobile, le coût de la communication réseau varie d'un périphérique à l'autre. Dans le chapitre IV, nous proposons l'extension de notre approche afin de prendre en considération la consommation d'énergie et l'hétérogénéité des périphériques mobiles en terme de batterie et de bande

passante. L'objectif de cette extension est de protéger les périphériques les plus faibles d'une surexploitation sur le réseau afin d'avoir une répartition équitable de la consommation énergétique. En effet, si l'on se place dans un contexte dans lequel tous les utilisateurs doivent pouvoir bénéficier de l'accès aux données disséminées le plus longtemps possible après leur réception, il est important de protéger les batteries les plus faibles autant que possible.

Dans le chapitre V, nous avons présenté une évaluation basée sur une simulation par événements discrets avec Peersim. Nous avons évalué les gains que nos approches apportent en fonction du paramétrage. Le simulateur a été calibré avec des mesures effectuées sur du matériel utilisé en production chez Magency afin qu'il soit réaliste. Ce mode de simulation a été motivé par le besoin d'évaluer rapidement les différentes approches exploitées et ce, sur différentes configurations. Les résultats des simulations ont montré que l'on pouvait par exemple avoir un gain de près de 40% sur un réseau de 100 périphériques mobiles pour la dissémination d'un fichier de 100 Mo, avec un degré de parallélisme autorisant jusqu'à 17 périphériques à échanger en parallèle sur le réseau (huit échanges D2D + un téléchargement depuis le serveur). Nous avons observé que la sollicitation excessive de périphériques mobiles pour la dissémination de données peut conduire à une consommation énergétique importante. Une limitation simple (voir section IV.1) du temps d'activité des périphériques mobiles sur le réseau permettrait de réduire la consommation. Cependant, celle-ci empêche le système d'atteindre son degré de parallélisme. Or, si l'on considère l'hétérogénéité des périphériques mobiles, l'objectif est de protéger les périphériques aux batteries les plus faibles. L'évaluation des approches pour la prise en compte de l'hétérogénéité des périphériques (voir chapitre IV) a montré que la limitation de l'activité du réseau sur les batteries les plus faibles permettait d'avoir un gain en temps de complétion de la dissémination mais aussi en temps d'activité sur le réseau. Cette restriction permettait d'augmenter le nombre de transferts de données plus efficaces, permettant d'acheminer la totalité de la donnée en un envoi. De plus, si l'on favorise des périphériques avec des bandes passantes élevées pour qu'ils échangent en début de dissémination. On peut très vite obtenir des sources de données complètes, ce qui permet de réduire le nombre de connexions D2D pour la dissémination, ce qui est plus efficace dans notre cas.

Bien que la calibration du simulateur soit faite par des mesures réelles, le réalisme est limité par le fait que les couches physiques du réseau ne sont pas simulées. Ce choix a été d'une part motivé par le besoin d'effectuer une évaluation rapide de l'impact de chaque paramètre du système sur les performances de la dissémination. D'autre part, l'évaluation par un simulateur réseau tel qu'OMNet++ est très coûteuse en raison du nombre de nœuds et du volume de données. Enfin, l'expérimentation en conditions réelles d'une dissémination sur un tel type de réseau aurait nécessité des moyens importants, notamment humains.

Durant ma thèse, j'ai eu l'opportunité de participer à un projet LIP6 conjoint entre mon équipe *DELYS* ainsi que l'équipe réseau *Phare*. Le chapitre VI présente les

travaux résultant de cette collaboration. L'objectif ici consiste en la mise en place de topologies hiérarchiques sur des réseaux Wi-Fi en utilisant la technologie Wi-Fi Direct. Notre intérêt s'est porté sur le cas d'étude des amphithéâtres au sein desquels les étudiants accèdent aux mêmes ressources de cours. Ce cas présente certaines similitudes avec le cas des événements à grande échelle. Cependant, les ressources réseaux dans ce cas peuvent être utilisées en intégralité pour la dissémination (il n'y a pas d'application concurrente à priori). De plus, les utilisateurs ne se déplacent pas ou alors très peu (ils sont assis à leur place). Nous exploitons cette propriété afin de découper le réseau en plusieurs groupes (clusters) regroupant les périphériques par zones géographiques. En effet, grâce à la connaissance des positions des périphériques, il devient assez aisé de mettre en œuvre un algorithme de clustering tel que celui des  $K$ -means. Une implémentation basée sur les SDN permet de gérer d'une manière plus simple le partitionnement du réseau ainsi que l'affectation de la puissance afin que les périphériques interfèrent le moins possible. Nous avons évalué cette approche en utilisant OMNet++ [127] sur un réseau de 50 périphériques mobiles pour l'envoi de fichiers de petites tailles (50Ko à s) afin d'évaluer les temps de dissémination et les taux de perte de paquets sur le réseau. L'évaluation a montré que l'on peut avoir jusqu'à 34% de gain en temps de complétion en utilisant le clustering et que le choix d'un nombre de clusters optimal permettrait non seulement d'avoir une répartition plus efficace de la bande passante sur le réseau mais aussi de meilleures performances.

## VII.2 Perspectives

Cette thèse ouvre de nombreuses perspectives, voici les principales :

**Expérimentations en conditions réelles :** Les résultats que l'évaluation d'EDWiN nous a permis d'obtenir sont très intéressants et prometteurs. Une expérimentation en conditions réelles nous permettraient de les valider. La réalisation de telles expérimentations nécessiterait de rassembler des périphériques capables de communiquer en D2D ainsi qu'un nombre important de personnes pour s'assurer du bon déroulement de la dissémination. Évaluer EDWiN en condition réelles nécessiterait d'utiliser une variante de Wi-Fi Direct offrant plus de possibilités de contrôle que les implémentations existantes. En effet, cela nous permettrait par exemple de choisir les canaux les moins encombrés pour les échanges D2D ou d'ajuster la puissance du signal.

**Adaptation d'EDWiN à l'environnement réseau :** Dans le cadre de déploiements à grande échelle durant lesquels les périphériques sont amenés à se déplacer, il serait intéressant d'étudier la mise en œuvre d'une dissémination de données qui exploiterait les informations que le périphérique collecte sur son voisinage [86]. On peut envisager l'utilisation de méthodes de géolocalisation en intérieur (indoor) pour mesurer la distance entre les périphériques qui échangent afin d'adapter la

puissance de leur signal pour qu'ils n'engendrent pas d'interférences dans leur voisinage. L'utilisation des informations de l'environnement permettrait aussi d'adapter le paramétrage la dissémination : adaptation du degré de parallélisme, changement du volume de données envoyées au périphériques, plusieurs niveaux de découpage, etc. Ils serait intéressant à l'avenir d'étudier de plus près ce cas et de voir quelles approches seraient plus adéquates pour une convergence rapide vers une configuration qui serait optimale et adaptée au système. Une telle approche devra être simple à mettre en œuvre et ne devra pas être intrusive pour les utilisateurs. Une architecture de type SDN pourrait donc être adaptée pour ce cas.

**Utilisation d'une approche combinant une dissémination point-à-point et hiérarchique :** Une des principales motivations d'EDWiN est qu'elle ne considère pas la présence des regroupements statiques de périphériques. A l'opposé, l'utilisation d'approches hiérarchiques sur des topologies fixes permettraient d'avoir une meilleure gestion de l'utilisation du signal Wi-Fi. Sur certains cas, on pourrait avoir ces deux propriétés sur un même réseau (certains périphériques très mobiles et d'autres fixes), il serait donc intéressant d'exploiter les deux types d'approches en créant des connexions en point-à-point opportunistes sur des périphériques qui se déplacent fréquemment et en formant des groupes Wi-Fi Direct en suivant une topologie hiérarchique sur ceux plutôt statiques. Ainsi la dissémination bénéficierait de l'adaptabilité de l'approche en point-à-point et de l'efficacité de l'utilisation du medium pour l'approche hiérarchique.



## Bibliographie

---

- [1] Cisco, “Cisco visual networking index : Global mobile data traffic forecast update, 2016-2021,” tech. rep., march 2017.
- [2] R. Toole, “BitTorrent Architecture and Protocol,” 2006.
- [3] J. Liang, R. Kumar, and K. W. Ross, “Understanding kaza,” 2004.
- [4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord : A Scalable Peer-to-Peer Lookup Service for Internet Applications,” in *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)*, pp. 149–160, 2001.
- [5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A Scalable Content-addressable Network,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 161–172, 2001.
- [6] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.
- [7] F. Rebecchi, M. D. De Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, “Data offloading techniques in cellular networks : a survey,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 580–603.
- [8] L. Al-Kanj, Z. Dawy, and E. Yaacoub, “Energy-aware cooperative content distribution over wireless networks : Design alternatives and implementation aspects,” *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1736–1760, Fourth 2013.
- [9] N. Saquib, E. Hossain, L. B. Le, and D. I. Kim, “Interference management in ofdma femtocell networks : Issues and approaches,” *IEEE Wireless Communications*, vol. 19, no. 3, 2012.
- [10] L. Hamidouche, S. Monnet, P. Sens, and D. Refauvelet, “Toward heterogeneity-aware device-to-device data dissemination over Wi-Fi networks,” in *ICPADS 2017 - International Conference on Parallel and Distributed Systems*, (Shenzhen, China), Dec. 2017.
- [11] T.-M.-T. Nguyen, L. Hamidouche, F. Mathieu, S. Monnet, and S. Iskounen, “SDN-based Wi-Fi Direct Clustering for Cloud Access in Campus Networks,” *Annals of Telecommunications, Springer*, 2017.
- [12] L. Hamidouche, S. Monnet, P. Sens, and D. Refauvelet, “Prise en compte de l'autonomie des batteries des périphériques mobiles pour la diffusion efficace de données en D2D.,” in *Compas'2017*, (Sophia Antipolis, France), 2017.



- [13] L. Hamidouche, S. Monnet, F. Bardolle, P. Sens, and D. Refauvelet, "Edwin : Leveraging device-to-device communications for efficient data dissemination over wi-fi networks," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, pp. 228–235, March 2017.
- [14] S. Iskounen, T. M. T. Nguyen, S. Monnet, and L. Hamidouche, "Device-to-device communications using wi-fi direct for dense wireless networks," in *2016 7th International Conference on the Network of the Future (NOF)*, pp. 1–3, Nov 2016.
- [15] L. Hamidouche, S. Monnet, and P. Sens, "EDWiN : Vers une diffusion efficace de données sur les réseaux Wi-Fi," in *Compas'2016*, (Lorient, France), 2016.
- [16] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, p. 29, 2003.
- [17] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010*, 2010.
- [18] D. Bindel, Y. Chen, P. Eaton, D. Geels, R. P. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao, and J. Kubiawicz, "Oceanstore : An architecture for global-scale persistent storage," *Asplos*, vol. 35, no. 11, pp. 190–201, 2000.
- [19] G. Shah, *Distributed Data Structures for Peer-to-peer Systems*. PhD thesis, Yale University, New Haven, CT, USA, May 2003.
- [20] A. Rowstron and P. Druschel, "Pastry : Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pp. 329–350, Springer, 2001.
- [21] M. Ripeanu, "Peer-to-peer architecture case study : Gnutella network," *Proceedings First International Conference on Peer-to-Peer Computing*, pp. 1–11, 2001.
- [22] S. Saroiu, K. P. Gummadi, and S. D. Gribble, "Measuring and analyzing the characteristics of napster and gnutella hosts," *Multimedia systems*, vol. 9, no. 2, pp. 170–184, 2003.
- [23] "SETI@HOME." <http://setiathome.ssl.berkeley.edu/>.
- [24] "Folding@Home." <http://folding.stanford.edu/>.
- [25] J. Liang, R. Kumar, and K. W. Ross, "The fasttrack overlay : A measurement study," *Computer Networks*, vol. 50, no. 6, pp. 842–858, 2006.
- [26] P. Maymounkov and D. Mazieres, "Kademlia : A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*, pp. 53–65, Springer, 2002.
- [27] P. Druschel and A. Rowstron, "Past : A large-scale, persistent peer-to-peer storage utility," in *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*, pp. 75–80, IEEE, 2001.

- [28] G. Wang and J. Tang, "The nosql principles and basic application of cassandra model," in *Computer Science & Service System (CSSS), 2012 International Conference on*, pp. 1332–1335, IEEE, 2012.
- [29] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv :1407.3561*, 2014.
- [30] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream : high-bandwidth multicast in cooperative environments," in *ACM SIGOPS Operating Systems Review*, vol. 37, pp. 298–313, ACM, 2003.
- [31] T. White, *Hadoop : The definitive guide*. " O'Reilly Media, Inc.", 2012.
- [32] S. Giordano, "Handbook of wireless networks and mobile computing," ch. Mobile Ad Hoc Networks, pp. 325–346, New York, NY, USA : John Wiley & Sons, Inc., 2002.
- [33] L. Liu, Y. Jing, Y. Zhang, and B. Xia, "A survey on p2p file sharing algorithms over manets," *Consumer Electronics Times*, vol. 2, no. 2, pp. 109–115, 2013.
- [34] Y. C. Hu, S. M. Das, and H. Pucha, "Exploiting the synergy between peer-to-peer and mobile ad hoc networks.," in *HotOS*, pp. 37–42, 2003.
- [35] H. Pucha, S. M. Das, and Y. C. Hu, "Ekta : An efficient dht substrate for distributed applications in mobile ad hoc networks," in *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, pp. 163–173, IEEE, 2004.
- [36] Y.-C. Tung and K. C.-J. Lin, "Location-assisted energy-efficient content search for mobile peer-to-peer networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pp. 477–482, IEEE, 2011.
- [37] S. Srinivasan, A. Moghadam, S. G. Hong, and H. Schulzrinne, "7ds-node cooperation and information exchange in mostly disconnected networks," in *Communications, 2007. ICC'07. IEEE International Conference on*, pp. 3921–3927, IEEE, 2007.
- [38] C. Lindemann and O. P. Waldhorst, "A distributed search service for peer-to-peer file sharing in mobile applications," in *Peer-to-Peer Computing, 2002.(P2P 2002). Proceedings. Second International Conference on*, pp. 73–80, IEEE, 2002.
- [39] A. Klemm, C. Lindemann, and O. P. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 4, pp. 2758–2763, IEEE, 2003.
- [40] F. Li and J. Wu, "Mops : Providing content-based service in disruption-tolerant networks," in *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on*, pp. 526–533, IEEE, 2009.
- [41] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 748–760, June 2008.

- [42] C. Boldrini, M. Conti, and A. Passarella, "Contentplace : social-aware data dissemination in opportunistic networks," in *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 203–210, ACM, 2008.
- [43] K. Chen, H. Shen, and H. Zhang, "Leveraging social networks for p2p content-based file sharing in disconnected manets," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 235–249, 2014.
- [44] K. Chandrasekaran, "Epidemic routing for partially-connected ad hoc networks," *University of Kansas, Lawrence*.
- [45] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 373–384, ACM, 2007.
- [46] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pp. 62–66, ACM, 2007.
- [47] J. Liu, M. Tang, and G. Yu, "Adaptive spray and wait routing based on relay-probability of node in dtn," in *Computer Science & Service System (CSSS), 2012 International Conference on*, pp. 1138–1141, IEEE, 2012.
- [48] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus : Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pp. 79–85, IEEE, 2007.
- [49] "Open Garden." <https://www.opengarden.com/>.
- [50] T. Kärkkäinen, M. Pitkänen, P. Houghton, and J. Ott, "Scampi application platform," in *Proceedings of the Seventh ACM International Workshop on Challenged Networks, CHANTS '12*, (New York, NY, USA), pp. 83–86, ACM, 2012.
- [51] C. Diot, M. Martin, and N. Erik, "Haggle project," 2004.
- [52] S. Farrell, S. Weber, A. McMahan, E. Meehan, and K. Hartnett, "An n4c dtn router node design," in *1st Extreme Workshop on Communication, Laponia, Sweden*, 2009.
- [53] "C3po - collaborative creation of contents and publishing using opportunistic networks," 2014-2017.
- [54] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," tech. rep., 2003.
- [55] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," tech. rep., 2003.
- [56] S. Dimatteo, P. Hui, B. Han, and V. O. K. Li, "Cellular traffic offloading through WiFi networks," *Proceedings - 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011*, pp. 192–201, 2011.

- [57] S. Andreev, A. Pyattaev, K. Johnsson, O. Galinina, and Y. Koucheryavy, "Cellular traffic offloading onto network-assisted device-to-device connections," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 20–31, 2014.
- [58] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "Proximity-based data offloading via network assisted device-to-device communications," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, IEEE, 2013.
- [59] H. Yoon, J. Kim, F. Tan, and R. Hsieh, "On-demand video streaming in mobile opportunistic networks," in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pp. 80–89, IEEE, 2008.
- [60] H. Lee, J.-Y. Yoo, and J. Kim, "Movi+ : Opportunity extension for mobile peer-to-peer video on demand," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pp. 247–252, IEEE, 2013.
- [61] L. Keller, C. Fragouli, and U. C. Irvine, "MicroCast : Cooperative Video Streaming on Smartphones Categories and Subject Descriptors," *MobiSys 2012*, pp. 57–69, 2012.
- [62] N. Golrezaei, M. Ji, A. F. Molisch, A. G. Dimakis, and G. Caire, "Device-to-device communications for wireless video delivery," *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, no. March, pp. 930–933, 2012.
- [63] Q. Gong, Y. Guo, Y. Chen, Y. Liu, and F. Xie, "Design and evaluation of a wifi-direct based lte cooperative video streaming system," in *Global Communications Conference (GLOBECOM), 2016 IEEE*, pp. 1–6, IEEE, 2016.
- [64] G. Lee, W. Saad, M. Bennis, A. Mehbodniya, and F. Adachi, "Online channel allocation for full-duplex device-to-device communications," in *Globecom Workshops (GC Wkshps), 2016 IEEE*, pp. 1–6, IEEE, 2016.
- [65] M.-S. Pan and Y.-P. Lin, "Efficient data dissemination for wi-fi peer-to-peer networks by unicasting among wi-fi p2p groups," *Wireless Networks*, pp. 1–19.
- [66] B. Chen and C. Yang, "Caching policy optimization for d2d communications by learning user preference," *arXiv preprint arXiv :1704.04860*, 2017.
- [67] T. Peng, Q. Lu, H. Wang, S. Xu, and W. Wang, "Interference avoidance mechanisms in the hybrid cellular and device-to-device systems," in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 617–621, Sept 2009.
- [68] S. Xu, H. Wang, T. Chen, Q. Huang, and T. Peng, "Effective interference cancellation scheme for device-to-device communication underlying cellular networks," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pp. 1–5, IEEE, 2010.
- [69] B. Kaufman and B. Aazhang, "Cellular networks with an overlaid device to device network," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pp. 1537–1541, IEEE, 2008.

- [70] P. Janis, V. Koivunen, C. Ribeiro, J. Korhonen, K. Doppler, and K. Hugl, "Interference-aware resource allocation for device-to-device radio underlaying cellular networks," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pp. 1–5, IEEE, 2009.
- [71] C. H. Yu, K. Doppler, C. B. Ribeiro, and O. Tirkkonen, "Resource sharing optimization for device-to-device communication underlaying cellular networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2752–2763, 2011.
- [72] R. R. Tyagi, F. Aurzada, K.-D. Lee, S. G. Kim, and M. Reisslein, "Impact of retransmission limit on preamble contention in lte-advanced network," *IEEE Systems Journal*, vol. 9, no. 3, pp. 752–765, 2015.
- [73] C. Xu, L. Song, Z. Han, D. Li, and B. Jiao, "Resource allocation using a reverse iterative combinatorial auction for device-to-device underlay cellular networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 4542–4547, IEEE, 2012.
- [74] G. Zhang, "Subcarrier and bit allocation for real-time services in multiuser ofdm systems," in *Communications, 2004 IEEE International Conference on*, vol. 5, pp. 2985–2989, IEEE, 2004.
- [75] M. Jung, K. Hwang, and S. Choi, "Joint mode selection and power allocation scheme for power-efficient device-to-device (d2d) communication," in *Vehicular technology conference (VTC Spring), 2012 IEEE 75th*, pp. 1–5, IEEE, 2012.
- [76] M. Belleschi, G. Fodor, and A. Abrardo, "Performance analysis of a distributed resource allocation scheme for d2d communications," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pp. 358–362, IEEE, 2011.
- [77] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications underlaying cellular networks," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3541–3551, 2013.
- [78] I. Kim, I.-S. Park, and Y. H. Lee, "Use of linear programming for dynamic subcarrier and bit allocation in multiuser ofdm," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1195–1207, 2006.
- [79] P. Cheng, L. Deng, H. Yu, Y. Xu, and H. Wang, "Resource allocation for cognitive networks with d2d communication : An evolutionary approach," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pp. 2671–2676, IEEE, 2012.
- [80] K. Vanganuru, S. Ferrante, and G. Sternberg, "System capacity and coverage of a cellular network with d2d mobile relays," in *MILITARY COMMUNICATIONS CONFERENCE, 2012-MILCOM 2012*, pp. 1–6, IEEE, 2012.
- [81] A. Asadi and V. Mancuso, "Energy efficient opportunistic uplink packet forwarding in hybrid wireless networks," in *Proceedings of the Fourth International Conference on Future Energy Systems, e-Energy '13, (New York, NY, USA)*, pp. 261–262, ACM, 2013.

- [82] A. Asadi and V. Mancuso, "On the compound impact of opportunistic scheduling and d2d communications in cellular networks," in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*, pp. 279–288, ACM, 2013.
- [83] S. C. Spinella, G. Araniti, A. Iera, and A. Molinaro, "Integration of ad-hoc networks with infrastructured systems for multicast services provisioning," in *Ultra Modern Telecommunications & Workshops, 2009. ICUMT'09. International Conference on*, pp. 1–6, IEEE, 2009.
- [84] N. Golrezaei, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," in *2012 IEEE International Conference on Communications (ICC)*, pp. 7077–7081, June 2012.
- [85] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Device-to-device collaboration through distributed storage," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 2397–2402, IEEE, 2012.
- [86] T. Iyer, R. Hsieh, N. B. Rizvandi, B. Varghese, and R. Boreli, "Mobile P2P trusted on-demand video streaming," *CoRR*, vol. abs/1204.0094, 2012.
- [87] J. Sung, D. Lee, Y. Bang, J. Lee, and J. K. K. Rhee, "Energy-aware algorithms for network-assisted device-to-device content delivery networks," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 469–471, Oct 2015.
- [88] F. Wang, C. Xu, L. Song, Q. Zhao, X. Wang, and Z. Han, "Energy-aware resource allocation for device-to-device underlay communication," in *2013 IEEE International Conference on Communications (ICC)*, pp. 6076–6080, June 2013.
- [89] Y. Zhao, Y. Li, H. Zhang, N. Ge, and J. Lu, "Fundamental tradeoffs on energy-aware d2d communication underlaying cellular networks : A dynamic graph approach," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 864–882, April 2016.
- [90] L. Nobach and D. Hausheer, *Towards Decentralized, Energy- and Privacy-Aware Device-to-Device Content Delivery*, pp. 128–132. Berlin, Heidelberg : Springer Berlin Heidelberg, 2014.
- [91] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "Ieee 802.11 wireless local area networks," *IEEE Communications magazine*, vol. 35, no. 9, pp. 116–126, 1997.
- [92] Bluetooth SIG, "BLUETOOTH SPECIFICATION Version 4.2," 2014.
- [93] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct : overview and experimentation," *Wireless Communications, IEEE*, vol. 20, no. 3, pp. 96–104, 2013.
- [94] M. A. Khan, W. Cherif, F. Filali, and R. Hamila, "Wi-fi direct research-current status and future perspectives," *Journal of Network and Computer Applications*, vol. 93, pp. 245–258, 2017.

- [95] K. Liu, W. Shen, B. Yin, X. Cao, L. X. Cai, and Y. Cheng, "Development of mobile ad-hoc networks over wi-fi direct with off-the-shelf android phones," in *Communications (ICC), 2016 IEEE International Conference on*, pp. 1–6, IEEE, 2016.
- [96] U. Botrel Menegato, L. Souza Cimino, S. E. Delabrida Silva, F. A. Medeiros Silva, J. Castro Lima, and R. A. R. Oliveira, "Dynamic clustering in wifi direct technology," in *Proceedings of the 12th ACM international symposium on Mobility management and wireless access*, pp. 25–29, ACM, 2014.
- [97] M. Jin, J.-Y. Jung, and J.-R. Lee, "Dynamic power-saving method for wi-fi direct based iot networks considering variable-bit-rate video traffic," *Sensors*, vol. 16, no. 10, p. 1680, 2016.
- [98] K.-W. Lim, Y. Seo, Y.-B. Ko, J. Kim, and J. Lee, "Dynamic power management in wi-fi direct for future wireless serial bus," *Wireless networks*, vol. 20, no. 7, pp. 1777–1793, 2014.
- [99] H. Je, D. Kwon, H. Kim, and H. Ju, "Mobile network configuration for large-scale multimedia delivery on a single wlan," in *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, pp. 1–6, IEEE, 2014.
- [100] R. Kanaoka and Y. Tobe, "Design of a data transfer system on smartphones using wi-fi direct and accelerometers," in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pp. 71–75, Oct 2014.
- [101] W. Cherif, M. A. Khan, F. Filali, S. Sharafeddine, and Z. Dawy, "P2p group formation enhancement for opportunistic networks with wi-fi direct," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, March 2017.
- [102] H. B. Li, R. Miura, and F. Kojima, "Channel access proposal for enabling quick discovery for d2d wireless networks," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1012–1016, Jan 2017.
- [103] "Wi-Fi Aware." <https://www.wi-fi.org/discover-wi-fi/wi-fi-aware>.
- [104] D. Mark, J. Varma, J. LaMarche, A. Horovitz, and K. Kim, "Peer-to-peer using multi-peer connectivity," in *More iPhone Development with Swift*, pp. 239–280, Springer, 2015.
- [105] M. Maity, B. Raman, and M. Vutukuru, "Tcp download performance in dense wifi scenarios : Analysis and solution," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 213–227, 2017.
- [106] "Magency." <http://www.magency.me/>.
- [107] "nginx." <https://nginx.org/en/>.
- [108] V. P. Kemerlis, E. C. Stefanis, G. Xylomenos, and G. C. Polyzos, "Throughput unfairness in tcp over wifi," in *WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services*, pp. 26–31, 2006.

- [109] “WiFi Performance Benchmark Testing in a Class-room Environment - Alethea.” [https://alethea.in/wp-content/uploads/Ubiquiti\\_HD\\_Test\\_Report.pdf](https://alethea.in/wp-content/uploads/Ubiquiti_HD_Test_Report.pdf).
- [110] “UniFi AP HD.” <https://unifi-hd.ubnt.com/>.
- [111] “R710 | Ruckus Wireless.” <https://www.ruckuswireless.com/products/access-points/ruckus-indoor/ruckus-r710>.
- [112] “802.11ac access point Aruba 320 Series.” <http://www.arubanetworks.com/products/networking/access-points/aruba-320-series/>.
- [113] “Cisco Meraki MR52.” <https://meraki.cisco.com/products/wireless/mr52>.
- [114] R. Riggio, M. K. Marina, and T. Rasheed, “Interference management in software-defined mobile networks,” in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pp. 626–632, IEEE, 2015.
- [115] O. Stiti, O. Braham, and G. Pujolle, “Virtual openflow-based sdn wi-fi access point,” in *Global Information Infrastructure and Networking Symposium (GIIS), 2015*, pp. 1–3, IEEE, 2015.
- [116] M. Seyedebrahimi, F. Bouhafs, A. Raschellà, M. Mackay, and Q. Shi, “Sdn-based channel assignment algorithm for interference management in dense wi-fi networks,” in *Networks and Communications (EuCNC), 2016 European Conference on*, pp. 128–132, IEEE, 2016.
- [117] A. Gupta, J. Min, and I. Rhee, “Wifox : Scaling wifi performance for large audience environments,” in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pp. 217–228, ACM, 2012.
- [118] K. V. Cardoso and J. F. de Rezende, “Increasing throughput in dense 802.11 networks by automatic rate adaptation improvement,” *Wireless Networks*, vol. 18, no. 1, pp. 95–112, 2012.
- [119] B. Bellalta, “Ieee 802.11 ax : High-efficiency wlans,” *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.
- [120] E. Perahia, C. Cordeiro, M. Park, and L. L. Yang, “Ieee 802.11 ad : Defining the next generation multi-gbps wi-fi,” in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pp. 1–5, IEEE, 2010.
- [121] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, “Hamming distance metric learning,” in *Advances in neural information processing systems*, pp. 1061–1069, 2012.
- [122] A. Montresor and M. Jelasity, “PeerSim : A scalable P2P simulator,” in *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P’09)*, (Seattle, WA), pp. 99–100, Sept. 2009.
- [123] Z. Yang, “Powertutor-a power monitor for android-based mobile platforms,” *EECS, University of Michigan, retrieved September*, vol. 2, p. 19, 2012.
- [124] A. J. Oliner, A. P. Iyer, I. Stoica, E. Lagerspetz, and S. Tarkoma, “Carat : Collaborative energy diagnosis for mobile devices,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, p. 10, ACM, 2013.



- [125] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha, "Appscope : Application energy metering framework for android smartphone using kernel activity monitoring.," in *USENIX Annual Technical Conference*, vol. 12, pp. 1–14, 2012.
- [126] M. A. Hoque, M. Siekkinen, K. N. Khan, Y. Xiao, and S. Tarkoma, "Modeling, profiling, and debugging the energy consumption of mobile devices," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 39, 2016.
- [127] S. Iskounen, T. M. T. Nguyen, and S. Monnet, "Wifi-direct simulation for inet in omnet++," *arXiv preprint arXiv:1609.04604*, 2016.
- [128] K. Benzekki, A. E. Fergougui, and A. E. Elalaoui, "Software-defined networking (sdn) : a survey," *Security and Communication Networks*, vol. 9, pp. 5803–5833, 12 2016.
- [129] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [130] J. A. Hartigan and M. A. Wong, "A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [131] D. Brélaz, "New methods to color the vertices of a graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251–256, 1979.
- [132] "OMNET++ Wi-Fi Direct Implementation Source Code. ." <https://www-phare.lip6.fr/~trnguyen/research/wifidirect/>.

## Liste des figures

---

II.1	Architecture de Napster . . . . .	11
II.2	Illustration d'un partage de fichier dans un réseau Gnutella. . . . .	12
II.3	Architecture de KaZaa/FastTrack . . . . .	13
II.4	Architecture de BitTorrent . . . . .	14
II.5	Architecture de GFS (crédits Ghemawat et al. [16]). . . . .	17
II.6	Topologie d'un réseau Wi-Fi à plusieurs points d'accès. . . . .	29
II.7	Topologie d'un groupe de communication Wi-Fi adhoc . . . . .	30
II.8	Topologie d'un groupe de communication Wi-Fi Direct. . . . .	31
II.9	Modes de formation de groupes Wi-Fi Direct (crédits Camps-Mur et al. [93]) . . . . .	32
II.10	Temps de complétion pour un fichier de 100 Mo en fonction du nombre de téléchargements parallèles sur une infrastructure Wi-Fi. . . . .	35
III.1	Représentation d'un réseau Wi-Fi à grande échelle . . . . .	41
III.2	Diagramme de séquence d'un téléchargement depuis le serveur. . . . .	53
III.3	Diagramme de séquence d'un échange D2D bidirectionnel. . . . .	54
IV.1	Description des différentes phases de dissémination. . . . .	59
IV.2	Prise en compte de la batterie et de la bande passante des périphériques : les phases de dissémination. . . . .	62
V.1	(a) Impact du degré de parallélisme sur le temps de complétion (b) Évolution du remplissage des chunks . . . . .	77
V.2	Impact de la taille du réseau sur le temps de complétion. . . . .	78
V.3	Impact de la taille des chunks sur le temps de complétion . . . . .	79
V.4	Impact du temps d'établissement de connexion sur les performances de la dissémination. . . . .	80
V.5	Impact du nombre de chunks téléchargés depuis le serveur $Max_{AP}$ . . . . .	81
V.6	Impact du nombre maximum de chunks par échange D2D $Max_{D2D}$ . . . . .	81
V.7	Comparaison des temps d'activité sur le réseau sans clustering. . . . .	82

V.8	(V.8a) Comparaison de l'évolution de la complétion des périphériques; (V.8b) Comparaison des temps d'activité sur le réseau des différentes approches. . . . .	83
V.9	Réduction de l'activité des périphériques sur le réseau avec la prise en compte de l'hétérogénéité des batteries. . . . .	84
V.10	(V.10a) Différence de batterie finale en fin de dissémination avec la prise en compte l'hétérogénéité des batteries; (V.10b) Évolution de la batterie minimale sur le réseau pour un réseau dont 20% des périphériques sont <i>mauvais</i> . . . . .	85
V.11	Comparaison des temps de complétion. . . . .	86
V.12	(V.12a) Phases de dissémination du serveur avec la prise en compte de l'hétérogénéité des batteries pour un réseau avec 40% de mauvais périphériques et (V.12b) évolution de la complétion des périphériques. . . . .	87
V.13	(V.13a) Phases de dissémination par serveur avec prise en compte de l'hétérogénéité des batteries pour un réseau avec 20% de <i>mauvais</i> périphériques et (V.13b) évolution de la complétion des périphériques. . . . .	88
V.14	Impact de la prise en compte de la bande passante sur le temps de complétion. . . . .	89
V.15	Gain énergétique de la prise en compte de la batterie et de la bande passante comparée à la prise en compte de la batterie (40% de <i>mauvais</i> périphériques). . . . .	90
VI.1	Architecture du réseau. . . . .	95
VI.2	Boucle de contrôle SDN. . . . .	96
VI.3	Topologie du réseau obtenue après le clustering . . . . .	97
VI.4	Contrôle de la puissance du signal avec le clustering . . . . .	99
VI.5	Évaluation du débit pour 802.11g(54Mbps) et 802.11n(150Mbps). . . . .	103
VI.6	Mesure de temps de calcul de clusters. . . . .	104
VI.7	Topologie de simulation sur OMNET++ (credits S. Iskounen [127]) . . . . .	105
VI.8	Temps moyen de téléchargement avec $K = 4$ . . . . .	107
VI.9	Impact de l'utilisation de clusters sur le taux d'erreurs et sur le temps de téléchargement. . . . .	107
VI.10	Nombre optimal de clusters. . . . .	108

## Liste des tableaux

---

V.1	Délais et débits moyens sur des tablettes Apple iPad Air 2 (20 mesures)	74
V.2	Configuration par défaut du simulateur . . . . .	75
V.3	Configuration du simulateur pour l'évaluation de la prise en compte de l'hétérogénéité des périphériques mobiles. . . . .	76
VI.1	Paramètres de simulation . . . . .	106

## Liste des Algorithmes

---

1	Initialisation et lancement de la dissémination . . . . .	47
2	Comportement du serveur . . . . .	48
3	Fonction <i>updateNST</i> du serveur . . . . .	49
4	Déclenchement d'un échange D2D . . . . .	49
5	Procédure des clients (périphériques mobiles) . . . . .	50
6	Dissémination avec prise en compte de la batterie . . . . .	60
7	Prise en compte de la batterie et de la bande passante . . . . .	64
8	Algorithme de sélection des transferts depuis le point d'accès . . . . .	65
9	Sélection des paires d'échanges D2D . . . . .	66
10	Algorithme des <i>K</i> -moyennes . . . . .	98
11	Affectation de clusters . . . . .	99

# Vers une dissémination efficace de données volumineuses sur des réseaux Wi-Fi denses

Lyes HAMIDOUCHE

## Résumé

Face à la prolifération des technologies mobiles et à l'augmentation du volume des données utilisées par les applications mobiles, les périphériques consomment de plus en plus de bande passante. Dans cette thèse, nous nous concentrons sur les réseaux Wi-Fi denses comme cela peut être le cas lors d'événements à grande échelle (ex : conférences, séminaire, etc.) où un serveur doit acheminer des données à un grand nombre de périphériques dans une fenêtre temporelle réduite. Dans ce contexte, la consommation de bande passante et les interférences engendrées par les téléchargements parallèles d'une donnée volumineuse par plusieurs périphériques connectés au même réseau dégradent les performances. Les technologies de communication Device-to-Device (D2D) comme Bluetooth ou Wi-Fi Direct permettent de mieux exploiter les ressources du réseau et d'améliorer les performances pour offrir une meilleure qualité d'expérience (QoE) aux utilisateurs.

Dans cette thèse nous proposons deux approches pour l'amélioration des performances de la dissémination de données. La première approche, plus adaptée à une configuration mobile, consiste à utiliser des connexions D2D en point-à-point sur une topologie plate pour les échanges de données. Nos évaluations montrent que notre approche permet de réduire les temps de dissémination jusqu'à 60% par rapport à l'utilisation du Wi-Fi seul. De plus, nous veillons à avoir une répartition équitable de la charge énergétique sur les périphériques afin de préserver les batteries les plus faibles du réseau. Nous avons pu voir qu'avec la prise en compte de l'autonomie des batteries et de la bande passante, la sollicitation des batteries les plus faibles peut être réduite de manière conséquente.

La deuxième approche, plus adaptée à des configurations statiques, consiste à mettre en place des topologies hiérarchiques dans lesquelles on regroupe les périphériques par clusters. Dans chaque cluster, un périphérique est élu pour être le relais des données qu'il recevra depuis le serveur et qu'il transmettra à ses voisins. Cette approche permet de gérer plus efficacement les interférences en adaptant la puissance du signal afin de limiter la portée des clusters. Dans ce cas, nous avons observé jusqu'à 30 % de gains en temps de dissémination.

Dans la continuité des travaux de cette thèse, nous discutons de plusieurs perspectives qu'il serait intéressant d'entreprendre par la suite, notamment l'adaptation automatique du protocole de dissémination à l'état du réseau et l'utilisation simultanée des deux types de topologie plate et hiérarchique.

## Abstract

We are witnessing a proliferation of mobile technologies and an increasing volume of data used by mobile applications. Devices consume thus more and more bandwidth. In this thesis, we focus on dense Wi-Fi networks during large-scale events (such as conferences). In this context, the bandwidth consumption and the interferences caused by the parallel downloads of a large volume of data by several mobile devices that are connected to the same Wi-Fi network degrade the performance of the dissemination. Device-to-Device (D2D) communication technologies such as Bluetooth or Wi-Fi Direct can be used in order to improve network performance to deliver better QoE to users.

In this thesis we propose two approaches for improving the performance of data dissemination. The first approach, more suited to a dynamic configuration, is to use point-to-point D2D connections on a flat topology for data exchange. Our evaluations show that our approach can reduce dissemination times by up to 60% compared to using Wi-Fi alone. In addition, we ensure a fair distribution of the energy load on the devices to preserve the weakest batteries in the network. We have observed that by taking into account the battery life and the bandwidth of mobile devices, the solicitation of the weakest batteries can be reduced significantly.

The second approach, more adapted to static configurations, consists in setting up hierarchical topologies by gathering mobile devices in small clusters. In each cluster, a device is chosen to relay the data that it receives from the server and forwards it to its neighbors. This approach helps to manage interference more efficiently by adjusting the signal strength in order to limit cluster reach. In this case, we observed up to 30% gains in dissemination time.

In the continuity of this thesis work, we discuss three perspectives which would be interesting to be undertaken, in particular the automatic adaptation of the dissemination to the state of the network and the simultaneous use of both topology types, flat and hierarchical.