HAL
open science

Systèmes pair-à-pair pour l'informatique opportuniste

Armel Esnault

To cite this version:

Armel Esnault. Systèmes pair-à-pair pour l'informatique opportuniste. Réseaux et télécommunications [cs.NI]. Université de Bretagne Sud, 2017. Français. NNT : 2017LORIS432 . tel-01455038v2

HAL Id: tel-01455038

https://theses.hal.science/tel-01455038v2

Submitted on 24 May 2017

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.

THÈSE / UNIVERSITÉ DE BRETAGNE SUD

UFR Sciences et Sciences de l'Ingénieur

sous le sceau de l'Université Européenne de Bretagne

Pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE SUD

Mention : Informatique

École Doctorale SICMA

présentée par

Armel Esnault

IRISA Institut de Recherche en Informatique et Systèmes Aléatoires

Systèmes pair-à-pair pour l'informatique opportuniste

Thèse soutenue le 20-01-2017,
devant la commission d'examen composée de :

F. Frederique Laforest
Professeur, Télécom Saint-Etienne / Président

M. David Bromberg
Professeur, Université de Rennes1 / Rapporteur

M. Etienne Rivière
Maître assistant, Université de Neuchâtel / Rapporteur

M. Frédéric Guidec
Professeur, Université de Bretagne-Sud / Directeur de thèse

M. Nicolas Le Sommer
Maître de Conferences, Université de Bretagne-Sud / Encadrant de thèse

Table des matières

1 Introduction **11**

1.1 Réseaux à un saut 12

1.2 Réseaux hybrides multi-sauts 12

1.3 Réseaux pair-à-pair et opportunistes à connectivité intermittente 13

1.4 Exemple de RHCI 16

1.5 Proposition de thèse 16

1.6 Plan du document 17

I État de l'art **19**

2 Communication opportuniste **21**

2.1 Vers des réseaux opportunistes hybrides à connectivité intermittente . . . 21

2.1.1 Réseaux sans fil à un saut 21

2.1.2 Réseaux mobiles multi-sauts 22

2.1.3 Réseaux sans fil à connectivité intermittente 23

2.1.4 Réseaux hybrides à connectivité intermittente (RHCI) 24

2.2 Technologies sans fil pour la communication opportuniste 25

2.2.1 Bluetooth 25

2.2.2 Wi-Fi 26

2.2.3 ZigBee 27

2.2.4 LTE direct 27

2.2.5 LoRaWAN 28

2.3 Protocoles de communication 28

2.3.1 Unique copie et délégation 28

2.3.2 Inondation 29

2.3.3 Historique des rencontres et analyse sociale 30

2.3.4 Position géographique 31

2.4 Déchargement de données et réseaux hybrides 31

2.5 Conclusion 33

3 Principes et mécanismes des réseaux pair-à-pair **35**

3.1 Applications pair-à-pair 35
3.1.1 Partage de fichiers et diffusion de données 35
3.1.2 Calcul réparti 36
3.2 Propriétés des réseaux pair-à-pair 36
3.2.1 Passage à l'échelle 36
3.2.2 Recherche d'information 36
3.2.3 Capacité de survie et tolérance aux pannes 36
3.3 Architecture pair-à-pair structurée 37
3.3.1 Chord 37
3.3.2 Pastry 38
3.3.3 Kademlia 38
3.3.4 CAN 39
3.3.5 T-MAN 40
3.4 Architecture pair-à-pair non structurée 41
3.4.1 Principe du *gossiping* 41
3.4.2 Cyclon 41
3.4.3 Napster 42
3.4.4 Fastrack 42
3.4.5 Gnutella 43
3.4.6 BitTorrent 43
3.4.7 Utilisation des filtres de Bloom dans les réseaux pair-à-pair non structurés 44
3.5 Conclusion 45

4 Vers l'utilisation des mécanismes pair-à-pair dans les réseaux opportunistes hybrides 47

4.1 Réseaux pair-à-pair et opportuniste : points communs et différences . . . 47
4.2 Utilisation des réseaux pair-à-pair structurés 49
4.3 Utilisation des réseaux pair-à-pair non structurés 50
4.4 Conclusion 51

II Contribution 53

5 Nephila : une plateforme pair-à-pair pour des réseaux hybrides à connectivité intermittente 55

5.1 Vue d'ensemble et architecture de la plateforme Nephila 55

5.2 Réseau de recouvrement pair-à-pair 56

5.3 Découverte des voisins 57

5.3.1 Partie ad hoc du réseau 58

5.3.2 Partie infrastructure du réseau 59

5.4 Calcul des chemins 62

5.5 Algorithme de transfert de messages 65

5.6 Mécanisme de « vaccination » 67

5.7 Conclusion 68

6 Échange de données avec Nephila 69

6.1 Communication point-à-point 69

6.2 Communication Anycast 71

6.3 Communication fondée sur le contenu 72

6.4 Conclusion 74

III Expérimentation et évaluation 77

7 Évaluation de la communication point-à-point 79

7.1 Scénario et paramètres de simulation 79

7.2 Résultats de simulation 82

7.2.1 Utilisation de l'infrastructure 85

7.2.2 Surcoût dû à l'échange des FBC 85

7.2.3 Influence de la date d'expiration 87

7.3 Conclusion 87

8 Évaluation de la communication anycast 89

8.1 Scénario et paramètres de simulation 89

8.2 Résultats de simulation 91

8.3 Conclusion 94

9 Évaluation de la communication basée sur le contenu **95**

9.1 Scénario et paramètres de simulation 95

9.2 Résultats de simulation 96

9.3 Conclusion 101

10 Conclusion et perspectives **103**

Table des figures

1.1 Exemple d'un réseau hybride multi-sauts 12
1.2 Exemple de catastrophes naturelles 14
1.3 Exemple d'un réseau hybride à connectivité intermittente 14
1.4 Architectures pair-à-pair 15

2.1 Réseaux sans fil à un saut 22
2.2 Exemple de réseau mobile multi-sauts en mode ad hoc 22
2.3 Connectivité du réseau 23
2.4 Formation d'un RHCI au sein d'une ville 24
2.5 Déchargement de données dans les parties sans fil du réseau 32

3.1 Exemple de réseau Chord avec 16 nœuds 38
3.2 Exemple d'arbre binaire Kademlia 39
3.3 Exemple de routage dans un réseau CAN à 2 dimensions 40
3.4 Organisation des pairs sous la forme d'un tore dans un réseau T-MAN . . 40
3.5 Architecture centralisée de Napster 42
3.6 Architecture de Fastrack 42
3.7 Architecture de Bittorent 43

4.1 Inadéquation entre le réseau logique et le réseau physique 49

5.1 Architecture de Nephila 56
5.2 Surcouche pair-à-pair 57
5.3 Découverte des voisins dans les parties ad hoc du réseau 58
5.4 Exemple d'une opération d'échange de nœuds 60
5.4 Exemple d'une opération d'échange de nœuds 61
5.5 Fonctions *reinforce* et *decay*. 63
5.6 Fonctions *merge* et *query*. 64
5.7 Processus d'échange de messages. 65

6.1 Exemple de propagation des FBCs 70
6.2 Example d'utilisation des groupes anycast dans Nephila 71

6.3 Exemple d'utilisation du mode de communication anycast pour implémenter un service permettant l'accès à Internet 72
6.4 Exemple de dissémination de données en utilisant l'approche basée sur le contenu. 74

7.1 Ville de Vannes 80
7.2 Illustration de l'environnement de simulation. 80
7.3 Échange de messages 81
7.4 Taux de délivrance des messages 83
7.5 Latence des messages 83
7.6 Nombre de sauts requis pour des messages ayant une date d'expiration de 20 minutes 84
7.7 Charge dans la partie infrastructure du réseau 84
7.8 Charge dans les parties sans fil du réseau 86
7.9 Nombre de messages dans les parties sans fil du réseau en fonction du temps, pour 2000 piétons 86
7.10 Nombre de FBC échangés dans les parties sans fil du réseau 86

8.1 Environnement de simulation 90
8.2 Scénario Anycast 90
8.3 Taux et latence de délivrance 92
8.4 Distribution du nombre de requêtes délivrées 93
8.5 Mécanisme de vaccination 94

9.1 Taux de délivrance des requêtes 96
9.2 Latence moyenne d'arrivée des requêtes 97
9.3 Nombre de sauts moyen des requêtes 97
9.4 Taux de délivrance des réponses 98
9.5 Latence moyenne d'arrivée des réponses 98
9.6 Nombre de sauts moyen des réponses 99
9.7 latence moyenne des couples requête/réponse 99
9.8 Nombre de sauts moyen des couples requête/réponse 100
9.9 charge moyenne par nœud par seconde 100

Liste des tableaux

7.1 Paramètres de simulation. 82

1 Introduction

L'Internet du futur sera sans nul doute celui des objets connectés [1]. Ces objets fixes ou mobiles aux fonctionnalités et caractéristiques diverses (e.g., smartphones, montres connectées, capteurs, actionneurs, robots) peuvent être interconnectés via des interfaces de communication sans fil de courte portée (e.g., Wi-Fi, Wi-Fi Direct, Bluetooth) ou longue portée (e.g., 3G/4G, Lora, SigFox). Ces objets reposent sur des infrastructures déployées par des opérateurs. La plupart des interfaces de communication de courte portée, comme Wi-Fi Direct ou Bluetooth, permettent quant à elles aux objets de communiquer directement entre eux sans recourir à une quelconque infrastructure. Les objets peuvent être utilisés pour collecter des données environnementales, suivre l'activité physique des personnes, les assister à domicile, leur permettre de communiquer ou de réaliser des tâches spécifiques, etc. Les objets connectés à Internet, dont le traitement et la collecte de données sont déportés vers des architectures centralisées de type Web ou Cloud, vont inéluctablement augmenter le trafic de données dans les réseaux des opérateurs. Il nous semble dès lors intéressant d'envisager d'autres types d'architectures réseaux et modes de communication pour supporter cette croissance du trafic de données.

Dans cette thèse, nous proposons d'étudier les communications dans les réseaux composés à la fois d'une infrastructure et de parties formées par des objets (aussi appelés nœuds du réseau) fixes ou mobiles qui communiquent directement entre eux via des interfaces de communication sans fils de courte portée. La mise en veille périodique de certains de ces objets pour des raisons d'économie d'énergie, ou la mobilité de ceux-ci, combinée à la faible portée de leur interface de communication entraînent de fréquentes ruptures de connectivité entre les objets et une topologie réseau très fluctuante. Par la suite, les réseaux de ce type seront qualifiés de réseaux hybrides à connectivité intermittente.

Pour offrir une vision homogène de la topologie de ces réseaux hybrides et des moyens de communication entre des objets connectés par intermittence, nous proposons une approche reposant sur un *overlay* pair-à-pair et des techniques de communication opportunistes. Cette approche vise à permettre de décharger le réseau des opérateurs en faisant passer une partie des échanges par des communications de proximité.

Elle présente par ailleurs des avantages certains en matière de tolérance aux ruptures de connectivité, aux zones faiblement peuplées, à la censure de la part d'un régime politique [2, 3] et aux défaillances d'équipements de communication, par exemple lors de catastrophes naturelles. Dans le reste de ce chapitre, nous présentons des cas d'utilisation

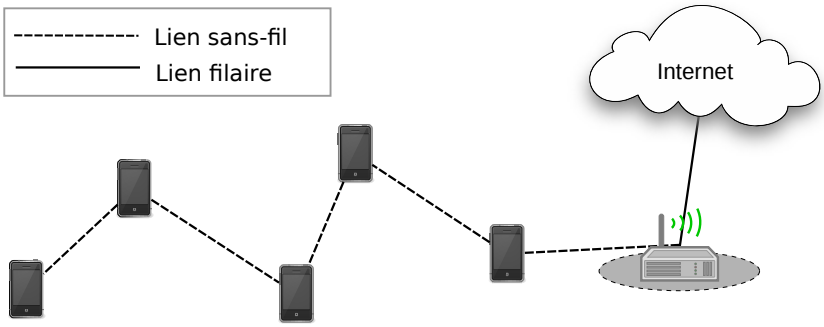


FIGURE 1.1 – Exemple d'un réseau hybride multi-sauts

de réseaux hybrides, le principe des communications opportunistes, des systèmes pair-à-pair, et montrons comment ceux-ci pourraient être efficacement utilisés dans l'Internet des Objets (IoT).

1.1 Réseaux à un saut

Traditionnellement les objets mobiles utilisent des points d'accès (Hotspot WiFi), ou des réseaux celullaires (3G/4G) pour accéder à Internet. Il n'y pas de nœud intermédiaire entre l'objet et le point d'accès, les communication s'effectuent donc en un saut dans la partie sans fil du réseau. Les points d'accès disposent d'une interface sans fil et d'une interface filaire connectée à l'infrastructure. Les objets mobiles voulant accéder à Internet via un point d'accès doivent se trouver à portée de communication de celui-ci. Ces Hotspot WiFi peuvent être installés dans des habitations, des bureaux, des commerces, des lieux publics (salles de concert, musées, bibliothèques, gares, hôtels, aéroports, aires de covoiturage, etc.).

1.2 Réseaux hybrides multi-sauts

La figure 1.1 illustre un réseau hybride multi-sauts . Dans cette figure, des objets mobiles, représentés par des téléphones portables, sont connectés via leur interface sans fil (e.g. Wi-Fi). L'un d'eux est à portée de communication d'un point d'accès Wi-Fi. Ce dernier est connecté à Internet. Les objets mobiles agissent comme des relais et retransmettent les données qu'ils reçoivent de leurs voisins. Ces voisins peuvent ainsi accéder à Internet, et cela même s'ils ne sont pas directement connectés au point d'accès Wi-Fi.

Des projets, tels que RoofNet [4], Serval [5] et OpenGarden [1], ont montré qu'il est possible de fournir une connexion à Internet à haut débit à des usagers nomades grâce à ce type de réseaux hybrides. Dans ces projets, un nombre limité de points d'accès Wi-Fi,

1. `http://opengarden.com/`

se comportent en tant que routeurs, et forment un cœur de réseau ; certains de ces routeurs étant connectés à Internet. Ces projets utilisent des protocoles de routage dynamiques tels que OLSR [6] *(Optimized Link State Routing Protocol)*, AODV [7] *(Ad hoc On-Demand Distance Vector)*, ou BATMAN [8] *(Better Approach To Mobile ad hoc Networking)*, pour assurer la communication dans les parties mobiles du réseau.

Dans le cadre de l'IoT les réseaux hybrides multi-sauts peuvent avoir un intérêt car ils permettent d'augmenter la capillarité des réseaux, d'intégrer plus facilement des objets utilisant des interfaces de communication de courte portée et de permettre des communications directes entre les objets.

1.3 Réseaux pair-à-pair et opportunistes à connectivité intermittente

Les réseaux opportunistes à connectivité intermittente [9, 10] permettent de délivrer des messages même lorsqu'une connexion de bout en bout est impossible entre des objets du réseau. Pour ce faire, ils reposent sur le principe du *store, carry and forward*. Ce principe exploite les opportunités de contacts et la mobilité de certains objets. Il permet à des objets à portée de communication d'échanger et de stocker (*store*) des messages destinés à d'autres objets au gré de leurs contacts (*carry*) et de délivrer ces messages lorsqu'ils rencontrent les destinataires ou d'autres objets intermédiaires plus à même, de délivrer ou de se rapprocher des destinataires (*forward*).

Les projets de réseaux hybrides multi-sauts présentent cependant certaines limitations. En effet dans la pratique, la topologie de ce type de réseaux est rarement connexe et très fluctuante, et ce du fait de la faible portée des interfaces de communication sans fil et de la mobilité des objets ou de la mise en veille de certains objets. L'utilisation de méthodes de communications opportunistes permet de tolérer les ruptures de connectivité résultant d'une disparation temporaire ou définitive de certains équipements dans la zone de communication des objets.

En outre, dans certaines circonstances exceptionnelles, par exemple lors du tsunami au Japon en 2011 (voir figure 1.2a), lors des ouragans Katrina (voir figure 1.2b) et Sandy aux États-Unis en 2005 et 2012, tout ou une partie de l'infrastructure de communication peut être détruite. Ces catastrophes ont montré le besoin de communiquer dans ces situations d'urgence malgré la disparition d'une partie de l'infrastructure de communication. Les techniques de communications opportunistes pourraient aussi aider dans la partie infrastructure. Un objet dans l'infrastructure peut stocker et relayer des messages aux objets mobiles même si les autres objets de l'infrastructure ne sont plus joignables. RoofNet, Serval et OpenGarden n'offrent pas cette possibilité.

La figure 1.3 présente un exemple de réseau hybride à connectivité intermittente (RHCI). Ce réseau est composé d'objets mobiles ayant la capacité de communiquer entre eux ou avec un point d'accès lorsqu'ils se trouvent à portée radio. Les points d'accès, en plus de leur interface de communication Wi-Fi, sont connectés à Internet. Certains objets

(a) Séisme au Japon en 2011 (b) Ouragan Katrina aux États-Unis

FIGURE 1.2 – Exemple de catastrophes naturelles

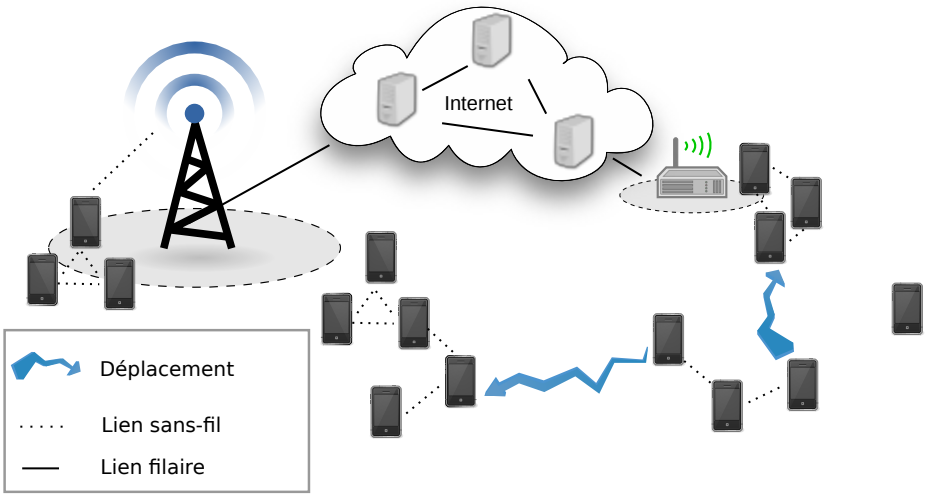
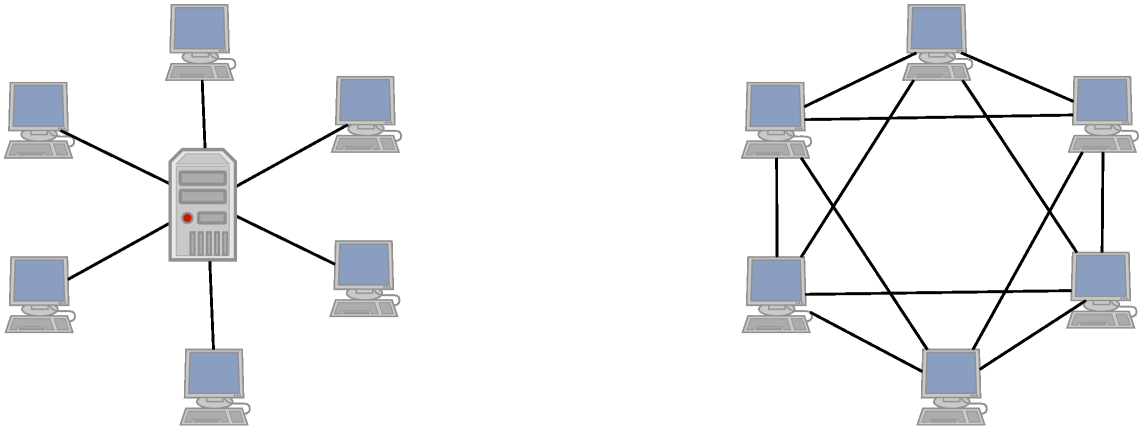


FIGURE 1.3 – Exemple d'un réseau hybride à connectivité intermittente



(a) Une architecture traditionnelle centralisée

(b) Une architecture pair-à-pair décentralisée

FIGURE 1.4 – Architectures pair-à-pair

mobiles disposent par ailleurs d'une connexion 3G/4G qui leur permet d'être connectés à Internet s'ils se trouvent à portée de communication d'une station de base.

La communication dans les réseaux opportunistes s'effectue directement d'équipement à équipement. Elle est par nature pair-à-pair (figure 1.4b). Ce modèle de communication est décentralisé et symétrique. N'importe quel pair peut initier une communication. Ce modèle est à opposer à l'approche traditionnelle client/serveur (figure 1.4a) dans laquelle ce sont les clients qui initient la communication. Les machines ont alors un rôle prédéfini : soit serveur soit client. Dans cette approche, toute la logique de calcul est centralisée sur le serveur ce qui rend le système inopérant en cas de défaillance de ce dernier. L'approche centralisée, bien que plus simple à mettre en œuvre présente des limites en terme de performance, de résistance à la censure, et de tolérance aux pannes. La bande passante disponible est limitée par celle du point central, ce qui peut impliquer une latence importante lorsqu'un grand nombre de terminaux se connectent à un serveur.

Les réseaux pair-à-pair [11, 12] ont émergé dans les réseaux d'infrastructure comme une alternative à l'approche client/serveur centralisée. Dans ces réseaux la charge de calcul, le stockage de données, ainsi que l'utilisation de la bande passante sont répartis entre les machines participantes afin d'éviter la surcharge de certaines d'entre-elles. La logique de calcul étant répartie entre tous les pairs, la défaillance de l'un d'entre eux ne compromet pas la totalité du système. D'autres pairs peuvent prendre le relais et effectuer les mêmes tâches que le pair défaillant.

Les applications réparties utilisant le modèle pair-à-pair sont devenues populaires grâce, notamment au succès des applications de partage de contenus sur Internet (e.g., Napster [13] ou Bittorrent [14]), de diffusion de flux audio et vidéos, et de téléphonie ou de visiophonie grâce au protocole WebRTC [15].

Les réseaux pair-à-pair restent un domaine très actif en recherche et reste plus compliqués à mettre en œuvre que l'approche client-serveur. Le choix de la topologie du système

pair-à-pair va influencer la capacité de passage à l'échelle du système, le temps et les techniques d'accès à l'information, ainsi que la résilience du système. De plus, l'utilisation des surcouches pair-à-pair dans les réseaux hybrides à connectivité intermittente reste un domaine encore très ouvert à la recherche.

1.4 Exemple de RHCI

Dans cette thèse, nous nous sommes intéressé à un exemple précis de réseaux à connectivité intermittente, à savoir ceux formés par des individus équipés de terminaux mobiles qui se déplacent au sein d'une ville de taille moyenne. Lors de déplacements, ils peuvent communiquer entre eux, accéder à des objets présents dans leur environnement, ou accéder à Internet.

Plus précisément dans cet exemple, des machines disposant de plusieurs interfaces de communication telles que des points d'accès Wi-Fi peuvent servir de relais entre les parties ad hoc du réseau et la partie infrastructure de celui ci. Les données peuvent être produites par des machines fixes, des objets connectés à Internet (e.g., capteurs) ou bien par des terminaux mobiles portés par des individus nomades. Le contenu créé peut ensuite être envoyé via des machines connectées à Internet, être déchargé vers la partie mobile du réseau ou être échangé directement entre les nœuds du réseau.

Les équipements composant ce réseau sont supposés être capables d'effectuer des communications de proximité directe en utilisant des technologies comme Wi-Fi en mode ad hoc, Wi-Fi P2P ou Bluetooth. Plusieurs schémas d'échange de données sont considérés dans cet exemple. L'échange point-à-point ; qui consiste à envoyer un message à un destinataire unique ; ce message pouvant être suivi éventuellement d'une réponse. Ce type d'échange peut être utilisé pour la mise en place d'un système de messagerie entre individus, accéder à un capteur ou servir de brique de base pour des scénarios plus complexes. Le schéma d'échange *Anycast* permet d'adresser un message à plusieurs destinataires simultanément susceptibles de répondre à cette requête. Dès lors, plusieurs destinataires sont susceptibles de répondre à ce message mais uniquement une seule réponse sera considérée. Ce mode de communication peut par exemple être utilisé pour interroger plusieurs objets connectés offrant un même service sans les connaître explicitement (e.g., capteur de température). Le dernier mode communication que nous considérons est une mode de communication basé sur le contenu qui permet d'exprimer le type de contenu plutôt qu'une destination. Ce mode de communication schéma d'échange peut être utilisé pour concevoir un système de partage de données.

1.5 Proposition de thèse

Les réseaux hybrides à connectivité intermittente, tels que ceux décrits précédemment, sont par nature dynamiques et hétérogènes : un grand nombre d'objets se connectent et se déconnectent du système en permanence. Les objets et points d'accès ont des ressources

et des capacités différentes en termes de puissance de calcul et d'autonomie. Certains équipements sont branchés au réseau électrique de façon permanente ou temporaire alors que d'autres fonctionnent sur batterie. La densité des objets au sein d'une zone peut varier géographiquement, et dans le temps dans les exemples de RHCI que nous considérons du fait de la mobilité de certains objets, et de l'activité menée dans ces zones (bureaux, zones commerciales, habitation, etc.).

Afin de fonctionner efficacement dans ce type de réseaux, un système servant de support pour la communication dans les RHCI devrait notamment

— ne pas avoir de point central afin de résister à la disparition d'objets ;
— abstraire les différences entre la partie infrastructure et les parties sans fil du réseau ;
— supporter la découverte des nœuds qui à la fois dans la partie infrastructure et dans les parties sans fil du réseau permet à tout nœud de s'envoyer des messages ;
— transférer des messages en tenant compte des fréquentes déconnexions dans les parties sans fil du réseau ;
— limiter le nombre de messages transférés afin d'économiser la batterie de certains nœuds, et éviter la congestion du réseau ;
— proposer différentes primitives de communication devraient être sous la forme d'un cadre de conception suffisamment générique et modulaire pour permettre le développement d'applications de différents types, par exemple des applications de communication entre individus ou des applications dédiées à l'IoT ;
— proposer des modes de communication tels que l'unicast, l'anycast ou un mode de communication basé sur le contenu. Ces modes de communication permettent par exemple d'interroger un capteur spécifique ou bien tous les capteurs d'un certain type.

Dans cette thèse, nous proposons un système pair-à-pair appelé Nephila pour l'échange de données fonctionnant dans les réseaux hybrides à connectivité intermittente. Ce système possède les propriétés décrites ci-dessus. Ce système a donné lieu à un prototype qui a été évalué en simulation.

1.6 Plan du document

Dans le chapitre 2 nous présentons l'état de l'art des communications opportunistes. Nous commençons par présenter les différents types de réseaux sans fil. Nous listons ensuite les différentes technologies de communication pouvant servir de support aux communications opportunistes.

Dans le chapitre 3, nous présentons l'état de l'art des réseaux pair-à-pair. Nous détaillons les différentes applications utilisant des réseaux pair-à-pair, les propriétés des réseaux pair-à-pair, les topologies pair-à-pair structurées et enfin les topologies pair-à-pair non structurées.

Dans le chapitre 4, nous synthétisons les points communs et les différences entre les réseaux opportunistes et les réseaux pair-à-pair. Nous montrons l'intérêt d'utiliser cer-

taines techniques des réseaux pair-à-pair dans les réseaux opportunistes.

Dans le chapitre 5, nous présentons notre approche reposant un overlay pair-à-pair pour l'informatique opportuniste et son implémentation dans une plateforme baptisé Nephila. Nous donnons une vue d'ensemble de la plateforme, et nous présentons les différents modules qui la compose.

Dans le chapitre 6, nous montrons comment Nephila peut être utilisé pour échanger des données. Nous détaillons les trois modes de communications utilisables dans Nephila, à savoir la communication en mode point-à-point, en mode anycast, et l'échange de données basé sur le contenu.

Dans les chapitres 7, 8 et 9 nous montrons les résultats de l'évaluation de Nephila en simulation. Nous présentons trois scénarios différents permettant d'évaluer chacun des modes de communication proposés.

Dans le chapitre 10, nous concluons et nous discutons des perspectives pour des travaux futurs.

Première partie

État de l’art

2
Communication opportuniste

Les communications dites opportunistes visent à exploiter les contacts réseaux entre des objets mobiles comme des opportunités d'acheminement de messages. Les objets, répartis dans l'environnement, forment un réseau qui souffre de fréquents partitionements. De ce fait, des connexions de bout en bout via plusieurs sauts entre des objets ne peuvent être garanties à tout instant, et de longues déconnexions entre les objets sont courantes.

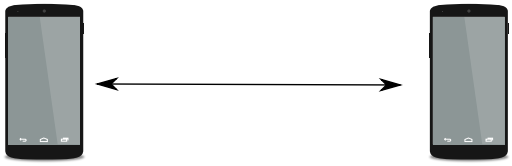
Dans la suite de ce chapitre, nous commençons par présenter les différents types de réseaux sans fil. Nous listons ensuite les différentes technologies de communication et protocoles de communication pouvant servir de support aux communications opportunistes.

2.1 Vers des réseaux opportunistes hybrides à connectivité intermittente

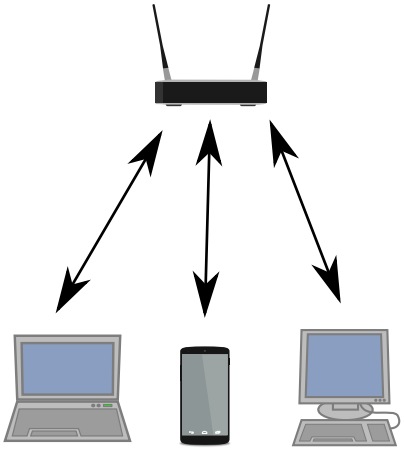
2.1.1 Réseaux sans fil à un saut

Les réseaux sans fil à un saut (voir figure 2.1) se caractérisent par une connexion directe entre des objets, mobiles ou non, et des éléments d'infrastructure (e.g., points d'accès Wi-Fi, stations de base).

Ces éléments d'infrastructure peuvent offrir un accès à Internet. Pour y accéder, un objet doit se trouver dans le rayon de communication de l'un de ces éléments. Les réseaux sans fil de ce type sont très largement répandus. Des points d'accès sans fils peuvent être installés dans des habitations (via la « *box* » du fournisseur d'accès à Internet), des bureaux, des commerces, des lieux à forte affluence (salles de concert, musées, bibliothèques, stades), des zones de transit (gares, aéroports, etc...). Les réseaux Wi-Fi offrent une bande passante élevée, mais une portée de communication inférieure à une centaine de mètres en pratique du fait des obstacles ou des interférences. Les réseaux dits « cellulaires »(3G/4G) offrent également un accès à Internet, avec des portées de communication nettement supérieures à celles des réseaux Wi-Fi. Leur coût de déploiement impose la commercialisation d'un service d'accès à Internet, qui est répercuté sous la forme d'un abonnement aux clients. L'installation et le choix du placement de stations de base restent



(a) Connexion directe entre deux terminaux mobiles



(b) Connexion entre terminaux mobiles et infrastructure

FIGURE 2.1 – Réseaux sans fil à un saut

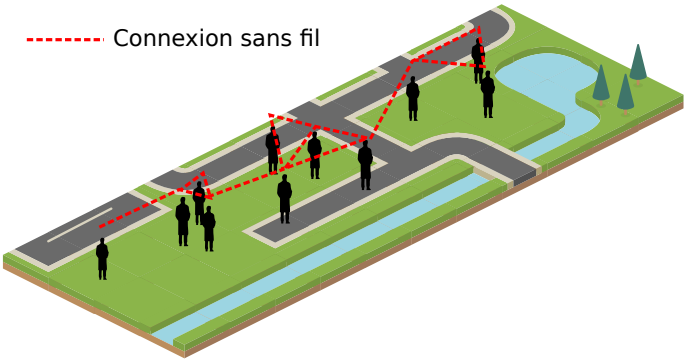


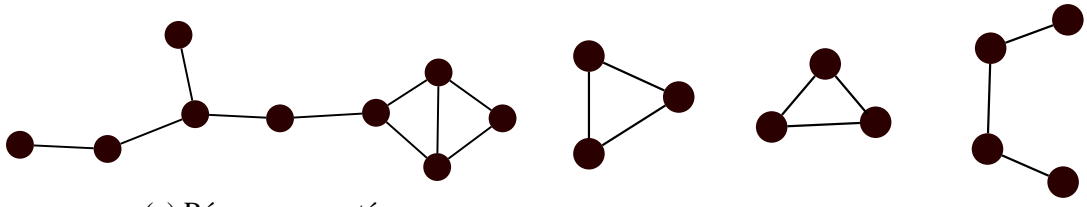
FIGURE 2.2 – Exemple de réseau mobile multi-sauts en mode ad hoc

donc conditionnés par l'attente d'un retour financier pour l'entreprise qui les exploite. Les zones peu denses se retrouvent donc peu ou pas couvertes par ces réseaux.

2.1.2 Réseaux mobiles multi-sauts

Les réseaux mobiles multi-sauts reposant sur des communications en mode ad hoc [16] (MANET : *Mobile Ad-hoc Networks*) sont des réseaux autonomes et auto-organisants composés d'objets mobiles utilisant des interfaces de communication sans fil pour échanger des données (voir figure 2.2). Dans ces réseaux, deux terminaux distants qui ne sont pas à portée de communication directe peuvent néanmoins communiquer via des terminaux intermédiaires qui jouent le rôle de relais. Plusieurs études [17, 18, 19] ont montré que l'utilisation du Wi-Fi en mode *Ad-Hoc* permettait de construire des réseaux multi-sauts viables.

Les domaines d'utilisation privilégiés des réseaux mobiles multi-sauts sont la communication dans des zones sans infrastructure, et l'extension de l'infrastructure par des



(a) Réseau connecté

(b) Réseau déconnecté

FIGURE 2.3 – Connectivité du réseau

moyens sans fils à faible coût. Serval [5], Open Garden [1], RoofNet [4] ont montré qu'il est possible de fournir une connexion à Internet à haut débit à des usagers nomades grâce à des communications multi-sauts, et grâce à un nombre limité de points d'accès Wi-Fi qui se comportent en tant que routeurs et qui forment un cœur de réseau.

Les MANETs peuvent être utilisés pour former des réseaux de terrain, par exemple des réseaux tactiques militaires [20, 21, 22]. Ces réseaux permettent aux soldats de se coordonner dans des environnements hostiles.

Le routage dans les réseaux MANETs s'effectue au moyen de protocoles de routage dynamique comme OLSR [6], AODV [7] ou ZRP [23]. OLSR est un protocole dit proactif. Les routes sont établies à l'avance grâce à un mécanisme de découverte de voisins. Chaque nœud échange périodiquement l'état de ses liens avec ses nœuds. Ainsi, une table de routage est construite dynamiquement, pour établir des routes avant même d'avoir à transférer un message. Par opposition un algorithme est dit réactif, si la décision de transfert est prise à l'émission d'un message. AODV entre dans cette catégorie. Les routes de communication sont établies *à la demande* en inondant le réseau de requêtes, jusqu'à ce que l'une d'entre elles atteigne la destination. Une réponse est alors envoyée à la source en utilisant le chemin inverse. Ce chemin sera utilisé pour transmettre les messages suivants.

2.1.3 Réseaux sans fil à connectivité intermittente

Dans les réseaux sans fil à connectivité intermittente, les nœuds du réseau sont le plus souvent mobiles, car ils peuvent être portés par des humains, par des animaux tels que dans les projets ZebraNet [24, 25, 26, 27] et SWIM [28], mais aussi être embarqués dans des véhicules comme dans le projet DieselNet [29]. Tout ou une partie des nœuds étant mobiles, une connexion de bout en bout entre tous les nœuds du réseau ne peut être garantie à tout instant. Le réseau n'est donc pas connexe (voir figure 2.3a), mais partitionné la plupart du temps (voir figure 2.3b). Pour faire face à ces ruptures de connexions qui peuvent être fréquentes et imprévisibles, des techniques reposant sur le principe du « *store, carry and forward* » [30, 31] sont utilisées. Ce principe consiste en effet à exploiter les opportunités de contact entre les objets fixes ou mobiles, leur capacité à stocker des messages, et leur mobilité afin de transporter ces messages entre les différentes parties du réseau [32, 33, 34, 35]. Deux objets peuvent donc communiquer même s'il n'existe pas de

1. `http://opengarden.com/`

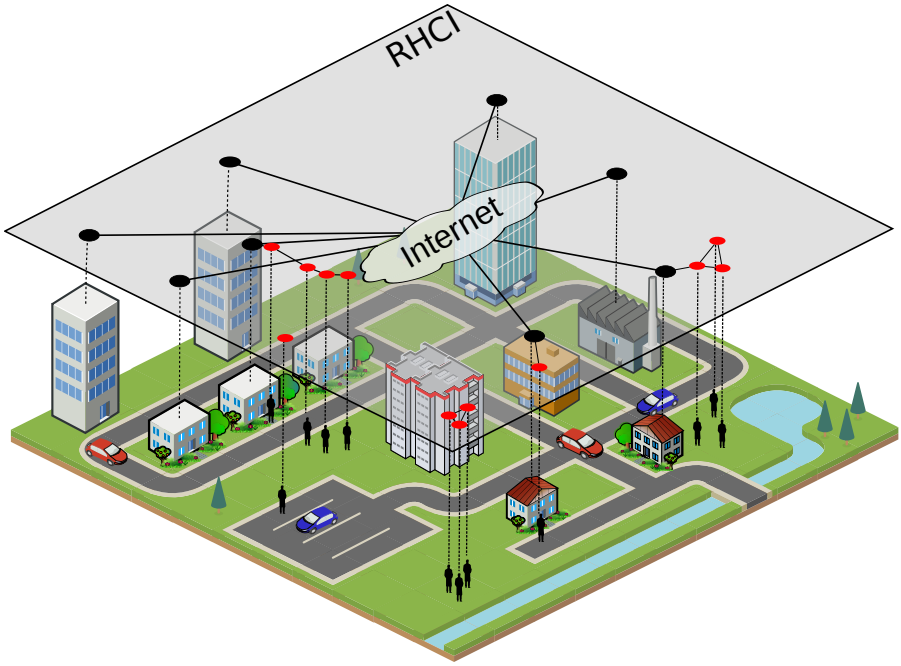


FIGURE 2.4 – Formation d'un RHCI au sein d'une ville

chemin de bout en bout entre eux. Des expériences récentes menées dans des conditions réelles ont montré que des applications asynchrones permettant l'échange de messages vocaux, de courriels, ou le partage de documents peuvent parfaitement fonctionner en utilisant ce mode de communication opportuniste [36, 37, 38, 39, 40].

2.1.4 Réseaux hybrides à connectivité intermittente (RHCI)

Une évolution intéressante des réseaux sans fils multi-sauts à connectivité intermittente est ce que nous appelons les réseaux hybrides à connectivité intermittente (RHCI). De façon analogue aux réseaux sans fils multi-sauts à connectivité intermittente, les RHCI intègrent des parties sans fil non connexes composées d'objets mobiles, mais également une partie infrastructure. Ces réseaux peuvent être complexes, couvrir une zone géographique importante (une ville par exemple), et être composés de nombreux équipements fixes et mobiles (voir figure 2.4). Les équipements mobiles peuvent être des terminaux mobiles utilisés par des personnes nomades, ou des systèmes embarqués dans des véhicules. Les équipements fixes peuvent être des objets connectés présents dans notre environnement quotidien, ou des routeurs qui peuvent être interconnectés pour former un cœur de réseau, offrir un accès à Internet, et établir des connexions avec les autres objets fixes et équipements mobiles du réseau. À l'instar des réseaux à connectivité intermittente, dans ces réseaux les communications doivent reposer en partie sur le principe général du *store, carry and forward* afin de supporter les ruptures de connectivité, et de faire face au partitionnement du réseau.

2.2 Technologies sans fil pour la communication opportuniste

Dans ce paragraphe, nous présentons des technologies sans fil permettant des communications directes de machine à machine nécessaires aux communications opportunistes.

Les caractéristiques étudiées sont la portée de communication, le temps de découverte, la bande passante et la consommation d'énergie.

Plus la portée de communication est importante, plus les objets mobiles pourront être en communication sur de longues périodes et avec un plus grand nombre d'objets, permettant ainsi d'acheminer plus de messages et plus rapidement. Une trop grande portée de communication peut toutefois entrainer une congestion de réseau lorsqu'un grand nombre d'objets sont à portée de communication et échangent des données.

Lorsque deux objets mobiles sont à portée de communication, ils doivent d'abord se découvrir mutuellement. Le temps de découverte réduit le temps disponible pour échanger des données. Lorsque des objets se déplaçant rapidement se croisent (e.g. objets embarqués dans des véhicules motorisés), il est important d'avoir un temps de découverte faible pour que les objets puissent avoir le temps d'échanger des données.

Le débit définit la quantité d'information pouvant être échangée entre objets. Cet aspect est particulièrement important lorsque l'on veut échanger de grandes quantités de données.

2.2.1 Bluetooth

Bluetooth [41] est une technologie de communication sans fil standardisée par l'IEEE sous la norme 802.15.1, fonctionnant dans la bande ISM 2.4-2.485 GHz. Cette technologie est principalement utilisée pour connecter des périphériques sans fil sur des ordinateurs personnels, des téléphones sans fil, ainsi que des consoles de jeux. La portée usuelle est d'environ 10 m (circuits intégrés de classe 2), mais peut atteindre jusqu'à 100 m pour les circuits intégrés de classe 1. À partir de la version 4.0 du Bluetooth, le débit peut atteindre 24 Mb/s.

Bluetooth est largement intégré dans les tablettes et les smarphones, et permet la communication directe de machine à machine, ce qui en fait un candidat intéressant pour les communications opportunistes. Cependant, la phase de découverte « bloque » l'émission et la réception de données pendant environ 7 secondes. Or pour ne pas manquer des opportunités de contact, le réseau doit être régulièrement sondé. On réservera donc l'utilisation du Bluetooth pour des communications opportunistes avec des porteurs peu mobiles.

2.2.2 Wi-Fi

Wi-Fi est le nom commercial de la norme IEEE 802.11x [42] et élaborée par la *Wi-Fi alliance* [2]. Selon les normes, les bandes utilisées sont ISM 900 Mhz, 2.4 Ghz ou 5.1 Ghz. Les objets mobiles (smarphones, tablettes, ordinateurs portables) embarquent tous un composant Wi-Fi. L'utilisation du Wi-Fi se fait très majoritairement dans le mode appelé *managed*, dans lequel des objets se connectent à un point d'accès.

2.2.2.1 Wi-Fi Ad Hoc

Wi-Fi en mode Ad Hoc [17], ou aussi appelé *Independent Basic Service Set* (IBSS), est un mode de communication qui contrairement au mode *managed*, ne nécessite pas de point d'accès et donc d'infrastructure pour communiquer entre équipements (nœuds).

Ce mode de communication semble donc plus pertinent que le mode traditionnel *managed* pour les communications opportunistes. Cependant son accès par les développeurs d'applications est très limité sur les objets mobiles tels que les smartphones. En effet, il est par exemple nécessaire sous Android d'avoir l'accès administrateur (root) sur l'objet. Quand bien même l'objet aurait la capacité de fonctionner en mode ad hoc, son support peut même être retiré du système d'exploitation sur certains objets tels que la tablette Nexus 7 (2013) conçue par Google. Cette tablette est pourtant la plateforme de référence pour les autres constructeurs de périphériques Android. Le mode de communication ad hoc sera donc de fait réservé à des scénarios où la flotte de terminaux utilisés est maîtrisée.

2.2.2.2 Wi-Fi Direct

Wi-Fi Direct [43] ou (ou Wi-Fi P2P) est défini par la *Wi-Fi Alliance* mais ne fait partie des spécifications IEEE. Cette norme permet à des équipements de communiquer sans être connectés à un point d'accès.

Les équipements négocient dynamiquement leur rôle à chaque connexion. Un équipement devient soit un *Group Owner* (GO), et se comporte dès lors comme un point d'accès, soit il fonctionne en tant que client.

Cette norme est supportée par tous les terminaux mobiles récents. De plus, en élisant dynamiquement un point d'accès, cette norme permet aux équipements clients de bénéficier du mode *Power Save*, c'est à dire la possibilité de se mettre en veille et donc d'économiser leur batterie. Ce mécanisme n'est pas disponible en mode ad hoc.

Les interfaces de programmation Wi-Fi Direct offrent aux développeurs d'application les méthodes pour découvrir les équipements ou des services à proximité, ainsi que la gestion de connexions sécurisées entre ces équipements.

Construire spontanément un réseau avec cette norme peut s'avérer difficile, dans la mesure où les GO ne peuvent être désignés explicitement ou connus à l'avance. Wi-Fi

2. http ://www.wi-fi.org/

Direct n'impose aucune règle sur ce point. Cette norme n'impose pas non plus qu'un équipement puisse être connecté avec plus d'un équipement Wi-Fi Direct à la fois. Les spécifications sont en constante évolutions et de prochaines versions des spécifications pourraient imposer le support d'un nombre minimum de connexions concurrentes.

2.2.2.3 Wi-Fi Aware

Wi-Fi Aware [3] est une spécification définie en 2015 par la *Wi-Fi alliance*. Cette spécification permet la découverte de services dans les réseaux Wi-Fi avant même que l'application ait à créer une connexion. Les domaines d'application énuméré par la *Wi-Fi Alliance* sont la messagerie instantanée, l'échange de photos, et de vidéos, ou les jeux multi-joueurs de proximité. Cette spécification a un fort potentiel pour l'établissement de connexions opportunistes. Cependant, comme pour Wi-Fi Direct, cette spécification n'est pas une spécification IEEE. Il n'existe pour le moment aucune implémentation mature permettant d'évaluer facilement son utilisation pour constituer un réseau opportuniste.

2.2.3 ZigBee

ZigBee [44] [4] est une implémentation de la spécification IEEE 802.15.4 introduite en 2004. La portée effective des équipements ZigBee varie entre 10 et 100 mètres avec un débit maximal de 250 Kbit/s. Cette spécification est parfaitement adaptée aux réseaux de capteurs et aux applications domotiques. ZigBee permet d'étendre la portée de communication en proposant la formation de réseaux maillés. ZigBee n'est pas disponible sur la majorité des smartphones ou des tablettes proposés au grand public et reste pour l'instant cantonné à des applications spécifiques utilisant des objets à faible consommation. Son faible débit ne permettra pas l'échange de données volumineuses telles que des photos ou des vidéos.

2.2.4 LTE direct

LTE direct [45], qui est standardisé dans 3GPP-R12 [5], est une technologie de découverte de services de proximité avec une portée maximale de 500 m. Bien que prometteuse, il n'existe pour l'instant aucune mise en œuvre concrète de cette technologie qui soit accessible publiquement. Cette technologie dépend des opérateurs de téléphonie mobile et son déploiement à grande échelle reste conditionné à l'existence d'un modèle économique pour satisfaire les intérêts financiers des opérateurs de télécommunication. Il n'est pas encore certain que cette technologie soit accessible pour tous les terminaux.

3. http ://www.wi-fi.org/discover-wi-fi/wi-fi-aware
4. http ://www.zigbee.org/
5. http ://www.3gpp.org/specifications/releases/68-release-12

2.2.5 LoRaWAN

LoRaWAN [6] est une technologie définie en 2015 pour former des réseaux permettant la communication *machine to machine* (M2M) à bas débit (0.3 kbps ; 50 kbps) avec une faible consommation énergétique. Cette technologie offre une portée pouvant atteindre plusieurs kilomètres et fonctionne en Europe sur les bandes ISM 433 Mhz ou 868 Mhz. Les équipements LoRa communiquent avec des équipements connectés à Internet grâce à des passerelles dédiées. Les communications sont donc centralisées autour des passerelles. Le système est de fait inopérant hors des zones de couverture des passerelles LoRa. Le passage à l'échelle de cette technologie de moyenne portée n'est pas encore prouvé, aucun déploiement impliquant des milliers d'équipements LoRa communicant dans une zone donnée n'ayant été réalisé à ce jour.

2.3 Protocoles de communication

Ces dernières années, plusieurs projets proposant des protocoles de communication pour les réseaux opportunistes ont été présentés tels que Scampi [46], Haggle [47], N4C [48], Sarah [49], C3PO [50] ou ASAWoO [51].

Une classification de ces protocoles peut être établie en fonction du comportement de leurs algorithmes de transfert de messages. Un état de l'art sur ces différents protocoles peut être trouvé dans [52]. Dans la suite de ce paragraphe, nous présentons les protocoles les plus connus.

2.3.1 Unique copie et délégation

Les algorithmes reposant sur la transmission d'une unique copie par délégation fonctionnent de manière similaire au routage de paquets IP sur Internet : pour chaque message créé, il existera une seule copie de ce message transitant sur le réseau. Ce mécanisme permet d'économiser la mémoire cache des nœuds et de limiter drastiquement l'utilisation de la bande passante. Les ressources des nœuds sont donc moins sollicitées, ce qui permet de faire des économies d'énergie.

Ces algorithmes sont efficaces quand les contacts entre nœuds sont prévisibles, comme c'est le cas dans la communication spatiale où la position des nœuds (satellites, sondes, etc.) peut être déterminée à l'avance avec précision, et ainsi de définir le meilleur chemin pour acheminer les données.

L'algorithme proposé par Spyropoulos et al. dans [53] permet uniquement une transmission directe des messages à leurs destinataires. Cet algorithme est très économe en énergie et en bande passante, mais présente de très faibles performances en taux de délivrance et une latence importante dès que le réseau devient peu dense. Une variante de cet algorithme est proposée par les auteurs [53]. Elle permet à un nœud de transférer

6. http ://lora-alliance.org/

des messages à l'un de ses voisins avec une certaine probabilité. Cet algorithme permet d'améliorer légèrement les performances, tout en augmentant aussi légèrement le trafic réseau.

Quand les contacts sont difficilement prévisibles, les performances se dégradent rapidement. C'est par exemple, le cas lorsque des piétons se déplacent dans une ville. En effet leur positions n'est pas forcément connues, et leur déplacements pas toujours réguliers. De plus, les algorithmes reposant sur la transmission de copie unique et la délégation ne sont pas prévus pour réagir efficacement face à des imprévus tels que l'arrêt de fonctionnement d'un terminal suite à l'épuisement de sa batterie.

2.3.2 Inondation

Epidemic Routing [54] est un algorithme de transfert de messages réactif inspiré du modèle mathématique de la propagation des maladies infectieuses. Tous les messages stockés dans le cache mémoire d'un nœud donné sont répliqués sur chacun des voisins. Cet algorithme ne requiert aucune heuristique complexe et ne fait aucune supposition quant à la mobilité des nœuds, ce qui le rend efficace quand la prédiction de déplacement de ces nœuds est difficile ou impossible. Cet algorithme présente les meilleures performances en termes de taux de délivrance et de latence sous réserve que le réseau ne soit pas congestionné et que les caches ne soient pas saturés de messages. En effet, dès que le nombre de nœuds ou de messages créés devient trop important, une consommation très importante de la bande passante est engendrée, pouvant aller jusqu'à une congestion de réseau et du cache des nœuds, ainsi qu'un épuisement plus rapide de la batterie. Cet algorithme est donc adapté aux réseaux de petite taille ayant un faible volume de messages créés.

RAPID [55] ordonne les messages selon une fonction d'utilité afin de maximiser une métrique spécifique (e.g, taux de délivrance, latence). PREP [56] qui est une variante de *Epidemic Routing*, assigne une priorité aux messages, qui est basée sur une estimation du coût de transmission et de la date d'expiration. PREP sélectionne sur la base de cette priorité les messages à transmettre ou à supprimer du cache en premier lorsque le cache des nœuds ou la bande passante sont limités.

Spray and Wait [57] est un algorithme utilisant la dissémination épidémique, mais avec une méthode de réduction drastique du nombre de messages. Lorsqu'un message est créé, un compteur est initialisé et embarqué dans les métadonnées du message. Ce compteur représente le nombre maximal de copies autorisées à circuler sur le réseau. Lorsque le message est transféré, le compteur est décrémenté. Une fois le compteur arrivé à 0 le message ne peut être transmis qu'à son destinataire.

Spray and Focus [58] utilise la même méthode de dissémination initiale que Spray and Wait, mais propose de remplacer la phase d'attente de Spray and Wait (i.e., lorsque le compteur de copie atteint 0). Le message peut être transféré en utilisant un mécanisme de copie unique ce qui permet de garder la même quantité de messages stockés dans le réseau. Une fonction d'utilité est utilisée pour déterminer si un message doit être transféré à un nœud voisin.

2.3.3 Historique des rencontres et analyse sociale

Les analyses sociales menées par Milgram [59, 60, 61] ont montré que la relation d'amitié entre les individus peut être modélisée par un graphe qui présente la propriété d'être quasiment connecté avec une topologie de type *petit monde* (*small world*). SimBet [62] utilise les relations sociales entre les individus pour favoriser les échanges entre les nœuds ayant le plus de similarités sociales.

Probabilistic Routing Protocol using History of Encounters and Transitivity (ProPHET) [63] est un algorithme de transfert de messages se basant sur l'historique des rencontres. L'hypothèse sur laquelle repose ProPHET est que si deux nœuds se sont rencontrés récemment, alors ils ont des chances de se rencontrer à nouveau dans le futur. Pour stocker ces informations de rencontre, chaque nœud dispose d'une matrice carrée ayant pour dimension le nombre de nœuds du réseau. Les valeurs de la matrice représentent les probabilités de rencontre. Elles sont augmentées à chaque rencontre entre deux nœuds et sont réduites au cours du temps. Ce système a pour effet de garder des valeurs plus élevées pour les rencontres récentes et des valeurs plus faibles lorsque des nœuds ne se sont pas rencontrés récemment. Ces valeurs sont par ailleurs calculées transitivement. Pour ce faire, les tables de valeurs sont échangées entre les nœuds afin que ces nœuds puissent déterminer si un de leurs voisins est un meilleur transporteur qu'eux même pour une destination donnée, et donc lui transférer une copie des messages pour lequel il sera considéré comme plus à même de transmettre le message.

MaxProp [64] ordonnance la transmission et la suppression des messages en fonction du nombre de sauts des messages transmis ainsi que des probabilités de rencontre basées sur les précédentes rencontres.

A History Based Routing Protocol for Opportunistic Network (HiBOp) [65] est un protocole utilisant le contexte de communication pour choisir les transporteurs de messages. Les informations de contexte peuvent par exemple contenir des informations personnelles sur l'utilisateur d'un terminal, telles que son nom, sa résidence personnelle ou professionnelle, ses loisirs (e.g. le sport, le cinéma). HiBOp peut ensuite décider en fonction de ces informations quels nœuds seront plus à même de délivrer un message. Par exemple HiBOp peut décider de transférer un message à un nœud dont la résidence du propriétaire se situe à proximité de celle du destinataire.

Context-Aware Routing (CAR) [66] exploite les informations de contexte de façon similaire à HiBOp, mais suppose un algorithme de routage pour réseaux mobiles multisauts comme OLSR ou AODV par composante connexe appelée *cloud*. Pour atteindre une partie non connectée du réseau, un nœud va choisir le nœud dans sa composante ayant la plus haute probabilité d'atteindre la destination. Les probabilités sont calculées de façon proactive et uniquement pour des destinations connues, puis sont disséminées dans le réseau.

2.3.4 Position géographique

La localisation d'un nœud peut être utilisée afin d'estimer des contacts futurs. Les techniques de localisation supposent le plus souvent la présence d'un système de localisation embarqué tel qu'un récepteur GPS. Bien que ces systèmes soient envisageables sur des objets disposant d'une autonomie énergétique suffisante tels que des véhicules, l'activation d'un tel dispositif sur des smartphones ou tablettes réduit fortement leur durée de fonctionnement.

GeoDTN+Nav [67, 68] utilise le système de navigation embarqué dans les véhicules. Les informations sur la destination ou la direction sont échangées entre véhicules. Ces informations permettent de déterminer les véhicules les plus à même de rencontrer la destination.

GeoSpray [69] utilise un système hybride basé sur le GPS et sur une dissémination d'un nombre limité de copies similaire à celle de Spray and Wait. GeoSpray dissémine initialement un nombre déterminé de copies, puis passe ensuite à un mécanisme de copie unique. Les copies ne sont transmises à un nœud que si celui est plus proche de la destination. Pour ce faire, GeoSpray, suppose que la position de la destination soit connue à l'avance.

Rao et al. [70] ne supposent pas que des nœuds connaissent sa position à tout instant (e.g, connexion GPS intermittente pour économiser la batterie). Les nœuds construisent un système de coordonnées virtuelles basé sur les contacts. Le routage des messages est ensuite effectué en fonction de ces coordonnées virtuelles. Les coordonnées sont construites par itérations successives. Périodiquement, chaque nœud i calcule ses coordonnées x_i and y_i en fonction de celle de ses voisins puis la diffuse afin que ses voisins puissent mettre à jour leurs coordonnées. L'équation de calcul est la suivante :

Cette solution ne nécessitant pas d'équipement de localisation supplémentaire et étant plus économe sur la consommation de la batterie semble intéressante, mais semble peu adaptée aux réseaux peu denses tels que ceux rencontrés à l'échelle d'une ville de taille moyenne.

2.4 Déchargement de données et réseaux hybrides

La croissance du marché des périphériques mobiles (téléphones portables, tablettes) a entraîné une hausse de l'utilisation des réseaux cellulaire (3G/4G). Le trafic de données pourrait atteindre 15.9 Exaoctets par mois en 2019 [71]. Ceci, sans compter le trafic engendré par les autres objets connectés. Selon ces projections, l'apport seul de la technologie 4G pourrait ne pas être suffisant pour absorber l'augmentation du trafic cellulaire. Des travaux proposent de décharger le réseau cellulaire en acheminant les données par d'autres moyens et notamment des communications sans fil de proximité [72].

La figure 2.5 présente les approches principales pour décharger le trafic des réseaux cellulaires vers les nœuds mobiles. La figure 2.5b montre comment les points d'accès

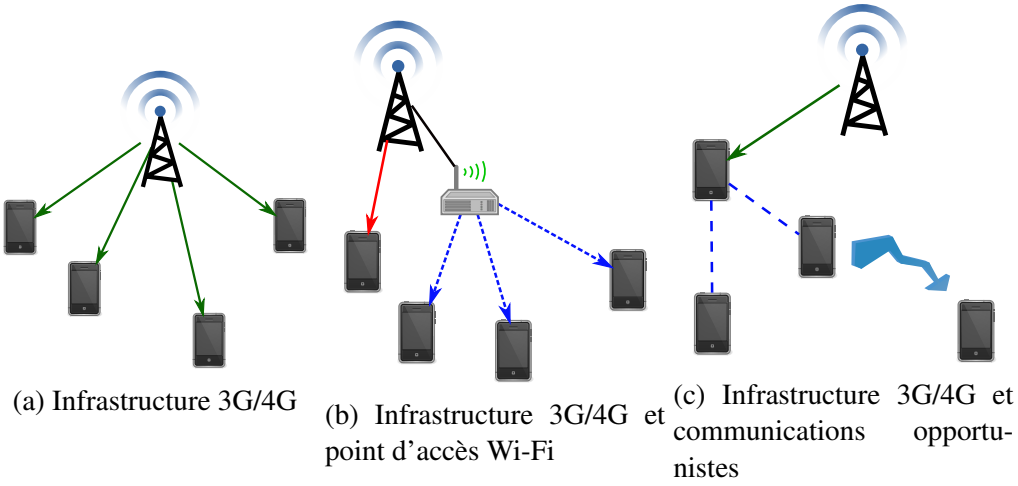


FIGURE 2.5 – Déchargement de données dans les parties sans fil du réseau

Wi-Fi peuvent être utilisés pour décharger une partie du trafic. La figure 2.5c montre l'utilisation des communications opportunistes entre nœuds mobiles. Dans la suite de ce paragraphe, nous considérons les approches utilisant les communications opportunistes pour décharger le trafic dont les applications peuvent tolérer des délais d'acheminement.

Push&Track [73] est une solution de déchargement de données utilisant un contrôleur de dissémination de données. Ce dernier injecte un certain nombre de copies en utilisant les interfaces de communication 3G/4G. Ces copies sont disséminées en utilisant des communications opportunistes Wi-Fi. Les nœuds ayant reçu ces copies notifient en retour le contrôleur en utilisant leur interface 3G/4G. Le contrôleur peut ainsi avoir une vue globale du nombre de copies présentes dans le réseau et réinjecter des copies si tous les destinataires n'ont pas reçu de copies.

RocNet [74] est une solution de déchargement de données des nœuds mobiles vers l'infrastructure. RocNet répartit la charge entre les différentes stations de base en utilisant des communications opportunistes. Dans les zones où les stations de base sont congestionnées, les communications opportunistes sont utilisées entre les nœuds mobiles afin d'acheminer les messages vers des stations de base qui ne sont pas congestionnées. Les « filtres à particules » [75] sont utilisés pour déterminer les nœuds les plus à même de se déplacer vers une station non congestionnée.

Mayer et al. [76] proposent un algorithme de routage utilisant l'infrastructure lorsque l'utilisation des communications opportunistes est supposée inefficace. Les messages sont initialement disséminés en utilisant les communications opportunistes. Lorsqu'il apparaît qu'un message met trop de temps ou bien à peu de chance d'atteindre son destinataire, l'algorithme de routage fait passer le message par l'infrastructure en utilisant les interfaces 3G/4G des nœuds.

TOMP [77] est un cadriciel de déchargement de données se basant sur la prédiction de mouvements entre les nœuds mobiles pour sélectionner les nœuds les plus à même de faire des rencontres. Ces nœuds serviront à disséminer les données en provenance

de l'infrastructure. Plusieurs métriques sont proposées pour prédire les opportunités de contact. La première se base sur la position actuelle des nœuds. La seconde considère les futures opportunités de rencontre à l'intérieur d'un périmètre défini. Enfin la troisième contraint le périmètre de la seconde métrique par un graphe de la cartographie routière.

2.5 Conclusion

Aucun de ces travaux ne propose d'abstraction permettant de gérer de façon uniforme les parties infrastructure et opportunistes d'un RHCI. La solution proposée devra être capable d'utiliser plusieurs technologies de communication sans fil. Il sera nécessaire de fournir dans le cadre de l'IoT un système permettant d'abstraire toutes ces technologies, de supporter les ruptures de communications, d'acheminer des données dans un RHCI et de supporter les communications de machine à machine. Les surchouches des réseaux pair-à-pair fournissent des abstractions et des mécanismes pour des machines dans des environnements hétérogènes et dynamiques. Ils seront par conséquent étudiés dans le chapitre suivant.

3

Principes et mécanismes des réseaux pair-à-pair

Les réseaux pair-à-pair sont des réseaux dans lesquels les machines sont appelées *pairs*. Ces machines jouent à la fois le rôle de serveur et de client, mais rarement les deux simultanément. Par opposition, les réseaux traditionnels sont composés de machines jouant le rôle de client ou de serveur. Les réseaux pair-à-pair sont par nature des réseaux coopératifs. Les machines partagent et mettent en commun leurs ressources telles que leur puissance de calcul, leur espace de stockage, leur bande passante, afin de rendre un service.

Dans un réseau pair-à-pair, les pairs sont interconnectés via un réseau de recouvrement logique (*overlay*). Ce réseau de recouvrement établit des liens entre les pairs. Les réseaux pair-à-pair peuvent être différenciés selon leur topologie. Cette topologie définit les méthodes utilisées pour répartir et rechercher des données, et pour tolérer les pannes. Il existe deux grandes familles de topologie : les topologies structurées et les topologies non structurées.

Dans la suite de ce chapitre, nous détaillons les différentes applications utilisant des réseaux pair-à-pair, les propriétés des réseaux pair-à-pair, les topologies pair-à-pair structurées et enfin les topologies pair-à-pair non structurées.

3.1 Applications pair-à-pair

3.1.1 Partage de fichiers et diffusion de données

Les réseaux de délivrance de contenu (CDN) ont pour vocation d'acheminer le plus rapidement possible des données vers l'utilisateur. Afin de répartir la charge et minimiser la latence, le contenu est répliqué en plusieurs points du réseau. Les mécanismes d'orchestration de la réplication peuvent être intégrés dans une architecture pair-à-pair. Le coût de réplication du contenu peut alors être réparti entre les nœuds du réseau.

3.1.2 Calcul réparti

La mise en commun des ressources, notamment la puissance de calcul, permet de construire des applications de calcul distribué exploitant en parallèle plusieurs processeurs afin de traiter de grands volumes de données.

Il peut s'agir parfois de machines grand public, comme c'est le cas dans le projet SETI@home [78]. Ce projet est un projet de recherche scientifique qui vise à détecter des formes de vie extraterrestre intelligente en analysant les signaux radio en provenance de l'espace. Le grand public est invité à participer en installant un logiciel exploitant les cycles processeurs inutilisés sur leurs ordinateurs personnels.

Pour le monde de la recherche et les entreprises, une organisation des nœuds sous la forme d'un réseau pair-à-pair dans un centre de calcul est plus économique que l'achat de superordinateurs pour traiter des volumes massifs de données.

3.2 Propriétés des réseaux pair-à-pair

3.2.1 Passage à l'échelle

Les systèmes pair-à-pair sont complètement répartis et peuvent atteindre des tailles jusqu'à plusieurs centaines de milliers de nœuds. À cette échelle, le système doit toujours présenter de bonnes performances. De plus, l'ajout de nœuds dans le système est vu comme une opportunité et non une contrainte. Les nouveaux nœuds apportent de nouvelles ressources dans le système, par exemple sous la forme de puissance de calcul ou encore de capacité de stockage. L'approche centralisée consistant à avoir un unique *maître* coordonnant les autres machines n'est plus viable dans les systèmes présentant un nombre important de nœuds.

3.2.2 Recherche d'information

L'utilisabilité des systèmes pair-à-pair dépend souvent de leur capacité à localiser et à récupérer efficacement des données. Ces systèmes étant capables de traiter de très grands volumes de données, une indexation par un unique serveur n'est souvent pas possible ni souhaitable. Les systèmes pair-à-pair ont recours à différentes stratégies pour rechercher du contenu en fonction de la topologie de leur vue logique, telle que la marche aléatoire, le parcours en largeur dans les réseaux non structurés, ou l'utilisation de tables de hachage réparties (DHT) dans les réseaux structurés.

3.2.3 Capacité de survie et tolérance aux pannes

La capacité de survie d'un système pair-à-pair est sa capacité à fonctionner en présence de fautes telles que la défaillance d'une machine, l'entrée et la sortie des nœuds

dans le système (*churn*), ou bien d'erreurs sur les liens réseaux. Ces erreurs peuvent survenir pour plusieurs raisons : malveillance (attaques informatiques, virus, déni de service), congestion réseau, épuisement de la batterie d'un terminal mobile.

3.3 Architecture pair-à-pair structurée

Les réseaux pair-à-pair utilisant une architecture structurée construisent une vue logique (*overlay*) dans laquelle les messages suivent un chemin déterministe. La délivrance des messages est assurée du succès en un petit nombre de sauts. Le réseau répartit des informations entre les pairs. Chaque pair indexe une partie des informations partagées dans le réseau. Dans le cas de la défaillance d'un nœud, l'information qu'il hébergeait est répartie entre les autres nœuds.

Une information est identifiée au moyen d'une *clé*. La clé est un bloc à taille fixe supposé unique et le plus souvent généré au moyen d'une fonction de hachage, comme c'est le cas dans Chord [79]. Celui-ci utilise une clé de 160 bits obtenu en appliquant la fonction SHA-1 [80] sur le contenu de l'information.

Les pairs sont aussi identifiés de façon unique dans le réseau. L'identifiant du pair peut être obtenu à partir d'un hash de son adresse IP [81], ou bien de ses coordonnées géographiques. L'identifiant d'un pair va déterminer l'espace des clés sous sa responsabilité. Par exemple un pair peut avoir la responsabilité des clés les plus *proches* de son identifiant.

La recherche et la publication d'information dans le réseau s'effectuent en utilisant une table de hachage répartie (*Distributed Hash Table*, DHT). Deux opérations sont disponibles pour manipuler les informations. *Put(key,val)* permet d'envoyer une information (*val*) avec sa clé (*key*). La récupération d'information s'effectue au moyen de l'opération *Get(key)* qui interroge le pair responsable de l'information et retourne la valeur associée à cette clé.

3.3.1 Chord

Dans le cas de Chord, les pairs sont organisés sous la forme d'un anneau en fonction de leur identifiant. Chaque pair maintient une table appelée *finger table* contenant une liste de pairs. La i^{eme} entrée du pair p contient l'identifiant le proche de $p+2^{i-1}mod(m)$, où m est la taille de la *finger table*. La figure 3.1 présente un exemple de réseau Chord comportant 16 pairs, dans lequel les relations de voisinage des pairs 2 et 11 sont représentées.

La recherche du pair responsable d'une clé particulière s'effectue en envoyant une requête à son voisin ayant l'identifiant le plus proche de la clé, qui à son tour va transférer la requête à son voisin le plus proche de la clé, et ainsi de suite jusqu'au pair responsable de la clé, lequel va répondre directement à l'émetteur initial de la requête.

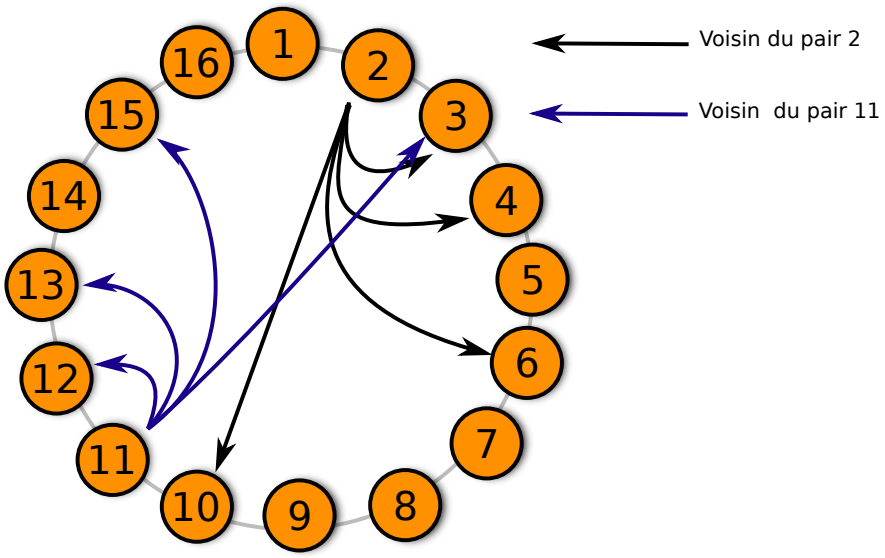


FIGURE 3.1 – Exemple de réseau Chord avec 16 nœuds

3.3.2 Pastry

Pastry [82] est un réseau pair-à-pair structuré similaire à Chord, représentant les pairs sur un anneau. Chaque nœud a un identifiant unique représenté sur 128 bits. Les identifiants sont choisis de façon uniforme, en appliquant par exemple une fonction de hachage sur l'adresse IP du nœud. Pastry se base sur le préfixe des adresses pour acheminer des messages. Chaque nœud maintient une table de routage, ainsi qu'une structure de données appelée *leaf set*. Cette structure de données contient les identifiants de nœuds les plus grands et les plus proches de l'identifiant du nœud local. Chaque entrée dans la table de routage de la ligne n partage les premiers n digits avec le nœud local. Le routage d'un message s'effectue en regardant si un identifiant de nœud dans le *leaf set* est plus proche de la clé du message que le nœud local. Si tel est le cas, le message est transféré à ce nœud, sinon le message est transféré au nœud partageant un préfixe commun avec la clé du message. Le processus est répété jusqu'à atteindre le nœud responsable de la clé.

3.3.3 Kademlia

Kademlia introduit la notion de distance entre nœuds, et essaie de réduire avec des recherches successives la distance vers une destination. La distance entre deux nœuds est calculée en utilisant la fonction XOR sur l'identifiant des nœuds. Cet identifiant est codé sur un espace de 160 bits. L'utilisation de la fonction XOR va avoir pour effet de privilégier les nœuds dont les bits de poids forts dans les identifiants sont différents de ceux du nœud local, et ainsi ordonner les nœuds sous la forme d'un arbre binaire.

Chacun des nœuds maintient une liste des autres nœuds appartenant au système appelé *k-bucket*. Cette liste contient k listes d'identifiants de nœuds (k étant un paramètre du système). Les nœuds dans la liste d'indice n sont ceux dont le bit d'indice n diffère du

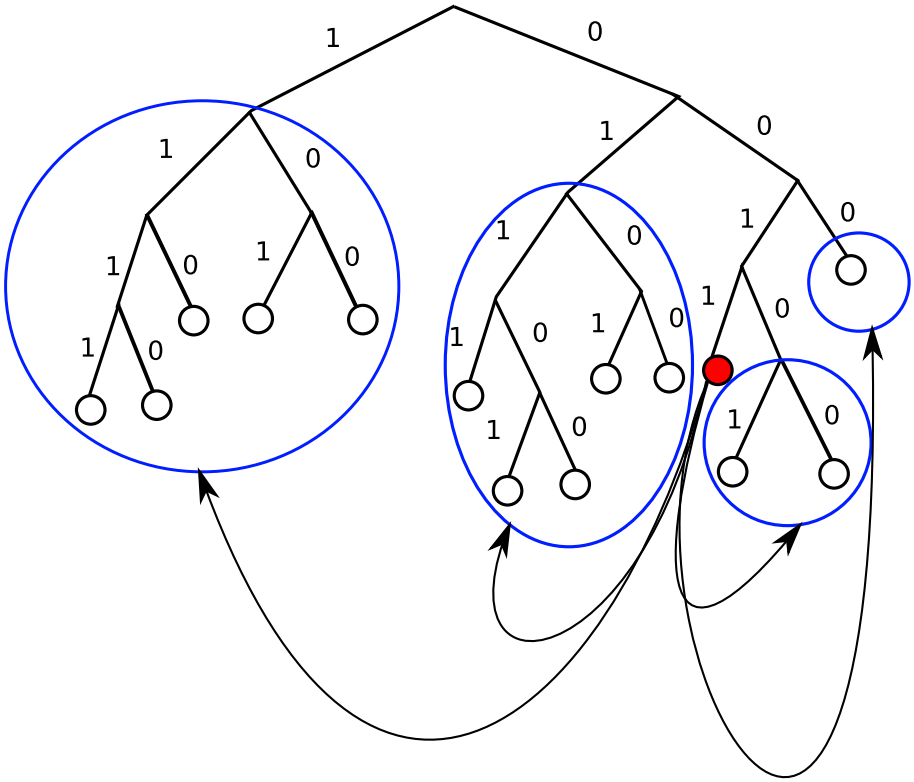


FIGURE 3.2 – Exemple d'arbre binaire Kademlia

nœud local, tandis que les $n-1$ bits restants sont communs avec le nœud local.

La figure 3.2 présente un exemple de réseau Kademlia représenté sous la forme d'un arbre binaire. Chaque feuille représente un nœud dans le système. Le nœud 0011 (représenté en rouge) dispose d'une liste de 4 éléments. Chacune des flèches pointe vers un sous-arbre dans lequel un des nœuds devrait être présent dans la liste des nœuds du nœud 0011.

3.3.4 CAN

CAN [83] utilise un espace à d dimensions pour identifier les pairs et les clés. Chaque pair est responsable d'une partition de cet espace. Le routage d'une requête s'effectue progressivement en parcourant la partition voisine la plus proche de la destination. Si un pair est défaillant, il peut être contourné en sélectionnant un autre pair proche. L'insertion d'un nœud s'effectue en sélectionnant un point dans l'espace, et en découvrant le pair responsable de cet espace puis en divisant cet espace en deux parties et enfin, en affectant au nouveau pair la responsabilité d'une des divisions. Finalement, les pairs voisins de cet espace sont notifiés de ces changements d'espace. La figure 3.3 présente un exemple de routage à partir d'un pair A vers une ressource r dont les coordonnées dans l'espace CAN (11, 7) sont sous la responsabilité du pair B.

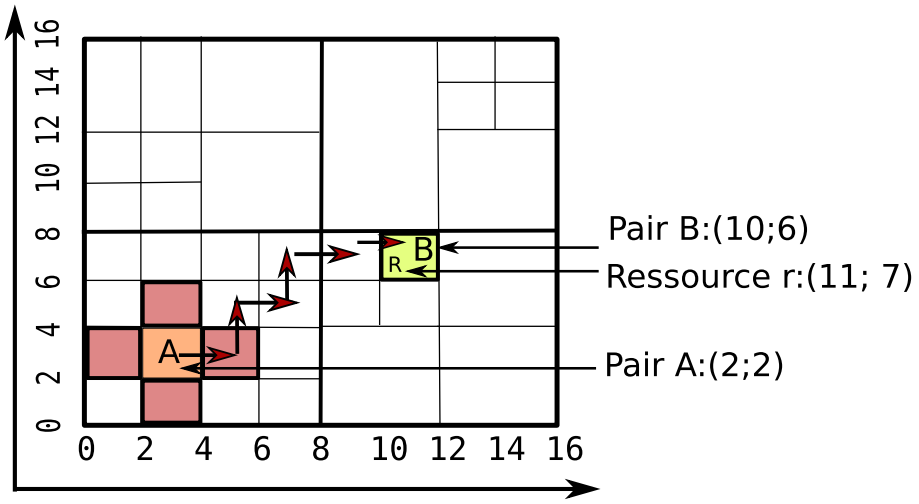


FIGURE 3.3 – Exemple de routage dans un réseau CAN à 2 dimensions

3.3.5 T-MAN

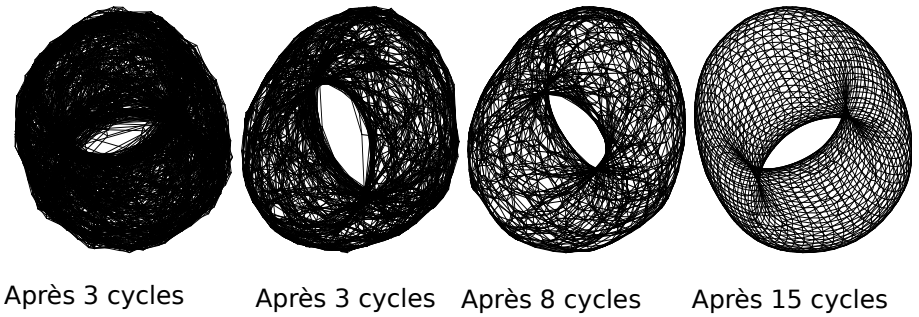


FIGURE 3.4 – Organisation des pairs sous la forme d'un tore dans un réseau T-MAN

Source: [84]

T-MAN [84] est un protocole permettant de construire des réseaux pair-à-pair ayant des structures convergentes. Les pairs peuvent être organisés sous différentes formes (e.g. ligne, anneau, tube, tore, arbre). La topologie dépend d'une fonction de classement fournit par l'utilisateur. Un mécanisme d'échange de liste de pairs périodique, organisé sous la forme de cycles permet à chaque pair de mettre à jour sa vue du réseau. La figure 3.4 présente une vue logique d'un réseau T-MAN dont les pairs sont finalement organisés sous la forme d'un tore. Plusieurs cycles sont nécessaires à la convergence de la structure.

3.4 Architecture pair-à-pair non structurée

3.4.1 Principe du *gossiping*

Les réseaux pair-à-pair non structurés n'imposent aucune structure *a priori*. Les connexions entre les pairs sont établies selon un processus stochastique. Les informations dans ce type de réseaux sont souvent diffusées en utilisant la méthode de communication par propagation de rumeur [85] (*gossiping*) aussi appelée propagation épidémique. Cette méthode permet de propager rapidement une information avec une charge de communication répartie entre les pairs [86].

L'algorithme général décrit dans [87] est présenté dans l'algorithme 1.

Algorithm 1 squelette de l'algorithme gossip

1: **loop**
2: wait()
3: p ← selectPeer()
4: send $state$ to p
5: receive $state_p$ from p
6: $state \leftarrow$ update($state_p$)
7: **end loop**

8: **procedure** ONRECEIVE(m)
9: $p \leftarrow m.sender$
10: $state_p \leftarrow m.state$
11: send $state$ to p
12: $state \leftarrow$update($state_p$)
13: **end procedure**

Chaque pair va périodiquement sélectionner un autre pair parmi ceux qu'il connaît déjà (ligne 2), envoyer *l'état* de l'application (ligne 3) attendre une éventuelle réponse (ligne 4) et mettre à jour son état en utilisant la fonction *update* fournie par l'application, et qui sera appliquée sur l'état reçu du pair. Lorsqu'un pair reçoit l'état d'un autre pair (procédure *onReceive()*), il envoie son propre état et met à jour le sien avec la même fonction *update*.

3.4.2 Cyclon

Cyclon [88] est un service d'échantillonnage de pair (*peer sampling service*). Les pairs s'échangent de façon aléatoire un sous-ensemble de leurs connaissances, appelé voisinage. Le processus est appliqué de façon proactive et périodique, permettant de garder en mémoire une liste de voisins non défaillants la plus à jour possible. Le graphe de connectivité résultant du processus d'échange de liste de pairs tend vers un graphe aléatoire à faible diamètre.

3.4.3 Napster

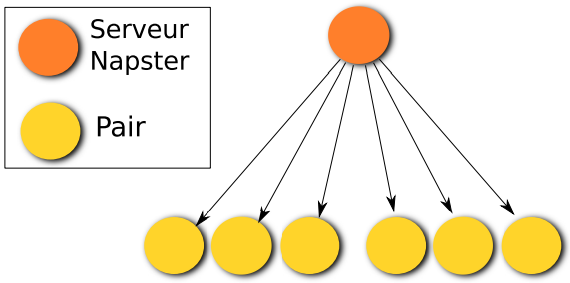


FIGURE 3.5 – Architecture centralisée de Napster

Napster [13] fut le premier système pair-à-pair de partage de fichiers. Dans Napster, un serveur central sert de point de rencontre et d'annuaire pour tous les nœuds du système. Lorsqu'un nouveau nœud désire entrer dans le système, il contacte le serveur central Napster et envoie alors une liste des fichiers qu'il possède et qu'il souhaite partager. La recherche de contenu s'effectue en interrogeant le serveur central, qui retourne une liste de nœuds partageant des fichiers qui correspondent à la requête. Une fois la liste de nœuds récupérée, le partage s'effectue en pair-à-pair entre les nœuds concernés.

Ce système est très simple à mettre en œuvre, mais présente un point de défaillance au niveau de l'annuaire central. De plus, la recherche étant centralisée, elle va présenter des problèmes de passage à l'échelle en termes de nombre de requêtes, de nombre de fichiers, et de pairs présents dans le réseau.

3.4.4 Fastrack

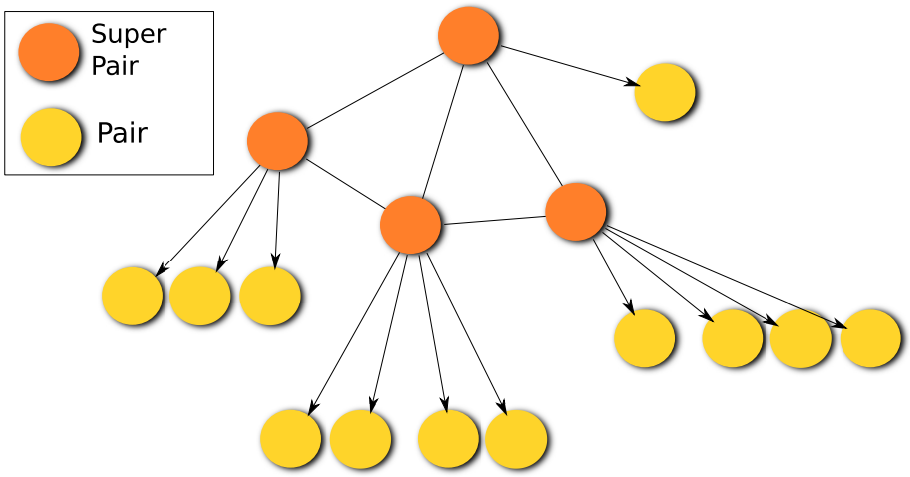


FIGURE 3.6 – Architecture de Fastrack

Fastrack est un protocole de partage de fichiers partiellement centralisé. Le système tire parti de l'hétérogénéité des nœuds pour accélérer la recherche d'information. Les

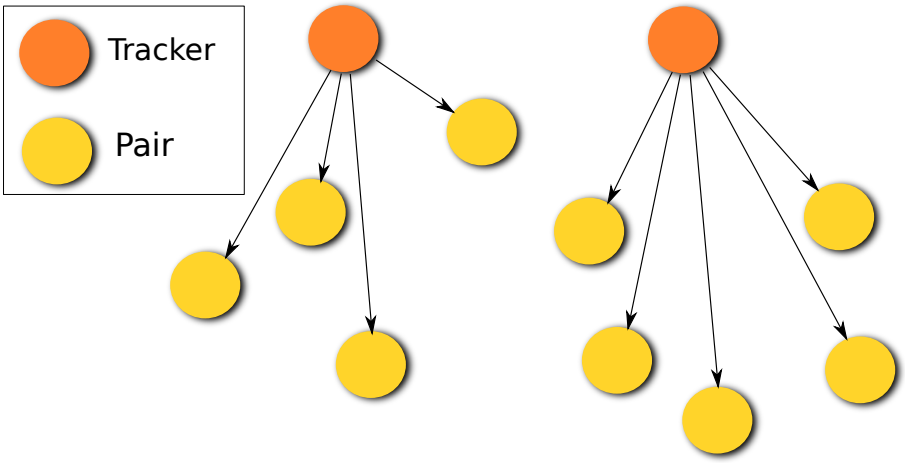


FIGURE 3.7 – Architecture de Bittorent

nœuds ayant des capacités supérieures aux autres en termes de capacité de stockage, de calcul et de bande passante sont désignés en tant que *super pairs*. Ces super pairs indexent les fichiers disponibles sur les nœuds et maintiennent des connexions vers d'autres super pairs. Un nœud cherchant un ficher va interroger les super pairs séquentiellement, jusqu'à trouver l'information recherchée.

3.4.5 Gnutella

Gnutella [89] est un système de partage de fichiers pair-à-pair utilisant initialement une architecture non structurée. Les requêtes de recherche de contenu dans le réseau sont diffusées de façon épidémique. Cette méthode, bien que très résistante aux pannes de machines, ne convient pas pour des réseaux accueillant un très grand nombre de machines. Si l'information recherchée n'est présente que sur un petit nombre de pairs, la requête peut devoir traverser un grand nombre de nœuds avant de rencontrer un pair possédant l'information recherchée. Un grand nombre de sauts implique une plus grande latence. De plus, la méthode de diffusion épidémique génère un grand nombre de messages dans le réseau, ce qui peut engendrer une congestion lorsqu'un grand nombre de requêtes traverse le réseau.

3.4.6 BitTorrent

BitTorrent est un service de partage de fichiers disposant d'un serveur central (*tracker*) pour coordonner le partage pour un fichier. Pour chaque fichier partagé, un fichier associé *.torrent* doit être créé, contenant notamment l'adresse du *tracker* responsable de la gestion des pairs partageant ce fichier. La diffusion et la récupération des fichiers *.torrent* ne font pas parties du protocole BitTorrent. Ces fichiers *.torrent* sont généralement diffusés sur des sites Web dédiés.

3.4.7 Utilisation des filtres de Bloom dans les réseaux pair-à-pair non structurés

Les filtres de Bloom sont des structures de données probabilistes, inventées par Burton Howard Bloom [90], permettant de tester la présence ou non d'un élément dans un ensemble. L'absence d'un élément peut être déterminée de façon certaine, alors que la présence est soumise à des faux positifs. Les filtres de Bloom permettent d'obtenir un compromis entre la fiabilité des informations retournées et la taille de la représentation en mémoire de l'ensemble.

Un filtre de Bloom est constitué d'un tableau de booléen T de taille m, et de k fonction(s) de hachage h_i tel que $1 \leq i \leq k$ et $0 \leq h_i(e) \leq m-1$ pour tout élément e. Les éléments du tableau sont initialisés à *Faux*. Un élément e est considéré présent dans le filtre de Bloom si pour toute fonction de hachage on a $T[h_i(e)] = Vrai$. L'insertion d'un élément e, met à *Vrai* les éléments du tableau dont les indices correspondent a l'application e sur chacune des fonctions de hachage : $\forall i, T[h_i(e)] := Vrai$. Il n'est pas possible de supprimer un élément dans un filtre de Bloom traditionnel puisqu'un indice donné peut correspondre à des éléments différents.

La probabilité de faux positif dans un filtre de Bloom est :

$$\left(1 - \left[1 - \frac{1}{m}\right]^{kn}\right)^{k}. \tag{3.1}$$

Le nombre de fonctions de hachage k qui minimise la probabilité de faux positifs est

$$k = \frac{m}{n} ln\ 2. \tag{3.2}$$

Les Filtres de Bloom ont été traditionnellement utilisés pour tester la présence d'un mot dans un dictionnaire, tout en gardant une représentation en mémoire de taille limitée. En effet stocker tous les éléments en mémoire peut être coûteux, et les filtres de Bloom sont un bon compromis entre la taille de la représentation mémoire et la probabilité de faux positifs.
Le filtre de Bloom est construit en insérant tous les éléments du dictionnaire un par un. Les Filtres de Bloom ont été utilisés pour implémenter des correcteurs orthographiques [91, 92] consommant peu de mémoire et ayant une faible probabilité de manquer des mots mal orthographiés.

Les filtres de Bloom à compteur [93] permettent de résoudre les problèmes de suppression d'éléments dans le filtre. Chaque entrée est stockée sur un entier positif plutôt que sur un bit. À l'insertion d'un élément, le compteur est incrémenté de 1, et lorsqu'un élément est supprimé celui-ci est décrémenté.

Les filtres de Bloom à décroissance exponentielle (EDBF) [94] ont été créés afin de traiter le problème de performance des recherches dans les réseaux pair-à-pair non structurés. Les recherches par inondation peuvent rapidement saturer le réseau et les marches

aléatoires peuvent nécessiter de traverser un grand nombre de nœuds. La quantité d'information stockée dans EDBF décroit exponentiellement avec la distance. En utilisant un algorithme de routage dans un réseau pair-à-pair non structuré tirant parti de cette structure, les auteurs ont montré qu'il était possible d'atteindre de meilleures performances que la marche aléatoire, et un surcoût réseau inférieur aux algorithmes de recherche par inondation.

3.5 Conclusion

Les réseaux pair-à-pair permettent de supporter des échanges de données ainsi que de répartir du calcul entre un grand nombre de nœuds qui peuvent apparaître et disparaître à n'importe quel moment du réseau. Ils sont très tolérants à ces déconnexions. Il existe deux familles de réseaux pair-à-pair : les réseaux pair-à-pair structurés et les réseaux pair-à-pair non structurés. Les réseaux pair-à-pair structurés organisent les nœuds sous la forme de structure tels que des anneaux, des tores ou des arbres. Ils sont généralement très performants dans la recherche d'informations, mais ont des coûts de maintenance plus importants. Les réseaux pair-à-pair non structurés n'imposent pas de structure et sont donc plus simple à mettre œuvre. Dans ce type de réseau, la diffusion d'information repose généralement sur le principe du *gossiping*.

4

Vers l'utilisation des mécanismes pair-à-pair dans les réseaux opportunistes hybrides

Dans ce chapitre, nous discutons des points communs et des différences existant entre les réseaux pair-à-pair et les réseaux opportunistes afin d'identifier les mécanismes et les techniques pair-à-pair qui sont susceptibles d'être utilisés dans les réseaux opportunistes hybrides à connectivité intermittente. En effet, les réseaux pair-à-pair ont été étudiés et déployés plus largement que les réseaux opportunistes. Certains mécanismes des réseaux pair-à-pair pourraient donc être utilisés dans les réseaux opportunistes pour améliorer la réplication des données dans les caches des nœuds fixes et mobiles, la recherche de données dans ces réseaux, et l'acheminement des messages entre les différents pairs.

4.1 Réseaux pair-à-pair et opportuniste : points communs et différences

Déconnexions Les connexions et déconnexions résultant de la mobilité ou de la mise hors tension des objets dans les réseaux opportunistes s'apparentent au *churn* dans les réseaux pair-à-pair. Les nœuds se connectent et se déconnectent du réseau, sans forcément avertir les autres membres du réseau à l'avance. Les réseaux pair-à-pair réagissent à la disparition de nœuds en reconfigurant leur réseau logique pair-à-pair.

Hétérogénéité des nœuds Dans les réseaux opportunistes, comme dans les réseaux pair-à-pair, les nœuds composant le réseau sont hétérogènes en termes de capacités de calcul, de capacité de stockage, de communication. Il convient de répartir la charge entre les nœuds du réseau afin de limiter le nombre de nœuds surchargés.

Connaissance limitée du réseau Dans les réseaux pair-à-pair reposant sur une infrastructure, la connaissance de chacun des nœuds est limitée par le système, afin d'éviter la gestion d'un grand nombre de connexions. Les réseaux opportunistes ont par nature une

vue limitée du système. Les nœuds ne peuvent être connectés entre eux que s'ils sont à portée radio.

Adressage des nœuds Pour contacter un nœud distant, il est nécessaire de connaître son identifiant. Dans les réseaux reposant sur l'infrastructure, l'adressage des nœuds du réseau est administré par des opérateurs disposant chacun d'une partie de l'espace d'adressage. Cependant, l'utilisation de certains mécanismes comme le NAT, l'adressage dynamique ou de la mobilité des nœuds peuvent faire varier l'adresse d'un nœud au cours du temps ou attribuer une même adresse à plusieurs nœuds. Afin d'identifier durablement les nœuds, les réseaux pair-à-pair ont recours à un adressage indépendant de l'adresse IP. Les identifiants peuvent être basés sur l'adresse MAC ou la clé publique d'un nœud. Dans les réseaux opportunistes, plusieurs technologies de communications sans fil peuvent être utilisées (e.g., Wi-Fi en mode ad hoc, Bluetooth). Leurs méthodes d'adressage sont incompatibles entre elles. À l'instar des réseaux pair-à-pair, l'utilisation d'identifiants indépendants de la technologie de communication, comme ceux utilisés dans les réseaux pair-à-pair, permet d'adresser des messages à des nœuds quelle que soit la technologie de communication, l'opérateur du réseau, ou la localisation du nœud.

Redondance des messages Dans les réseaux pair-à-pair large échelle, la défaillance de machines est inévitable : extinction de la machine, pannes matérielles, etc. Pour pallier ces problèmes, les réseaux pair-à-pair répartissent et dupliquent les informations pour que la perte de nœuds ne compromette par l'intégrité du système. La redondance de messages sert également à diminuer le nombre de sauts qu'un message doit traverser avant d'atteindre sa destination. Dans les réseaux opportunistes incluant des objets connectés grands publics fonctionnant sur batterie, la disparition arrivera tôt ou tard. Les réseaux opportunistes utilisent la redondance de messages entre pairs pour pallier des défaillances, et pour augmenter la probabilité que le porteur d'un message rencontre son destinataire. Dans certains réseaux pair-à-pair, les données sont réparties géographiquement pour améliorer le temps d'accès à celles-ci en fonction des requêtes et de la localisation des clients.

Routage Les réseaux pair-à-pair d'infrastructure supposent l'existence d'un mécanisme de routage permettant de contacter tout nœud dont l'adresse est connue. Les réseaux opportunistes ne disposent pas de mécanisme de routage intégré. Une solution de routage spécifique doit donc être utilisée en fonction des caractéristiques du réseau. Le routage peut être effectué par inondation, par calcul des probabilités de rencontre se basant sur l'historique des contacts, par analyse sociale ou bien la localisation des nœuds.

Latence Les réseaux pair-à-pair ont été à l'origine conçus pour les réseaux d'infrastructure ayant de faibles latences. Tout nœud est capable d'en contacter un autre avec des latences de l'ordre de la centaine de millisecondes. Dans les réseaux connectés de façon intermittente, un message peut être délivré à son destinataire en plusieurs minutes ou heures du fait de la mobilité et de la distance des nœuds. Il est crucial qu'une surcouche

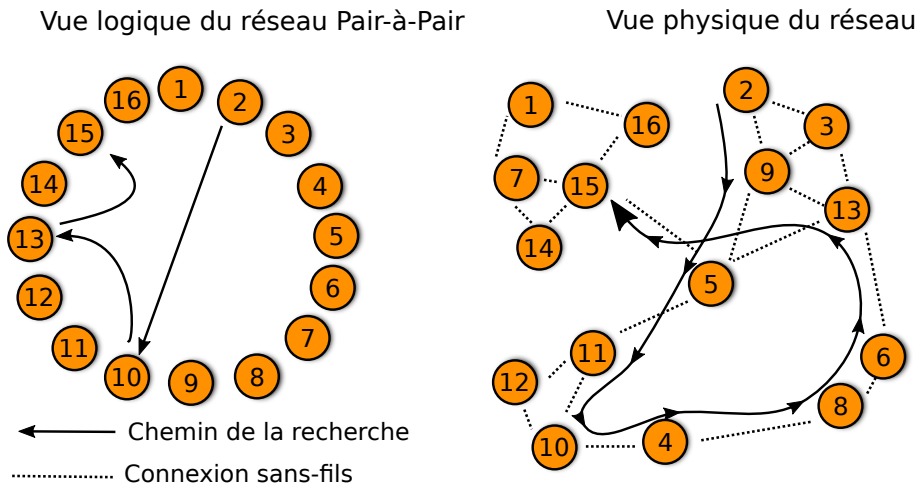


FIGURE 4.1 – Inadéquation entre le réseau logique et le réseau physique

construite au-dessus d'un réseau opportuniste tienne compte de la localisation des nœuds afin de réduire le temps d'acheminement.

4.2 Utilisation des réseaux pair-à-pair structurés

Le routage dans une table de hachage répartie (*Distributed Hash Table*, DHT) est généralement basé sur les identifiants des nœuds, et ne prend pas en compte la position des nœuds pour acheminer les messages. Ces nœuds peuvent être potentiellement très éloignés physiquement. La topologie de la vue logique devient donc inadaptée par rapport au réseau physique sous-jacent. Un exemple de topologie est illustré dans la figure 4.1. Dans cet exemple, un nœud (i.e., 2) cherchant une information émet une requête devant traverser trois nœuds pour trouver l'information. Bien que le nombre de sauts soit faible, la répartition géographique des nœuds à contacter rend le système inefficace dans le cas des réseaux opportunistes. Pour traverser ces trois nœuds de la vue logique (10,13,15) la requête devra physiquement emprunter le chemin (9,5,11,10,4,8,6,13,5,15).

Dans le cas de Chord, chaque nouvel arrivant doit établir $O(logN)$ nouvelles connexions et $O(logN)$ participants doivent altérer leur table de routage. De plus, les informations enregistrées doivent être redistribuées entre les pairs. Les fréquentes apparitions/disparitions de nœuds dans un réseau opportuniste, pourraient engendrer un surcoût prohibitif de maintenance de la DHT.

Les performances des réseaux pair-à-pair structurés sont fortement dépendantes de la qualité des informations stockées dans la DHT. Lors des déconnexions, le réseau peut se retrouver partitionné, et plusieurs DHT peuvent coexister. Une fois les connexions rétablies entre les nœuds, les DHT doivent être fusionnées afin que les nœuds des différentes partitions puissent communiquer à nouveau.

Les DHTs peuvent être adaptées pour prendre en compte la nature dynamique des

réseaux mobiles [95, 96, 97, 98]. Ekta [99] et MADPastry [100] tiennent compte de la localisation des nœuds dans la construction de la DHT en utilisant des nœuds servant de point de repère pour créer des grappes de nœuds physiquement proches. Cependant ces adaptations reposent sur les algorithmes de routage AODV, OLSR, ou DSR pour acheminer leurs messages. Ces algorithmes ne sont pas efficaces lorsqu'il n'existe pas ou peu de connexions de bout en bout entre les nœuds et donc, ne sont pas adaptés pour les réseaux opportunistes.

4.3 Utilisation des réseaux pair-à-pair non structurés

Les réseaux pair-à-pair non structurés n'imposent pas d'organiser les nœuds selon une topologie stricte. Ils sont plus tolérants à la disparation de nœuds que les réseaux pair-à-pair structurés. Le coût de maintenance de la vue logique est plus faible que dans les réseaux structurés.

Dans les réseaux opportunistes, le routage doit s'effectuer sur chacune des machines. Les techniques de routage applicatif tels qu'utilisés dans les réseaux pair-à-pair pour faire de la diffusion de données multicast pourraient être utilisées aussi pour les communications opportunistes.

Des solutions comme ORION [101], MPP [101] ZP2P [102], P2PSI [103] sont des adaptations de réseaux pair-à-pair non structurés pour les MANETs. Ces solutions supposent des connexions de bout en bout et reposent sur des algorithmes de routage tels que AODV [104], DSR [104] ou ARA [104] pour acheminer leurs messages. Ces solutions ne sont donc pas adaptées pour les réseaux opportunistes.

Le principe du *gossiping* dans lequel des informations sont échangées avec des nœuds choisis aléatoirement peut s'appliquer aux réseaux opportunistes en choisissant des nœuds à portée de communication.
Ce principe a servi de support pour concevoir un mécanisme de synchronisation entre nœuds [105] dans les réseaux pair-à-pair. La synchronisation d'horloge dans les réseaux opportunistes est un prérequis important pour mettre en œuvre certains mécanismes d'économie d'énergie telle que la mise en veille périodique des nœuds. Afin de ne pas manquer des messages, les nœuds doivent être réveillés au même moment. Ceci requiert des horloges aussi synchronisées que possible. L'utilisation de cette technique serait donc intéressante pour synchroniser les nœuds dans les réseaux opportunistes.
Ce principe a aussi été utilisé pour permettre le calcul de valeur globale [106]. Ce mécanisme pourrait servir à calculer l'espace disque total disponible dans le réseau ainsi son utilisation moyenne pour mettre en œuvre des algorithmes de répartition de cache disque dans les réseaux opportunistes.

Dans les réseaux pair-à-pair non structurés, certaines techniques de recherche d'information comme l'inondation ou la marche aléatoire peuvent être perçues comme similaires aux techniques de dissémination épidémique des réseaux opportunistes. En effet, la topologie de ce type de réseau ne leur permet pas de localiser facilement une ressource dans

le réseau. De façon similaire aux réseaux opportunistes, l'inondation simple peut rapidement saturer le réseau, et la marche aléatoire se révèle être peu performante. L'utilisation de filtres de Bloom a été proposée comme alternative efficace pour orienter la recherche d'information dans les réseaux pair-à-pair non structurés [107, 108, 109, 110]. Ce mécanisme est applicable à la fois aux parties sans fil et à la partie infrastructure du réseau, passe à l'échelle, et s'adapte aux environnements dynamiques tels que les RHCI.

4.4 Conclusion

Dans le cadre de l'IoT, l'utilisation des techniques permettant de créer des surcouches logiques dans les réseaux pair-à-pair permettrait de masquer les différences entre les différents types de réseau (e.g. Ad-Hoc, Infrastructure) et les différentes technologies de communication sans fil. L'abstraction du voisinage dans laquelle les nœuds communiquent uniquement avec un sous-ensemble appelé leurs voisins est une surcouche répondant à ce besoin.

Les adaptations des réseaux pair-à-pair structurés et non structurés pour les MANETs reposent sur l'utilisation d'algorithme de routage tels que AODV ou OLSR. Ces algorithmes sont inefficaces lorsqu'il n'existe pas de connexions de bout en bout entre les nœuds et qu'ils sont déconnectés sur de longues périodes comme c'est le cas dans les réseaux à connexion intermittente. Une solution de routage spécifique doit donc être utilisée pour les parties sans fil du réseau. Cette solution devrait reposer sur le principe général du *store, carry and forward* pour supporter les connexions intermittentes entre les nœuds et sur la diffusion de multiples copies pour augmenter la probabilité de délivrance et diminuer le délai de transmission. Dans la partie infrastructure du réseau, un *peer sampling service* serait utilisé pour se conformer à l'abstraction de voisinage. Cyclon est un protocole de *peer sampling service* robuste et simple à mettre en œuvre qui répond à ce besoin.

Dans la surcouche pair-à-pair les données pourront être diffusées en utilisant le principe du *gossiping* afin de répartir la charge entre les pairs et acheminer les données le plus rapidement possible.

Afin d'obtenir les meilleures performances, la solution devrait tenir compte de la localisation des nœuds dans le réseau. Les filtres de Bloom sont une structure appropriée répondant à ce besoin.

Dans les chapitres suivants, nous proposons une approche reposant sur ces différents mécanismes.

Deuxième partie

Contribution

5

Nephila : une plateforme pair-à-pair pour des réseaux hybrides à connectivité intermittente

Dans ce chapitre, nous présentons Nephila, une plateforme pair-à-pair pour les réseaux hybrides à connectivité intermittente (RHCI). Nous donnons dans un premier temps une vue d'ensemble de celle-ci, puis nous détaillons la structure du réseau de recouvrement, les mécanismes de détection des voisins, et de calcul des routes dans le réseau de recouvrement.

5.1 Vue d'ensemble et architecture de la plateforme Nephila

Nephila est un système de communication pair-à-pair décentralisé et non structuré pour RHCI, qui offre une vue homogène du réseau en masquant les disparités de connectivité existant entre les différents nœuds composant le réseau. Ce type de réseaux peut en effet impliquer de nombreux protocoles et technologies de communication. Il fournit un ensemble de fonctions permettant d'élaborer différentes stratégies de transfert de messages.

L'architecture modulaire de Nephila est présentée dans la figure 5.1. Nephila est mis en œuvre sous la forme d'un intergiciel. Le système que nous avons conçu se veut simple et générique. Nous ne faisons aucune hypothèse sur la mobilité des nœuds.

Nephila implante un mécanisme permettant à chaque nœud du réseau de découvrir quels sont ses voisins. Dans cette optique, Nephila effectue une découverte proactive de nœuds fixes et mobiles. Ce processus de découverte est une fonctionnalité originale de Nephila, et par conséquent est détaillé dans un paragraphe spécifique dans la suite de ce chapitre.

Afin de supporter les ruptures de connectivité, l'algorithme de transfert de messages de Nephila met en œuvre le principe général du « store, carry and forward ». Cet algorithme utilise, sur chaque nœud, un cache de messages permettant de stocker temporaire-

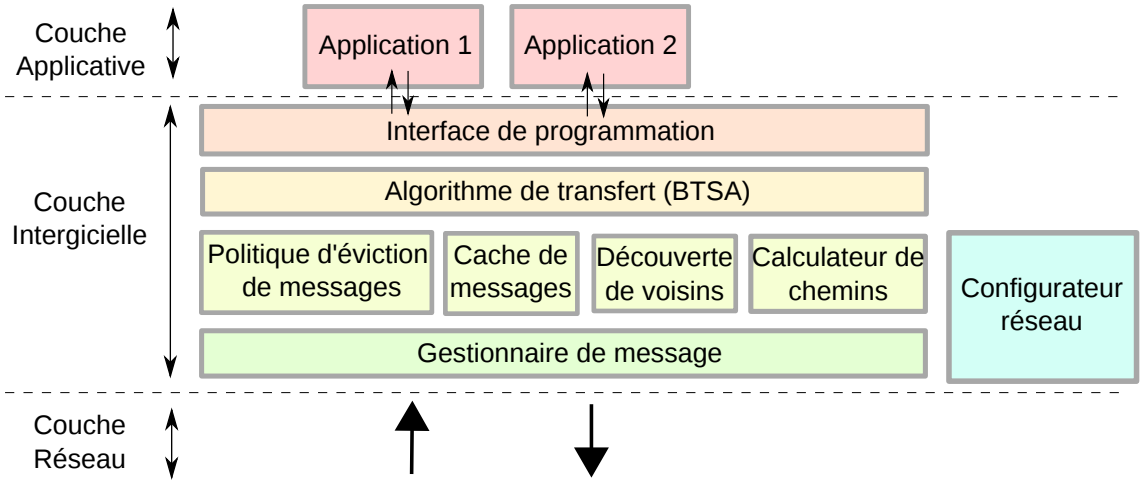


FIGURE 5.1 – Architecture de Nephila

ment des messages à destination d'autres nœuds. Il décide quand et à qui sont transférés les messages du cache. Cet algorithme cherche à maximiser le taux de délivrance de messages, minimiser la latence, et réduire la consommation de bande passante. Chaque nœud mobile et fixe dispose d'un cache de messages permettant la réplication des messages.

La politique d'éviction de messages supprime du cache les messages expirés ou déjà délivrés, et les messages les plus anciens lorsque le cache est plein.

Le calculateur de chemin quant à lui fournit pour chacun des nœuds du réseau une liste de valeurs nommées valeurs de chemin (VC). Une valeur de chemin reflète la capacité d'un nœud à faire parvenir des messages à une destination donnée, soit directement, soit via des nœuds intermédiaires. Ces valeurs de chemin sont utilisées par l'algorithme de transfert de messages pour déterminer les nœuds les plus à même de délivrer un message à son destinataire. Ces valeurs de chemin sont regroupées dans une structure de données appelée filtre de Bloom à décroissance exponentielle.

5.2 Réseau de recouvrement pair-à-pair

Le réseau de recouvrement permet de masquer les différences entre les parties ad hoc et la partie infrastructure du réseau. Le but est de faciliter la programmation aux couches logicielles de niveau supérieur. Ceci permet par exemple à l'algorithme de transfert de messages de traiter tous les nœuds de la même façon qu'ils soient sur des parties filaires ou sans fil, reliées à l'infrastructure ou sur des parties ad hoc du réseau.

Le réseau de recouvrement de Nephila doit être capable de couvrir des réseaux contenant plusieurs milliers de nœuds, qu'ils soient présents dans les parties ad hoc ou infrastructure du réseau. Les mécanismes mis en œuvre doivent donc être capables de passer à cette échelle.

Le réseau de recouvrement est mis en œuvre au moyen d'une surcouche pair-à-pair.

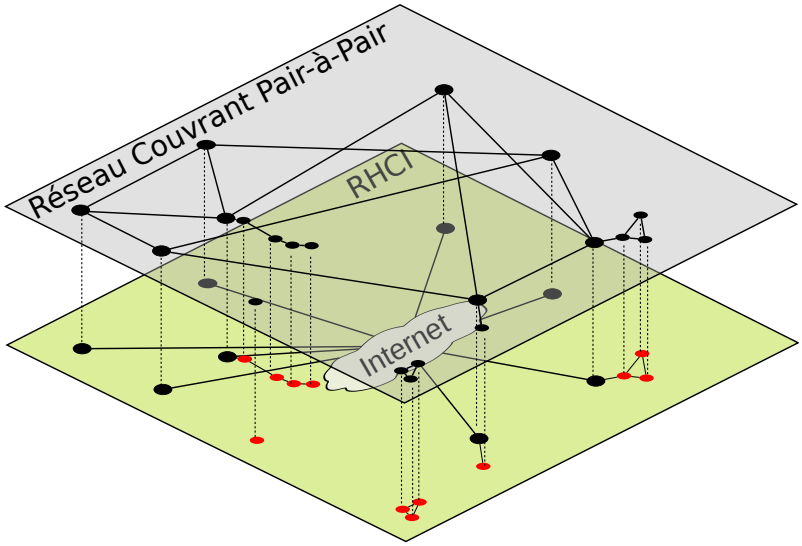


FIGURE 5.2 – Surcouche pair-à-pair

Une vue uniformisée du réseau est construite, permettant à chaque nœud de communiquer, et ce indépendamment de la couche réseau qu'il utilise. Nephila fournit à chaque nœud une vue logique du réseau appelée voisinage dans laquelle seuls les nœuds dits *voisins* sont autorisés à communiquer. Dans cette abstraction, deux voisins sont supposés être capables de communiquer avec une faible latence et peu de perte de paquets. Le contrôle du voisinage par Nephila permet de limiter le nombre de connexions qu'un nœud doit maintenir pour permettre un passage à l'échelle, et de vérifier la qualité du lien réseau permettant de communiquer.

5.3 Découverte des voisins

Nephila met en œuvre deux mécanismes distincts de découverte des nœuds voisins. Le premier mécanisme est dédié à la découverte des nœuds qui constituent le cœur du RHCI (i.e., les routeurs). Le second est dédié à la découverte des nœuds dans les parties sans fil du réseau.

Dans un RHCI, il est difficile de se baser uniquement sur l'adresse IP des objets pour transférer des messages, et en particulier dans les parties ad hoc du réseau du fait de la mobilité. De même, dans la partie infrastructure l'adresse IP peut être attribuée dynamiquement et changée périodiquement, ou partagée par plusieurs nœuds présents dans le même réseau local (adressage privé). Nephila attribue donc un identifiant supposé unique à chacun des nœuds. La construction de cet identifiant est le résultat d'une application d'une fonction de hachage sur la clé publique d'un nœud. Cette clé publique est générée à l'installation de Nephila sur un terminal et pourra servir pour chiffrer des messages. Afin de transférer les messages aux couches réseau inférieures, Nephila maintient une table de correspondance entre l'identifiant Nephila d'un voisin et son adresse IP ou MAC. Cette correspondance est établie pendant la découverte des nœuds et est supprimée lorsque deux

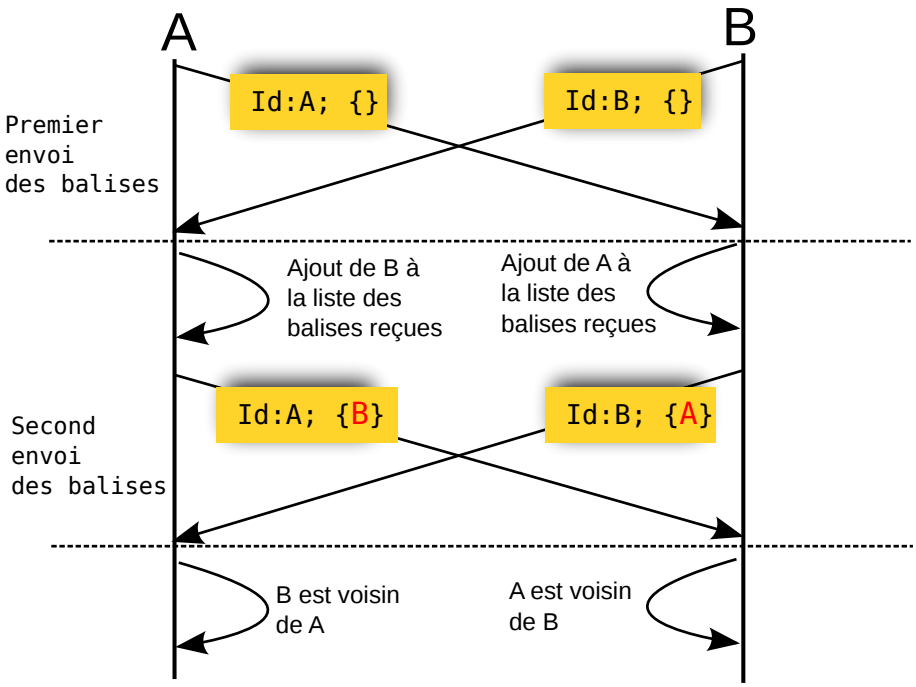


FIGURE 5.3 – Découverte des voisins dans les parties ad hoc du réseau

nœuds ne sont plus considérés comme voisins.

5.3.1 Partie ad hoc du réseau

Dans les parties ad hoc du réseau, deux nœuds sont considérés comme étant voisins s'ils sont capables de communiquer sans utiliser de nœuds intermédiaires.

La découverte de nœuds voisins (qui présentée dans la figure 5.3 est effectuée en utilisant un mécanisme de diffusion de messages de balisage. Chaque nœud diffuse périodiquement un message (appelé balise) incluant son propre identifiant, et les identifiants des autres nœuds dont il a récemment reçu une balise. Un nœud recevant une balise contenant son propre identifiant peut en déduire qu'il est à portée de communication du nœud ayant émis cette balise. A travers ce mécanisme on souhaite s'assurer qu'il y a bien une symétrie entre les deux nœuds en termes d'émission et de réception.

Si un nœud ne reçoit plus ou n'apparaît plus dans les balises d'un de ses voisins pendant un certain temps alors il ne considère plus ce nœud comme faisant partie de son voisinage.

Afin de garantir que deux nœuds voisins sont à portée directe de communication, les balises sont exclues du principe de communication «store, carry and forward». Elles ne sont ni stockées dans le cache des nœuds, ni retransmises.

5.3.2 Partie infrastructure du réseau

Dans la partie infrastructure, les voisins sont découverts en utilisant un service d'échantillonnage de pairs basé sur Cyclon [111, 112]. Cyclon permet de construire une surcouche pair-à-pair tolérant un nombre important de connexions et déconnexions de nœuds dans le système (appelés *churn*). Cyclon permet de garder un graphe connecté même lorsque la moitié des nœuds quittent le système. L'arrivée et le départ massif de nœuds, appelés *flash crowd*, peuvent se produire aux heures de départ ou d'arrivée au domicile.

Cyclon fournit à chaque nœud un sous-ensemble des membres du système appelé *voisins*. Le voisinage de chaque nœud est modifié dans le temps afin de faciliter la dissémination d'information. Le nombre maximal de voisins de chaque nœud est limité et paramétrable.

Chaque pair échange périodiquement avec un de ses voisins un sous-ensemble aléatoire de son voisinage. Pour chaque voisin une variable *age* est gardée en mémoire. Le rôle de cette variable « age » est de borner la date à laquelle un lien sera choisi pour effectuer une opération d'échantillonnage de pairs. Ceci permet d'éliminer les nœuds ne faisant plus partie du système, sans qu'ils n'aient à avertir eux-mêmes de leur départ. L'opération d'échantillonnage de pairs est répétée périodiquement selon un paramètre λt.

Opération d'échantillonnage de pairs P

1. Incrémenter le champ *age* pour chacun des voisins dans le cache de P
2. Selectionner le sous ensemble composé du plus vieux voisin Q selon sa variable *age* et de $l-1$ voisins choisis aléatoirement. l étant un paramètre décrivant le nombre de nœuds maximum devant être échangés à chaque opération.
3. Remplacer dans le sous-ensemble sélectionné Q par l'identifiant P avec son *age* 0
4. Envoyer ce sous-ensemble au nœud Q
5. Recevoir de Q un sous-ensemble choisi aléatoirement des voisins de taille i avec $0 \leq i \leq l$
6. Supprimer dans ce sous-ensemble toute entrée déjà contenue dans P ou ayant P comme identifiant.
7. Mettre à jour les entrées de P en ajoutant les entrées restantes du sous-ensemble. Tout d'abord ajouter les entrées jusqu'à ce que la taille maximale de voisinage soit atteinte. Remplacer les entrées du sous-ensemble envoyé à Q.

(a) Le nœud 1 démarre une opération d'échange de nœuds

Nœud 1

Nœud	Age
12	5
48	0
16	7
14	8
5	3
3	8
63	12
20	9
72	7
15	14

(b) L'âge des voisins est incrémenté de 1

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

(c) Sélection d'un sous-ensemble

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Nœud	Age
15	15
5	4
14	9

(d) Remplacement du plus vieux voisin

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Nœud	Age
1 ~~15~~	0 ~~15~~
5	4
14	9

(e) Envoi du sous-ensemble

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Nœud	Age
1	0
5	4
14	9

Nœud 15

Nœud	Age
1	0
5	4
14	9

Nœud	Age
12	6
41	7
5	8
56	11
21	3
27	6
35	2
11	8
89	1

FIGURE 5.4 – Exemple d'une opération d'échange de nœuds

(f) Sélection aléatoire de 3 voisins

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Nœud	Age
1	0
5	4
14	9

Nœud 15

Nœud	Age
1	0
5	4
14	9

Nœud	Age
12	6
41	7
5	8
56	11
21	3

Nœud	Age
27	6
12	6
41	7

Nœud	Age
27	6
35	2
11	8
89	1

(g) Envoi de la réponse

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Nœud	Age
1	0
5	4
14	9

Nœud	Age
27	6
12	6
41	7

Nœud 15

Nœud	Age
1	0
5	4
14	9

Nœud	Age
12	6
41	7
5	8
56	11
21	3

Nœud	Age
27	6
12	6
41	7

Nœud	Age
27	6
35	2
11	8
89	1

(h) Supression des entrées déjà connues

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Noeud	Age
1	0
5	4
14	9

Nœud	Age
27	6
~~12~~	~~6~~
41	7

Nœud 15

Nœud	Age
1	0
~~5~~	~~4~~
14	9

Nœud	Age
12	6
41	7
5	8
56	11
21	3

Nœud	Age
27	6
12	6
41	7

Nœud	Age
27	6
35	2
11	8
89	1

(i) Remplissage des entrées libres

Nœud 1

Nœud	Age
12	6
48	1
16	8
14	9
5	4
3	9
63	13
20	10
72	8
15	15

Nœud	Age
1	0
5	4
14	9

Nœud	Age
27	6
41	7

Nœud 15

Nœud	Age
~~1~~	~~0~~
14	9

Nœud	Age
12	6
41	7
5	8
56	11
21	3
1	0

Nœud	Age
27	6
12	6
41	7

Nœud	Age
27	6
35	2
11	8
89	1

(j) Remplacement des entrées

Nœud 1

Noeud	Age
12	6
48	1
16	8
14	9
41 ~~5~~	7 ~~4~~
3	9
63	13
20	10
72	8
27 ~~15~~	6 ~~15~~

Noeud	Age
1	0
5	4
14	9

Noeud	Age
~~27~~	~~6~~
~~41~~	~~7~~

Nœud 15

Noeud	Age
~~14~~	~~9~~

Noeud	Age
12	6
41	7
5	8
56	11
21	3
1	0
14 ~~27~~	9 ~~6~~

Noeud	Age
27	6
12	6
41	7

Noeud	Age
35	2
11	8
89	1

(k) Fin de l'opération

Nœud 1

Noeud	Age
12	6
48	1
16	8
14	9
41	7
3	9
63	13
20	10
72	8
27	6

Nœud 15

Noeud	Age
12	6
41	7
5	8
56	11
21	3
1	0
14	9
35	2
11	8
89	1

FIGURE 5.4 – Exemple d'une opération d'échange de nœuds

Exemple d'opération d'échantillonnage de pairs La figure 5.4 un exemple d'une opération d'échantillonnage de pairs entre deux nœuds. Chaque nœud possède un cache pouvant contenir au maximum dix identifiants de nœuds. Le nœud 1 initie une opération d'échange de nœuds (figure 5.4a). Il incrémente de 1 l'âge de tous les nœuds présents dans son cache (figure 5.4b). Il sélectionne un sous-ensemble de nœuds composé du nœud le

plus âgé de son cache qui est le nœud 15 avec un âge de 15 ainsi que deux autres nœuds choisis aléatoirement : les nœuds 5 et 14 (figure 5.4c). Il remplace dans ce sous-ensemble l'entrée correspondant au nœud le plus âgé par son identifiant (i.e. 1) d'âge 0 (figure 5.4d). Ce sous-ensemble est envoyé au nœud le plus âgé (figure 5.4e). 15 sélectionne un sous-ensemble de 3 nœuds choisi aléatoirement (figure 5.4f) et l'envoie en retour au nœud 1 (figure 5.4g). Chacun des nœuds 1 et 15 va intégrer les sous-ensembles reçus dans leur propre cache. Le nœud 15 supprime l'entrée du nœud 5 et le nœud 1 supprime l'entrée correspondant au nœud 12 dans leur sous-ensemble reçu puisque ces entrées existent déjà dans leur cache respectif (figure 5.4h). Le nœud 15 intègre l'entrée du nœud 1 dans l'entrée non utilisée (figure 5.4i). Les entrées restantes sont intégrées en remplaçant les entrées qui ont été envoyées à l'autre nœud (figure 5.4j). Pour le nœud 15, l'entrée correspondant au nœud 27 est remplacée par le nœud 14. Pour le nœud 1 l'entrée 41 remplace 5 et 27 remplace 15.

5.3.2.1 Amorçage du réseau et recouvrement d'isolation

Cyclon étant un système entièrement décentralisé, il est nécessaire d'avoir une méthode pour joindre la surcouche pair-à-pair. Cette méthode est appelée *amorçage*. Il suffit pour un nœud désirant joindre le système d'en connaître un autre qui est déjà connecté à la surcouche pair-à-pair. En effet, en effectuant successivement des opérations d'échantillonnage de pairs, avec un membre du système, un nœud pourra découvrir progressivement ses voisins.

Nous supposons l'existence d'un petit groupe de nœuds, connus à l'avance, dont au moins un d'entre eux est toujours joignable. Ces nœuds sont appelés nœuds d'amorçage.

Dans le cas où tous les voisins d'un nœud quittent le système, celui-ci se retrouve isolé. Il lui suffit d'effectuer une nouvelle découverte de voisins avec des nœuds d'amorçage pour rejoindre le système.

5.4 Calcul des chemins

Les valeurs de chemins représentent une estimation de la capacité d'un nœud à transmettre un message à un autre, directement ou en utilisant d'autres nœuds comme intermédiaires. Ces valeurs sont utilisées pour mettre en œuvre des algorithmes de transfert de messages. Elles sont échangées entre voisins pendant le processus de découverte de voisins, et ensuite périodiquement.

La dissémination de ces valeurs peut s'avérer coûteuse, puisque les nœuds peuvent avoir des valeurs de chemin pour un très grand nombre de destinations possibles, voire dans le pire des cas pour tous les nœuds présents dans le réseau. Afin de réduire le volume d'information transmis, nous utilisons une version modifiée des filtres de Bloom à décroissance exponentielle (FBDE). Un FBDE est une extension des filtres de Bloom traditionnels, qui permet d'encoder de manière compacte des tables de routage probabi-

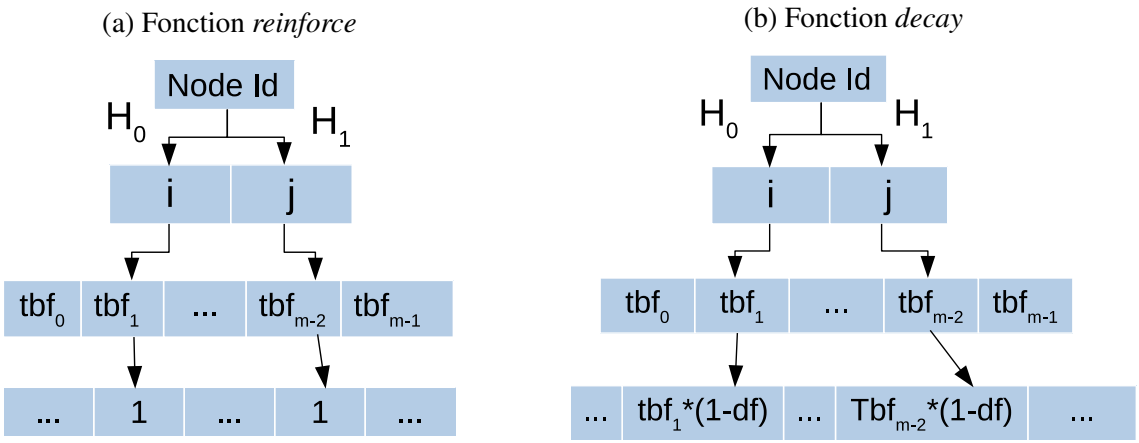


FIGURE 5.5 – Fonctions *reinforce* et *decay*.

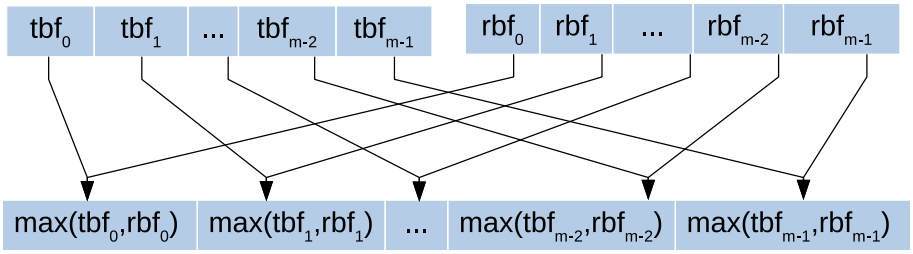
listes. Ces filtres ont été conçus pour stocker et propager des informations sur les contenus hébergés par des nœuds dans des réseaux pair-à-pair. Ils sont généralement associés à un processus de découverte de données. Nous avons conçu une version modifiée de ces FBDE pour stocker et disséminer efficacement les valeurs de chemin de chaque nœud. Cette nouvelle version est appelée FBC (pour filtre de Bloom de chemin). Par la suite, nous présentons une nouvelle mise en œuvre basée sur les FBDE, les filtres de bloom de chemin (FBC).

Pour stocker des informations de routage de façon probabiliste dans les RHCI, nous proposons les Filtres de Bloom de Chemin (FBC), qui sont une variante des filtres de Bloom à décroissance exponentielle. De façon similaire aux filtres de Bloom traditionnels et ceux à décroissance exponentielle, nous définissons un FBC comme un tableau de m éléments noté $[fbc_0, ..., fbc_{m-1}]$. k fonctions de hachage sont définies, et supposées être indépendantes. Elles sont notées $h_0, h_i, ..., h_{k-1}$. Ces fonctions sont aussi bien utilisées pour insérer les valeurs de chemin dans le FBC que pour les trouver. Les indices du tableau m correspondant à un nœud x sont obtenus en évaluant simultanément chacune des k fonctions $h_0, h_i, ..., h_{k-1}$. Contrairement aux autres filtres de Bloom, un FBC contient des réels positifs compris entre 0 et 1.

Quatre fonctions ont été définies pour opérer sur le FBC : *decay(fbc,n,df)*, *reinforce(fbc,n), merge(fbc,rfbc)* and*query(fbc,n)*. n est l'identité d'un nœud, *fbc* le FBC du nœud local, df le facteur de décroissance, et *rfbc* le FBC reçu d'un nœud distant.

Afin de prendre en compte la perte de confiance au fil du temps dans les valeurs de chemin qui servent pour le routage, la fonction *decay* (Figure 5.5b) est appliquée périodiquement par chaque nœud sur son FBC. Lorsqu'un nœud détecte un nœud dans son voisinage, la fonction *reinforce* (Figure 5.5a) est appliquée sur son FBC avec l'identifiant du nœud détecté comme paramètre. Cette fonction fixe à la valeur maximum (i.e., 1) la valeur de chemins correspondant à ce nœud. Ceci reflète la forte capacité à relayer une information pour un nœud à portée de communication. La fonction *reinforce* est aussi appelée périodiquement pour chacun des voisins. Tant que deux nœuds se considèrent comme voisins, ils s'envoient leur FBC local. Le FBC reçu d'un voisin est appelé $rtbf$

(a) Fonction *merge*



(b) Fonction *query*

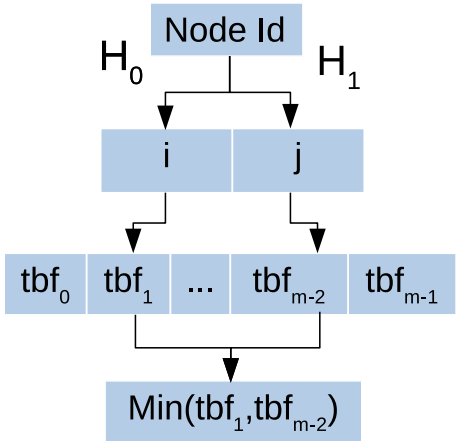


FIGURE 5.6 – Fonctions *merge* et *query*.

(remote trail bloom filter). Il est stocké par le nœud local, ce qui permet à celui-ci de comparer ses valeurs de chemin avec celles de son voisin. Le FBC reçu d'un voisin est intégré dans celui du nœud local au moyen de la fonction *merge*, qui retient pour chaque élément du tableau la valeur maximum. La fonction *decay* est préalablement appliquée sur le FBC du voisin afin de représenter la perte de confiance dans l'information due au saut supplémentaire.

Afin de trouver une valeur de chemin associée à un nœud dans un FBC, la fonction *query* (présentée en Figure 5.6b avec deux fonctions de hachage H_0 et H_1) est utilisée. En invoquant cette fonction sur son propre FBC et sur le FBC reçu de ses voisins, un nœud peut comparer les valeurs de retour d'appels de la fonction *query* et déterminer lequel de lui ou de ses voisins est le meilleur transporteur pour délivrer un message à un nœud donné.

Les FBCs permettent de propager de façon transitive les valeurs de chemin dans le réseau. Ces valeurs de chemin sont réduites proportionnellement avec le nombre de sauts et dans le temps. Une valeur de chemin en n sauts est définie par $(1-df)^n * (1-df)^{\left\lfloor \frac{e-a}{p} \right\rfloor}$ où e et a sont respectivement la date d'émission par un nœud situé à $n-1$ sauts et la date de réception du FBC par un nœud voisin, et est p la période de décroissance.

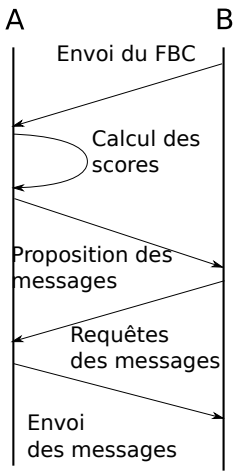


FIGURE 5.7 – Processus d'échange de messages.

En assignant aux nœuds fixes connectés à l'infrastructure une valeur de chemin minimale par défaut supérieure à celle des nœuds mobiles, il est possible de privilégier le transfert de ces messages par l'infrastructure. Le message sera dirigé vers un nœud fixe et sera donc stocké par l'infrastructure jusqu'à découverte de la destination ou de nœuds plus à même de délivrer le message. Les nœuds connectés à l'infrastructure sont supposés être plus à même de délivrer un message, en particulier lorsque la destination est éloignée de l'émetteur du message.

5.5 Algorithme de transfert de messages

Le processus d'échange de messages de Nephila est présenté dans la figure 5.7. Lorsqu'un nœud découvre un nouveau voisin et qu'il reçoit un FBC de ce dernier, il regarde tout d'abord dans son cache local s'il existe des messages qui lui sont directement destinés, et les lui envoie le cas échéant. Grâce au FBC, il peut en outre calculer les scores pour chacun des autres messages de son cache, et sélectionner les messages pour lesquels ce voisin présente un score supérieur au sien pour délivrer lesdits messages. Un vecteur de résumé est construit pour tous les messages sélectionnés. Ce vecteur contient les identifiants de ces messages. Il est envoyé au voisin. Le voisin sélectionne alors parmi les identifiants des messages proposés ceux qui ne sont pas déjà présents dans son cache, et envoie en retour une requête contenant les identifiants de ces messages. Le nœud recevant la requête peut donc envoyer tous les messages dont les identifiants apparaissent dans la requête. Le même processus est appliqué sur le nœud voisin.

BTSA (pour *Best Trail Selection Algorithm*) est un algorithme de transfert de messages que nous avons conçu en utilisant les fonctions présentées précédemment (voir algorithme 2). Il repose sur les valeurs de chemin, et exploite la propriété de transitivité du FBC. Il transfère les messages aux nœuds voisins qui possèdent la plus grande valeur de chemin pour les destinations visées. Lorsqu'un nœud reçoit un FBC de l'un de ses

voisins, il évalue, pour chaque message qu'il possède localement, la capacité de ce voisin à pouvoir les transmettre. Afin d'éviter une dissémination épidémique des messages, qui pourrait résulter de la sélection de nœuds intermédiaires possédant une valeur de chemin comparable à celle du nœud local, une valeur de seuil a été introduite dans BTSA. Un message ne sera donc transféré à un nœud voisin que si celui-ci possède une valeur de chemin pour la destination recherchée supérieure à celle du nœud local augmentée de la valeur de seuil (*best-treshold*).

Algorithm 2 Algorithme de sélection des meilleurs chemins (BTSA)

```
 1: function ON-TRAIL-RECEIVED(trail,from)
 2:     for all messages m do
 3:         if ((query(trail,recipient(m)) - query(local-trail,recipient(m)) > best-treshold))
    then
 4:             propose(m,from)
 5:         else if copies(m) > 0  then
 6:             decrement-copies(m)
 7:             propose(set-copies-to-zero(m),from)
 8:         end if
 9:     end for
10: end function

11: function ON-PROPOSE-RECEIVED(message-id,from)
12:     if (not in-cache(message-id)) then
13:         request(message-id,from)
14:     end if
15: end function

16: function ON-REQUEST-RECEIVED(message-id,from)
17:     forward(find-in-cache(message-id),from)
18: end function

19: function ON-MESSAGE-RECEIVED(m,from)
20:     for all neigbors n do
21:         if query(trail_n,recipient(m)) > query(local-trail,recipient(m)) then
22:             propose(m,n)
23:         else if copies(m) > 0  then
24:             decrement-copies(m)
25:             propose(set-copies-to-zero(m),from)
26:         end if
27:     end for
28: end function
```

Il peut être risqué de transmettre exclusiment des copies de message à des nœuds considérés comme de meilleurs candidats. Lorsque le réseau est peu dense, pas ou très

peu de voisins peuvent être considéré comme de bons candidats, pour transmettre les messages ; réduisant ainsi les chances de délivrer les messages. BTSA permet de résoudre ce problème en transmettant délibérément un nombre limité de copies d'un message à de « mauvais » nœuds intermédiaires (i.e., des nœuds possédant une valeur de chemin plus faible que celle du nœud local). De façon analogue à Spray and Wait [57], le compteur de copies à destination de mauvais intermédiaires est stocké dans chaque message, et décrémenté par la fonction *decrement-copies* à chaque transfert vers un mauvais intermédiaire.

Lorsqu'une décision de transfert de message doit être prise pour un message dont la destination n'a pas été rencontrée, ou est éloignée en temps ou en nombre de sauts (i.e. valeur de chemin proche de 0), le message peut être transféré à des voisins n'étant pas beaucoup plus à même de délivrer le message.

5.6 Mécanisme de « vaccination »

Les algorithmes de transfert de messages utilisés dans les réseaux opportunistes disséminent souvent plusieurs copies d'un même message afin d'augmenter la probabilité de délivrance, ou de diminuer le temps de délivrance. De façon similaire, BTSA réplique des copies de messages sur les nœuds qu'il considère comme étant plus les plus aptes à délivrer un message, ou sur d'autres nœuds lorsque le réseau est peu dense. Un des premiers mécanismes mis en place pour éviter que la multiplication de ces copies ne congestionne le réseau est d'associer à chaque message créé une date d'expiration correspondant à une date au-delà de laquelle tout nœud possédant une copie d'un message peut le supprimer de son cache. Le choix de ce paramètre est critique pour le bon fonctionnement du réseau. Une date d'expiration trop proche pourrait empêcher certains messages d'atteindre leur destinataire, et une date d'expiration trop lointaine provoque l'accumulation de messages dans le réseau. Le choix de ce paramètre est difficile, en particulier lorsqu'on ne suppose aucune connaissance sur la nature des messages, le nombre de nœuds, leur mobilité, ou leur zone de déplacement.

Nous proposons un mécanisme complémentaire au mécanisme de date d'expiration, basé sur le principe de l'infection/vaccination. Ce mécanisme permet d'éliminer des caches des nœuds les messages délivrés avant même qu'ils n'arrivent à expiration. Ce mécanisme permet donc de choisir une date d'expiration plus lointaine, permettant ainsi d'augmenter les chances de délivrance, tout en éliminant au plus tôt les messages délivrés. Le principe de « l'infection/vaccination »repose sur le mécanisme suivant : Lorsqu'une copie de message est transférée sur un nœud, ce dernier est alors marqué comme *infecté* par ce message. Un nœud recevant un message qui lui est destiné crée un *vaccin* pour ce message et ses copies. Ce vaccin est ensuite propagé en retour dans le réseau, tout d'abord au nœud ayant délivré le message puis aux autres nœuds par propagation de tables de vaccination. Tout nœud recevant ce vaccin peut supprimer le message correspondant de son cache. Un nœud vacciné ne peut recevoir une nouvelle copie du message correspondant.

Notre implémentation de ce principe utilise le mécanisme du *gossiping*. Une table de vaccination contenant des identifiants de messages délivrés ainsi que leur date d'expi-

ration est diffusée périodiquement, et à chaque rencontre d'un voisin. Lorsqu'un nœud détecte un nouveau voisin, il lui envoie sa table de vaccination. De même, un nœud partage sa table de vaccination périodiquement avec tous ses voisins. Une table reçue est fusionnée avec la table locale en faisant l'union sur l'ensemble des identifiants. Ceci permet de garder et de disséminer des informations à jour sur les messages délivrés. Une opération de maintenance est effectuée à chaque expiration de message : son identifiant est supprimé de la table de vaccination afin d'éviter que la taille de cette table n'augmente indéfiniment.

5.7 Conclusion

Dans ce chapitre, nous avons présenté la plateforme Nephila. Cette plateforme est composée de différents modules, à savoir 1) un mécanisme permettant à chaque nœud du réseau de découvrir quels sont ses voisins ; 2) un algorithme de transfert de messages qui implémente le principe général du « store, carry and forward » et permet d'acheminer des messages dans les RHCI ; 3) un processeur de calcul de chemin qui quant à lui calcule et fournit pour chacun des nœuds du réseau une liste de valeurs nommées valeurs de chemin ; 4) et un mécanisme de vaccination qui permet de supprimer les messages délivrés au plus tôt afin d'éviter l'accumulation de ceux-ci dans le réseau. Dans le chapitre suivant nous présentons comment ces mécanismes sont utilisés par Nephila pour échanger des données.

6
Échange de données avec Nephila

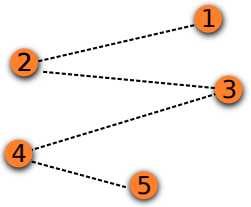
Dans ce chapitre, nous présentons les différents modes de communication offerts par le système Nephila. Ces différents modes de communications ont pour but de faciliter la tâche des développeurs lors de la mise en œuvre d'applications modernes faisant communiquer des objets connectés (e.g. échange de données entre nœuds, invocation de services, partage de contenus). De plus, ces modes de communication sont spécifiquement optimisés pour exploiter les mécanismes offerts par Nephila (filtre de Bloom de chemin, algorithme de vaccination, algorithme de transfert de messages BTSA).

Trois modes distincts de communication sont proposés, à savoir le mode point-à-point, le mode anycast et le mode basé sur le contenu. Ces trois modes de communication sont détaillés dans la suite de ce chapitre.

6.1 Communication point-à-point

La communication point-à-point permet d'échanger des messages entre deux nœuds du réseau. Dans le cadre de l'IoT, ce mode de communication permet, par exemple, d'obtenir la valeur d'un capteur ou bien d'activer un actionneur. La figure 6.1 présente un exemple de propagation des Filtre de Bloom de Chemin (FBC) dans un réseau constitué par Nephila. Chaque nœud dispose d'un FBC de 8 éléments, et deux fonctions de hachage. Le paramètre de décroissance exponentielle est de 0.2. Tous les FBCs sont initialisés à 0. Chaque nœud met à jour son FBC en fonction des nœuds présents dans son voisinage. Par exemple, le nœud 3, qui est voisin des nœuds 2 et 4, mettra à 1 les indices 1, 2 et 5 de son FBC en utilisant la méthode *reinforce*. Les indices 1, 2 et 5 sont obtenus via les deux fonctions de hachage. Chaque nœud envoie son FBC à ses voisins. Chaque FBC est stocké et intégré (avec la fonction *merge*) dans le FBC local après que la fonction de décroissance (*decay*) aie été appliquée sur tous les éléments. La table des valeurs de chemins montre une vue globale des capacités de routage pour chacun des nœuds. La table des valeurs de chemins dernière montre les valeurs qui seront utilisées par l'algorithme de transfert de messages BTSA. Par exemple, un message émis par le nœud 4 à destination du nœud 1 sera tout d'abord transféré au nœud 3. En effet, le nœud 3 à une valeur de Chemin (VC) de 0.8 pour le nœud 1, alors que le nœud 4 a une VC de 0.64, et le nœud 5 de 0. Une fois transmise au nœud 3, une copie sera envoyée au nœud 2. En effet, le nœud 2 étant voisin

Topologie du réseau



Valeurs des fonctions de hachage

Nœud	1	2	3	4	5
H_0	0	1	3	2	7
H_1	4	5	6	1	3

Index / Nœud	0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0

Mise à jour Des Voisins

Index / Nœud	0	1	2	3	4	5	6	7
1	0	1	0	0	0	1	0	0
2	1	0	0	1	1	0	1	0
3	0	1	1	0	0	1	0	0
4	0	0	0	1	0	0	1	1
5	0	1	1	0	0	0	0	0

Premier échange de FBC

Index / Nœud	0	1	2	3	4	5	6	7
1	0.8	1	0	0.8	0.8	1	0.8	0
2	1	0.8	0.8	1	1	0.8	1	0
3	0.8	1	1	0.8	0.8	1	0.8	0.8
4	0	0.8	0.8	1	0	0.8	1	1
5	0	1	1	0.8	0	0	0.8	0.8

Second échange de FBC

Index / Nœud	0	1	2	3	4	5	6	7
1	0.8	1	0.64	0.8	0.8	1	0.8	0
2	1	0.8	0.8	1	1	0.8	1	0.64
3	0.8	1	1	0.8	0.8	1	0.8	0.8
4	0.64	0.8	0.8	1	0.64	0.8	1	1
5	0	1	1	0.8	0	0.64	0.8	0.8

Vue globale des valeurs de chemins

Nœud	1	2	3	4	5
1		1	0.8	0.64	0
2	1		1	0.8	0.64
3	0.8	1		1	0.8
4	0.64	0.8	1		1
5	0	0.64	0.8	1	

FIGURE 6.1 – Exemple de propagation des FBCs

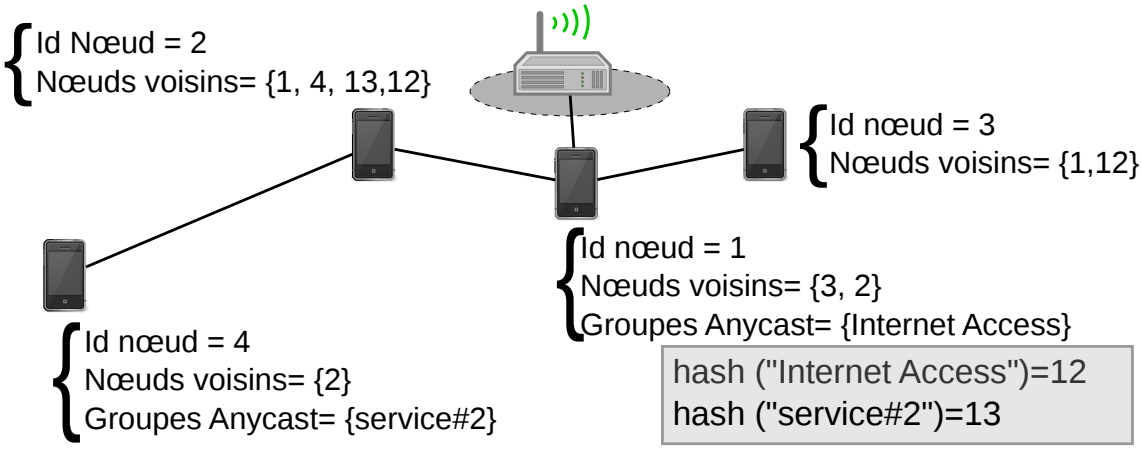


FIGURE 6.2 – Example d'utilisation des groupes anycast dans Nephila

du destinataire il possède une valeur de chemin de 1 pour ce nœud.

6.2 Communication Anycast

Nous proposons un schéma d'échange de données en *Anycast*. Ce mode de communication peut être utilisé pour désigner un service (voir figure 6.2). Les messages à destination d'objets fournissant un service spécifique seront dirigés vers celui présentant le plus haut score, permettant ainsi de réduire la latence et d'augmenter les probabilités de délivrance du service demandé. Le modèle de communication *anycast* dans Nephila n'impose aucune convention de nommage spécifique. Ce mécanisme est notamment utilisé par Nephila pour identifier les nœuds capables d'offrir un accès à Internet grâce à une connexion filaire à l'infrastructure, via une connexion 3G/4G, ou via un point d'accès. Un message peut donc être reçu, stocké, transporté et transféré par différents nœuds intermédiaires pour finalement être délivré à sa destination sur Internet par un nœud qui est membre du groupe *anycast Internet access*.

Nephila n'intègre pas de mécanisme de coordination permettant de garantir la délivrance d'un message à au plus un membre du groupe *anycast*. Plusieurs copies d'un message peuvent donc être délivrées à différents nœuds appartenant au même groupe anycast. L'algorithme de vaccination permet toutefois de supprimer rapidement les copies restantes une fois le message délivré. Nous supposons que les applications tolèrent la réception de copies multiples, ou qu'elles implantent elles-mêmes un mécanisme de coordination spécifique garantissant la délivrance d'au plus un message. Chaque nœud annonce son appartenance à un groupe *anycast* à ses voisins dans les messages de balisage. À l'instar du mode de communication point-à-point, les identifiants de ces groupes *anycast* seront ajoutés dans les FBCs.

Un exemple est présenté dans la figure 6.2. Cette figure présente le résultat de l'application de l'unique fonction de hachage aux noms des groupes anycast *Internet Access* et

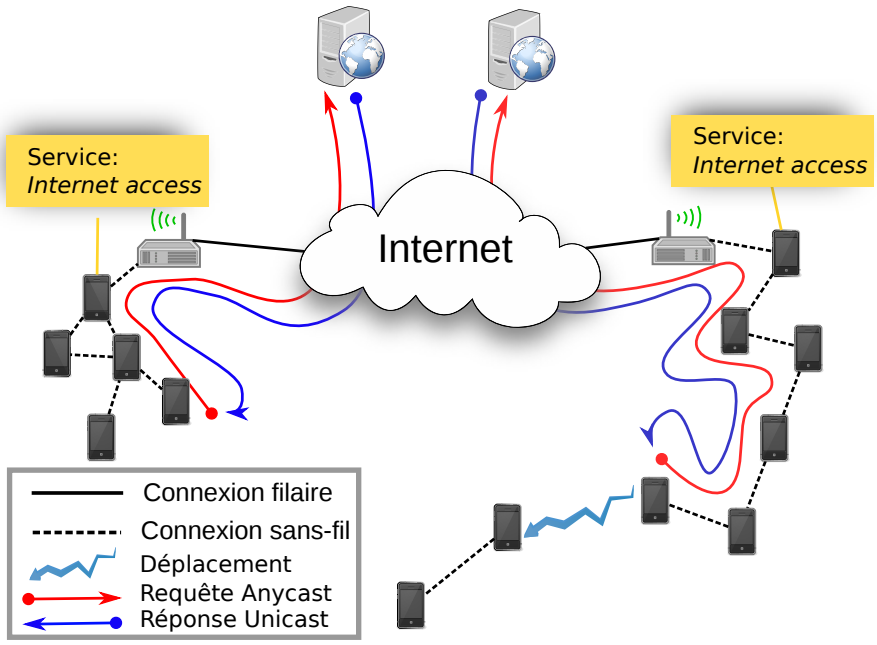


FIGURE 6.3 – Exemple d'utilisation du mode de communication anycast pour implémenter un service permettant l'accès à Internet

Service#2. Les valeurs obtenues sont respectivement 12 et 13. Dans cet exemple, le nœud 3 peut ajouter 1 et 12 dans sa liste de voisins, le nœud 2 peut ajouter 1,4,12,13. Le nœud 4 peut être atteint par le nœud 2, en utilisant son adresse (i.e. 4) ou bien son groupe anycast (i.e. 13).

Les messages applicatifs à destination de nœuds distants accessibles sur Internet seront encapsulés dans des messages Nephila, avec comme adresse de destination le groupe anycast *Internet access* (voir dans la figure 6.3). Ces messages seront ensuite extraits par un nœud membre de ce groupe. Une réponse peut être reçue en effectuant l'opération inverse : l'hôte distant qui n'est pas un nœud du réseau pair-à-pair créé par Nephila peut répondre au membre du groupe anycast lui ayant transmis le message. Ce dernier à son tour encapsulera la réponse dans un message Nephila (unicast) à destination de l'émetteur initial du message. L'utilisation du schéma de transfert anycast est transparent pour l'algorithme de transfert BTSA, aucune modification n'est nécessaire.

6.3 Communication fondée sur le contenu

L'échange de données se fait traditionnellement entre des machines dont les destinataires sont clairement identifiés. Lorsqu'une donnée est désirée, mais que sa localisation n'est pas connue à l'avance un mécanisme externe est nécessaire pour découvrir l'adresse des nœuds hébergeant cette donnée. Le mode de communication basée sur le contenu permet directement de spécifier le type de contenu désiré sans connaître sa localisation. Ce mode de communication permet par exemple d'implanter un système de partage de

fichiers distribué et décentralisé, ou bien de servir de cache de données dans le cadre du déchargement de données. Dans le mode de communication *basé sur le contenu,* l'utilisateur spécifie directement le type de contenu qui l'intéresse, et c'est au système sous-jacent d'effectuer la phase de recherche, et la récupération du contenu. Le mode de communication basé sur le contenu permet une plus grande flexibilité sur l'échange de données. La spécification des données étant indépendante de l'adressage des nœuds ou du réseau servant du support à sa transmission, elle est particulièrement bien adaptée aux réseaux dont la topologie est très dynamique tels que les RHCI.

Nous avons intégré dans le système Nephila un système d'échange de données basé sur le contenu adapté pour les réseaux hybrides à connectivité intermittente. Ce système est basé sur un unique échange de requête/réponse afin de limiter la latence. Il est basé sur l'utilisation des FBC, et utilise les modes de communication *anycast* et *unicast* pour acheminer les messages. Le contenu peut provenir de machines accessibles sur Internet ou être produit par les nœuds mobiles eux-mêmes et être partagé avec les autres nœuds mobiles ou d'autres machines accessibles sur Internet.

Nous avons spécifiquement basé notre approche de dissémination de données sur un schéma de requête/réponse afin de pouvoir bénéficier des avantages du mécanisme de vaccination (présenté en section 5.6). En effet, nous pensons que l'utilisation des dates d'expiration seules est insuffisante pour limiter la propagation des messages. Le schéma requête/réponse que nous utilisons permet d'identifier lorsqu'un message a été délivré à son destinataire. Ceci permet d'envoyer un message de vaccination en retour, et d'inscrire l'identifiant du message dans la table de vaccination.

Un nœud désirant obtenir un contenu particulier émet une requête *anycast* (telle que présentée en section 6.2) avec comme destinataire l'identifiant du contenu désiré. Un contenu est identifié de façon analogue aux services *anycast* et donc n'impose donc aucune convention de nommage spécifique. Lorsque la requête est délivrée à un nœud hébergeant ce contenu, celui génère une réponse en *unicast* à destination de l'émetteur de la requête.

Si le contenu doit être spécifiquement récupéré ou envoyé sur un nœud ne faisant pas partie du réseau formé par Nephila (e.g. un serveur localisé sur Internet) un nœud fournissant un service de passerelle peut être utilisé. Dans ce cas, la requête de demande de contenu va être encapsulée dans un message Nephila anycast avec comme destination l'identifiant du group anycast correspondant à un service de passerelle. Tout nœud membre du réseau Nephila et disposant à la fois d'une interface de communication sans-fils et d'une connexion à Internet est potentiellement capable de fournir ce service de passerelle.

La figure 6.4 présente un exemple de dissémination d'une donnée en utilisant l'approche basée sur le contenu. Dans cet exemple, le réseau est composé de 5 nœuds, et dont l'un d'entre eux (le nœud 2) possède initialement la donnée *contenu#1*. Le nœud 4 désire obtenir la donnée *contenu#1*. Il émet alors une requête anycast ayant comme destinataire *contenu#1* (étape 1). Cette requête sera transférée au nœud 2 puis au nœud 3 grâce à l'algorithme BTSA et les FBCs (étape 2). En effet, la capacité d'atteindre le *contenu#1* sera

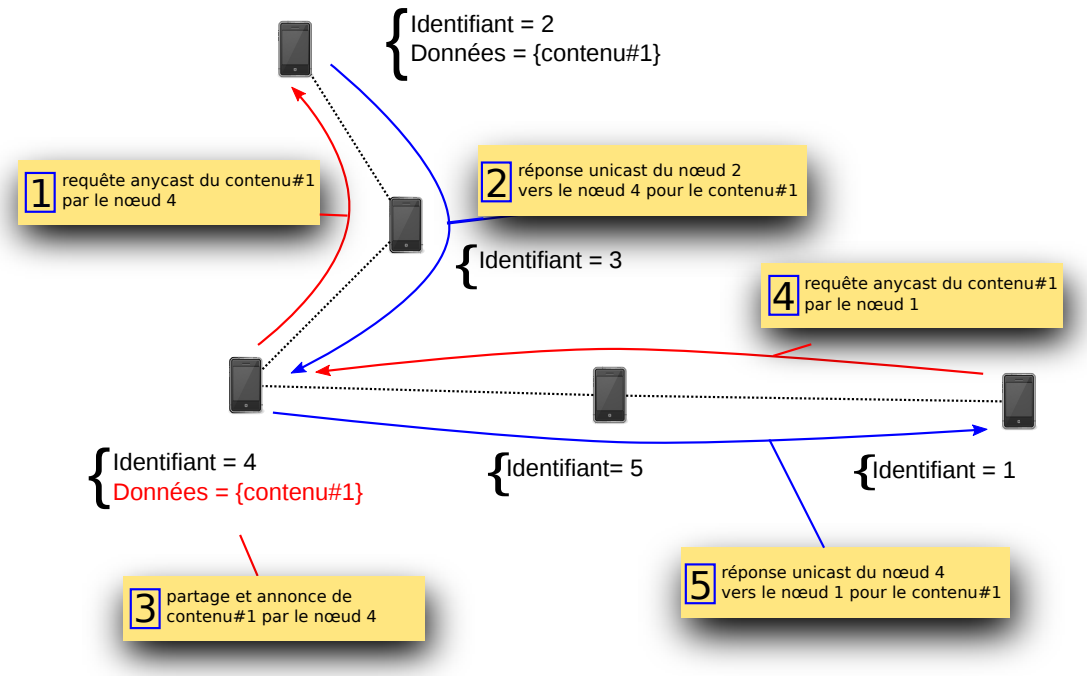


FIGURE 6.4 – Exemple de dissémination de données en utilisant l'approche basée sur le contenu.

initialement annoncée dans le FBC du nœud 2, puis diffusée au nœud 3, qui par transitivé l'annoncera dans son propre FBC. En comparant sa valeur de chemin pour le contenu *contenu#1* avec celle de ses voisins, le nœud 4 s'apercevra que le nœud 3 est le plus à même de délivrer la requête et va donc lui transférer une copie de cette requête. Le nœud 3 pourra délivrer la requête au nœud 2. Une réponse unicast à destination du nœud 4 sera générée. Une fois délivrée à l'émetteur de la requête, ce dernier annoncera sa capacité dans son FBC (étape 3). Le nœud 1 désirant alors obtenir ce même contenu émettra une requête qui sera dirigée vers le nœud 4 (étape 4). Le nœud 4 et le nœud 2 sont tous les deux capables de fournir la donnée, mais le nœud 4 étant plus proche en terme de nombre de sauts, il recevra en priorité la requête. Le mécanisme de vaccination supprimera la requête empêchant ainsi des copies inutiles. Une réponse sera finalement envoyée et délivrée au nœud 1 (étape 5).

6.4 Conclusion

Dans ce chapitre, nous avons présenté les trois modes de communication proposés par le système Nephila, à savoir 1) un mode de communication point-à-point, qui permet d'échanger des données avec un nœud particulier du réseau ; 2) un mode anycast, qui permet de transmettre un même message simultanément parmi un ensemble de nœuds qui offrent un service commun ; 3) un mode basé sur le contenu, qui permet de s'échanger

des messages en fonction du contenu de ceux-ci et pas de leur identifiant respectif. Dans les chapitres suivants de ce mémoire, nous présentons les résultats des évaluations de ces trois modes de communication.

Troisième partie

Expérimentation et évaluation

7

Évaluation de la communication point-à-point

Dans ce chapitre, nous présentons les résultats d'évaluation du mode de communication point-à-point implanté dans le système Nephila, ainsi que le scénario de simulation qui a été utilisé pour cette évaluation.

7.1 Scénario et paramètres de simulation

Le scénario défini se veut être aussi réaliste que possible. Il implique un nombre variable de piétons possédant chacun un smartphone et se déplaçant librement dans les rues d'une ville française de taille moyenne, à une vitesse variant entre 1 m/s et 1.6 m/s

Afin d'obtenir un schéma de mobilité réaliste pour un grand nombre de piétons, nous avons choisi d'utiliser des traces générées par le simulateur « ONE » [113]. En effet, il n'existe pas de jeux de traces récoltées impliquant un grand nombre de piétons se déplaçant dans une ville. Le schéma de mobilité retenu est un schéma dans lequel les nœuds se déplacent librement dans les rues de la ville. Nous avons ajouté le support de l'infrastructure à ce simulateur afin de pouvoir simuler des réseaux hybrides.

Nous avons choisi la ville de Vannes (présentée dans la figure 7.1) comme environnement de simulation. Cette ville s'étend sur environ $25\,km^2$ et est peuplée d'environ 50000 habitants. Les données cartographiques ont été extraites du service de cartographie collaboratif OpenStreetMap [114].

Les positions des points d'accès ont été extraites de données réelles publiées par la société Fon. Cette une société permet de partager des connexions à Internet dans le monde grâce à des routeurs Wi-Fi dédiés, ou via les points d'accès de certains fournisseurs d'accès à Internet, comme SFR en France.

Dans le scénario de simulation que nous considérons, 100 piétons utilisent leur smartphone pour communiquer. Ils émettent chacun 5 messages toutes les 20 minutes. Les smartphones des autres piétons se comportent uniquement en tant que relais de messages.

Nous avons effectué des simulations avec 100, 500, 1000 et 2000 piétons. Pour évaluer

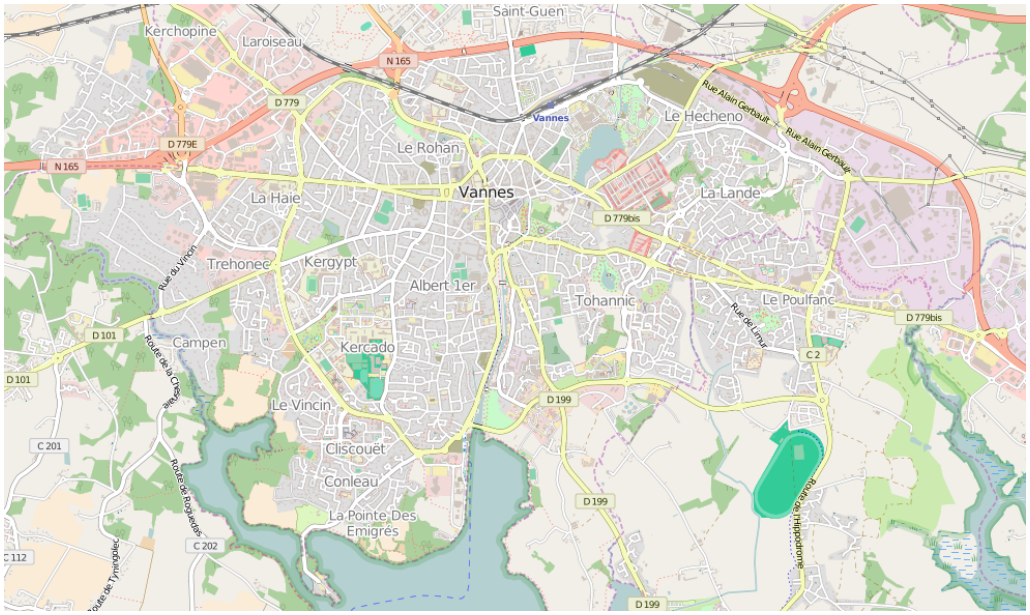


FIGURE 7.1 – Ville de Vannes

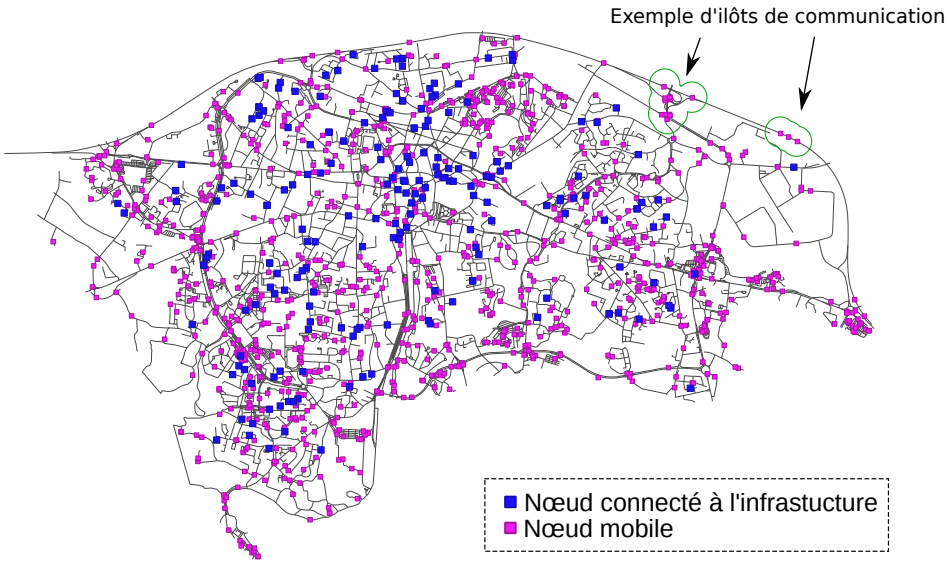
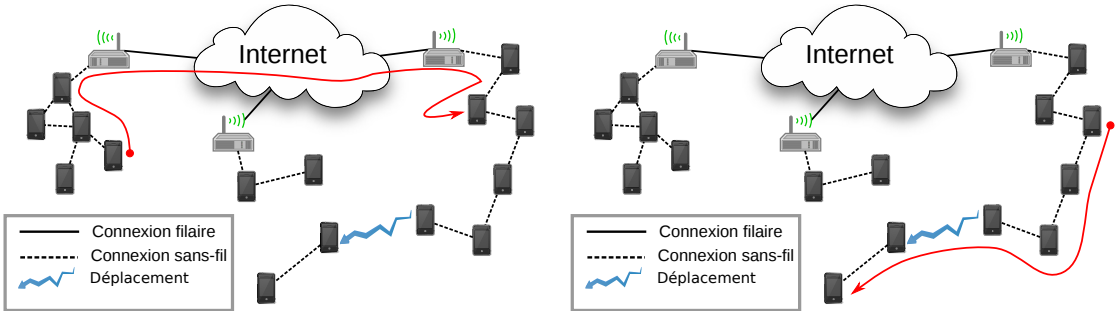


FIGURE 7.2 – Illustration de l'environnement de simulation.



(a) Utilisation de l'infrastructure (b) Utilisation de la mobilité

FIGURE 7.3 – Échange de messages

l'impact de la taille du cœur de réseau sur les performances de Nephila, nous avons réalisé des simulations avec 0, 100 et 200 points d'accès fixes. Ces points d'accès sont supposés être connectés à Internet. L'algorithme BTSA est utilisé pour acheminer les messages. Ces différentes configurations seront respectivement identifiées par BTSA 0, BTSA 100 et BTSA 200 par la suite. L'environnement de simulation est illustré dans la figure 7.2.

Nous supposons que les nœuds du réseau ont une taille de cache limitée (10 Mo pour les terminaux mobiles et 40 Mo pour les nœuds fixes). Nous considérons en outre que les messages ont un nombre de sauts maximum fixé à 30, et une taille maximale de 200 ko. La simulation dure 80 minutes. Pendant les 10 premières minutes, les terminaux fixes et mobiles réalisent une découverte de leur voisinage, et calculent et disséminent leur Filtre de Bloom de Chemin (FBC). Ce délai écoulé, les terminaux mobiles commencent à émettre des messages. L'émission des messages est arrêtée 10 minutes avant la fin de la simulation afin de permettre aux derniers messages émis d'arriver à leur destinataire. La portée de communication des interfaces sans fil est limitée à 50 mètres, et leurs débits de transmission à 10 Mbit/s. Le débit des liens filaires est limité à 20 Mbit/s. Les paramètres de configuration de Nephila utilisés pour les simulations sont donnés dans le tableau 7.1.

Ces simulations visent à comprendre d'une part comment se comporte notre système lors de l'échange de données dans les parties mobiles du réseau, et d'autre part d'étudier l'impact du nombre de points d'accès fixes constituant le cœur de réseau sur la délivrance des données. Dans cette optique, nous avons réalisé des simulations pour les protocoles PRoPHET [115] et Spray-and-Wait [57] dans un environnement dépourvu de points d'accès fixes. Ces deux protocoles ont été présentés dans le chapitre 2. Le nombre de copies maximum pour Spray-and-Wait a été fixé à 10. PRoPHET a quant à lui été configuré avec ses paramètres par défaut ($Pinit$: 0.75, β : 0.25, γ : 0.98) [115]. Les probabilités de délivrance calculées par PRoPHET peuvent être assimilées aux valeurs de chemin estimées dans Nephila. Une comparaison de ces protocoles avec notre solution est fournie ci-après.

Paramètre	Valeur
FBC	
Période de décroissance	20 s
Période de renforcement	20 s
Facteur de renforcement	0.2
Période de diffusion	30 s
VC min pour piétons	0.1
VC min pour un point d'accès	0.3
Nombre d'éléments	600
Nombre de fonctions de hachage	2
BTSA	
Seuil de transfert	0.1
Nombre de copies	10
Cyclon	
Taille du voisinage	13
Taille des échantillons	3
Période d'échange	120 s

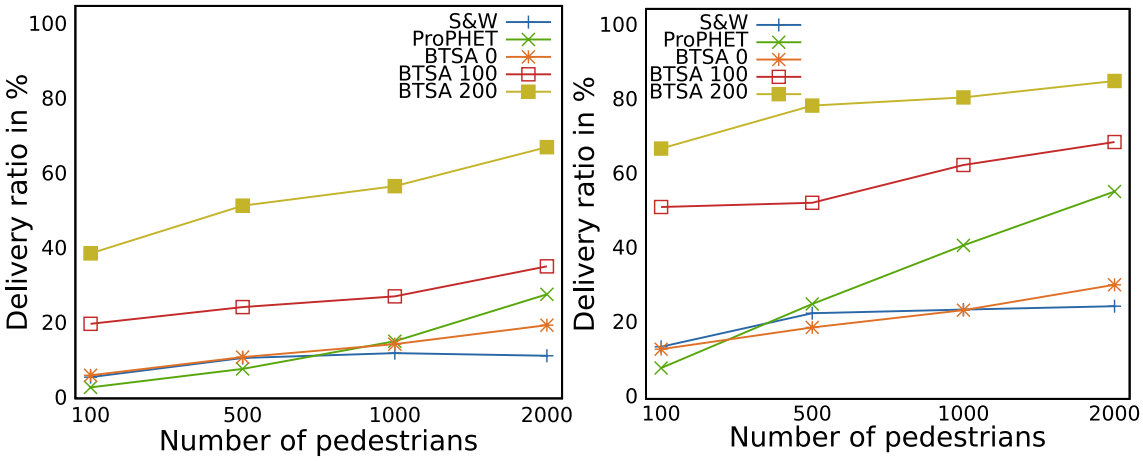
TABLE 7.1 – Paramètres de simulation.

7.2 Résultats de simulation

Les performances de Nephila et BTSA sont comparées avec celles de PRoPHET et Spray-and-Wait suivant trois métriques, à savoir le temps moyen d'acheminement des messages, le taux de messages délivrés, et la charge du réseau. Le taux de délivrance est le nombre de messages délivrés divisé par le nombre total de messages envoyés par les 100 émetteurs. Il reflète la capacité du système à délivrer les messages à leur destination avant leur date d'expiration. La charge du réseau est le nombre total de messages échangés entre les nœuds durant la simulation. Dans les parties sans fil du réseau, seuls les messages transportant des données ont été comptabilisés, puisque les implantations de PRoPHET et de Spray-and-Wait dans le simulateur « the One »n'intègrent pas de processus de découverte du voisinage.

Le taux de délivrance pour un message est le nombre de messages délivrés divisé par le nombre de messages émis (un message étant délivré au plus une fois). Chaque message possède une date d'expiration après laquelle il est supprimé du cache de tous les noeuds. Le taux de délivrance permet donc d'évaluer la capacité du système à acheminer et délivrer des messages à leur destinataire dans un temps imparti.

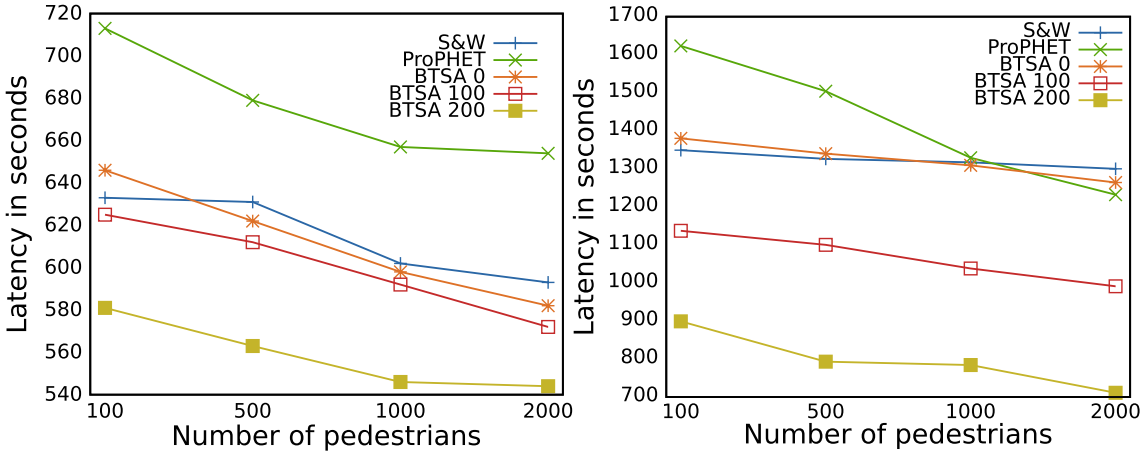
Les figures 7.4, 7.5, et 7.6 montrent, en fonction du nombre de piétons, le temps moyen et le nombre moyen de sauts nécessaires pour transférer un message à sa destination, ainsi que le pourcentage de messages délivrés avec succès. Les figures 7.7, 7.8, et 7.9 montrent le nombre de messages disséminés dans les parties sans fil et filaire du réseau. BTSA 0 (BTSA sans point d'accès) et Spray-and-Wait ont des performances similaires



(a) Date d'expiration de 20 minutes

(b) Date d'expiration de 60 minutes

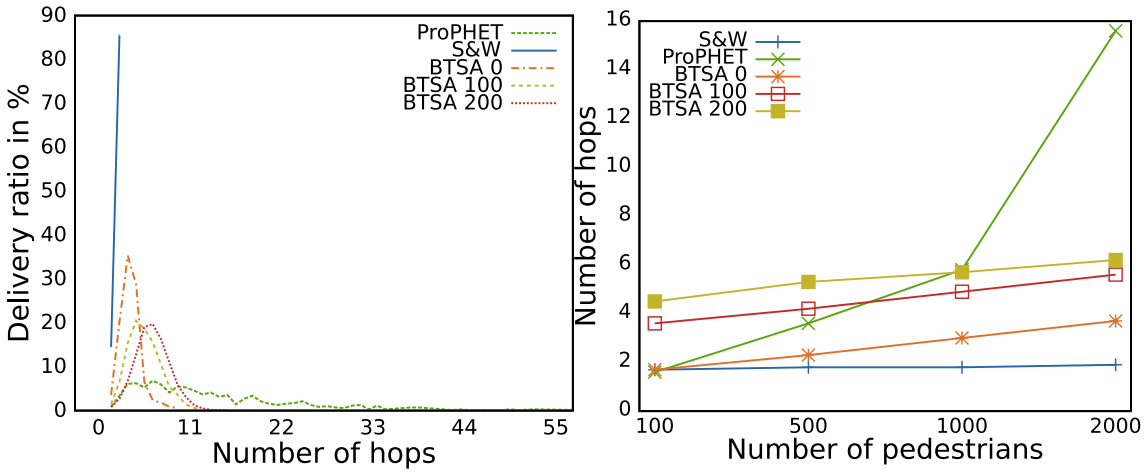
FIGURE 7.4 – Taux de délivrance des messages



(a) Date d'expiration de 20 minutes

(b) Date d'expiration de 60 minutes

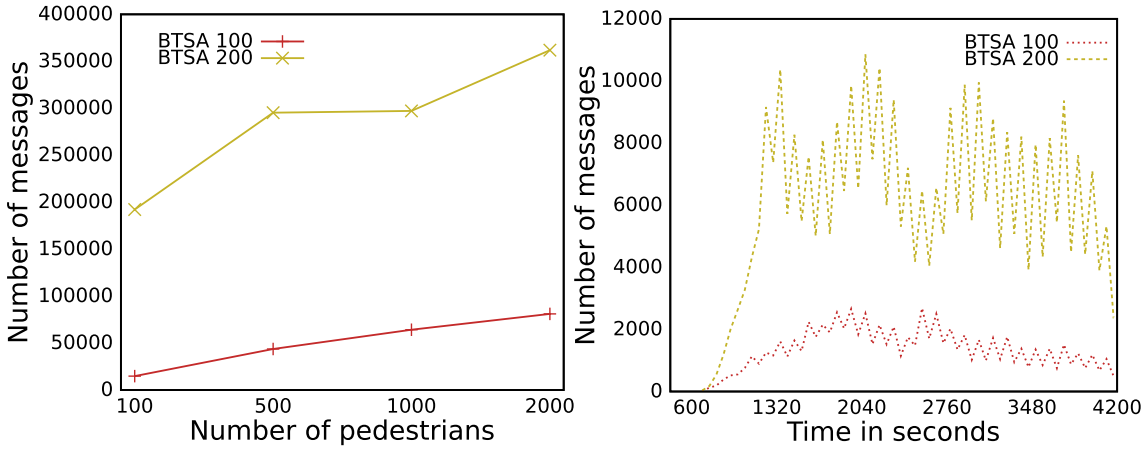
FIGURE 7.5 – Latence des messages



(a) Distribution du nombre de sauts requis pour 2000 piétons

(b) Nombre de sauts moyen requis en fonction du nombre de piétons

FIGURE 7.6 – Nombre de sauts requis pour des messages ayant une date d'expiration de 20 minutes



(a) Nombre de messages transmis en fonction du nombre de piétons

(b) Nombre de messages transmis pour 2000 piétons en fonction du temps

FIGURE 7.7 – Charge dans la partie infrastructure du réseau

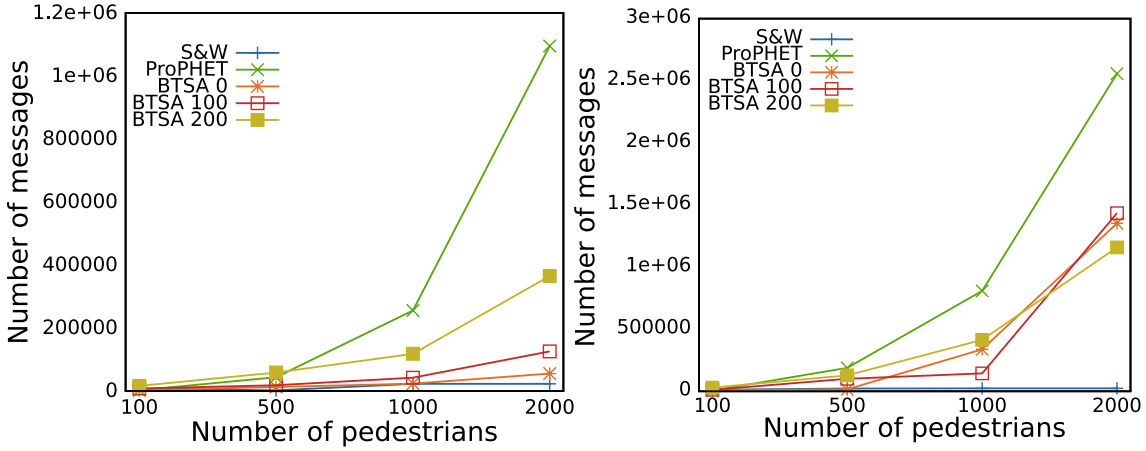
en ce qui concerne le temps moyen d'acheminement, le nombre de sauts et nombre des messages. BTSA 0 et Spray-and-Wait sont tous les deux meilleurs que PRoPHET de 100 à 500 piétons. Ceci est dû au fait que dans des réseaux peu denses, les contacts radio ne sont pas fréquents, et par conséquent les probabilités de délivrance estimées par PRoPHET ne sont pas pertinentes, beaucoup de messages étant avec le protocole PRoPHET délivrés par l'émetteur lui-même. BTSA 0 et Spray-and-Wait ne présentent pas ces limitations. Ils peuvent en effet transférer des copies de messages à des nœuds n'ayant pas été identifiés comme étant de bons transporteurs, qui à leur tour vont transférer une copie à d'autres nœuds, augmentant ainsi les probabilités de rencontres ultérieures avec de bons relayeurs. Le nombre de contacts augmentant avec le nombre de piétons, PRoPHET fournit un taux de délivrance des messages supérieur à celui de BTSA 0 et de Spray-and-Wait entre 1000 et 2000 piétons (voir figure 7.4). Cependant, il présente l'inconvénient d'augmenter le nombre de copies de messages disséminées dans le réseau de manière significative (voir figure 7.7). En étant contraint par le nombre de copies de messages à transmettre, Spray-and-Wait ne bénéficie pas autant que PRoPHET de l'augmentation du nombre de piétons (voir figure 7.9). Pour 2000 piétons, BTSA 0 offre de meilleures performances que Spray-and-Wait. Le réseau est plus dense et donc lui permet de calculer des scores plus pertinents, créant ainsi des chemins en plusieurs sauts plus efficaces (voir figures 7.4, 7.6,7.5).

7.2.1 Utilisation de l'infrastructure

Les figures 7.4 et 7.5 et montrent l'impact très significatif du cœur du réseau constitué par les points d'accès fixes sur le taux de délivrance et le temps moyen d'acheminement des messages, et cela sans introduire de surcoût majeur dans les parties sans fil du réseau. Un nombre plus important de nœuds fixes permet de couvrir une zone géographique plus large, et par conséquent augmente la probabilité de délivrer un message à une destination éloignée tout en réduisant le temps de délivrance. Ces effets sont atténués graduellement lorsque la densité du réseau et la durée de vie des messages augmentent. En effet, les opportunités de contact entre nœuds mobiles augmentant, plus de messages sont délivrés par les nœuds mobiles sans avoir recours à l'infrastructure. La figure 7.7 montre qu'un nombre important de messages est transmis à travers le cœur du réseau.

7.2.2 Surcoût dû à l'échange des FBC

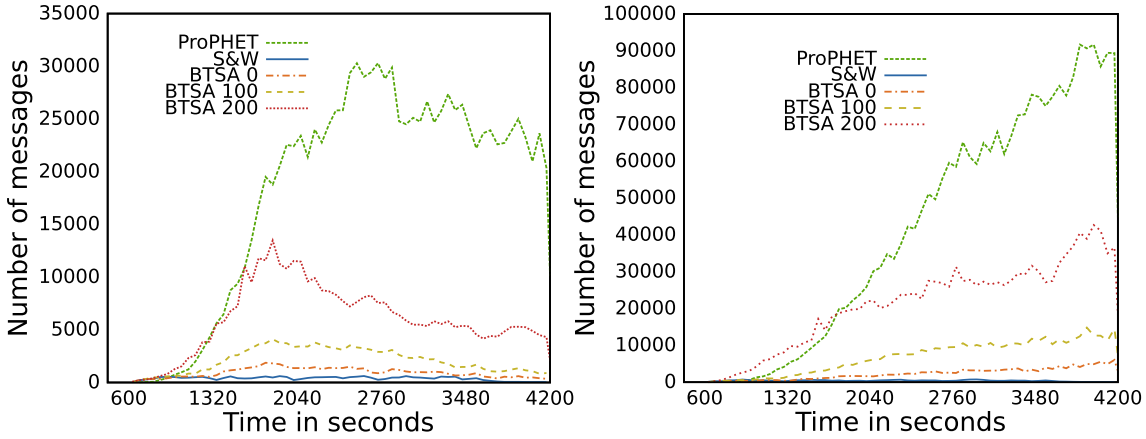
La figure 7.10 montre le surcoût introduit par l'échange de FBC dans les parties sans fil du réseau. Ce surcoût dépend du nombre de contacts et de leur durée. Il est donc plus important dans un réseau dense que dans un réseau clairsemé. Pour les simulations impliquant 2000 piétons, la moyenne du débit de données dans les parties sans fil du réseau pendant la simulation représente environ 2 kbit/s par nœud. Dans les parties filaires il est d'environ 305 Mo pour BTSA 100 et d'environ 629 Mo pour BTSA 200, ce qui représente en moyenne 770 bits/s par nœud fixe. Les nœuds fixes ayant un nombre maximal de voisins borné ainsi qu'une période d'échange prédéterminée, l'augmentation de leur



(a) Date d'expiration de 20 minutes

(b) Date d'expiration de 60 minutes

FIGURE 7.8 – Charge dans les parties sans fil du réseau



(a) Date d'expiration de 20 minutes

(b) Date d'expiration de 60 minutes

FIGURE 7.9 – Nombre de messages dans les parties sans fil du réseau en fonction du temps, pour 2000 piétons

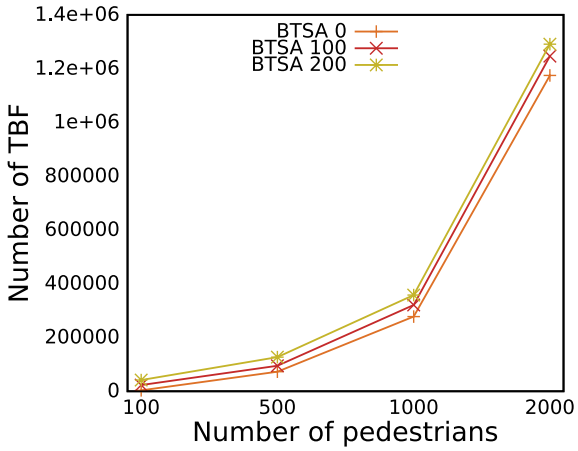


FIGURE 7.10 – Nombre de FBC échangés dans les parties sans fil du réseau

nombre ne fera pas varier le surcoût introduit par les FBC sur chaque nœud. Les nœuds échangent leur FBC uniquement avec leurs voisins directs, la durée de vie des messages n'a donc aucune conséquence sur leur nombre.

7.2.3 Influence de la date d'expiration

Augmenter la durée de vie des messages de 20 à 60 minutes permet d'obtenir un meilleur taux délivrance des messages (figures 7.4a et 7.4b), mais se traduit par un délai de délivrance moyen plus élevé (figures 7.5a et 7.5b). Le nombre total de messages (figures 7.8a et 7.8b) augmente aussi de façon continue pour une durée de vie de 60 minutes, étant donné que les messages ne sont pas supprimés du cache des nœuds et donc continuent à être répliqués dans le réseau pendant la simulation.

7.3 Conclusion

Les résultats de simulation ont montré que dans le scénario considéré, notre proposition permettait de délivrer plus de messages en moins de temps que des solutions existantes tels que *PRoPhet* ou *Spray and Wait*, tout en réduisant la charge réseau des terminaux mobiles. L'utilisation de l'infrastructure permet d'améliorer significativement les performances en termes de taux de délivrance ainsi que de délai de délivrance.

8

Évaluation de la communication anycast

Dans ce chapitre nous présentons les évaluations du mode de communication anycast. Nous considérons le même environnement que dans le chapitre précédent.

8.1 Scénario et paramètres de simulation

Sur le modèle du scénario utilisé pour évaluer le mode communication point-à-point, un ensemble variable de points d'accès sont répartis dans la ville. Pour des raisons de réalisme, nous avons divisé cet ensemble en 4 groupes distincts, qui sont supposés être gérés par différents opérateurs (présenté dans la figure 8.1). La communication entre un nœud mobile et un point d'accès ne peut être établie que si le porteur du nœud dispose d'un compte chez le fournisseur gérant ce point d'accès. Une fois une connexion établie avec un point d'accès, un nœud pourra agir en tant que relais Nephila, et transmettre des messages pour le compte de n'importe quel autre nœud Nephila sur la partie filaire ou la partie infrastructure.

Dans cette simulation (présenté dans la figure 8.2), un sous-ensemble de piétons (100) émet des messages contenant des requêtes vers des serveurs fixes sur Internet. Lorsqu'une requête est reçue par un serveur, une réponse est retournée en « unicast » à destination de l'émetteur de la requête. Ces requêtes sont émises en utilisant le mode de communication anycast. Les nœuds disposant d'une connexion à Internet et d'une connexion sans fil annoncent leur capacité à relayer un message connecté à Internet en diffusant une annonce *Internet access* dans leur FBC. Les requêtes émises vers un nœud sur Internet ne faisant pas partie du réseau Nephila sont encapsulées dans des messages Nephila avec pour destination le service anycast *Internet access*.

La durée totale de la simulation est fixée à 103 minutes. Les 3 premières minutes permettent de découvrir le voisinage de chaque nœud, ainsi que de disséminer les Filtres de Bloom de Chemin (FBCs). Après cette période, certains terminaux mobiles démarrent une application émettant des messages pendant une heure. L'émission de messages est arrêtée environ 40 minutes avant la fin de la simulation afin de permettre aux derniers messages émis d'arriver à leur destinataire, puis de recevoir une réponse. Ceci permet ainsi de mesurer combien de requêtes et réponses ont finalement été transmises.

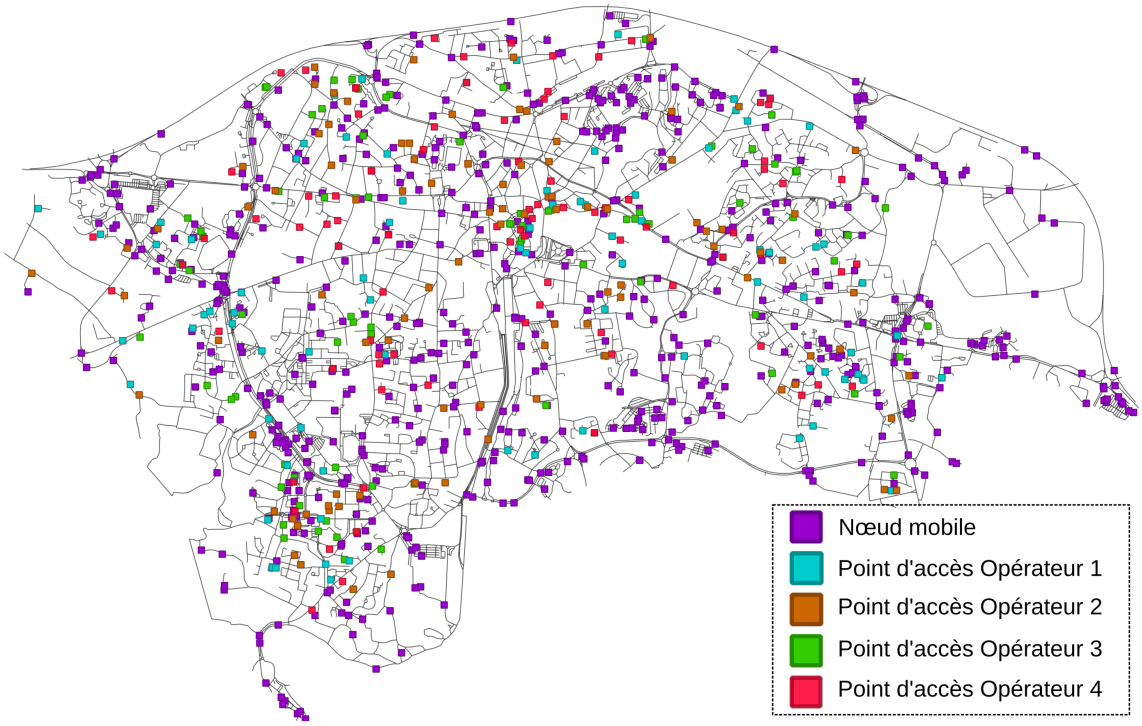


FIGURE 8.1 – Environnement de simulation

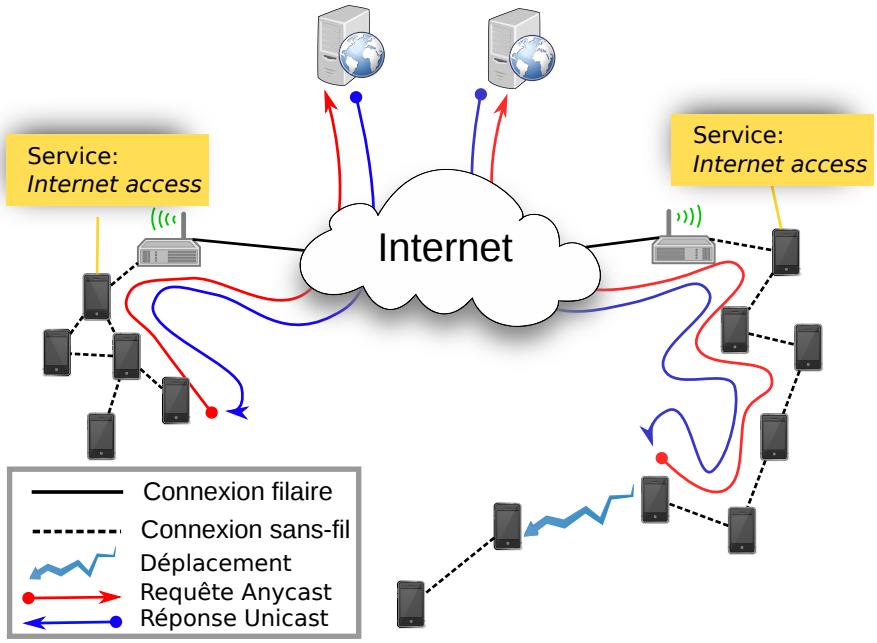


FIGURE 8.2 – Scénario Anycast

Dans cette simulation, les piétons se déplacent à une vitesse variant entre 1 m/s et 1.6 m/s. Les simulations ont été effectuées avec 100, 500, 1000, 2000, 4000 piétons et 100, 200, 400, 800 points d'accès. La durée de vie de chaque message est fixée à 20 minutes. La portée de communication des interfaces sans fil est limitée à 50 mètres avec un débit maximal de 10 Mbit/s. Le débit des interfaces filaires est limité à 20 Mbit/s.

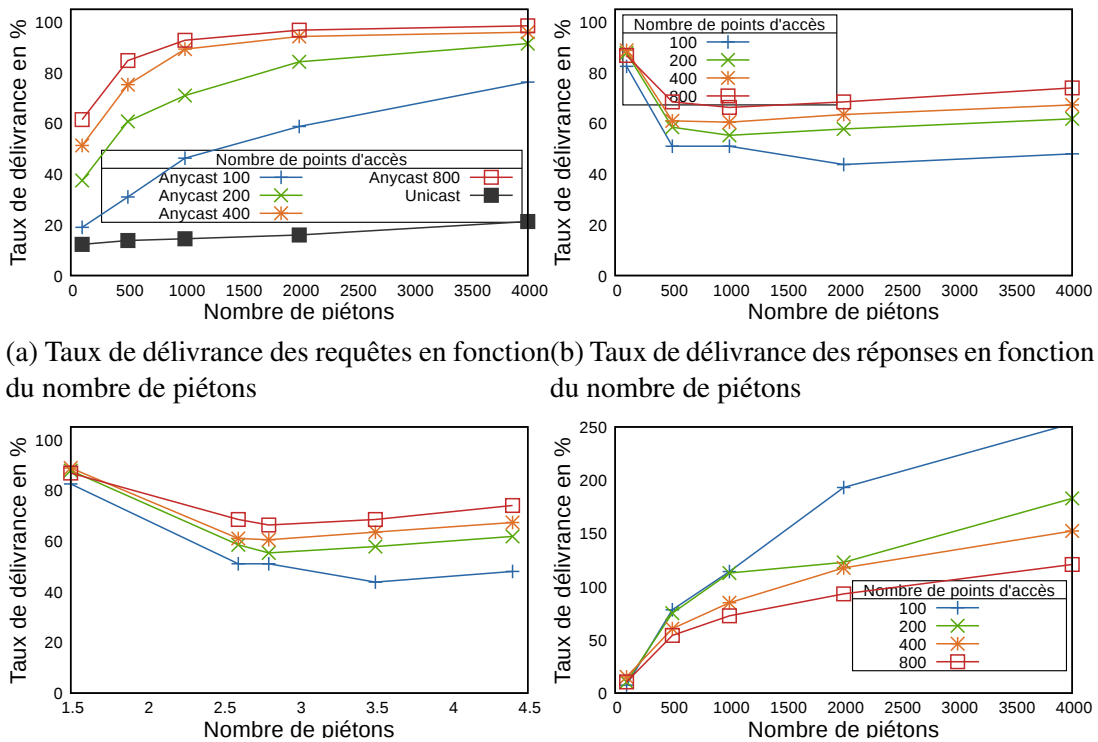
Afin de confirmer l'intérêt du mécanisme de vaccination présenté dans le chapitre 5.6, les simulations ont été effectuées avec et sans ce mécanisme.

Les paramètres de Nephila utilisés pour la simulation sont les mêmes que ceux définis dans le chapitre 7 (tableau 7.1). Le nombre d'éléments maximum pour la table de vaccination est de 1000.

8.2 Résultats de simulation

La figure 8.3 montre les résultats obtenus pour la délivrance des requêtes et des réponses. Un message envoyé en anycast est considéré comme délivré s'il a atteint au moins un des destinataires. Lorsque le nombre de points d'accès et de piétons augmente, le délai et le taux de délivrance des requêtes augmentent. Un maximum de 98% de requête sont délivrées pour 800 points d'accès et 4000 piétons. Ce résultat est cohérent avec ce qui est attendu dans ces circonstances. En effet, la probabilité de rencontrer des points d'accès et de trouver un chemin menant à la destination visée est plus élevée dans un réseau dense que dans un réseau clairsemé. Ceci est confirmé par les résultats présentés dans la figure 8.4. Quand le nombre de nœuds mobiles est faible (100), la plupart des requêtes sont délivrées directement (i.e., en un saut). Cette proportion croît lorsque le nombre de points d'accès augmente. Quand le nombre d'objets mobiles et de points d'accès augmente, on peut observer que la plupart des requêtes sont délivrées en 4 sauts ou moins. Par exemple, pour 4000 piétons et 800 points d'accès, 80% des requêtes sont délivrées en au plus quatre sauts. De plus, pour 800 points d'accès, le temps de délivrance de la plupart des requêtes est compris entre 0 et 200 secondes (voir figure 8.4b). Le nombre des requêtes délivrées dans une courte période de temps augmente significativement avec le nombre de terminaux mobiles. La figure 8.3a fournit une comparaison entre l'utilisation des mécanismes anycast et unicast pour le transfert des messages de requêtes par Nephila pour 800 points d'accès.

Concernant le transfert des réponses, il peut être observé que 90% d'entre elles sont délivrées lorsque le nombre d'objets mobiles est réduit à 100. Ce pourcentage diminue lorsque ce nombre passe à 500, puis augmente avec un nombre plus important de piétons (voir figure 8.3b). L'explication réside dans le fait que dans un RHCI composé de seulement 100 piétons, les requêtes sont le plus souvent délivrées directement par l'émetteur du message lui-même, ce qui lui permet ainsi de recevoir directement la réponse. C'est pourquoi le délai de délivrance est aussi très faible pour 100 piétons, et qu'il augmente lorsque plus de piétons sont impliqués dans le transport des messages (voir figures 8.4a and 8.3d). Ce délai augmente de façon plus significative encore lorsque le nombre de points d'accès est faible. Les chemins entre la source et la destination sont en effet plus



(a) Taux de délivrance des requêtes en fonction du nombre de piétons

(b) Taux de délivrance des réponses en fonction du nombre de piétons

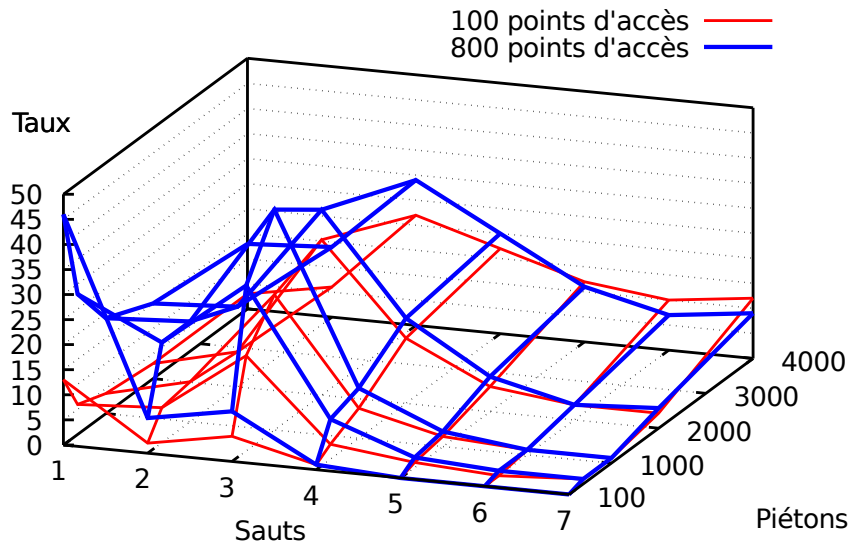
(c) Taux de délivrance des réponses en fonction du nombre moyen de sauts

(d) Latence moyenne des réponses en fonctions du nombre de piétons

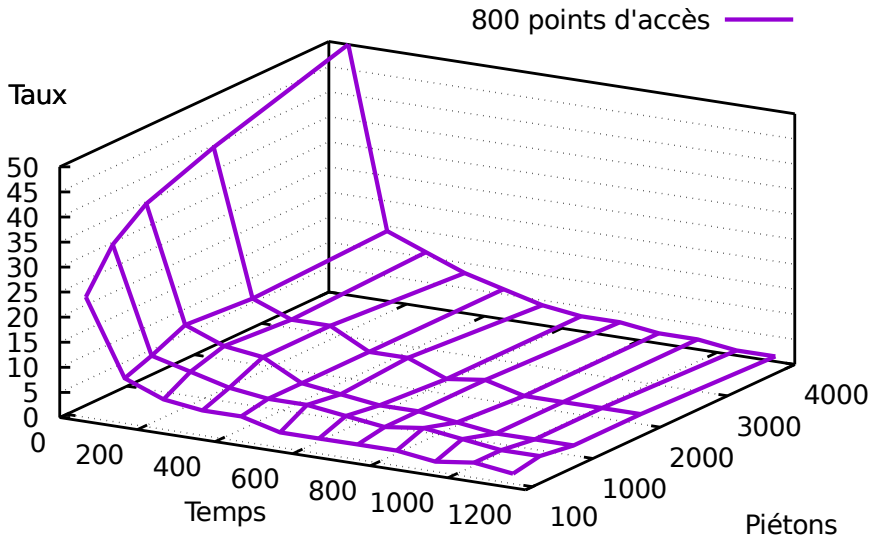
FIGURE 8.3 – Taux et latence de délivrance

discontinus dans un RHCI composé d'un faible nombre de points d'accès que dans un RHCI qui en comprend un grand nombre.

Mécanisme de vaccination Le coût du mécanisme de vaccination sur le volume de messages disséminés dans la partie mobile du RHCI est présenté dans la figure 8.5. L'efficacité de ce mécanisme dépend du nombre de messages délivrés, du nombre de copies de messages stockées sur chaque objet, et du nombre de contacts entre objets, étant donné que chaque contact est une opportunité pour que deux objets échangent leurs tables de vaccination. L'efficacité augmente lorsque le nombre d'objets augmente. Ce mécanisme permet dans ce scénario d'économiser jusqu'à 175 Go pour 4000 piétons et 800 points d'accès. La charge additionnelle générée par l'échange des tables de vaccination est présentée dans la figure 8.5b. Dans le pire cas, cela représente environ 8.5 kbit/s par objet (soit 384 Mo pendant toute la durée de la simulation) ce qui est plutôt faible comparé aux capacités de communication des interfaces Wi-Fi.

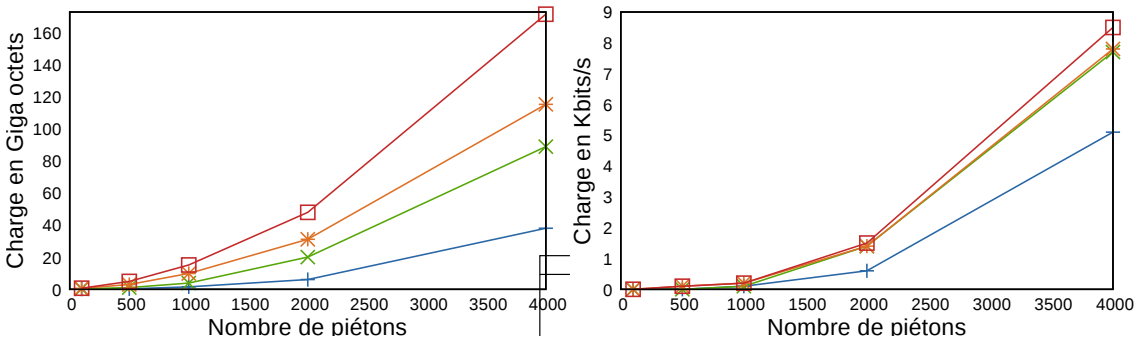


(a) En fonction du nombre de sauts et de piétons pour 100 et 800 point d'accès



(b) En fonction de la latence et du nombre de piétons en utilisant 800 point d'accès.

FIGURE 8.4 – Distribution du nombre de requêtes délivrées



(a) Nombre de messages supprimés en fonction du nombre de piétons

(b) Charge des messages de vaccination en fonction du nombre de nœuds

FIGURE 8.5 – Mécanisme de vaccination

8.3 Conclusion

Les résultats confirment l'efficacité de notre mécanisme de transfert de messages en mode anycast, étant donné que 98% des requêtes sont délivrées alors que seulement 20% le sont avec le mécanisme unicast. On peut en conclure que le mécanisme de transfert des messages par anycast implémenté dans Nephila est efficace pour transmettre avec un faible nombre de sauts des messages vers des points d'accès, avec un fort taux de délivrance et une faible latence.

9

Évaluation de la communication basée sur le contenu

Dans ce chapitre, nous présentons les évaluations du mode de communication basée sur le contenu. Nous considérons le même environnement que dans le chapitre 7.

9.1 Scénario et paramètres de simulation

Notre scénario met en œuvre un cas de partage et de déchargement de contenu disponible sur Internet vers un réseau sans fil à l'échelle d'une ville de taille moyenne.

De façon similaire à l'évaluation pour le mode de communication unicast et anycast, le mode de communication basé sur le contenu a été évalué en utilisant le simulateur réseau « ONE ». Les simulations ont été effectuées avec une population de 100, 500, 2000 et 4000 piétons, ainsi que 100, 200, 400 et 800 points d'accès Wi-Fi. Les points d'accès sans-fil ont une position fixe, et sont interconnectés par Internet.

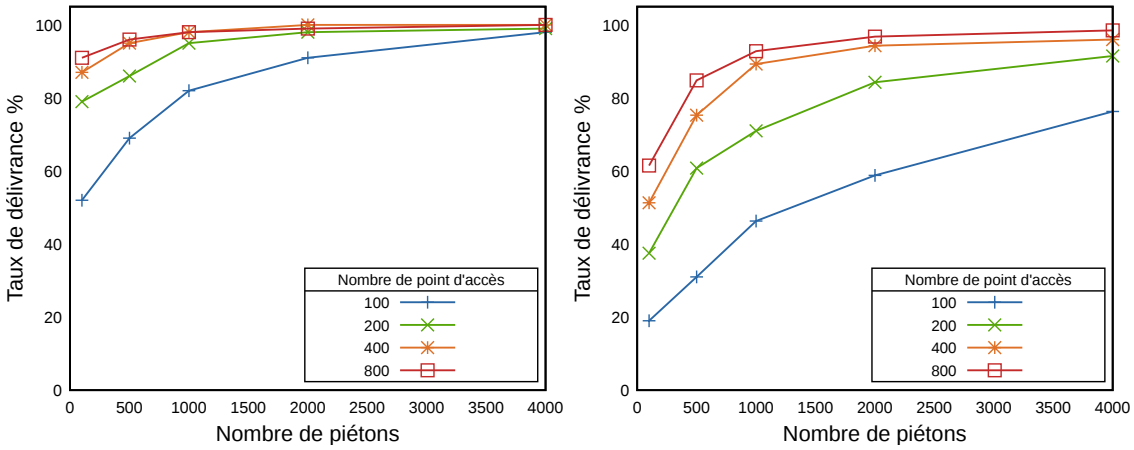
Dans cette simulation, 100 piétons émettent des messages contenant des requêtes pour un type de contenu particulier, choisi parmi 100 contenus générés aléatoirement. Lorsqu'une requête est reçue par un nœud hébergeant ce contenu, un message de réponse est alors créé, avec pour destination l'émetteur de la requête. Une fois le message de réponse reçu, un nœud devient capable de répondre directement à un autre nœud émettant une requête pour le même contenu.

Les performances du mode de communication basé sur le contenu sont comparées avec celle de l'approche classique, dans laquelle seul le nœud hébergeant initialement un certain contenu pourra émettre un message de réponse, et cela même si d'autres nœuds ont déjà récupérés ce contenu.Pour cette comparaison les requêtes sont transmises en utilisant la méthode de communication anycast. Seul le traitement des réponses diffère. Dans la suite de ce chapitre, nous appellerons notre approche l'approche « avec partage » et l'approche classique l'approche « sans partage ».

Le mécanisme de vaccination est utilisé pour supprimer au plus tôt les copies des requêtes/réponses qui ont été délivrées à leur destinataire.

Les paramètres de Nephila utilisés pour la simulation sont les mêmes que ceux définis dans le chapitre 7 (tableau 7.1).

9.2 Résultats de simulation



(a) Approche avec partage

(b) Approche sans partage

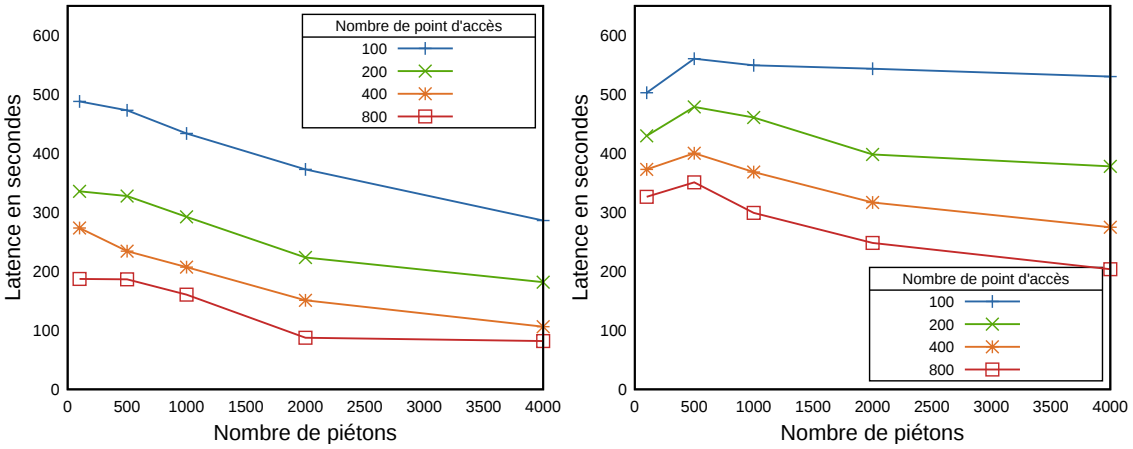
FIGURE 9.1 – Taux de délivrance des requêtes

Ls figure 9.1 montre les taux de délivrance avec et sans partage, en fonction du nombre de piétons et de points d'accès. Nous considérons qu'une requête est délivrée lorsqu'au moins un nœud capable de répondre à cette requête reçoit le message.

Le taux de délivrance pour chaque résultat avec partage est supérieur à son équivalent sans partage. Pour 4000 piétons, le mode de communication basé contenu présente un taux de délivrance proche de ou égal à 100%, et cela quel que soit le nombre de points d'accès.

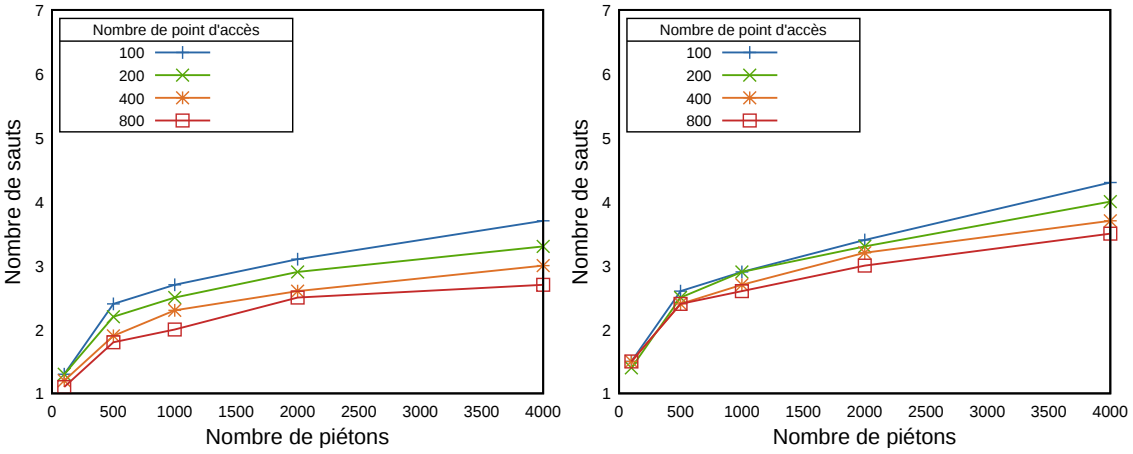
Grâce à l'approche avec partage, un nœud ayant reçu une réponse va annoncer être capable de répondre à des requêtes pour ce même contenu dans son FBC. Plus le contenu est populaire, plus il sera facile de trouver un nœud proche capable de répondre à une requête pour ce contenu. La courbe du taux de délivrance des requêtes sans partage est similaire à celle présentée dans la figure 8.3a pour le scénario évaluant le mode anycast. En effet, dans les deux scénarios les requêtes sont transmises en utilisant la méthode anycast.

Les latences moyennes d'arrivée des requêtes sont présentées dans la figure 9.2. Le nombre de points d'accès influence la latence plus fortement que le nombre de piétons, que ce soit avec l'approche avec partage ou sans partage. En effet les points d'accès disposent tous d'un accès direct aux contenus, ce qui leur permet de répondre directement aux requêtes. L'approche avec partage offre en revanche une latence plus faible que l'approche classique. Dans l'approche classique, la latence augmente légèrement lorsque le nombre de nœuds augmente de 100 à 500. Ceci est dû à une forte augmentation du taux



(a) Approche avec partage (b) Approche sans partage

FIGURE 9.2 – Latence moyenne d'arrivée des requêtes



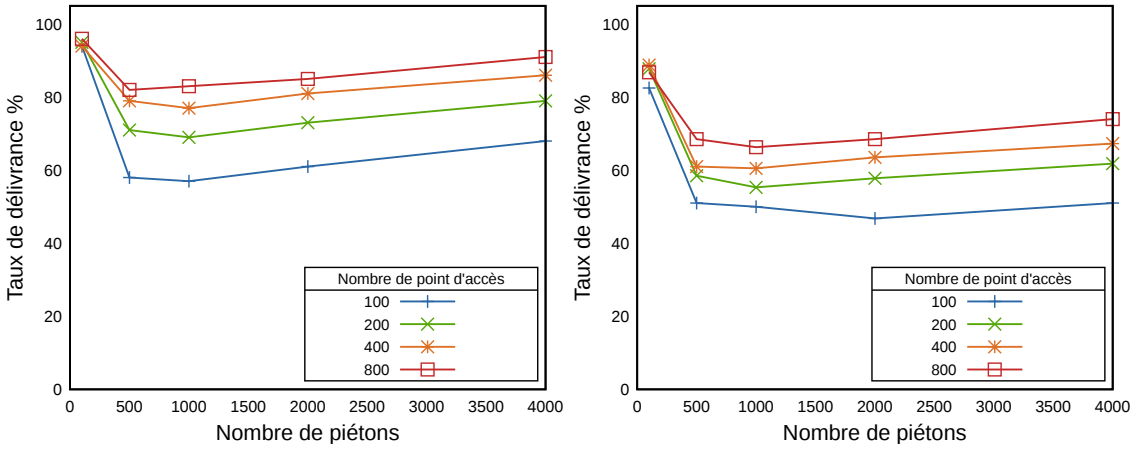
(a) Approche avec partage (b) Approche sans partage

FIGURE 9.3 – Nombre de sauts moyen des requêtes

de délivrance (voir figure 9.1b) grâce à l'augmentation du nombre du nombre de contacts permettant d'atteindre des nœuds plus éloignés en plusieurs sauts (voir figure 9.3b).

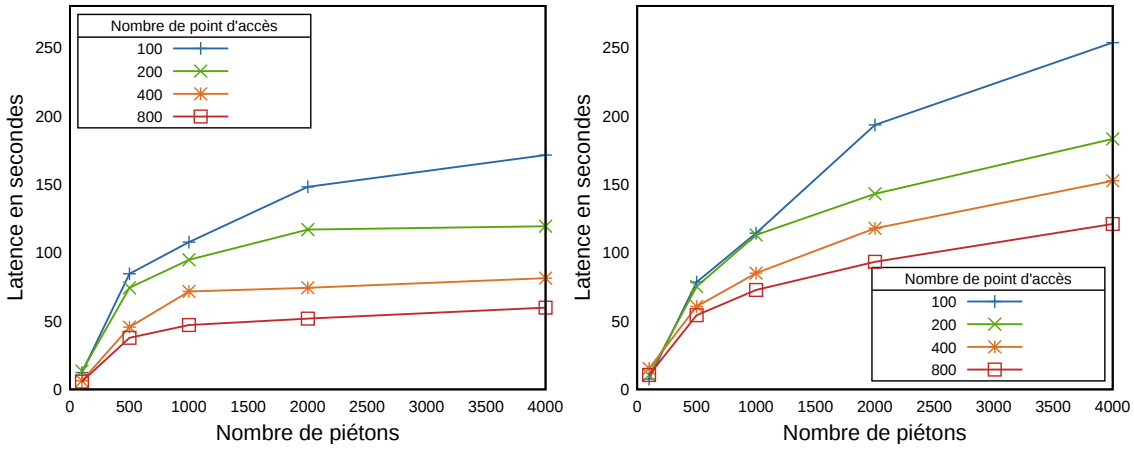
Le nombre moyen de sauts nécessaires pour qu'une requête soit délivrée est présenté dans la figure 9.3. Pour les deux approches, le nombre de sauts augmente, lorsque le nombre d'objets augmente et décroît lorsque le nombre de points d'accès augmente. Les nœuds mobiles servant de transporteurs intermédiaires, ils permettent d'atteindre des nœuds éloignés des points d'accès en plusieurs sauts. Plus le nombre de points d'accès augmente, moins il est nécessaire d'utiliser des nœuds mobiles comme intermédiaires.

Le taux de délivrance des réponses est présenté dans la figure 9.4. Cette figure montre un taux de délivrance de 96% avec partage, et de 89% sans partage pour 100 piétons, puis un taux inférieur pour 500 piétons. Ceci est dû au fait que lorsque le nombre de piétons est très faible, la plupart des requêtes sont délivrées en 1 saut (figure 9.3), ce qui permet à



(a) Approche avec partage (b) Approche sans partage

FIGURE 9.4 – Taux de délivrance des réponses



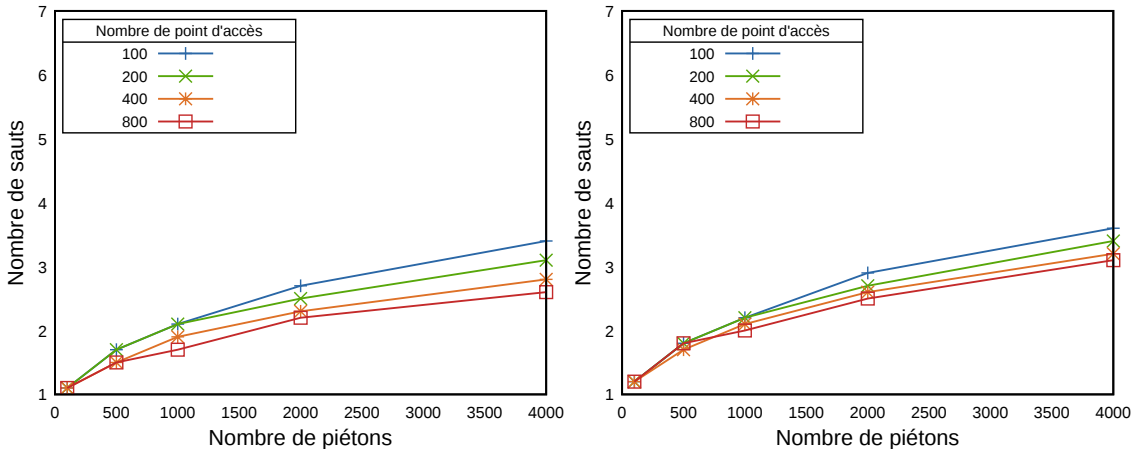
(a) Approche avec partage (b) Approche sans partage

FIGURE 9.5 – Latence moyenne d'arrivée des réponses

la réponse d'être délivrée immédiatement. Ceci est cohérent avec la latence moyenne des réponses, qui est de l'ordre de la dizaine de secondes quand le système est composé de seulement 100 piétons (voir figure 9.2).

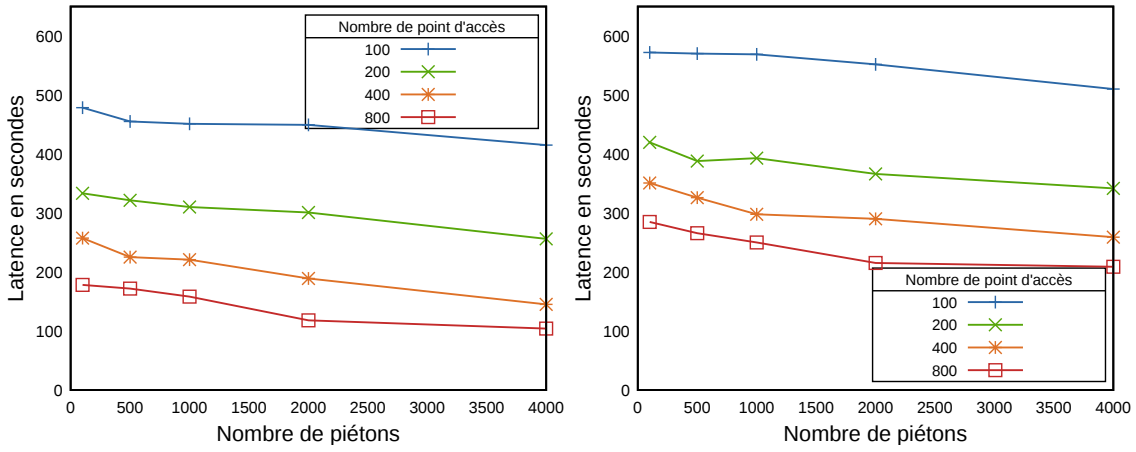
L'évolution de la latence d'arrivée des réponses montrée dans les figures 9.5b et 9.5a croît avec le nombre de points d'accès et d'objets. Ceci s'explique par le fait que l'augmentation du nombre d'objets et de points d'accès aide à délivrer des messages pour des destinataires plus éloignés de l'émetteur. La figure 9.6 montre le nombre de sauts moyen nécessaires pour acheminer une réponse à son destinataire. Elle montre également une augmentation du nombre de sauts quand le nombre d'objets mobiles et de points d'accès augmente.

L'évolution de la moyenne ainsi que le nombre de sauts des couples requête/réponse sont présentés dans les figures 9.7, et 9.8. Ces courbes montrent le temps total moyen



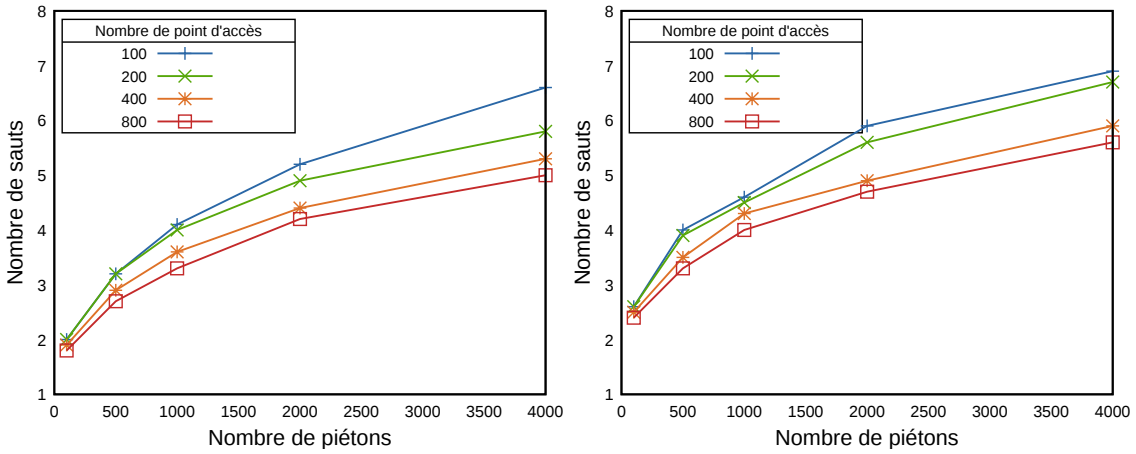
(a) Approche avec partage (b) Approche sans partage

FIGURE 9.6 – Nombre de sauts moyen des réponses



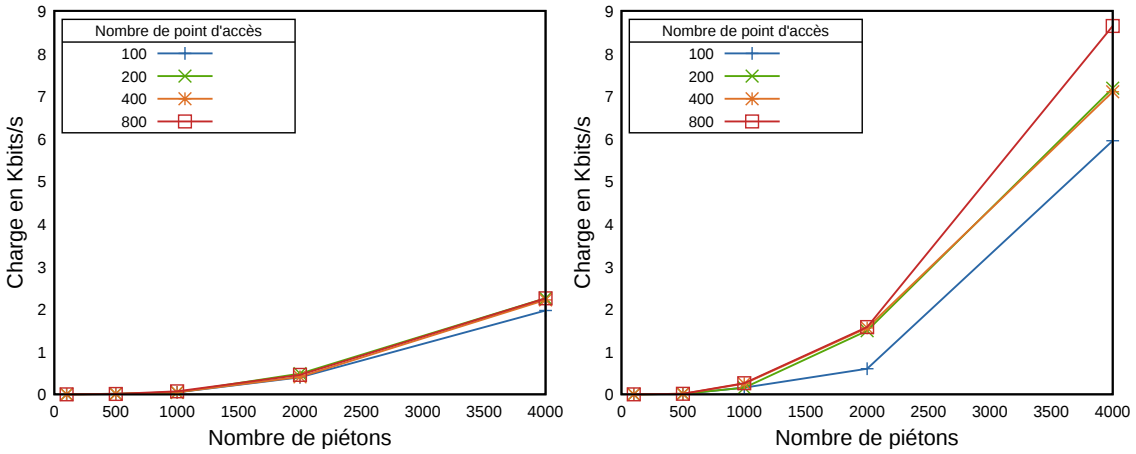
(a) Approche avec partage (b) Approche sans partage

FIGURE 9.7 – latence moyenne des couples requête/réponse



(a) Approche avec partage (b) Approche sans partage

FIGURE 9.8 – Nombre de sauts moyen des couples requête/réponse



(a) Approche avec partage (b) Approche sans partage

FIGURE 9.9 – charge moyenne par nœud par seconde

requis pour qu'un client obtienne le contenu demandé. Le nombre de points d'accès joue un rôle majeur dans la diminution de la latence. Par exemple, pour 4000 piétons et 800 points d'accès la latence passe de 200 à 100 secondes.

La charge moyenne par piéton par seconde est présentée dans la figure 9.9. L'utilisation du partage permet de réduire significativement la charge sur chaque noeud du système lorsque le nombre de nœuds est supérieur ou égal à 2000. Pour 4000 piétons et 800 points d'accès, la charge représente environ 2.5 kbit/s lorsque les piétons partagent leur contenu alors que celle-ci est de 8.5 kbit/s sans partage.

9.3 Conclusion

Dans ce chapitre, nous avons présenté les résultats de l'évaluation du mode de communication basé sur le contenu. Nous avons décrit un scénario mettant en œuvre la récupération de contenu populaire sur Internet par les nœuds mobiles. Les résultats montrent la pertinence du mode de communication basé sur le contenu. Notre approche collaborative, dans laquelle les nœuds sont capables de partager le contenu déjà acquis, permet de délivrer plus de messages, et en moins de temps, qu'une approche non collaborative. De plus, notre approche est plus économe en terme de nombre de messages disséminés que l'approche non collaborative.

10
Conclusion et perspectives

Conclusion

Le contexte de cette thèse se situe dans le cadre de la communication entre objets connectés. Ces derniers, dont le nombre est en constante augmentation, vont inéluctablement augmenter le trafic de données dans les réseaux des opérateurs. Pour des raisons de passage à l'échelle, de tolérance à la perte d'infrastructure, et de résistance à la censure, nous défendons l'idée qu'un système de communication entre objets connectés ne devrait pas reposer seulement sur l'infrastructure. L'utilisation des communications opportunistes via des interfaces de communication sans fil de courte portée (e.g. Wi-Fi P2P, Wi-Fi Ad-Hoc, Bluetooth) comme complément ou alternative à l'infrastructure semble prometteur pour pallier les faiblesses de l'infrastructure.

Dans les chapitres 5 et 6, nous avons présenté un système de communication pair-à-pair capable de fonctionner dans les réseaux hybrides à connexion intermittente baptisé Nephila. Ce système est capable de tirer parti des interfaces permettant des communications objet à objet (e.g. Wi-Fi P2P, Wi-Fi ad hoc, Bluetooth), ainsi que des interfaces de permettant d'établir des connexions avec une infrastructure (3G/4G, Wi-Fi *Managed*) réseau. Nephila met en œuvre un mécanisme permettant à chaque nœud du réseau de découvrir quels sont ses voisins, et les changements intervenant dans leur voisinage. Nous avons développé un algorithme de transfert de messages (BTSA) capable de supporter les ruptures de connectivité. Pour ce faire, BTSA met en œuvre le principe général du *store, carry and forward*. Cet algorithme utilise un cache de messages permettant à chaque nœud de stocker temporairement des messages à destination d'autres nœuds, et décide quand et à qui sont transférés les messages du cache. Afin d'aider BTSA à prendre les meilleures décisions, nous avons créé un algorithme de calcul de chemin. Ce dernier a pour but d'évaluer les routes (appelées valeurs de chemins) entre les nœuds. Ces valeurs sont échangées entre les nœuds en utilisant le principe du *gossiping*. Afin de passer à l'échelle, une structure de données particulière est utilisée pour stocker et propager ces valeurs : les Filtres de Bloom de Chemin. Cette structure probabiliste offre un compromis intéressant entre sa taille et le nombre de collisions.
Nephila fournit plusieurs modes d'échange de données. Le mode de communication point-à-point permet d'adresser des messages à n'importe quel nœud du réseau Nephila. Le mode de communication anycast permet de diriger les messages vers le nœud le plus

proche, lorsque plusieurs nœuds sont susceptibles d'héberger la donnée. Ce mécanisme nous a permis de fournir un service capable de relayer des messages à destination de nœuds accessibles sur Internet et ne faisant pas partie du réseau Nephila. Enfin le mode de communication basé sur le contenu permet à l'utilisateur de spécifier un type de contenu sans avoir à connaître l'identifiant des nœuds hébergeant ce contenu.

Dans les chapitres 7, 8, et 9 nous avons présenté les résultats de l'évaluation de Nephila pour différents scénarios. Nous avons décrit et évalué trois différents scénarios. Le premier scénario est la communication entre des nœuds mobiles. Ce scénario permet d'évaluer le mode de communication point-à-point. Les résultats montrent la capacité de Nephila d'acheminer efficacement les messages tout en tirant parti des points d'accès Wi-Fi lorsque ceux-ci sont disponibles. Nephila, sans infrastructure, obtient de meilleures performances que *Spray and Wait* et est plus économe en volume de messages que *PRoPHET*. Le second scénario évalue le mode de communication anycast. Des objets mobiles communiquent avec des serveurs fixes sur Internet, via un schéma de requête/réponse. L'approche anycast est comparée à l'approche unicast, et les résultats montrent les performances supérieures de l'approche anycast. Le troisième scénario évalue l'approche d'échange de données basée sur le contenu. Ce scénario met en œuvre la récupération de contenu populaire, avec un partage des données récupérées par les nœusd mobiles. Les résultats montrent la supériorité de notre approche de diffusion basée sur le contenu en termes de taux de délivrance et de latence des messages délivrés par rapport à une approche non collaborative.

Perspectives

Nous concluons ce mémoire de thèse en présentant certains problèmes ouverts qu'il serait intéressant d'étudier dans le prolongement de cette thèse.

Le système Nephila pourrait être amélioré en implantant une dissémination géographique des données, une différentiation du trafic, une structure pair-à-pair s'adaptant à la mobilité des nœuds ou encore en prenant en considération des aspects de sécurité.

Expérimentations en conditions réelles Les résultats obtenus en simulation mériteraient d'être validés dans un futur proche par des expérimentations en conditions réelles. De telles expérimentations n'ont pas été réalisées car elle nécessitent de pouvoir installer le système sur un grand nombre de smarphones portés par des piétons, et de pouvoir recruter ceux-ci pour tester le système. Pour tester Nephila en conditions réelles, une solution consisterait à intégrer celui-ci sur des applications mobiles et des intergiciels tels que ceux développés dans les projets C3PO [50]. En effet, dans le projet C3PO, des campagnes d'expérimentations en conditions réelles ont étés menées et le seront encore prochainement.

Dissémination géographique La dissémination géographique [116, 117, 118] permet de confiner certains messages dans une zone géographique. Ce mode de dissémination semble particulièrement adapté à la diffusion de données lors d'événements sociaux limités à une zone géographique, tels que des manifestations culturelles ou sportives. En effet, certaines données telles que des résultats sportifs ou l'agenda de l'événement sont susceptibles d'intéresser un grand nombre de personnes assistant à ces événements. Ce mode de dissémination permettrait également d'avertir les utilisateurs de leur arrivée dans une zone de danger, ou bien de disséminer une requête à un ensemble d'objets connectés (e.g. des capteurs) dans une zone donnée. Caractériser la zone géographique d'arrivée d'un message permet de développer des algorithmes de routage spécialisés qui limiteraient la diffusion des messages hors des zones spécifiées.

Différentiation du trafic Différencier le trafic permet d'attribuer des niveaux de priorité différente à certaines données. Dans cette thèse, nous avons traité les messages de façon uniforme. Notre algorithme de routage (BTSA) pourrait être adapté pour tenir compte d'une valeur de priorité associée à chaque message. Cette valeur serait par exemple fournie par l'utilisateur de l'intergiciel Nephila. La transmission des messages serait ordonnancée en fonction de leur priorité, et la politique de suppression des messages éliminerait d'abord les messages les moins prioritaires.

Structure pair-à-pair s'adaptant à la mobilité des nœuds Dans cette thèse, nous avons choisi de considérer un réseau pair-à-pair non structuré pour sa capacité à s'adapter à la mobilité et aux défaillances des nœuds. Il pourrait être intéressant d'étudier la construction de réseau de recouvrement s'adaptant à la dynamicité et au partitionnement du réseau. Le problème d'inadéquation entre la vue logique des réseaux pair-à-pair structurés et leur vue physique reste ouvert dans les réseaux opportunistes. T-MAN est un cadriciel permettant de former des réseaux pair-à-pair structurés. Contrairement aux autres approches qui permettent de former des réseaux pair-à-pair structurés, T-MAN [84] repose le principe du *gossiping* ce qui le rend plus robuste et plus rapidement convergent que les autres approches. T-MAN pourrait être adapté pour les réseaux opportunistes et serait donc un candidat intéressant pour poursuivre l'étude de la formation de structure logique s'adaptant aux spécificités des réseaux fortement dynamiques tels que les réseaux mobiles.

Sécurité Dans ce mémoire nous n'avons pas traité de la problématique de la sécurité dans les RHCI. Les premières mesures à prendre seraient de s'assurer de l'authencité, l'intégrité et la confidentialité du contenu des messages transportés dans le réseau. L'utilisation d'algorithmes de cryptographie asymétrique [119] et de tiers de confiance permet de résoudre ces problèmes sur Internet. Cependant les fréquentes déconnexions rendent plus difficile l'utilisation d'autorités de certification dans un RHCI. Des mécanismes basés sur un système de confiance et de réputation [120, 121, 122], qui ne nécessitent pas d'autorité centrale à tout instant, semble une approche intéressante. Des mécanismes basés sur la réputation peuvent aussi traiter la problématique des nœuds dits « égoïstes »(i.e.,

nœuds profitant du transport de leurs messages par des nœuds intermédiaires sans participer eux-mêmes).

Bibliographie

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things : A survey. *Computer Networks*, 54(15) :2787–2805, 2010.

[2] Alan Knight. Networked Journalism in the Arab Spring. *REASSESSING JOURNALISM*, page 107, 2013.

[3] Matt J Duffy. Smartphones in the Arab spring. *IPI Report : Media and Money. Vienna : International Press Institute*, pages 53–56, 2011.

[4] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pages 121–132, New York, NY, USA, 2004. ACM.

[5] P. Gardner-Stephen. The serval project : Practical wireless ad-hoc mobile telecommunications. *Rural, Remote & Humanitarian Telecommunications Fellow, Flinders University and Founder, Serval Project, Inc*, 2011.

[6] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (OLSR). Technical report, 2003.

[7] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (AODV) routing. Technical report, 2003.

[8] Axel Neumann, Corinna Aichele, Marek Lindner, and Simon Wunderlich. Better approach to mobile ad-hoc networking (BATMAN). *IETF draft, October*, 2008.

[9] M. Conti and M. Kumar. Opportunities in opportunistic computing. *Computer*, 43(1) :42–50, 2010.

[10] M. Conti, S. Giordano, M. May, and A. Passarella. From opportunistic networks to opportunistic computing. *Communications Magazine, IEEE*, 48(9) :126–139, 2010.

[11] Rüdiger Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, pages 101–102. IEEE, 2001.

[12] Ralf Steinmetz and Klaus Wehrle. 2. What Is This "Peer-to-Peer" About ? In *Peer-to-Peer Systems and Applications*, pages 9–16. Springer, 2005.

[13] Stefan Saroiu, Krishna P Gummadi, and Steven D Gribble. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia systems*, 9(2) :170–184, 2003.

[14] Bram Cohen. Bittorent protocol.

[15] Alan B Johnston and Daniel C Burnett. *WebRTC : APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC, 2012.

[16] Scott Corson and Joseph Macker. Mobile ad hoc networking (MANET) : Routing protocol performance issues and evaluation considerations. Technical report, 1998.

[17] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. Wi-fi in ad hoc mode : a measurement study. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 145–154. IEEE, 2004.

[18] Yawen Barowski, Saad Biaz, and Prathima Agrawal. Towards the performance analysis of IEEE 802.11 in multi-hop ad-hoc networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 1, pages 100–106. IEEE, 2005.

[19] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. IEEE 802.11 ad hoc networks : performance measurements. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 758–763. IEEE, 2003.

[20] Jack L Burbank, Philip F Chimento, Brian K Haberman, and William T Kasch. Key challenges of military tactical networking and the elusive promise of MANET technology. *IEEE Communications Magazine*, 44(11) :39–45, 2006.

[21] Xiaofeng Lu, Yung-chih Chen, Ian Leung, Zhang Xiong, and Pietro Liò. A novel mobility model from a heterogeneous military MANET trace. In *International Conference on Ad-Hoc Networks and Wireless*, pages 463–474. Springer, 2008.

[22] Lionel Barrère. *Étude et proposition de services dans les réseaux mobiles militaires de type MANet*. PhD thesis, Bordeaux 1, 2009.

[23] Zygmunt J Haas, Marc R Pearlman, and Prince Samar. The zone routing protocol (ZRP) for ad hoc networks. 2002.

[24] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking : Design tradeoffs and early experiences with ZebraNet. In *ACM Sigplan Notices*, volume 37, pages 96–107. ACM, 2002.

[25] Pei Zhang, Christopher M Sadler, Stephen A Lyon, and Margaret Martonosi. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238. ACM, 2004.

[26] Ting Liu, Christopher M Sadler, Pei Zhang, and Margaret Martonosi. Implementing software on resource-constrained mobile sensors : Experiences with Impala and ZebraNet. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 256–269. ACM, 2004.

[27] Margaret Martonosi. Embedded systems in the wild : Zebranet software, hardware, and deployment experiences. *ACM Sigplan Notices*, 41(7) :1–1, 2006.

[28] Tara Small and Zygmunt J Haas. The shared wireless infostation model : a new ad hoc networking paradigm (or where there is a whale, there is a way). In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 233–244. ACM, 2003.

[29] Hamed Soroush, Nilanjan Banerjee, Mark Corner, Brian Levine, and Brian Lynn. A retrospective look at the UMass DOME mobile testbed. *ACM SIGMOBILE Mobile Computing and Communications Review*, 15(4) :2–15, 2012.

[30] Kevin Fall and Stephen Farrell. DTN : an architectural retrospective. *IEEE Journal on Selected Areas in Communications*, 26(5) :828–836, 2008.

[31] Alex McMahon and Stephen Farrell. Delay-and disruption-tolerant networking. *IEEE Internet Computing*, 13(6) :82, 2009.

[32] Marco Conti and Silvia Giordano. Multihop Ad Hoc Networking : The Reality. *IEEE Communications Magazine*, 45(4) :88–95, April 2007.

[33] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking : data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11) :134–141, 2006.

[34] Stefano Ferretti. Shaping opportunistic networks. *Computer Communications*, 36(5) :481–503, 2013.

[35] Vinícius FS Mota, Felipe D Cunha, Daniel F Macedo, José MS Nogueira, and Antonio AF Loureiro. Protocols, mobility models and tools in opportunistic networks : A survey. *Computer Communications*, 48 :5–19, 2014.

[36] Hervé Ntareme, Marco Zennaro, and Björn Pehrson. Delay Tolerant Network on Smartphones : Applications for Communication Challenged Areas. In *3rd Extreme Workshop on Communications*, September 2011.

[37] Stephen Farrell, Alex McMahon, Stefan Weber, Kerry Hartnett, Aidan Lynch, and Eoin Meehan. Report on DTN Applications During Arctic Summer 2010 Trial. In *1st International Workshop on Opportunistic and Delay/Disruption-Tolerant Networking*, October 2011.

[38] Md. Tarikul Islam, Anssi Turkulainen, Teemu Kärkkäinen, Mikko Pitkänen, and Jörg Ott. Practical Voice Communications in Challenged Networks. In *1st Extreme Workshop on Communications*, August 2009.

[39] Costas Tziouvas, Lambros Lambrinos, and Chrysostomos Chrysostomou. A Delay Tolerant Platform for Voice Message Delivery. In *1st International Workshop on Opportunistic and Delay/Disruption-Tolerant Networking*, pages 1–5, 2011.

[40] Yves Mahéo, Nicolas Le Sommer, Pascale Launay, Frédéric Guidec, and Mario Dragone. Beyond Opportunistic Networking Protocols : a Disruption-Tolerant Application Suite for Disconnected MANETs. In *4th Extreme Conference on Communication (ExtremeCom'12)*, pages 1–6, Zürich, Switzerland, March 2012. ACM.

[41] Jaap C Haartsen. The Bluetooth radio system. *IEEE personal communications*, 7(1) :28–36, 2000.

[42] Brian P Crow, Indra Widjaja, LG Kim, and Prescott T Sakai. IEEE 802.11 wireless local area networks. *IEEE Communications magazine*, 35(9) :116–126, 1997.

[43] Daniel Camps-Mur, Andres Garcia-Saavedra, and Pablo Serrano. Device-to-device communications with Wi-Fi Direct : overview and experimentation. *IEEE wireless communications*, 20(3) :96–104, 2013.

[44] Patrick Kinney et al. Zigbee technology : Wireless control that simply works. In *Communications design conference*, volume 2, pages 1–7, 2003.

[45] Arash Asadi and Vincenzo Mancuso. WiFi Direct and LTE D2D in action. In *Wireless Days (WD), 2013 IFIP*, pages 1–8. IEEE, 2013.

[46] Teemu Kärkkäinen, Mikko Pitkänen, Paul Houghton, and Jörg Ott. Scampi application platform. In *Proceedings of the seventh ACM international workshop on Challenged networks*, pages 83–86. ACM, 2012.

[47] James Scott, Jon Crowcroft, Pan Hui, and Christophe Diot. Haggle : A networking architecture designed around mobile users. In *WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services*, pages 78–86, 2006.

[48] Stephen Farrell, Stefan Weber, Alex McMahon, Eoin Meehan, and Kerry Hartnett. An n4c dtn router node design. In *1st Extreme Workshop on Communication, Laponia, Sweden, August 8–14 2009*, 2009.

[49] Julien Haillot, Frédéric Guidec, Serge Corlay, and Jacques Turbert. Disruption-Tolerant Content-Driven Information Dissemination in Partially Connected Military Tactical Radio Networks. In *28th IEEE Military Communication Conference (MILCOM'2009)*, pages 2326–2332, Boston, United States, October 2009. IEEE CS.

[50] Frédérique Laforest, Nicolas Le Sommer, Stéphane Frénot, François De Corbière, Yves Mahéo, Pascale Launay, Christophe GRAVIER, Julien Subercaze, Damien Reimert, Etienne Brodu, Idris Daikh, Nicolas Phelippeau, Xavier Adam, Frédéric Guidec, and Stéphane Grumbach. C3PO : a Spontaneous and Ephemeral Social Networking Framework for a collaborative Creation and Publishing of Multimedia Contents. In *International conference on selected topics in Mobile and Wireless Networking (MoWNet 2014)*, Procedia Computer Science, pages 1–6, Rome, Italy, September 2014. Elsevier.

[51] Michael Mrissa, Lionel Médini, Jean-Paul Jamont, Nicolas Le Sommer, and Jérôme Laplace. An Avatar Architecture for the Web of Things. *Internet Computing, IEEE*, page 30, March 2015.

[52] Vinìcius F. S. Mota, Felipe D. Cunha, Daniel F. Macedo, José M. S. Nogueira, and Antonio A. F. Loureiro. Protocols, Mobility Models and Tools in Opportunistic Networks : A Survey. *Computer Communications*, March 2014.

[53] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 235–244. IEEE, 2004.

[54] Amin Vahdat, David Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.

[55] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 373–384. ACM, 2007.

[56] Ram Ramanathan, Richard Hansen, Prithwish Basu, Regina Rosales-Hain, and Rajesh Krishnan. Prioritized epidemic routing for opportunistic networks. In *Procee-*

dings of the 1st international MobiSys workshop on Mobile opportunistic networking, pages 62–66. ACM, 2007.

[57] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait : an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.

[58] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi 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*, pages 79–85. IEEE, 2007.

[59] S. Milgram. The small world problem. *Psychology today*, 2(1) :60–67, 1967.

[60] W. Hsu and A. Helmy. On nodal encounter patterns in wireless LAN traces. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–10. IEEE, 2006.

[61] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684) :440–442, 1998.

[62] E.M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[63] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *Service Assurance with Partial and Intermittent Resources*, pages 239–254, 2004.

[64] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine. Maxprop : Routing for vehicle-based disruption-tolerant networks. In *Proc. ieee infocom*, volume 6, pages 1–11. Barcelona, Spain, 2006.

[65] Chiara Boldrini, Marco Conti, Jacopo Jacopini, and Andrea Passarella. Hibop : a history based routing protocol for opportunistic networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–12. IEEE, 2007.

[66] Mirco Musolesi and Cecilia Mascolo. Car : context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, 8(2) :246–260, 2009.

[67] Pei-Chun Cheng, Jui-Ting Weng, Lung-Chih Tung, Kevin C Lee, Mario Gerla, and Jerome Haerri. GeoDTN+ Nav : a hybrid geographic and DTN routing with navigation assistance in urban vehicular networks. *MobiQuitous/ISVCS*, 2008.

[68] Pei-Chun Cheng, Kevin C Lee, Mario Gerla, and Jérôme Härri. GeoDTN+ Nav : geographic DTN routing with navigator prediction for urban vehicular environments. *Mobile Networks and Applications*, 15(1) :61–82, 2010.

[69] Vasco NGJ Soares, Joel JPC Rodrigues, and Farid Farahmand. GeoSpray : A geographic routing protocol for vehicular delay-tolerant networks. *Information Fusion*, 15 :102–113, 2014.

[70] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 96–108. ACM, 2003.

[71] Cisco VNI Mobile. Cisco visual networking index : global mobile data traffic forecast update, 2011–2016. *San Jose, CA*, 1, 2014.

[72] Antonino Masaracchia. Opportunistic traffic Offloading Mechanisms for Mobile/4G Networks. In *Proceedings of the 2015 on MobiSys PhD Forum*, pages 15–16. ACM, 2015.

[73] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M.D. De Amorim. Push-and-track : Saving infrastructure bandwidth through opportunistic forwarding. *Pervasive and Mobile Computing*, 2012.

[74] Haruki Izumikawa and Jiro Katto. RoCNet : Spatial mobile data offload with user-behavior prediction through delay tolerant networks. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2196–2201. IEEE, 2013.

[75] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2) :174–188, 2002.

[76] Christoph P Mayer and Oliver P Waldhorst. Offloading infrastructure using delay tolerant networks and assurance of delivery. In *Wireless Days (WD), 2011 IFIP*, pages 1–7. IEEE, 2011.

[77] Patrick Baier, Frank Dürr, and Kurt Rothermel. TOMP : Opportunistic traffic offloading using movement predictions. In *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, pages 50–58. IEEE, 2012.

[78] David P Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. SETI@ home : an experiment in public-resource computing. *Communications of the ACM*, 45(11) :56–61, 2002.

[79] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM Computer Communication Review*, volume 31, pages 149–160. ACM, 2001.

[80] D Eastlake 3rd and Paul Jones. US secure hash algorithm 1 (SHA1). Technical report, 2001.

[81] Jon Postel. Internet protocol. 1981.

[82] Antony Rowstron and Peter Druschel. Pastry : Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001*, pages 329–350. Springer, 2001.

[83] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. *A scalable content-addressable network*, volume 31. ACM, 2001.

[84] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. T-Man : Gossip-based fast overlay topology construction. *Computer networks*, 53(13) :2321–2339, 2009.

[85] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12. ACM, 1987.

[86] AM Kermarrec and M van Steen. ACM SIGOPS operating systems review 41. *Special issue on gossip-based networking*, 2007.

[87] O. Babaoglu, M. Jelasity, O. Babaoglu, and M. Jelasity. Self-* properties through gossiping. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 366(1881) :3747–3757, 2008.

[88] Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten Van Steen. Gossip-based peer sampling. *ACM Transactions on Computer Systems (TOCS)*, 25(3) :8, 2007.

[89] Matei Ripeanu. Peer-to-peer architecture case study : Gnutella network. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, pages 99–100. IEEE, 2001.

[90] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7) :422–426, 1970.

[91] M McIlroy. Development of a spelling list. *IEEE Transactions on Communications*, 30(1) :91–99, 1982.

[92] James K Mullin and Daniel J Margoliash. A tale of three spelling checkers. *Software : Practice and Experience*, 20(6) :625–630, 1990.

[93] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z Broder. Summary cache : a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking (TON)*, 8(3) :281–293, 2000.

[94] A. Kumar, J. Xu, and E.W. Zegura. Efficient and scalable query routing for unstructured peer-to-peer networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1162–1173. IEEE, 2005.

[95] Tobias Heer, Stefan Götz, Simon Rieche, and Klaus Wehrle. Adapting Distributed Hash Tables for Mobile Ad Hoc Networks. In *PerCom Workshops*, pages 173–178. Citeseer, 2006.

[96] Piotr Karwaczyński, Dariusz Konieczny, Jaka Mocnik, and Marko Novak. Dual proximity neighbour selection method for peer-to-peer-based discovery service. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 590–591. ACM, 2007.

[97] Ngoc Ben Dang, Son Tung Vu, and Hoai Son Nguyen. Building a low-latency, proximity-aware DHT-based P2P network. In *Knowledge and Systems Engineering, 2009. KSE'09. International Conference on*, pages 195–200. IEEE, 2009.

[98] Qiang Xu, Lechang Sun, and Jingju Liu. Topology-aware kademlia based on distributed clustering in self-organizing mode. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 1, pages V1–267. IEEE, 2010.

[99] Himabindu Pucha, Saumitra M Das, and Y Charlie 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*, pages 163–173. IEEE, 2004.

[100] Thomas Zahn and Jochen Schiller. DHT-based unicast for mobile ad hoc networks. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pages 5–pp. IEEE, 2006.

[101] Alexander Klemm, Christoph Lindemann, and Oliver 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*, volume 4, pages 2758–2763. IEEE, 2003.

[102] Wolfgang Kellerer and Rüdiger Schollmeier. Proactive search routing for mobile peer-to-peer networks : Zone-based p2p. In *Proc. of ASWN*. Citeseer, 2005.

[103] Nadir Shah and Depei Qian. An efficient unstructured p2p overlay over manet using underlying proactive routing. In *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*, pages 248–255. IEEE, 2011.

[104] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.

[105] O. Babaoglu, T. Binci, M. Jelasity, and A. Montresor. Firefly-inspired heartbeat synchronization in overlay networks. In *Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on*, pages 77–86. IEEE, 2007.

[106] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems (TOCS)*, 23(3) :219–252, 2005.

[107] John Byers, Jeffrey Considine, Michael Mitzenmacher, and Stanislav Rost. Informed content delivery across adaptive overlay networks. *ACM SIGCOMM Computer Communication Review*, 32(4) :47–60, 2002.

[108] Christopher Rohrs. Query routing for the Gnutella network. *lime Wire LLC, http ://www.limewire.com/developer/queryrouting/keywordrouting.htm*, 2001.

[109] Michael T Prinkey. An efficient scheme for query processing on peer-to-peer networks. *aeolus Research, Inc., http ://aeolusres.homestead.com/files/index.html*, 2001.

[110] Sean C Rhea and John Kubiatowicz. Probabilistic location and routing. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1248–1257. IEEE, 2002.

[111] Spyros Voulgaris, Daniela Gavidia, and Maarten Van Steen. Cyclon : Inexpensive membership management for unstructured p2p overlays. *Journal of Network and Systems Management*, 13(2) :197–217, 2005.

[112] François Bonnet, Frédéric Tronel, and Spyros Voulgaris. Brief announcement : Performance analysis of cyclon, an inexpensive membership management for un-

structured p2p overlays. In *International Symposium on Distributed Computing*, pages 560–562. Springer, 2006.

[113] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09 : Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.

[114] Mordechai Haklay and Patrick Weber. Openstreetmap : User-generated street maps. *Pervasive Computing, IEEE*, 7(4) :12–18, 2008.

[115] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic Routing in Intermittently Connected Networks. In *Proceedings of the The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, Fortaleza, Brazil, August 2004.

[116] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing : A recursive data dissemination protocol for wireless sensor networks. Technical report, Citeseer, 2001.

[117] Tomasz Imielinski and Julio C Navas. Geographic Addressing, Routing, and resource discovery with the global Positioning system. *published by Computer Science Department, University of Rutgers*, 1996.

[118] Julio C Navas and Tomasz Imielinski. GeoCast—geographic addressing and routing. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–76. ACM, 1997.

[119] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[120] Gianluca Dini and Angelica Lo Duca. A reputation-based approach to tolerate misbehaving carriers in delay tolerant networks. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 772–777. IEEE, 2010.

[121] Lifei Wei, Haojin Zhu, Zhenfu Cao, and Xuemin Sherman Shen. MobiID : A user-centric and social-aware reputation based incentive scheme for delay/disruption tolerant networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 177–190. Springer, 2011.

[122] Gianluca Dini and Angelica Lo Duca. Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network. *Ad Hoc Networks*, 10(7) :1167–1178, 2012.

Résumé

La multiplication du nombre d'objets, qui ont vocation à être connectés à Internet (e.g., smartphones, capteurs), et la croissance des échanges de données effectués par des individus en situation de mobilité ont conduit, et conduiront encore, à une augmentation significative du trafic de données dans les réseaux, et en particulier dans les réseaux cellulaires. Les récents progrès réalisés au niveau de la couche physique pour accroître les débits dans ces réseaux pourraient s'avérer insuffisants dans le futur avec l'émergence d'un Internet des objets. Il nous semble dès lors intéressant d'étudier des architectures réseau alternatives ou complémentaires.
Les réseaux hybrides à connectivité intermittente (RHCI), qui sont constitués d'une infrastructure et de parties formées par des objets fixes ou mobiles communiquant en mode ad hoc, font partie de ces architectures qui méritent d'être étudiées.
Dans cette thèse, nous étudions les bénéfices que pourrait apporter l'utilisation des techniques des réseaux pair-à-pair et des communications opportunistes dans les RHCI. Nous proposons une architecture pair-à-pair décentralisée et non structurée qui permet d'assurer les communications entre des objets dans des RHCI de grande taille via différents modes de communication. Un prototype de plateforme, baptisé Nephila a été développé, pour évaluer cette approche en simulation.

Abstract

The number of devices that are likely to get connected to the Internet (e.g., smartphones, sensors), and the amount of data produced by people using these devices grow continuously, especially in cellular networks. Latest developments performed on the physical layer to increase the networks' bandwidth might be insufficient in the future, because of the emergence of the Internet of things. Therefore, it seems to be interesting to study new or complementary network architectures.
Intermittently-Connected Hybrid Networks (ICHN), which are composed both of an infrastructure part and of parts formed by mobile device communicating using ad hoc mode, are examples of those architectures that deserve to be studied.
In this thesis, we study benefits that peer-to-peer mechanisms and opportunistic networking techniques could bring to ICHN. We propose a decentralized unstructured peer-to-peer overlay architecture that supports communications between devices in wide ICHNs. A prototype named Nephila has been developed to evaluate this approach in simulation.

n d'ordre : 432

Université de Bretagne Sud
Centre d'Enseignement et de Recherche Y. Coppens - rue Yves Mainguy - 56000 VANNES
Tél : + 33(0)2 97 01 70 70 Fax : + 33(0)2 97 01 70 70