

UNIVERSITÉ D'ÉVRY-VAL-D'ESSONNE

LABORATOIRE DE RÉSEAUX ET SYSTÈMES MULTIMÉDIA

THÈSE

Pour obtenir le grade de

Docteur de l'Université d'Évry-Val-d'Essonne

Spécialité : *Informatique*

Présentée et devrait être soutenue publiquement

Le 25 Mars 2009

Optimisation des applications de streaming peer to peer pour des réseaux ad hoc mobiles

Par

Mehdi NAFA

Jury

Directeur	M. Nazim AGOULMINE	Professeur à l'Université d'Évry-Val-d'Essonne
Rapporteurs	M ^{me} Francine KRIEF	Professeur à l'ENSEIRB
	M. Ahmed KARMOUCH	Professeur à l'Université d'Ottawa
Examineurs	M. Djamel BELAID	Professeur à Télécom SudParis
	M. Chadi BARAKAT	Chargé de recherche à l'INRIA Sophia
	M ^{me} Brigitte KERVELLA	Maitre de conférences à l'Université d'Amiens

Dédicace

Je dédie cet humble travail,

A celle qui m'a toujours soutenu, supportant mes sauts d'humeur et mon stress, inconditionnellement présente auprès de moi, à toi ma Ouafa,

A celui que j'appelle mon « p'tit bonheur » qui à changer ma dernière année de thèse ainsi que ma vie en une véritable allégresse, oui à toi mon petit Nazim.

A ma chère maman, qui a toujours été là pour moi, m'a toujours soutenu, compris et fait tout ce qu'il faut pour me faciliter la vie, merci maman,

A mon adorable papa, sans qui je ne serai jamais arrivé là, cet homme qui a toujours su être présent pour moi, à tout moment et en toutes circonstances, faisant ainsi tout les sacrifices, soucieux toujours de me propulser vers le meilleur, un très grand merci papa,

A mon frère avec qui j'apprends toujours de la vie, des acquis que je n'aurais jamais su avoir sans toi, merci Dhikrane,

A vous mes amis, vous qui ensoleillez ma vie, et avec qui j'ai ce très grand plaisir de partager des instants de bonheur continu, et en particulier : Billy, Sana, Loops, Yacine, Abdoul, Makdoudi, Malek, Mounir, Rahim, Nadir, Jeddi, Teicire.

A vous mes camarades de parcours les précieux amis du LRSM : Elyes, Gondi, Hajer, Alaa, Nada, Halim, Ismail, Hui, Dang et Toan.

Encore merci du fond du cœur,

Mehdi

Remerciement

Ce travail de thèse n'aurait jamais aboutit sans l'aide, les encouragements et l'implication de certaines personnes à qui j'exprime à travers ces quelques phrases modestes toute ma gratitude, en particulier :

Je suis très reconnaissant envers Monsieur Ahmed KARMOUCH, Professeur à l'Université d'Ottawa (Canada) ainsi que Madame Francine KRIEF Professeure à l'ENSERB (France) pour m'avoir fait l'honneur de juger ma thèse et de rapporter mon travail je vous remercie pour toute l'attention portée durant cette évaluation.

Je remercie chaleureusement Monsieur Djamel BELAID, Professeur à Télécom SudParis (France), Monsieur Chadi BARAKAT, Chargé de recherche à l'INRIA Sophia (France), Madame Brigitte KERVELLA, Maitre de conférences à l'Université d'Amiens (France). D'avoir accepter d'être parmi les membres de jury de ma thèse et d'avoir examiner mes travaux.

Toute ma gratitude va à mon Professeur, Monsieur Nazim AGOULMINE, Professeur à l'Université d'Evry Val d'Essonne (France) pour m'avoir accueilli dans son laboratoire de recherche LRSM (Laboratoire de Réseaux et Systèmes Multimedia) et de m'avoir permis d'effectuer cette thèse dans de bonne conditions. Je le remercie aussi pour m'avoir donné toute ces idées, ainsi que pour toutes les riches réunions de travail, sans quoi je ne serai jamais arrivé à bout de cette thèse, je ne vous remercierai jamais assez pour m'avoir fait profiter de votre expérience et de m'avoir témoigné tant de bienveillance.

Je remercie O. Bensaada et K. Nafaa pour avoir bien voulu faire une relecture critique de ce manuscrit

Abstract

Video content production has reached, nowadays, a big volume of traffic over the internet. Emerging social and professional networks (i.e. Facebook) and video portals (i.e. Youtube) acted as veritable accelerators for this phenomenon. In addition to IPTV and VoD applications, personal video sharing is also in the run.

Our initial concern in this thesis was infrastructure wireless networks. We were focusing on how to provide a seamless mobile multimedia streaming service during horizontal handovers over cellular WLAN networks. This includes the proposal of a prediction based buffering scheme to cope with the video service disruption during the blackout periods. The work ended with a comparison between simulated and real test-bed implemented results.

After file sharing and IP telephony, p2p streaming applications are getting a great concern from both academia and industry. The main goal of this technology is to allow nodes to cooperate and exchange contents among themselves. A lot of p2p streaming applications are currently used, over the Internet, for both video on demand and real-time streaming services like IPTV such as PPStream, UUSee, SopCas, TVants and Joost.

At the same time we are seeing the development of new type of infrastructure-less communications using different types of devices PDA, telephones, computers, etc called Mobile Ad Hoc Networks. With this communication technology, networks can be created spontaneously at any place (home, office, street, bus, etc.). In a second phase our aim was to show how these MANET and P2P technologies can stimulate the emergence of new types of experience sharing in the society. We can easily imagine a scenario where mobile users build their own swarm, without resorting to the infrastructure, to exchange videos among them with no cost. Relying on their built-in wireless communication interfaces, they form a spontaneous network based on their preferences and including their friend, colleagues, etc. The two last contribution chapters are targeting the modeling and simulation of a mobile p2p streaming solution for MANETs. Relying on a mobility model based on the social communities we have shown the feasibility of P2P streaming over MANETs.

Résumé

De nos jours, La de production de contenu vidéo génère un grand volume de trafic sur Internet. Une panoplie de réseaux sociaux émergent (tels que Facebook) et de portails vidéo communautaires (tels que Youtube) ont agi comme de véritables accélérateurs de ce phénomène. En plus des applications IPTV et de VoD, le partage de vidéos personnelles est également dans la course.

Notre première préoccupation dans cette thèse est la mobilité dans les réseaux sans fil dits infrastructure. Nous nous sommes concentrés sur la façon de fournir un service de streaming multimédia sans coupures lors des transferts horizontaux d'un mobile traversant des réseaux WLAN cellulaires. Cela comprend la proposition d'un système de mise en cache (buffering) fondé sur la prédiction du handover afin de faire face à l'interruption du service vidéo au cours de la période de la perte de connexion par le terminal. Le travail s'est terminé par une comparaison entre des résultats obtenus par simulation d'un coté et ceux mesurés sur un banc d'essai réel.

Après le partage de fichiers et la téléphonie IP, les applications de streaming p2p reçoivent un grand intérêt à la fois académique et industriel. L'objectif principal de cette technologie est de permettre aux nœuds de coopérer et d'échanger les contenus entre eux. Un grand nombre d'applications de streaming p2p sont actuellement utilisées, sur Internet, à la fois pour la vidéo à la demande et en temps réel comme les services de télévision IP, tels que PPStream, UUSee, SopCas, TvAnts et Joost.

En même temps, nous assistons au développement d'un nouveau modèle de communication sans infrastructures présentes dans différents types de dispositifs mobile (tels que les PDA, téléphones, ordinateurs portable, etc) appelée Mobile Ad Hoc Networks. Avec cette technologie de communication, les réseaux peuvent être créés spontanément à n'importe quel endroit (domicile, bureau, rue, bus, etc.) Dans une deuxième phase, notre objectif était de montrer comment ces technologies P2P MANET et peuvent stimuler l'émergence de nouvelles formes de partage d'expérience dans la société. Nous pouvons facilement imaginer un scénario où les mobiles des utilisateurs construire leur propre essaim, sans recourir à l'infrastructure, à l'échange de vidéos entre eux sans frais. S'appuyant sur leurs interfaces de communications sans fils, ils forment un réseau spontané en fonction de leurs préférences et y compris leurs amis, leurs collègues, etc. Les deux derniers chapitres de contribution portent sur la modélisation et la simulation d'une solution de streaming p2p pour MANETs. S'appuyant sur un modèle de mobilité basé sur des communautés sociales, nous avons pu démontrer la faisabilité du streaming P2P sur MANETs.

Liste des tableaux

Tableau 1 – Comparaison entre topologies Mesh et Tree	23
Tableau 2 – MANETs vs.P2P: Différences	35
Tableau 3 – MANETs vs.P2P:Similitudes	37
Tableau 4 autres paramètres du modèle	85
Tableau 5. Statistiques sur les nœuds	90
Tableau 6 – Paramètres de simulation.....	100

Liste des figures

Figure 1: Réseau P2P de 1 ^{ère} génération (index central)	10
Figure 2: Réseau P2P de 2 ^{ème} génération (inondation de requête)	11
Figure 3: Table de hachage distribuée (DHT)	12
Figure 4: Le super peering.....	13
Figure 5: Parts des clients (BitTorrent, Kazaa, eMule et Gnutella) sur le trafic P2P file sharing [6]	18
Figure 6: topologies Mesh vs. Tree.....	23
Figure 7: Push vs. Pull.....	26
Figure 8: Classification des familles de protocoles ad hoc	27
Figure 9:Algorithme de prédiction du Tbho.....	52
Figure 10: Prédiction du Tbho et évolution du buffer	57
Figure 11: Schéma du banc d'essai.....	58
Figure 12:Schéma du buffering.....	59
Figure 13 Composantes logicielles de SM ² Stream	61
Figure 14 Evolution des RSS reçus	62
Figure 15: Tbho prédit et évolution du buffer	63
Figure 16 Couches du modèle	70
Figure 17 : Comparaison des résultats des équations maîtresse et de taux [109].....	73
Figure 18: Un exemple de réseau social [121]	75
Figure 19: Exemple d'une Matrice d'affinité représentant un réseau social.	76
Figure 20:Exemple de matrice d'adjacence représentant le réseau social.	76
Figure 21: Exemple d'une communauté mobile [121].	77
Figure 22: Distribution des degrés entrant et sortant.	86
Figure 23: Evaluations du délai (cas fixe).	87
Figure 24: Evaluations du délai (cas mobile sans considération de l'overhead).	89

Figure 25: Distribution du délai (cas mobile avec considération de l'overhead)	90
Figure 26: Vue d'ensemble de MadTorStream.....	98
Figure 27: Vue en couches de MadTorStream.	98
Figure 28: La stratégie Closest Deadline First.....	99
Figure 29: Résultats avec Closest Deadline First.....	101
Figure 30: Résultats avec Rarest First.....	101

Liste des algorithmes

Algorithme 1: Collecte d'information sur les réseaux avoisinants	59
Algorithme 2 : prédiction des T_{bho} et T_{bmo}	60
Algorithme 3 : mise en cache.....	60
Algorithme 4: exécution du handover	61
Algorithme 5: modification de l'algorithme principal de Groover	84

Table des matières

Dédicace	i
Remerciement	iii
Abstract	v
Résumé.....	vii
Liste des tableaux	ix
Liste des figures.....	xi
Liste des algorithmes.....	xiii
Table des matières	xv
Chapitre 1. Introduction	1
1.1. Streaming peer to peer	1
1.2. Mobile ad hoc networks	2
1.3. Contexte & Motivations.....	3
1.4. Contributions de la thèse	4
1.5. Structure du document	5
Chapitre 2. Etat de l'art	7
2.1. Le streaming vidéo	7
2.1.1. Transport de flux sur le réseau	8
2.1.2. Sur la couche applicative	8
2.1.2.1. <i>Mesure de la performance</i>	9
2.1.3. Utilisation du 802.11 pour le streaming.....	9
2.2. Les applications p2p.....	10
2.2.1. L'indexation et la recherche.....	10
2.2.1.1. <i>Approche centralisée</i>	10
2.2.1.2. <i>décentralisée (distribuée)</i>	11
a. Tables de hachages distribuées DHT:	11
b. Hybride	12

2.2.2.	Application de partage de contenu	13
2.2.2.1.	<i>Partage de fichier (File sharing)</i>	13
a.	Gnutella.....	14
b.	FastTrack.....	15
c.	eDonkey.....	16
d.	BitTorrent	16
e.	Statistiques.....	17
2.2.2.2.	<i>Voix sur IP</i>	18
2.2.2.3.	<i>Streaming audio et vidéo</i>	19
a.	Pplive	20
b.	Joost.....	21
2.2.2.4.	<i>Les deux topologies du streaming P2P</i>	22
a.	Topologie mesh.....	22
b.	Topologie Arbre.....	22
c.	Comparaison.....	23
2.2.2.5.	<i>Stratégies de sélection de bloc et de nœuds</i>	24
a.	La sélection de pairs	24
b.	La sélection de bloc	25
2.2.2.6.	<i>Technique de transfer des blocs</i>	25
a.	Le mode push	25
b.	Le mode Pull	26
c.	Le mode hybride.....	26
2.3.	Les réseaux ad hoc mobiles.....	27
2.3.1.	Classification des protocoles de routage.....	27
2.3.1.1.	<i>Protocoles proactifs</i>	27
a.	OLSR (Optimized Link-State Routing).....	28
2.3.1.2.	<i>Réactif</i>	28
a.	AODV (Ad Hoc On-demand Distance Vector)	28

2.3.1.3. <i>Hybride</i>	29
a. ZRP (Zone Routing Protocol)	29
2.3.2. Modèles de mobilité	30
2.3.2.1. <i>Modèles synthétiques</i>	30
a. Modèles basés sur la marche aléatoire	30
b. Le Random Way-point	31
2.3.3. Group Mobility Model	31
2.3.4. Community based mobility model	32
2.4. P2P sur MANETs quelle motivation ?.....	32
2.4.1. Différences	33
2.4.2. Similitudes	35
2.4.3. Synergie entre les systèmes P2P et MANET	37
2.4.4. P2P MANETS	38
2.4.5. Approches de conception du P2P MANET	39
2.4.5.1. <i>Stratifiée et non structurée</i>	39
2.4.5.2. <i>Intégrée et non structurée</i>	39
2.4.5.3. <i>Stratifiée et structurée</i>	40
2.4.5.4. <i>Intégrée et structurée</i>	40
2.5. Mobile P2P Streaming	40
Chapitre 3. SM ² Stream: Diffusion multimédia dans les réseaux disruptifs	43
3.1. Résumé.....	43
3.2. Introduction	43
3.2.1. Contexte	43
3.2.2. Problématique.....	44
3.3. Introduction à la mobilité dans l'infrastructure sans fil:	44
3.4. Travaux relatifs.....	45
3.5. La mise en cache (buffering)	47
3.6. La prédiction du handover	48
3.6.1. Handover horizontal dans WLAN.....	49

3.6.2.	Présentation du filtre grey GM(1,1)	50
3.6.3.	Prédiction du temps avant handover.....	51
3.6.4.	Prédiction du temps avant l'abandon de la cellule.....	54
3.7.	Evaluations	55
3.7.1.	Simulations	55
3.7.2.	Résultats et Discussions.....	56
3.7.3.	Expérimentation	57
3.7.3.1.	<i>Description du banc d'essai</i>	57
3.7.3.2.	<i>Algorithmes réalisés dans VLC</i>	58
3.7.3.3.	<i>Résultats</i>	61
3.8.	Conclusion	63
Chapitre 4.	MADP2PStream: Modèle de streaming peer to peer sur réseaux ad hoc mobiles..	65
4.1.	Introduction	65
4.1.1.	Objectif de cette contribution.....	66
4.2.	Travaux existants	67
4.3.	Modélisation du p2p streaming basé sur le modèle de Carra <i>et al.</i>	68
4.3.1.	Principes de base	68
4.3.2.	Analyse des applications streaming fondées sur le mesh.....	69
4.3.3.	Comportement du système ciblé.....	71
4.3.3.1.	<i>Le streaming</i>	71
4.3.3.2.	<i>Arrivée, départ & mise à jour</i>	72
4.3.4.	Modèle mathématique	72
4.4.	Modélisation de la mobilité par Musolesi <i>et al.</i>	74
4.4.1.	Matrice d'affinité (affinity).....	75
4.4.2.	Modèle social de la communauté	76
4.5.	Prise en compte de l'overhead dans le calcul des délais.....	79
4.5.1.	Cas de protocole réactif	80
4.5.2.	Cas de protocole proactif.....	81
4.6.	Représentation de la mobilité dans le graphe de streaming p2p.....	82

4.7.	Evaluations	84
4.7.1.	Groover: solveur d'équations différentielles.....	84
4.7.2.	Application du modèle MADP2PStream	84
4.7.3.	Analyse du degré	85
4.7.4.	Analyse du délai	86
4.7.5.	Analyse de la qualité.....	90
4.8.	Discussion et conclusions.....	91
Chapitre 5.	: <i>MadTorStream</i> - un protocole pour le streaming p2p mobile	93
5.1.	Introduction	93
5.1.1.	Persistence et réplication pour palier à la mobilité.....	94
5.1.2.	Mobilité augmente la diversité.....	94
5.2.	Travail effectué	94
5.2.1.	Travaux relatifs	95
5.2.2.	Manet modèles de routage et de la mobilité	96
5.2.3.	Mobile peer-to-peer streaming	96
5.2.4.	Difference entre file sharing et streaming p2p.....	96
5.3.	Vue d'ensemble de MadTorStream	97
5.3.1.	Manet modèles de routage et de la mobilité	98
5.3.2.	Au niveau peer-to-peer (couche de partage vidéo).....	99
5.3.3.	Application du Modèle de mobilité.....	99
5.4.	Simulations.....	100
5.5.	Discussion des résultats.....	100
5.6.	Conclusions	102
Chapitre 6.	Conclusions et perspectives	103
6.1.	Synthèse du travail réalisé	103
6.2.	Résumé des contributions	104
6.2.1.	Streaming sans coupure en mobilité sur WLAN « SM ² Stream ».....	104
6.2.2.	Modèle de streaming p2p sur MANET « MADP2PStream ».....	104
6.2.3.	Streaming bittorrent sur MANET « MadTorStream »	105

6.3.	Perspectives	105
6.3.1.	Adaptation de paquets vidéo pour handover	106
6.3.2.	Mécanisme de réplication pour streaming P2P sur MANET	106
6.3.3.	Clustering basé sur la session de lecture.....	107
6.3.4.	Localisation P2P sémantique de contenu vidéo (DHT, MPEG-7 et Ontologies) ...	107
	Références	109
	Annexe A.....	123
A.1.	Liste des publications.....	123
A.1.1.	Articles de recherche.....	123
A.1.2.	Rapports de projet.....	124
A.2.	Index.....	125
A.3.	Glossaire	127
A.4.	Media Streaming dans VLC.....	129

Chapitre 1. Introduction

Les réseaux de télécommunications ont, en cette dernière décennie, connu une remarquable croissance. L'usage des nouvelles technologies mobiles tend à s'imposer en acteur quotidien de l'activité socio économique et culturelle des sociétés. L'envoi de messages et de photographies, l'accès à l'information, les jeux et le partage de fichiers sont des applications qui font aujourd'hui partie intégrante des appareils mobiles. Ce comportement humain consacre la rupture de la dépendance à la localisation géographique d'alors assurant une concordance lors de la mobilité, avec l'omni-disponibilité de l'information.

Plusieurs phénomènes fascinants caractérisent l'évolution actuelle d'Internet, notamment avec l'avènement du multimédia, tant sur le plan professionnel que celui du loisir. Sans aucun doute YouTube, FaceBook, mais aussi, la diffusion de chaînes télévisées sur internet sont les parfaits exemples de cette mutation.

1.1. Streaming peer to peer

D'un autre coté, les internautes se sont retournés vers une nouvelle structure distribuée dédiée au partage des informations connue sous le nom des systèmes P2P (peer to peer ou pair à pair); un réseau où les entités communiquent directement et indépendamment de toute autorité centrale pour assurer leurs interactions. La popularité d'une telle technologie révèle la tendance manifeste d'Internet vers un système réparti qui supporte plus que de simples applications client/serveur. Cette distribution du contrôle et des objets constitue la pierre angulaire des réseaux pair-à-pair. En général, cette décentralisation s'effectue de manière à ce que les pairs soient fortement autonomes et indépendants les uns des autres. L'avantage de cette approche réside dans sa capacité de passer à l'échelle: Un bon réseau pair à pair peut facilement atteindre une taille de quelques millions de pairs, qui peuvent parfaitement le rejoindre ou le quitter, à tout moment, sans détériorer sérieusement la qualité globale du service.

Depuis leur apparition, les protocoles P2P connaissent un succès commercial considérable ainsi qu'un grand intérêt académique. Ces applications P2P utilisent actuellement des protocoles de

plus en plus complexes avec des comportements qui suscitent, depuis leur émergence, l'intérêt des chercheurs et des industriels. Dans les activités de recherches sur des systèmes P2P, les problématiques et les aspects les plus abordés sont manifestement ceux liés à la découverte de ressources, la dynamique, la gestion des contenus, le routage ainsi que l'amélioration de la qualité du service. Ces études se basent sur l'optimisation et la théorie des graphes afin de construire des topologies robustes qui garantissent le passage à l'échelle et des propriétés de tolérance aux pannes. L'indexation par l'utilisation des tables de hachage distribuées a en outre, révolutionné les environnements P2P impliquant ainsi un accès rapide à l'information avec un routage efficace dans le réseau. Ces techniques continuent à être développées et connaissent, de ce fait, une extension vers les réseaux ad hoc mobiles.

Un aspect crucial de l'optimisation des réseaux pair à pair à grande échelle, est la gestion de l'interaction inter-pairs et de la distribution du contrôle, dans un environnement où les pairs arrivent et partent à tout moment. ; C'est une propriété de la conception où la distribution des données et des contrôles a lieu d'une manière automatisée, ne nécessitant aucune gestion centralisée. Ceci donne en effet naissance à la conception de *systèmes auto organisés*.

Les réseaux P2P étaient, tout au départ, dominés par les applications de messagerie instantanée et de partage de fichiers ; plus tard, c'est au tour de la Voix sur IP (VoIP) et aux grilles de calcul (Grid Computing). Depuis peu, la télévision a rejoint le pair à pair : les systèmes de IPTV (TéléVision sur IP) ont migré d'un schéma client /serveur vers une architecture distribuée, tirant profit de l'agrégation des ressources respectives de chaque nœud qui forme le réseau (t.q. Puissance de calcul, stockage et bande passante) pour partager des flux vidéos à travers des communautés coopératives. Différentes solutions P2P peuvent être distinguées pour la distribution de vidéo, avec différents fournisseurs et utilisateurs.

Actuellement, les développements de différentes solutions de streaming peer to peer ont vu le jour, engendrant un ensemble de clients propriétaires, que ce soit pour la Vidéo à la demande (VoD) ou en directe (live streaming), des projets de IPTV ambitieux pour la diffusion de TV internationales en directe à travers le monde en combinant du P2P et du client/serveur.

1.2. Mobile ad hoc networks

Dans les communications classiques, dites d'infrastructure, des appareils mobiles utilisent la communication radio à un saut pour accéder à une station de base qui le connecte directement à l'infrastructure. À contrario, les communications ad hoc n'exigent aucune infrastructure pour communiquer. Les nœuds ad hoc mobiles fonctionnent, à la fois, comme hôtes, serveurs et relais. Ils communiquent entre eux directement, par des liaisons à un seul ou plusieurs sauts 'ils ne sont pas à portée.

Comme ces nœuds sont mobiles, la création du routage est affectée par la perte et la création de liens. La topologie du réseau peut donc changer rapidement et de façon imprévisible. Les MANETs (Mobile Ad-hoc NETworks) sont utiles dans de nombreux environnements (catastrophe naturelle, communication en combat, réseau de capteurs) et n'ont nul besoin d'infrastructure. Les communications dans les régions plus petites peuvent être mise en place grâce aux MANETs.

1.3. Contexte & Motivations

Aujourd'hui on peut constater les facteurs suivants qui affectent de manière importante les réseaux : multimédia, mobilité. Le trafic vidéo est classé parmi les applications les plus consommatrices en bande passante sur internet. Ce trafic qui a même détrôné le partage de fichier, dans le classement des application les plus gourmandes, avec l'apparition de différent moyens « libres » de production et de publication de contenus audiovisuels sur internet utilisant des serveurs dédiés à leur stockage. Ce phénomène de production individuelle et massive de vidéos a généré de gros volumes de données qui posent alors un problème aux FAI (Fournisseur d'accès à Internet) notamment par sa consommation démesurée de la bande passante. Ce qui n'a cessé de s'amplifier avec l'apparition du phénomène peer to peer streaming sur la toile. Malgré les avancées dans les domaines des méthodes de compression numérique, le volume croissant des vidéos mises en ligne, par les internautes, ne cesse de préoccuper les opérateurs.

La consommation d'énergie est une autre caractéristique contraignante dans les réseaux et plus particulièrement pour les appareils mobiles. En effet, le fait que ces derniers comptent sur leurs batteries pour s'alimenter en puissance, les rend plus vulnérables que leurs homologues fixes, qui eux ont une source d'énergies permanente. Ce qui fait que le réseau doit prendre cet handicap en considération, pour éviter que les terminaux fragiles n'épuisent complètement leurs batteries à cause du routage et de la persistance des informations.

Aujourd'hui de plus en plus de terminaux sont utilisés de manière mobile, les téléphones deviennent de vrais ordinateurs connectés à l'Internet, les ordinateurs portables jadis fonctionnant en autonome sont maintenant quasiment connectés à Internet tout le temps grâce aux clés 3G.

Les capacités bornées, de la mémoire et de la puissance de calcul, des dispositifs mobiles les contraignent à supporter une qualité d'encodage vidéo (BitRate) limitée. Ceci est à prendre en compte dans la conception d'applications destinées à un déploiement sur des réseaux ad hoc mobiles. Depuis leurs apparitions, les équipements mobiles ne cessent de s'améliorer. Ils reçoivent de plus en plus de puissance de calcul avec des processeurs qui sont équivalant à ceux des premiers ordinateurs de bureau, leurs permettant ainsi une certaine puissance d'exécution tel que le décodage vidéo entre autres.

Dans ce travail de thèse nous nous sommes intéressés au support de la QoS des applications multimédia durant la mobilité. Nous avons voulu traiter, d'une part, la diffusion de service vidéo dans les réseaux sans fils infrastructure ou dans les réseaux ad hoc mobiles. Les problèmes principaux qui apparaissent sont le délai, la gigue ainsi que la qualité de l'expérience.

L'approche P2P a apporté une nouvelle manière d'assurer la QoS sur le filaire. Nous avons décidé de la porter sur les réseaux ad hoc pour faire du streaming. Après les avoir vu migrer le streaming vidéo des réseaux filaires vers le sans fil, nous restons très convaincus qu'il y a un réel besoin de porter ces applications peer to peer sur les MANETs, pour les raisons suivantes: (a) La couche routage des MANETs est parfois inappropriée quand il s'agit de fournir les services demandés par des applications mobiles sophistiquées (gourmandes en bande passante et puissance de calcul), et; (b) la nature imprévisible du canal radio ainsi que la mobilité des utilisateurs posent des problèmes majeurs au routage, nécessitant ainsi l'intervention d'une couche supérieure. La stratégie employée de nos jours, est celle de garder une certaine simplicité dans la définition des protocoles de transport et de routage MANET, et au besoin; de les compléter en utilisant des fonctions au niveau applicatif via des overlays ou des réseaux pair à pair.

1.4. Contributions de la thèse

Cette thèse est consacrée à l'étude de l'impact de la mobilité dans les réseaux sans fils avec et sans infrastructure. Il s'agit d'abord, d'analyser dans ce contexte ce phénomène dans les réseaux à infrastructure, ensuite, de développer cette approche aux réseaux sans infrastructure. Dans le cadre de deux projets de recherches européens^{1,2}, nous avons pu proposer et valider, en collaboration avec notre équipe, des solutions pour les deux cas de figure.

Dans le premier travail, nous nous sommes intéressés au maintien de la QoS de l'application multimédia à des utilisateurs mobiles connectés au réseau infrastructure WLAN (Wireless Local Area Network). Lors de ses déplacements le terminal de l'utilisateur se connecte à différents points d'accès alors que l'utilisateur continue à recevoir le service. Notre contribution consisté à proposer SM²Stream: un mécanisme de *mise en cache* (buffering) dynamique, guidée par la prédiction du handover, pour palier à la coupure du flux vidéo (streaming) lors d'un handover horizontal entre des cellules Wifi.

Dans le second cas, l'étude porte sur la diffusion vidéo sur des réseaux ad hoc mobiles. Tout particulièrement, les différents problèmes qu'engendre la dynamique générée par la mobilité. En

¹ SUMO (Service Ubiquity in Mobile and Wireless Realm) est un projet collaboratif du programme Européen ITEA.

² ExpeShare (Experience sharing in mobile peer communities) est un projet collaboratif du programme Européen ITEA.

plus de perturbations causées par les départs/arrivées des nœuds dans les réseaux peer to peer classiques (fixe). Ces méfaits peuvent se répercuter directement sur la qualité du service peer to peer en général et la qualité de l'expérience perçue par l'utilisateur dans le cas du streaming vidéo. C'est dans cette optique que nous avons proposé deux solutions appelées MADP2PStream et MadTorStream. Respectivement, un modèle analytique et une plate-forme de diffuser vidéo sur les réseaux ad hoc mobiles.

Le modèle se base sur une proposition de graphe pour le peer to peer streaming dont la dynamique est accentuée par une contrainte supplémentaire, celle des états des liens sans fils. Nous employons un modèle de mobilité basé sur les comportements sociaux. Ce modèle sera par la suite, validé par simulation dans des conditions de fonctionnement réelles des réseaux, en prenant en compte différents paramètres, et en utilisant NS-2, dans deux scénarios: le premier avec un protocole de routage proactif et le second avec un protocole de routage réactif.

1.5. Structure du document

La suite de ce manuscrit est structurée en six chapitres, après ce chapitre 1 introductif:

Le Chapitre 2: État de l'art, introduit les différents concepts et définitions en rapport avec l'objet de cette thèse et en rappelle les principaux résultats de travaux de recherches y afférents effectués dans le domaine des réseaux mobiles. Ainsi, sont présentées les différentes avancées dans le domaine : le streaming vidéo, les réseaux sans fils porteurs pour les MANETs et les systèmes peer to peer streaming tout en dressant un état des lieux qui se veut le plus exhaustif possible de tout ce qui se pratique dans cet art.

Le Chapitre 3: SM²Stream - Streaming Multimédia en Mobilité sans coupure est consacré à l'étude et réalisation du système de prédiction et d'exécution du Hand Over horizontal, tout en assurant la continuité du service vidéo lors de la mobilité entre des cellules WiFi (Wireless LAN). Cette étape porte sur la présentation de l'algorithme retenu pour estimer le temps avant ce HandOver, les détails de son implémentation ainsi que celle du mécanisme de mise en cache adaptatif pour palier aux coupures durant le changement de cellule.

Le chapitre 4 : MADP2PStream - un modèle pour le streaming p2p mobile s'intéresse à la modélisation des réseaux peer to peer streaming sur internet. Sur la base de l'étude des propriétés de la mobilité, nous proposons une extension de ce modèle pour supporter une mobilité communautaire, basée sur les réseaux sociaux. Et ce afin de démontrer la faisabilité de la solution de streaming peer to peer, sur les réseaux ad hoc mobile. Des évaluations analytiques seront exposées dans la deuxième partie de ce chapitre.

Dans le *Chapitre 5 : MadTorStream - un protocole pour le streaming p2p mobile*, nous étudions tout d'abord les propriétés nécessaires à un protocole de streaming P2P pour qu'il fonctionne sur un réseau MANET. Nous proposons l'étude d'un protocole de partage de fichier existant pour envisager le streaming sur un réseau MANET. Nous traitons un scénario sur un protocole de routage proactif. Le même modèle de mobilité sera utilisé: *community based mobility model*. Le but étant l'évaluation de la qualité du streaming peer to peer dans différentes variations de la vitesse des nœuds.

Le Chapitre 6 est un récapitulatif de toutes nos contributions, apportées précédemment, ainsi qu'une analyse de l'expérience tirée de nos travaux de recherches. Enfin, une discussion sur les perspectives de nos travaux futures clôturera cette thèse.

Chapitre 2. Etat de l'art

Dans ce chapitre nous introduisons différentes notions de base que nous avons étudié pour le besoin de cette thèse : le streaming vidéo sur réseaux sans fils, les applications peer to peer, les MANETs et enfin la synergie entre ces trois axes. Ceci nous permettra surtout de nous positionner par rapport à ces divers aspects faisant intervenir plusieurs domaines desquels nous nous sommes inspirés pour mener à bien nos travaux.

2.1. Le streaming vidéo

Le streaming est le processus de réception et d'affichage de l'information pendant que celle-ci est envoyée par un fournisseur. Les flux transmis et affichés sont du type multimédia (vidéo, audio ou les deux). D'autres types de médias (photos, livres électroniques, applications, etc) n'ont pas besoin de cette méthode de transfert, car n'ayant aucune contrainte de séquentialité, ils sont téléchargés entièrement avant d'être traités, exécutés ou affichés. Les motivations pour faire du streaming sont multiples:

Les flux audio et vidéo sont généralement volumineux, ce qui suppose un délai important de transmission. Cette latence peut ne pas être acceptable pour l'utilisateur. De plus, l'objet doit être stocké dans le disque de l'équipement récepteur, qui peut parfois être un problème à cause de la taille.

Par ailleurs, les flux multimédia ne sont pas souvent lus dans leur totalité. Les usagers visionnent généralement le début d'une vidéo ou d'une chanson pour en connaître le contenu avant de l'arrêter, puis l'arrêter au milieu. Dans cette situation il est inutile de télécharger tout le fichier audio ou vidéo et ainsi de gaspiller la bande passante du réseau. Les événements dits en direct, audio ou vidéo, créés et transmis en temps réel, doivent être transmis au fur et à mesure de leurs productions. Dans ce cas, il est impossible d'enregistrer préalablement le contenu à transmettre.

Le streaming est considéré comme un défi technologique, de part ses contraintes temps réel: une sorte de synchronisation (horodatage) des données doit être fournie par la source de l'information et correctement traitée à la réception afin d'afficher les données (vidéo ou audio) de façon adéquate. Le flux vidéo empruntent évidemment aujourd'hui le moyen de transmission le plus

économique, qu'est le réseau Internet. Ce streaming peut être envoyé de deux manières différentes: unicast ou multicast. L'unicast implique l'envoi d'un flux vidéo séparément à chaque destinataire, alors qu'en multicast, le même flux est envoyé dans le réseau, et est dupliqué proportionnellement au nombre d'utilisateurs.

2.1.1. Transport de flux sur le réseau

De nombreux protocoles de transfert et de méthodes dites d'encapsulation, ont été proposés pour le streaming: les canaux DVB (Digital Video Broadcast) utilisent le MPEG-TS (Motion Pictures Expert Group – Transport Stream), alors que la diffusion de médias sur Internet passe souvent par RTP (Real-time Transport Protocol), avec l'utilisation supplémentaire de RTCP (Real-time Transport Control Protocol), et RTSP (Real-Time Streaming Protocol).

Il existe également d'autres moyens d'envoyer des informations en mode streaming: directement sur UDP (User Datagram Protocol), qui propose un contrôle physique du réseau disponible; et sur HTTP (HyperText Transfert Protocol) via TCP (Transfer Control Protocol) permettant de traverser les pare-feux du réseau et d'exploiter l'utilisation commune et l'implémentation du protocole HTTP. RTSP est utilisé en extension de HTTP pour supporter le déplacement fluide dans le flux pendant sa lecture.

Malgré les propositions multiples de protocoles applicatifs, dédiés au streaming, la majorité des applications utilisent toujours TCP ou UDP, ceci peut s'expliquer par le fait que la couche transport est implémentée au niveau des systèmes d'exploitation qui se sont limités, dans le passé, aux protocoles TCP et UDP. Donc, dans le cadre de notre étude, nous utiliseront ces protocoles largement déployés actuellement sur les réseaux sans fils 802.11 [1] de l'IEEE.

2.1.2. Sur la couche applicative

Jusqu'à présent, les applications et les protocoles de niveau applicatif se contentaient d'exploiter les services qui sont disponibles au niveau des couches inférieures. Étant donné que les conditions de transmission sont dynamiques (débit, taux de perte, latence, gigue), les applications multimédia subissent les variations de ces conditions avec des réactions limitées, à leur niveau, en utilisant des mécanismes de QoS présentés précédemment. De nouvelles interactions, entre la couche application et les couches réseaux, permettront à ces mécanismes d'être plus dynamiques et plus adaptatifs pour une configuration adéquate suivant l'état instantané du réseau. De plus, les applications multimédia doivent être informées des mécanismes QoS déployés dans le réseau pour assurer une certaine qualité des informations importantes, par exemple une correspondance entre la couche de base dans le codage hiérarchique et les classes de service prioritaires au niveau IP et au niveau MAC.

2.1.2.1. *Mesure de la performance*

Dans le streaming multimédia en général, quelques paramètres interviennent dans la mesure de la qualité de ce service à savoir: Le débit, est la bande passante disponible de bout en bout sur le réseau. Le taux de perte BER (Bit Error Rate), est la proportion de paquets perdus, lors de leurs transmissions, sur le nombre total des paquets constituant la vidéo. La latence ou délai de transmission, représente le temps nécessaire pour traverser le réseau de bout-en-bout, de l'émetteur jusqu'au récepteur. Quant à la gigue, c'est la variation de la latence d'acheminement de bout-en-bout des paquets vidéo sur le réseau. Une métrique supplémentaire vient s'ajouter aux paramètres précédents est celle de la qualité perçue par les utilisateurs (QoE: Quality of Experience).

2.1.3. Utilisation du 802.11 pour le streaming

Dans toute cette thèse, les travaux que nous présenteront portent sur le streaming sur 802.11. Le WLAN sera utilisé dans les cas infrastructure et MANETs. Notre choix peut être justifié par le fait que ce standard de réseau sans fil soit le plus répandu et surtout le plus polyvalent quant aux applications qu'il peut avoir.

L'utilisation de liaisons sans fil pour le streaming vidéo sur Internet devient de plus en plus courante aujourd'hui. Cette combinaison ramène l'exigence des applications multimédia temps réel (intolérantes aux ruptures), à la nature très imparfaite - et capricieuse - des liaisons radio. Un grand effort est nécessaire pour joindre ces deux mondes, tels que le délai, la gigue, la perte pour que les exigences des applications multimédias puissent être satisfaites par l'instabilité et le manque de fiabilité des liens radio.

Les liens sans fil constituent des goulots d'étranglement pour un certain nombre de raisons. Tout d'abord, la communication sur un canal sans fil n'est tout simplement pas en mesure d'atteindre la même qualité (débit, taux d'erreur, etc), assurée par son homologue filaire, ce qui réduit la qualité du contenu multimédia livré. Deuxièmement, dans un environnement mobile, les conditions du canal peuvent changer rapidement en raison de la variation de la distance entre les stations (la mobilité des utilisateurs), l'évanouissement de Rayleigh, et les interférences. Alors que l'application multimédia doit livrer son contenu en temps réel, elle est très sensible à la gigue dans le transfert des paquets engendrée par les retransmissions des protocoles de transport. Troisièmement, le streaming multimédia peut se faire sur un support tel 802.11 et interférer avec les autres utilisateurs qui sont, par exemple, entrain de partager des fichiers.

2.2. Les applications p2p

Depuis leur apparition, les réseaux peer to peer, pair à pair ou p2p n'ont cessé de faire parler d'eux. Ce modèle de communication est passé par des différentes améliorations pour arriver aux applications que l'on connaît aujourd'hui (BitTorrent, Kazaa, Skype, Joost...). Dans ce chapitre nous retracerons les différents changements en citant surtout les travaux de recherche effectués dans ce domaine.

2.2.1. L'indexation et la recherche

Considéré comme précurseur, le développement de la recherche p2p a connu trois grandes phases, motivé chaque fois par un manque de performance dans ces protocoles de localisation de contenu. Sans trop s'attarder, nous citons les principales caractéristiques de cette évolution.

2.2.1.1. Approche centralisée

Appelée aussi à index central, ce système est basé sur un serveur central qui joue le rôle d'annuaire qui répertorie tous les pairs qui proposent du contenu. Les autres pairs du système consultent l'annuaire, comme un index, pour récupérer les coordonnées (adresse IP) de leurs futurs voisins. Napster [5] constitue l'exemple type d'approche centralisée

Soulignons toute fois, que cette approche n'est pas considérée comme totalement distribuée du fait de la présence du serveur central d'adresses, sans lequel le réseau ne pourrait fonctionner. Néanmoins, les fichiers sont directement échangés par les pairs.

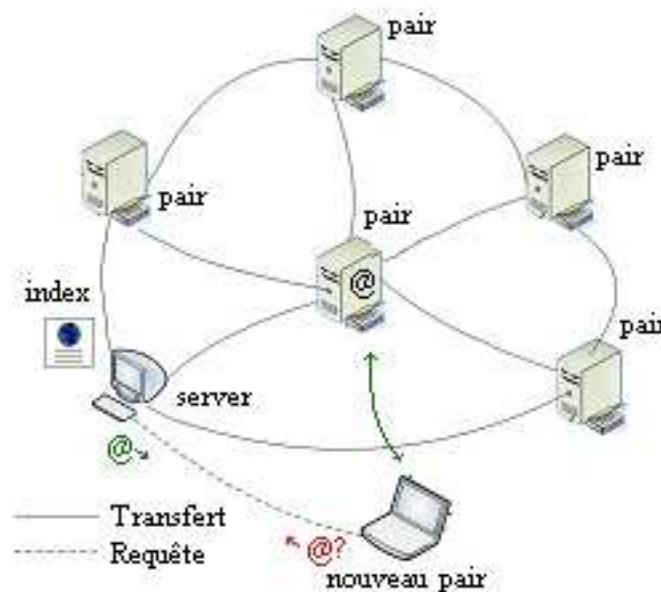


Figure 1: Réseau P2P de 1^{ère} génération (index central)

2.2.1.2. *décentralisée (distribuée)*

Chaque pair indexe ses propre fichiers il n'y a donc plus nécessité d'un serveur central. La demande d'un fichier transite de proche en proche dans tout le réseau, rendant ainsi le système plus robuste. Cependant, la recherche est plus difficile, car elle exige un nombre de messages, proportionnel au nombre de nœuds du réseau. Cette approche a été introduite et appliquée par le système Gnutella [2].

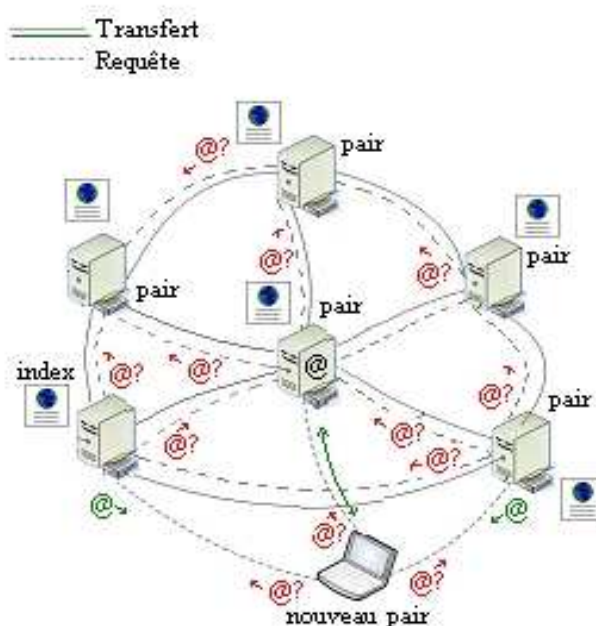


Figure 2: Réseau P2P de 2^{ème} génération (inondation de requête)

Contrairement aux applications de première génération, cette architecture est un véritable réseau P2P, puisqu'elle peut se passer totalement de l'entité centrale. Mais cette deuxième génération s'est vite éteinte; car le réseau était, du point de vue bande passante, plus ou moins hétérogène car, constitué à la fois d'ordinateurs connectés à haut débit et d'autres connectés par modem, les premiers s'imposaient donc avec leur vitesse de connexion ce qui ralentissait manifestement l'ensemble du réseau ; générant ainsi, une inertie globale du système.

a. *Tables de hachages distribuées DHT:*

Le principe de ces tables de hachage repose sur l'utilisation d'une fonction de hachage pour faire correspondre à chaque nom de fichier son empreinte; c'est en fait une sorte de *résumé crypté* du fichier, sous forme d'une chaîne de bits (une succession de 0 et de 1) de longueur fixe (souvent 128 ou 160 bits). Ce nombre devient l'*identifiant* du fichier. Les nœuds en charge d'une empreinte sont ceux dont l'identifiant est le plus proche.

Cette technique permet l’optimisation de l’architecture précédente, ce qui assurerait la réalisation des recherches par un nombre de messages en croissance logarithmique par rapport au nombre de pairs. Le réseau OVERNET [3] est un réseau précurseur en matière de DHT [4].

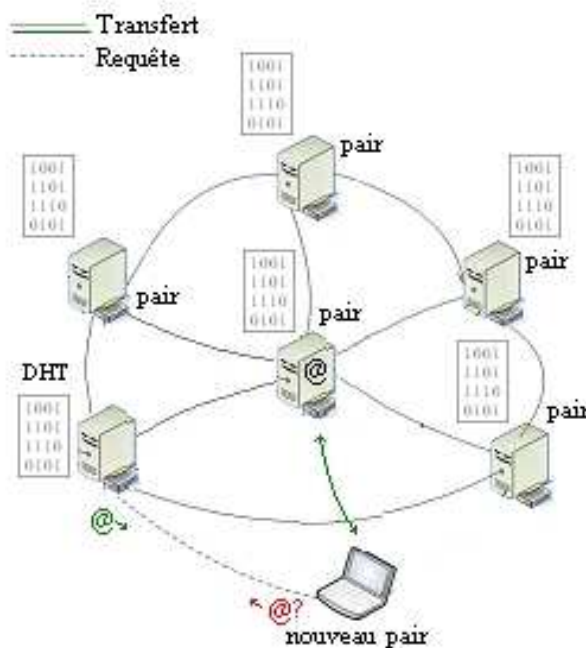


Figure 3: Table de hachage distribuée (DHT)

Dans cette approche, grâce à la structuration des connaissances, les nœuds en charge de l’empreinte sont retrouvés en routant la requête de proche en proche en un nombre d’étapes logarithmique [4] par rapport au nombre de nœuds. La recherche est donc beaucoup plus efficace: une dizaine d’échanges de messages au lieu de quelques centaines voire quelques milliers si la donnée cherchée n’est pas répliquée.

Pour annoncer le partage d’un fichier, la source doit contacter le nœud en charge de l’empreinte du fichier pour lui indiquer son adresse IP. Un pair qui recherche ce fichier n’aura qu’à contacter ce même nœud pour retrouver l’adresse IP du pair (ou des pairs) le partageant.

b. Hybride

Il s’agit d’un compromis entre les deux générations précédentes. En effet, une distinction est apportée aux pairs sur deux niveaux : des pairs à haut débit et des pairs à connexion classique.

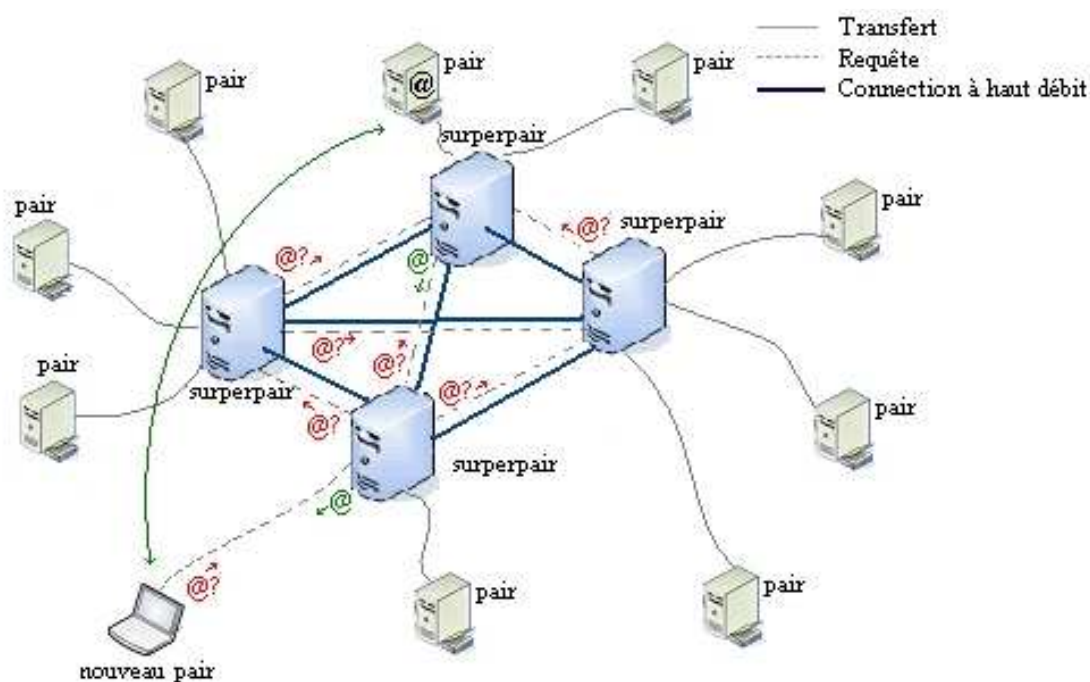


Figure 4: *Le super peering*

Les ordinateurs à connexion classique se connectent à ceux de haut débit, des superpeers qui indexent les données et qui agissent en tant que serveur distribué. Ces supernœuds agissent entre eux comme les pairs de la deuxième génération, le transfert de messages entre eux est plus rapide vu qu'ils ont des connexions à haut débit. La connexion directe entre pairs ne se fait qu'une fois l'adresse recherchée est obtenue.

2.2.2. Application de partage de contenu

Nous présentons différents types d'applications p2p pour le partage de contenus. Nous en traiterions celles qui ont eu un grand développement que ce soit pour l'échange de fichier, la voix sur IP ou le multimédia streaming ; sans pour autant omettre de rappeler qu'il y eut un grand intérêt scientifique quant à l'étude de ces programmes. Ceci nous oriente sur les investigations à entreprendre et leurs corollaires : la compréhension, l'analyse et surtout l'interprétation des résultats de nos recherches, car ce que nous étudierons plus tard, n'est qu'une adaptation des notions utilisées dans ces applicatifs. Ce qui nous orientera, petit à petit, vers nos propositions de streaming pour MANET.

2.2.2.1. Partage de fichier (*File sharing*)

Depuis l'apparition de Napster [5], bon nombre d'applications de file-sharing on vu le jour à travers le monde. Avec des différentes philosophies et modes opératoires. Parmi les plus populaires nous verrons: Gnutella, Fasttrack, eDonkey et BitTorrent.

a. *Gnutella*

Gnutella [2], Ce protocole a été imaginé en 2000 par Tom Pepper et Justin Frankel, des programmeurs pour la société NullSoft. Diffusé peu de temps sur Internet, il put être adopté par plusieurs programmeurs et sur différentes plateformes [6]. L’idée de ce protocole vient du fait que les réseaux qui existaient (i.e. Napster) à l’époque, étaient vulnérables; et n’assuraient pas une liberté totale à cause des serveurs centraux, souvent espionnés. Gnutella est l’un des réseaux dits purement p2p car il n’utilise pas de serveur central, avec un contrôle totalement distribué. Parmi les logiciels exploitant ce réseau: MLDonkey, Bearshare, Shareaza. Dans la première version de Gnutella (0.4), un pair est connecté à un ensemble de voisins, qu’il interroge en leur envoyant un message de recherche. Ses voisins le renvoient à leurs voisins, et ainsi de suite.

Un TTL (Time To Live), représentant le nombre de retransmissions, est associé à cette requête. Quand sa valeur s’annule, le message correspondant n’est plus retransmis ; Cependant, les pairs qui possèdent le fichier recherché, répondent par un message (contenant le nom de ce fichier et leurs adresse IP) qui parcourt le chemin inverse jusqu’au pair qui a lancé cette requête. Ce mécanisme, connu sous le nom d’inondation (flooding), est coûteux en bande passante, et les recherches sont plus lentes qu’en centralisé et augmente exponentiellement avec le nombre de pairs [7][8]. Cette limite a incité les développeurs à proposer la notion d’ultrapeers ; celle-ci a été incluse dans la version 0.6 de Gnutella avec le client LimeWire en 2001.

Dans le réseau Gnutella [6], Les pairs sont classés en deux catégories : Les ultrapeers sont choisis parmi les nœuds les plus stables du réseau, du point de vue connectivité. Et les feuilles (leafs), raccordés à trois ultrapeers sachant que chacun des ces ultrapeers peut se connecter jusqu’à 45 nœuds et 30 autres ultrapeers. Les ultrapeers indexent les fichiers de leurs voisins et se chargent de répondre aux requêtes sur tout cet index. Ces messages sont reçus par les ultrapeers ne sont retransmis qu’à leurs homologues ; cela constitue une protection (shield) aux clients feuille, et préserve la performance des connectés à bas débit.

GUESS (Gnutella UDP Extension for Scalable Searches) est un mécanisme grâce auquel les clients feuille contrôlent le nombre d’ultrapeers à interroger et donc réduisent leur bande passante. En fonction de la grande (resp. petite) popularité de la ressource recherchée le client peut diminuer (resp. augmenter) le nombre d’ultrapeers à interroger. Avec le mécanisme par inondation, le nombre d’ultrapeers interrogé est incontrôlable. GUESS propose alors, de laisser à l’initiateur de la recherche le soin d’interroger successivement un ensemble d’ultrapeers, jusqu’à ce qu’il obtienne un nombre de résultats satisfaisants.

Lorsqu’un nouveau client arrive sur le réseau il ne possède aucune information sur les pairs. Afin d’éviter ce problème, un script appelé *bootstrap*, est installé sur les serveurs web, et permet au

client Gnutella de récupérer une liste des pairs à qui se connecter. Ce script est accessible via une URL via le protocole HTTP, et possède deux types de requêtes : *GET* : Récupère une liste d'adresse IP des nœuds Gnutella et d'URLs d'autres serveurs GwebCache. *UPDATE* : Informe le GWebCache qu'un nouveau ultrapeer ou serveur GWebCache est présent sur le réseau. La liste du serveur GWebCache est de taille limitée, il n'y insert que les dernières adresses reçues, pour garantir leurs disponibilités et augmenter la chance qu'elles soient encore accessibles à la prochaine re-connexion.

b. FastTrack

Ce protocole, utilisé par trois clients populaires: Kazaa, Grokster et iMesh, a été introduit en Mars 2001 par Nokklas Zenstörm, Janus Friis et Jaan Talinn. FastTrack [9] est un protocole P2P de seconde génération, utilisant la notion de superpeers. Si un peer assez puissant (CPU, Débit) se connecte au réseau il est promu supernœud, et agit comme un serveur temporaire d'index pour les autres clients.

Pour pouvoir se connecter au réseau, chaque nœud contacte les supernœuds, stockés dans une liste dans le logiciel utilisé. Dès qu'il obtient une adresse d'un supernœud actif il lui demande la liste codée d'adresse d'autres supernœuds actifs, qu'il utilisera lors de ses prochaines reconnections. Le client choisi alors l'un des supernœuds comme son « ascendant », et lui envoie la liste des fichiers qu'il désire partager afin qu'ils y soient indexés. Il lui envoie également des requêtes sur des fichiers qu'il veut obtenir.

Le supernœud communique avec d'autres supernœuds afin de satisfaire ces requêtes. Une fois la réponse trouvée elle remonte jusqu'au client concerné, celui-ci se relie directement au pair pour télécharger le fichier en HTTP.

Pour permettre les téléchargements de sources multiples, FastTrack utilise l'algorithme de brouillage UUHash. Cet algorithme permet la vérification de fichiers volumineux en peu de temps, même sur des ordinateurs lents, mais il ne peut pas détecter une corruption massive d'un contenu [10]. Ce qui a encouragé plus d'un utilisateur à diffuser des fichiers corrompus sur le réseau.

Le protocole FastTrack utilise un cryptage qui n'a pas été documenté par ses créateurs. De plus, les premiers clients étaient propriétaires. Cependant, les données d'initialisation pour les algorithmes de cryptage sont envoyées en clair et sans clé publique ; ainsi son analyse a été relativement facile. En 2003, des programmeurs open-source ont réussi à disséquer la partie du protocole traitant la communication client-supernœud, mais celle qui parle de transmission supernœud-supernœud demeure en grande partie inconnue.

c. eDonkey

Lancé en Septembre 2000 par Meta-Machine Inc., Le réseau eDonkey fait partie des réseaux p2p hybride tels que Napster [11] et est composée de clients et de serveurs. Un serveur eDonkey joue le rôle d'un hub de communication et d'un serveur d'indexes qui distribue les adresses des autres serveurs aux clients qui le sollicitent. Les clients, eux, téléchargent et partagent les fichiers. Étant un protocole ouvert, eDonkey possède une douzaine de versions de ses clients et serveurs dont le client le plus populaire était eDonkey2000.

Cependant, depuis 2005, eMule, un logiciel gratuit basé sur le client eDonkey à été publié en Mars 2002 par Hendrik Breikreuz et est devenu le plus populaire de clients. Dans ses premiers jours, eDonkey comptait sur des serveurs d'utilisateurs volontaires pour garantir le bon fonctionnement de son réseau. Néanmoins, ce petit cluster était souvent surchargé et vulnérable aux attaques. Pour résoudre le problème de surcharge des serveurs et de passage à l'échelle, la DHT Kademia [3] a été intégrée aux deux clients, Les réseaux eMule et eDonkey2000 sont ainsi appelés respectivement Kad et Overnet.

d. BitTorrent

BitTorrent [12], conçu par Bram Cohen en 2002, est un protocole ouvert écrit initialement en python, mais disponible aujourd'hui dans différents langages de programmations. Pour partager des fichiers avec BitTorrent, il faut tout d'abord créer un Torrent; un fichier de méta informations qui décrit le contenu à partager. Ce méta-fichier porte des renseignements sur: le *tracker* (superviseur du partage), et les hashes (identifiant crypté) des blocs du fichier échangé. Le protocole spécifie la façon dont ce fichier est distribué. En fait, le torrent est publié via le web (site, forums, lieux de recherche spécifiques...). Pour obtenir un fichier, le client BitTorrent doit commencer par introduire le fichier torrent correspondant. Il contacte alors le tracker, qui peut être unique ou distribué (cas de la version trackerless). Ce nœud obtient alors une liste de pairs partageant le même fichier. À tout moment, basé sur un mécanisme d'incitation appelé tit-for-tat, BitTorrent donne des blocks de son fichier aux voisins qui lui ont en le plus offert dans le passé. Puis, il en sélectionne un au hasard pour assurer ce qu'il appelle *optimistic unchoking*.

Un client BitTorrent télécharge le bloc le plus rare du voisinage. Cette stratégie, Local Rarest First, augmente la probabilité de la distribution des nouveaux blocks et donc une utilisation efficace du débit montant de chaque nœud.

Un certain nombre de caractéristiques uniques ont permis à BitTorrent, un dernier venu dans le partage de fichiers P2P, de devenir un utilitaire de partage de fichiers très populaire. Tout d'abord, BitTorrent lui-même n'offre aucun mécanisme de recherche pour trouver les fichiers partagés, il se dégage ainsi de l'accusation de violation de droit d'auteur. Deuxièmement, son

mécanisme de partage de contenu Tit-for-Tat, bien que simple, est remarquablement efficace et encourage les utilisateurs de BitTorrent, à contribuer leurs ressources. Ce qui le rend, par conséquent, moins vulnérable au problème de *free-riding* qui affecte les autres applications de partage de fichiers P2P.

En troisième lieu, grâce à l'enregistrement des haches des blocs dans le torrent, BitTorrent permet d'éviter la pollution [10] qui menace les applications de partage de fichiers. Quatrièmement, le fait de diviser le fichier en plusieurs blocs, et grâce à une stratégie privilégiant le téléchargement les blocs les plus rares en premier, permet d'optimiser l'utilisation de la bande passante des pairs et de parvenir à une efficacité élevée. Effectivement, BitTorrent est considéré comme la première application P2P de partage de fichiers largement déployée [6], offrant une telle efficacité et robustesse. BitTorrent a généré beaucoup d'enthousiasme quant à la distribution de fichiers P2P dans les universités et l'industrie.

Bon nombre de projets de logiciels libres, par exemple, OpenOffice.org, SUSE et Ubuntu, utilise BitTorrent pour diffuser leur codes source et binaires compilés. World of Warcraft, un jeu de rôle en ligne, utilise BitTorrent pour distribuer ses patches. Warner Brothers Entertainment prévoit de distribuer des films et émissions de télévision sur Internet via BitTorrent. Un certain nombre de conférences ont aussi distribué leurs enregistrements vidéo par le biais de BitTorrent.

e. Statistiques

D'après CacheLogic [6], BitTorrent, eDonkey, FastTrack et Gnutella sont les clients de partage de fichiers en p2p les plus populaires aujourd'hui. Dans la figure 5 nous montrons les parts de trafic dans différents pays pour ces quatre utilitaires d'échanges p2p. Généralement, BitTorrent et eDonkey sont de loin les plus populaires, le réseau FastTrack vient bien après en troisième position pour laisser la quatrième et dernière place à Gnutella. Ce classement reste différent d'un pays à l'autre. Par exemple, en Corée du Sud eDonkey détrône BitTorrent, alors que c'est l'inverse à Singapour. Ils sont presque ex aequo aux États Unis et en Chine.

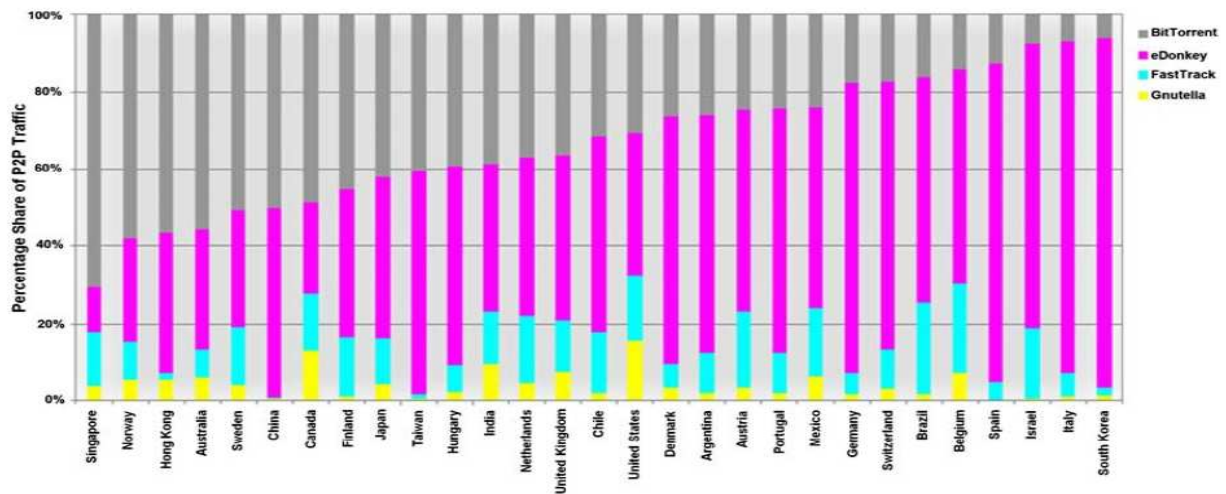


Figure 5: Parts des clients (*BitTorrent, Kazaa, eMule et Gnutella*) sur le trafic P2P file sharing [6]

2.2.2.2. Voix sur IP

Skype [13] est le client p2p VoIP le plus populaire. Il a été développé par les créateurs de Kazaa. Skype permet à ses utilisateurs de passer des appels et envoyer des messages texte à d'autres utilisateurs. En terme de fonctionnalité, il est très similaire aux applications de messagerie instantanée MSN et Yahoo, comme l'appel voix, messagerie instantanée, les conférences audio et les listes d'amis. Toutefois, les protocoles et les techniques qu'il emploie sont très différents.

Skype peut fonctionner de façon transparente dans presque tout type de NAT et pare-feu, et avec une meilleure qualité de la voix que les autres clients VoIP. Il crypte ses appels de bout en bout, et stocke les informations de ses membres de façon décentralisée.

Il existe deux types de nœud dans le réseau Skype, des nœuds ordinaires et des supernœuds (SN). Le nœud ordinaire est un client Skype, qui effectue des appels et envoie des messages texte. Tout nœud avec une adresse IP publique, ayant suffisamment de CPU, mémoire et bande passante du réseau est un super nœud potentiel. Une fois authentifié via le serveur de connexion de Skype, un nœud se connecte à un supernœud. Le serveur de connexion est une entité dans le réseau de Skype gérant les noms d'utilisateurs et leurs mots de passe. Ce serveur assure que les noms de connexion de Skype sont uniques dans l'ensemble de l'espace de nom Skype.

Outre le serveur de connexion, il y a les serveurs SkypeOut et SkypeIn [2] qui fournissent, respectivement les ponts de PC-PSTN et de PSTN-PC. Ceux là n'interviennent pas dans les appels PC à PC. Ainsi, nous pouvons dire que le serveur de connexion est le seul élément central dans le réseau P2P Skype. Les informations des utilisateurs, en ligne et hors ligne, sont stockées et propagées de façon décentralisée. Chaque nœud de Skype utilise une variante de STUN [14], un protocole qui détermine les types des NAT et pare-feu derrière lesquels il se trouve le nœud.

Le réseau de Skype est un Overlay. Par conséquent, chaque client Skype (SC) a besoin de construire et d'actualiser une table de nœuds accessibles. Dans Skype, ce tableau est appelé host cache (HC) et contient l'adresse IP et le numéro de port des supernœuds. Skype met en place une structure P2P de troisième génération ou Global Index³, garantissant de trouver un utilisateur si ce dernier s'est connecté au réseau au cours des 72 dernières heures. Skype utilise une technique de recherche d'utilisateurs, combinant hachage et inondations périodiques limitées, pour obtenir des informations sur les utilisateurs en ligne. Pour les recherches infructueuses Skype lance une requête au serveur de connexion.

2.2.2.3. Streaming audio et vidéo

Les informations publiées par CacheLogic [6] montrent également que la taille moyenne d'un fichier partagé en P2P est en constante augmentation, et la majorité du volume du trafic P2P est générée par des objets avec une taille moyenne de plus de 1 GB. Cela suggère que le partage de fichiers P2P évolue fortement vers le partage de vidéo/musique. Jusqu'à ce que nous entrons dans l'ère du streaming P2P, un mode de transfert beaucoup plus favorable pour les fichiers multimédia volumineux. Inspiré du principe de partage de fichier, cette technologie consiste en un ensemble de clients qui coopèrent pour s'échanger du contenu vidéo entre eux. En d'autres termes, un nœud est entrain de lire une vidéo pendant qu'il retransmet les paquets déjà reçus à ses voisins dans l'overlay. Par rapport au partage de fichiers, une contrainte additionnelle vient s'ajouter, celle de la date limite avant laquelle ces paquets doivent arriver dans le buffer du client.

Actuellement, plus d'une douzaine de sociétés travaillent activement dans ce domaine. Certaines entreprises (Abacast⁴, PPLive⁵, PPStream⁶, UUSee⁷, Roxbeam⁸, Mysee⁹, etc).

Logiciel P2P orientés vers la vidéo en streaming, parfois appelé Internet TV), recrée un flux localement et le lit ensuite via Windows Media Player, Real Player, VLC ou d'autres lecteurs multimédias. Les clients P2P actuels sont: PPLive, PPstream, TvAnts¹⁰, SopCast¹¹, et Joost¹². Certains d'entre eux, comme TvAnts, fondé sur la technologie BitTorrent, permettent de regarder

³ Global Index (GI): http://www.skype.com/skype_p2pexplained.html

⁴ Abacast Online, URL: <http://www.abacast.com/>

⁵ PPLive.com, URL: <http://www.pplive.com/>

⁶ PPStream, URL: <http://www.ppstream.com/>

⁷ UUSee, URL: <http://www.uusee.com/>

⁸ Roxbeam media network, URL: <http://www.roxbeam.com/>

⁹ Mysee, URL: <http://www.mysee.com/>

¹⁰ TVAnts, URL: www.tvants.com

¹¹ Sopcast, URL: www.sopcast.com

¹² Joost. URL: www.joost.com

les médias (chaînes TV). Joost est un client hybride conjuguant multicast et p2p pour le streaming des chaînes télévisée du monde entier. Dans ce qui suit nous détaillerons le fonctionnement de deux clients p2p streaming, les plus utilisés. Puis nous rappellerons les notions relatives au fondement du p2p streaming.

a. Pplive

En utilisant PPLive, sans doute l'un des clients de streaming P2P les plus populaires de nos jours. Vers la fin 2005, PPLive enregistre 20 millions de connectés, et 1 million de téléspectateurs par jour. Il supporte plus de 200.000 utilisateurs avec un débit entre 400 et 800 kbps pour le Festival du printemps 2006 lors du Gala du Nouvel An chinois le 28 Janvier 2006. En 2007, le nombre d'utilisateurs simultanés pour une session PPLive s'est élevé à 1,5 million [15].

Ceci correspond à un débit global (par agrégation) avoisinant les 600 Gbit / s, 540TB transférés en 2 heures d'événement. Au cours duquel, le serveur PPLive n'a fournit 10 Mbps de bande passante (0,0017% du débit global). D'après une analyse préliminaire en [16], PPLive fonctionne comme suit. Lorsque le client PPLive est lancé, il récupère d'un serveur de chaîne les méta-informations de toutes les chaînes. Actuellement, PPLive offre environ 300-400 chaînes, avec tous des téléspectateurs regardant la même chaîne et sont à peu près au même point du flux. Le client PPLive présente une liste de chaînes à l'utilisateur, qui choisit une chaîne à regarder.

Après la sélection, le clients PPLive contacte le tracker de ce canal, et récupère une liste des pairs auxquels il se connecte, et commence à échanger des données. Pour se faire, le flux multimédia est découpé en morceaux d'une seconde chacun. Le pair PPLive échange une carte de disponibilité des blocks dans sa fenêtre de lecture, qui est de l'ordre de quelques minutes, et prend la décision sur les morceaux à récupérer et/ou envoyer grâce à un algorithme propriétaire. Les morceaux reçus sont stockés dans le buffer du client Pplive, et sont acheminé via un pipe Http vers Windows Média player ou Real player, selon le type du fichier source.

Dans l'analyse de [15], nous apprenons que l'actuelle plate-forme PPLive provoque un gel de lecture d'environ minute, avec une fréquentes de 4 incidents en 7 minutes. L'expérience de PPLive démontre que le streaming vidéo à grande échelle peut être réalisé. Mais il y a encore de l'amélioration à faire, pour augmenter son efficacité et sa robustesse. PPLive encourt également un long retard de lecture. Aussi, à son démarrage, PPLive peut entraîner un retard d'environ 20s à 30s pour les chaînes populaire, et qui atteint les 2 minutes pour un canal impopulaire, certains pairs peuvent regarder des images avec quelques minutes de décalage par rapport à d'autres.

b. Joost

Joost est l'un des premiers clients de streaming p2p commerciaux. Il peut fournir de la télévision de haute qualité, à la demande. Nous avons tenté de découvrir [17] les divers aspects de fonctionnement de ce protocole en étudiant le comportement de ses nœuds, et en analysant le trafic réseau et les logiciels libres utilisés dans ce client. Sans surprise, nous avons découvert que Skype et Joost ont certains mécanismes et techniques p2p en commun.

Ceci nous a permis de: (i), déduire l'architecture globale de Joost et certains de ses composants clés en se basant sur une étude méticuleuse de son trafic réseau, (ii) approfondir nos connaissances quant au déroulement du streaming et le comportement des pairs, (iii) Évaluer sa performance et son utilisation de la bande passante.

Bien que Joost soit une technologie peer-to-peer de distribution vidéo, il s'appuie fortement sur quelques serveurs centralisés pour fournir les contenus vidéo. Nous pensons que le caractère centralisé de Joost est le principal facteur influant sur son niveau de considération de la localité et son faible ratio d'équité. Du point de vue du réseau, Joost consomme environ 700 kbps en download et 120 kbps en upload, quelle que soit la capacité totale du réseau. C'est en supposant que la capacité du réseau en amont, est de plus de 1Mbps. Joost utilise le port 33333 pour envoyer des données vidéo à partir de sa baie de serveurs (une aux US et une en Europe). Pour réduire considérablement la quantité de données dans le réseau, il faut bloquer le port 33333.

Joost repose principalement sur ses serveurs dédiés de contenu en tant que nœuds source afin de distribuer la vidéo. Les technologies P2P sont utilisées pour l'assister, assurant ainsi la distribution de vidéo et le passage à l'échelle. Dans Joost, les supernœuds sont seulement utilisés pour contrôler le trafic, ils ne relayent jamais les contenus vidéo. Les principaux flux sont envoyés à partir des Joost Seeders. Tout le trafic est crypté afin de partager du contenu vidéo sécurisé contre le piratage. Les pairs mettent en cache le contenu reçu et le retransmettent quand cela est nécessaire. Aidant, ainsi, les autres pairs à recouvrir les blocs perdus suite à une congestion.

Joost n'utilise pas les ressources des pairs de manière efficace, en particulier lorsque des nœuds de grande capacité sont disponibles. En d'autres termes, Joost pourrait être légèrement plus demandeur en termes de débits montants, vis-à-vis des pairs à grandes capacités. En effet, dans des moments de grandes affluences des nœuds, le protocole pourrait envisager l'utilisation de ces aptitudes pour récupérer, à la volée, la liste des chaînes, actuellement téléchargée depuis le serveur, ce qui permet de décharger le serveur. Joost fournit actuellement chaque client avec la même qualité de la vidéo. Ceci provoque une utilisation inefficace des ressources dans la mesure où certains clients sont incapables de supporter une telle qualité de la vidéo. Au final, nous pensons

que des techniques d'adaptation en couches de la vidéo, conjugués à certaines stratégies d'incitation, devrait être introduits dans Joost.

2.2.2.4. Les deux topologies du streaming P2P

Deux modèles régissent le streaming p2p, que ce soit pour la vidéo à la demande ou le streaming. Ces topologies sont le Mesh-Based, un réseau maillé et le Tree-Based, un arbre de diffusion applicatif.

a. Topologie mesh

Provenant de la recherche sur les protocoles épidémiques, l'approche fondée sur le maillage (dans ce qui suit nous utiliseront *mesh*) repose sur l'échange périodique d'information sur l'état des nœuds, ce qui conduit à une diffusion de l'information à tous les nœuds. BitTorrent, protocole de partage de fichiers, est celui qui a vulgarisé l'utilisation du mesh [12]. En effet, ce protocole construit un mesh non structuré pour échanger des fichiers de données. Le fichier est divisé en blocs, échangés par les nœuds dans un mode pull (voir définition dans 2.2.2.6-b) jusqu'à ce que les nœuds puissent reconstituer le fichier original. Pour le streaming vidéo sur le mesh, la topologie est construite aléatoirement: chaque nouveau nœud se connecte à une liste de voisins, obtenus au hasard, avec qui, il commence directement à échanger des blocs de la vidéo. Ces paquets circulent dans le réseau d'une manière aléatoire qui dépend de l'algorithme de sélection de nœuds et de blocs.

b. Topologie Arbre

Dans cette approche de distribution de contenu, les nœuds s'organisent en une sorte d'arbre multicast applicatif (ALM Application-Layer Multicast). ESM [18], Scattercast [19] et Overcast [20]. Sont des travaux qui emploient ce type d'arbre, où la racine est la source du stream, les autres nœuds coopèrent dans la re-transmission des pièces de la vidéo déjà reçues. Dans ces systèmes, comme CoopNet [21], SplitStream [22], MutualCast [23] et FastReplica [24], le nœud source crée l'arbre, et les nœuds qui arrivent deviennent ses descendants directs, et ainsi de suite. En fonction de son temps d'arrivée le pair s'attache le plus profondément dans l'arbre. Une livraison en cascade de la vidéo entre les nœuds de l'arbre, à sens unique des parents vers les enfants. En d'autres termes, cela consiste en une procédure de réplication engagée par la racine, qui maintient overlay, jusqu'au dernier nœud feuille de l'arbre. Chaque nœud de l'arbre retransmet autant de copies du bloc reçu que d'enfant il possède. Cette approche n'est pas évolutive car en arrivant, un nœud s'attache à un seul parent et à un certain nombre fixe de fils. Aussi, dans le cas du départ d'un nœud, la maintenance sera très complexe causant une rupture dans la distribution de la vidéo à tous ses descendants.

c. Comparaison

Pour avoir une idée sur le fonctionnement des deux topologies, dans le tableau suivant, nous présentons une comparaison entre les mesh et les arbres.

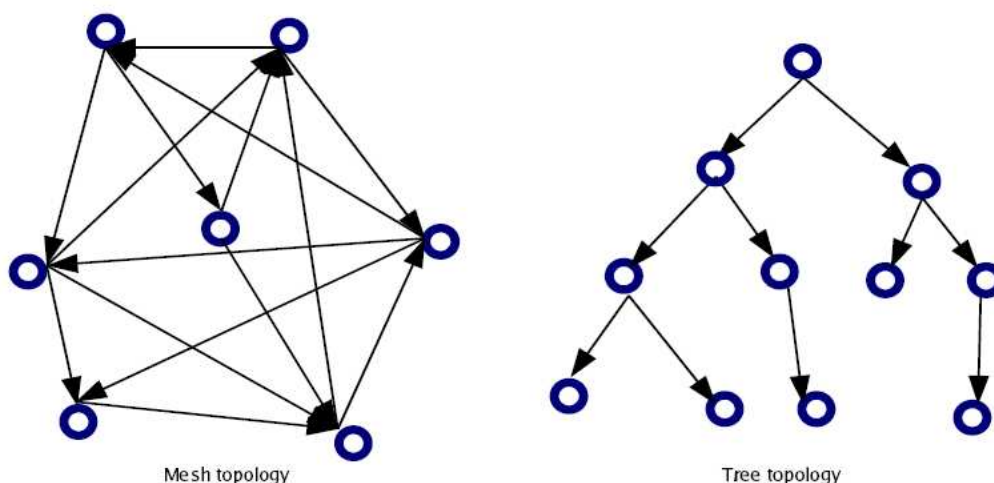


Figure 6: topologies Mesh vs. Tree.

Aspects	Topologies	
	Arbre	Mesh
Source	Un seul nœud offre la vidéo	Plusieurs nœuds contribuent
auto-organisation	A l'arrivée il s'attache en bas de l'arbre, en profondeur	A l'arrivée le nœud obtient une liste de nœuds avec qui il va coopérer
Maintenance	Difficile à maintenir, après le départ d'un parent, tous ses descendants deviennent orphelins.	Messages hello (ping) suffisent à chaque nœud pour maintenir ses voisins.
Dynamique	La stabilité du système est détériorée par la dynamique des nœuds.	Pas d'effets sur la topologie grâce aux connexions aléatoires renouvelées à chaque départ/arrivée
Utilisation optimale de la bande passante	Pas de contribution de débit montant des nœuds feuilles	Bonne coopération du débit montant et descendant. Surtout que chaque nœud envoie/reçoit à/de plusieurs voisins à la fois.
Mobilité	La topologie des arbres ne résiste pas au départ des nœuds même s'il est active autre part.	Pendant son déplacement le nœud prévient ses voisins et obtient une nouvelle liste dès qu'il arrive à sa destination.

Tableau 1 – Comparaison entre topologies Mesh et Tree

De ce tableau comparatif il apparaît clairement que la topologie mesh est beaucoup plus robuste que celle basée sur les arbres. Dans la structure d'arbre, un nœud ne reçoit que de son parent direct. Donc, lorsque celui-ci tombe en panne, se déplace ou quitte tout simplement les feuilles, tous le sous-arbre (composé par l'ensemble de descendants du nœud en question) ne parvient pas à obtenir le flux jusqu'à ce que l'arbre soit reconstruit. Dans les réseaux streaming sur mesh, la vidéo est obtenue à partir de plusieurs voisins, et dans ce cas, si l'un d'entre eux échoue, d'autres continueront à fournir le stream, encore faut-il qu'au moins une copie de chaque bloc existe. De ce fait, nous concluons que le système orienté mesh demeure le meilleur candidat.

Néanmoins, un problème que rencontrent les structures en mesh, en général est celui de ne pas compter sur une structure réseau prédéfinie et sont donc, plus difficiles à étudier que les topologies arbre. Or, dans [25], les auteurs montrent que, même dans les mesh, des motifs d'arbres de diffusion apparaissent dynamiquement dans l'overlay. Ces modes de diffusion peuvent alors être étudiés de la même manière que les structures basées sur les arbres.

La contribution de ce travail est de construire une topologie de streaming p2p sur MANETs. Compte tenu des similitudes entre ces derniers, la structure mesh est donc plus adaptée dans ce scénario. Par ailleurs, il a été prouvé que la réduction des hauteurs d'arbres de diffusion dans un mesh de streaming réduit considérablement le retard subit par la mise en cache de la vidéo [25] ce qui motive davantage le choix de cette topologie.

2.2.2.5. *Stratégies de sélection de bloc et de nœuds*

Que ce soit en mode push ou pull, le nœud doit décider d'une stratégie de sélection de bloc, c'est-à-dire, quel bloc demander/offrir, et de/à qui le bloc est reçu/donné. Nous appelons cela la sélection de pairs et la sélection de blocs.

D'un protocole p2p à un autre, que ce soit pour le streaming ou le file sharing, les nœuds utilisent différentes stratégies pour sélectionner des nœuds et envoyer ou demander des blocs. Chaque stratégie possède ses avantages et inconvénients. Ce qui nous intéresse est de savoir quels sont les changements à faire, sur la stratégie d'échange de blocs, quand nous passons du file-sharing au streaming. Ceci sera étudié en détail dans notre contribution dans le paragraphe 5.2.4.

a. *La sélection de pairs*

Dans cette sélection un pair choisit le nœud à qui il va donner des blocs. Pour le mode pull, le récepteur choisit généralement les pairs émetteurs en fonction de leurs capacités d'envoi. Par exemple, PeerStreaming [25] choisit des pairs qui offrent un moindre délai entre le moment de la demande et celui de la réponse. Pour le mode push, l'expéditeur peut sélectionner ses destinataires sur la base de leurs équités ou performances. Par exemple, BitTorrent utilise une stratégie

récompensant l'équité, qui est le tit-for-tat. Le nœud BitTorrent donne aux voisins qui lui ont donné le meilleur upload. Cela empêche le free riding, et encourage les pairs à contribuer.

Une stratégie alternative repose sur la mesure de la performance. Par exemple, DISCOVER [26] alloue plus de bande passante à l'envoi aux pairs qui ont pratiquement fini de remplir leur buffer. Cette stratégie de sélection permet aux nœuds avec des ressources importantes d'épauler les pairs avec peu de ressources, et améliore ainsi la qualité de service dans l'ensemble du système.

b. La sélection de bloc

Cette stratégie choisit un bloc sur une liste de blocs disponibles chez un voisin. Quelques schémas de sélection bloc [27]: (a) Sequential: Sélectionne le premier bloc disponible; (b) Rarest: Choisie le bloc le plus rare dans le voisinage local du pair. Il n'existe aucune méthode pour départager les blocs avec la même rareté. BitTorrent utilise cette stratégie de sélection de blocs. (c) Random: Sélectionne un bloc au hasard; d) Random rarest: Choisi le bloc le plus rare dans le voisinage local du pair. Si plusieurs blocs sont de même rareté, un bloc est sélectionné au hasard.

Les résultats de BitTorrent et BulletPrime [27] montrent que la stratégie de sélection random rarest est la meilleure en termes d'efficacité, la sélection random et rarest succèdent de près, tandis que la stratégie de sélection séquentielle vient en dernière position pour ce qui est du file sharing. En fonction du sens de la circulation des blocs, ceux là peuvent être soit demandés, offerts ou les deux. Les trois modes de diffusion sont utilisés dans le P2P.

2.2.2.6. Technique de transfert des blocs

a. Le mode push

Dans le mode push (aussi appelé le mode sender-driven), l'expéditeur est à l'initiative, et envoie le bloc déjà reçu aux voisins choisis auparavant. Nous notons que le mode push dans un arbre est assez différent de celui du mesh. Le premier n'a pas besoin de consulter les récepteurs pour la validité du bloc, ce qui permet de réaliser un plus faible retard d'acheminement. Nous observons également que, dans les distributions orientées mesh ou hybride (paragraphe suivant) sont plus efficaces en utilisation des débits montants du réseau P2P. Parce que dans les deux cas, l'expéditeur est l'initiateur, il peut facilement faire en sorte que son débit sortant disponible soit pleinement utilisé. DISCOVER [26], une plate-forme P2P combinant partage de fichiers et streaming multimédia, utilise un mesh en mode push. Le pair collecte les informations relatives à la disponibilité des blocs chez ses voisins, et propose un ensemble de blocs qu'il souhaite leur envoyer. À la réception de cette liste, le voisin rejette certaines propositions de blocs s'ils sont déjà proposés ou en cours de téléchargement, et confirme pour le reste. Le transfert de blocs est alors initié par l'expéditeur après avoir reçu la confirmation de la liste. Avalanche [28], un outil de

partage de fichiers P2P utilisant le network coding (mécanisme mélangeant les blocs lors de leur livraison), est aussi un protocole push dans une topologie mesh. Dans lequel, l'expéditeur propose d'abord au récepteur le vecteur du code du bloc transmis. Le récepteur vérifie si ce vecteur contient de nouvelles informations, et l'accepte ou le rejette. L'expéditeur et lui envoie alors le bloc codé après sa confirmation.

b. Le mode Pull

Dans le mode pull (aussi appelé axé-récepteur 'receiver-driven'), le récepteur est à l'initiative de prendre le bloc qu'il veut de son voisin. Ce mode est plus avantageux par rapport au maintien de la qualité de service (QoS) pour le récepteur. Parce que celui-ci peut choisir les blocs à recevoir, il a donc un meilleur contrôle du processus de livraison, et peu facilement assurer la livraison à temps d'un bloc ceci est essentielle dans le multimédia streaming P2P. Par exemple, BitTorrent fonctionne selon un modèle pull. Un nœud échange avec ses voisins des informations, sur les blocs qu'il détient, et décide quel bloc récupérer parmi ceux disponibles dans son voisinage.

c. Le mode hybride

Dans un mode hybride, où à la fois, le récepteur et l'émetteur peuvent prendre l'initiative de négocier le transfert de blocs. Les réseaux P2P basés sur les arbres adoptent généralement un mode push, dans lequel les pairs poussent simplement les blocs reçus vers les pairs qui se trouvent en aval. Dans les mesh P2P le choix du mode de transfert est plus souple, les nœuds peuvent soit utiliser le pull, le push ou bien combiner les deux.

GridMedia [29], une des premières plateformes P2P streaming, adopte ce mode hybride de push-pull sur une topologie orientée mesh. Les blocs y sont classés en deux catégories: des blocs pull et des blocs push. Lorsqu'il rejoint le réseau, le nœud utilise le pull pour obtenir les premiers blocs, puis utilise le mode push pour envoyer les blocs à ses voisins afin de réduire les délais.

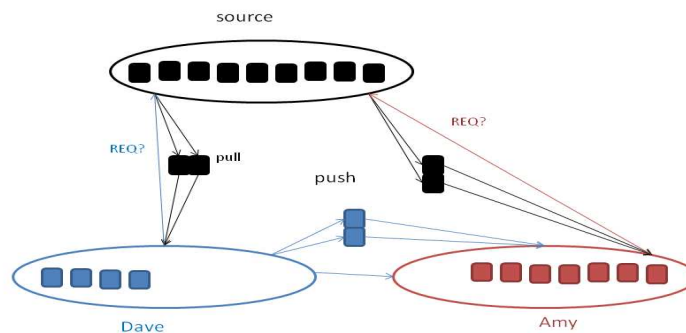


Figure 7: Push vs. Pull.

2.3. Les réseaux ad hoc mobiles

Un réseau *ad-hoc mobile* est un réseau créé par une réunion spontanée de mobiles capables de communiquer grâce à leurs interfaces sans fils, sans aucune infrastructure ni configuration préalable. Les nœuds d'un tel réseau coopèrent pour la gestion du réseau. Un MANET (Mobile Adhoc NETWORK), se caractérise par sa méthode de découverte de nœuds, son algorithme de routage, et le modèle de mobilité régissant le mouvement des mobiles.

2.3.1. Classification des protocoles de routage

Depuis leurs apparitions, les MANETs ont été très étudiés surtout les aspects : routage, découverte et maintenance des nœuds mobiles. Une multitude de protocoles de routage a été proposée dans la foulée. Ces derniers sont classés en trois grandes familles: (a) les proactifs (b) les réactifs, et (c) les hybrides (Figure 8). Ne pouvant pas tous les citer, dans ce qui suit, nous allons présenter en détail un protocole de routage pour chacune des familles précédemment citées.

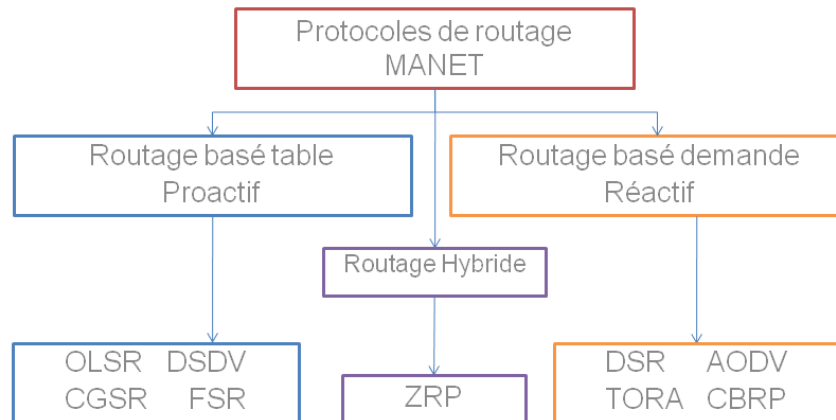


Figure 8: Classification des familles de protocoles ad hoc

2.3.1.1. Protocoles proactifs

Dans ce type de routage, chaque nœud construit une table de routage par la découverte d'autres nœuds dans tout le réseau, une opération coûteuse pour le réseau à haute dynamique, parmi ces protocoles OLSR et DSDV.

a. OLSR (Optimized Link-State Routing)

Un des protocoles réactifs les plus utilisés dans la communauté, OLSR [30] est proposé par des chercheurs de l'INRIA et fait l'objet de travaux de standardisation depuis 2003 au sein de la IETF dans le groupe MANET. C'est un protocole proactif basé sur une approche modifiée, dont la version originale où chaque nœud propage des informations sur l'état des liens à tous les nœuds du réseau. L'altération apportée par OLSR consiste à réduire l'overhead en ne diffusant cette information que par quelques nœuds du réseau.

Dans ce protocole proactif, chaque nœud transmet sa liste de voisins en émettant périodiquement une sorte de beacon. De cette manière chaque nœud est au courant des nœuds qui circulent dans son voisinage à 2 sauts. Il utilise un algorithme de sélection de relais multi-points MPR (Multi-Point Relay). L'ensemble des relais d'un nœud N donné est construit à partir de l'ensemble minimal des voisins à un saut de N . De telle sorte que chaque voisin à deux sauts de N ait au moins un même MPR. Chaque nœud dans OLSR choisit indépendamment ses relais multi-points. La seule connaissance dont il a besoin est celle relative à ses voisins à deux sauts.

Quand un nœud diffuse un message, tous ses voisins le reçoivent. Les MPR n'ayant pas reçu ce message auparavant le rediffusent. Donc, l'overhead causé par l'inondation diminue considérablement dans OLSR. Le nœud diffuse périodiquement l'information sur l'état des liens avec leurs MPRs dans le réseau. La périodicité de l'envoi de ces messages est sujette à la détection du changement de l'ensemble des MPRs. Dans ce cas précis, la période prend alors sa valeur minimale. Dans le cas contraire cette valeur augmente jusqu'à ce qu'elle atteigne la valeur de l'intervalle de rafraîchissement (refresh interval). Chaque nœud reçoit, en plus, une information sur la topologie du réseau et construit alors sa table de routage grâce au message d'état de lien. Ce routage utilisé dans OLSR n'inclut que les MPR comme nœud intermédiaire entre deux nœuds donnés. Plusieurs implémentations prêtes à déployer existent ainsi qu'un bon nombre de travaux de tests de performance de OLSR et de son support de la QoS.

2.3.1.2. Réactif

Cette approche propose de ne construire une route vers la destination que si un nœud a besoin de communiquer. Ceci permet de résister à la mobilité mais génère une surcharge supplémentaire (overhead) dans le réseau à chaque tentative de communication.

a. AODV (Ad Hoc On-demand Distance Vector)

Dans la famille des routages réactifs, le protocole AODV [31] est l'un des plus célèbres. Dans cette approche, quand un nœud source S a des données à envoyer au nœud destination D , mais ne possède pas tout le cheminement vers celle-ci, il doit lancer une Route REQuest (RREQ). Ce

paquet est alors inondé à travers le réseau. Chaque nœud recevant une RREQ, la fait suivre à ses voisins, s'il ne l'a pas déjà fait au moins une fois, ou s'il n'est pas le nœud destination de ce message, et sous la condition que le compteur Time to Live (TTL) ne soit pas à zéro.

Le paquet RREQ comporte l'identifiant de la source (SrcID), l'identifiant de la destination (DestID), numéro de séquence de la source (SrcSeqNum), numéro de séquence de la destination (DesSeqNum), l'identification de la diffusion (BcastID), et le TTL. Le DesSeqNum indique la fraîcheur du chemin accepté par la source. Lorsqu'un nœud reçoit un paquet RREQ, et s'il en est le destinataire ou connaît un chemin valide jusqu'au destinataire, il renvoie à la source du message une réponse Route REPLY (RREP). Sinon, il retransmet le RREQ à ses voisins.

Le nœud intermédiaire est en mesure de déterminer si un cheminement est valide ou non, en comparant son propre numéro de séquence avec le DesSeqNum dans le RREQ. Un nœud intermédiaire est en mesure de savoir si le RREQ est déjà traité ou non, en comparant le BcastID au SrcID. Lors de la retransmission du RREQ, chaque nœud intermédiaire y injecte l'adresse du nœud le précédant et son BcastID.

Un timer est utilisé pour supprimer cette entrée dans le cas où un RREP paquet n'est pas reçue avant sa date d'expiration. Lors de la réception d'un paquet RREP, un nœud intermédiaire mémorise aussi des informations sur le nœud à partir duquel ce paquet est reçu. Ce nœud étant le saut suivant lors de la transmission des données au nœud destination qui a initié le RREP.

2.3.1.3. *Hybride*

Il résulte d'une combinaison des avantages des deux familles précédentes, ce qui offre un bon compromis entre persistance de la route et surcharge du réseau. En général, les propositions hybrides exploitent la hiérarchisation du réseau en utilisant des protocoles réactifs et proactifs et sont alors intégrés à différents niveaux de la hiérarchie.

a. ZRP (Zone Routing Protocol)

Ce protocole, proposé par Haas et al. [32] en 2002, combine les deux approches vues précédemment, proactive et réactive, pour en tirer le maximum d'avantages et combler ainsi les désavantages de chacune par les points forts de l'autre. Il consiste en la division du réseau, en fonction de la distance entre les nœuds, en plusieurs parties appelées zones de routage. ZRP utilise deux approches proactive (IARP) pour le routage dans la même zone (intra-zone routing protocol) et réactive (IERP) entre les zones (inter-zone routing protocol). L'IARP maintient une information sur l'état de lien des nœuds à une distance d . Dans le cas où le nœud source et la destination sont dans la même zone, la route est alors disponible immédiatement. La plupart des protocoles proactifs existants pourraient être utilisés en tant que IARP dans ZRP. Lorsque la

source et la destination appartiennent à deux zones différentes, l'IERP effectue une découverte de route de la même manière que c'est dans les protocoles réactifs, sauf qu'elle passe par les nœuds périphériques.

2.3.2. Modèles de mobilité

Largement utilisés dans la mesure de performance des réseaux ad hoc mobiles en particulier et sans fils en général, les modèles de mobilité sont souvent utilisés pour simuler et évaluer la performance des algorithmes et des protocoles dans les réseaux ad hoc mobiles. La définition réaliste du modèle de mobilité est l'un des aspects les plus critiques et difficiles de la simulation d'applications et de systèmes conçus pour les environnements mobiles.

Essentiellement, Il existe deux types de schémas de mobilité: ceux basés sur les traces et les synthétiques [33]. Les traces sont obtenues au moyen de mesures de déploiements des systèmes et sont généralement constitués des traces de la connectivité ou des informations localisées, alors que les modèles synthétiques sont des modèles mathématiques, tels que des systèmes d'équations, qui essaient de capturer le mouvement des mobiles.

Un survol des modèles de mobilités proposés pour MANET peut être consulté dans [33], [34]. Ces modèles varient en fonction de leur complexité, leur structure de corrélation ainsi que des restrictions sur les mouvements. Cependant, aucun d'eux n'est assez détaillé pour s'adapter à toutes les situations.

Actuellement, les données qui existent sont des captures de traces de circulation des personnes, issues des traces GPS et Bluetooth (c'est-à-dire, contenant les identifiants Bluetooth des mobiles se trouvant dans la portée d'un appareil). Par exemple, les chercheurs des Laboratoires de recherche d'Intel et de l'université de Cambridge, ont distribué des équipements Bluetooth à des personnes, afin de recueillir des données sur les mouvements humains et d'en extraire les caractéristiques des modèles de leurs mouvements dans le réseau. Des expérimentations ont été menées en premier lieu par des étudiants et des chercheurs de Cambridge [35] et, ensuite, par des participants de INFOCOM 2005 [36]. Exemples de projets similaires sont le Wireless Topology Discovery à UCSD [38] et le Campus-Wide Wi-Fi Traffic mené au Dartmouth College [39].

2.3.2.1. *Modèles synthétiques*

a. Modèles basés sur la marche aléatoire

Le Random Walk est un modèle de mobilité simple [40][41], aussi appelé le Brownian motion [40], il est largement utilisé comme modèle pour représenter les mouvements purement aléatoire des entités d'un système dans les différentes disciplines de la physique à la météorologie. Toutefois, il

ne peut être considéré comme un modèle pour simuler des environnements sans fil, car les mouvements humains ne présentent pas de continuité dans les changements de direction qui caractérisent ce modèle de mobilité.

Dans [42][43], les auteurs présentent Random Trip, une généralisation des Random Walk et Random Way-Point. Les auteurs introduisent une technique qui dérive l'état initial de la simulation d'un régime stationnaire (une méthode qui est généralement appelée simulation parfaite) sur la base du Palm Calculus [44] afin de résoudre le problème de stationnarité-temporelle. Cette méthode de simulation du Random Way-Point a été proposée par Navidi et Camp dans [45].

b. Le Random Way-point

Dans ce modèle, un nœud mobile se déplace de sa position actuelle vers une nouvelle destination choisie au hasard dans l'espace de simulation, avec une vitesse aléatoire distribuée uniformément entre deux vitesses, minimale et maximale $[V_{\min}, V_{\max}]$. Le Random Way-Point inclut le temps de pause lors d'un changement de vitesse et de direction. Dès qu'un nœud mobile arrive à sa nouvelle destination, il se met en pause pour une période de temps (temps de pause) avant de reprendre son mouvement.

Les propriétés analytiques du Random Way-Point ont été traitées dans plusieurs travaux du point de vue de la distribution stationnaire des nœuds [46][47], leurs distributions spatiales [48] et de l'évolution de la répartition des nœuds par le biais d'équations aux dérivées partielles [49].

2.3.3. Group Mobility Model

Nous avons vu que ces modèles sont utilisés pour représenter les mouvements des nœuds mobiles seuls, cependant, dans certaines situations, le comportement des hôtes mobiles se déplaçant ensemble, tels que des pelotons de soldats, un groupe d'élèves ou des collègues en conférence doivent être pris en compte lors de la modélisation de la mobilité. Pour ces raisons, les modèles de mobilité de groupe ont été proposés, comme le Reference Group mobility model [50], le modèle Reference Velocity Group [51] et le modèle de mobilité Structured Group [52]. Ces modèles sont basés sur une série d'équations qui relient les mouvements d'un nœud à la position d'un sous-ensemble des autres nœuds du réseau.

Ces modèles sont utiles pour reproduire les scénarios caractérisés par la présence de groupes d'individus, cependant, les mouvements générés ne reflètent pas ceux observés dans la réalité, dans lesquels les groupes se déplacent de façon aléatoire dans l'espace de la simulation. Les mécanismes d'appartenance sont également figés, un nœud unique ne peut pas rejoindre d'autres groupes durant la simulation. Récemment, Piorkowski et Alii ont proposé un modèle synthétique

appelé Heterogeneous Random Walk [53], qui est capable de reproduire la présence de clusters observés dans les traces réelle. L'objectif de ce modèle est de disposer d'un modèle mathématique permettant d'étudier et d'expliquer l'émergence de réseaux en clusters.

2.3.4. Community based mobility model

Dans [54][55], Musolesi et al. décrivent un nouveau modèle de mobilité basé sur la théorie des réseaux sociaux. Issue de la combinaison des modèles synthétiques basés sur les groupes, des traces réelles et des réseaux sociaux. Partant du constat que les appareils mobile sont transportés par des humains, leurs mouvements sont donc fortement affectés par leurs relations.

Ce modèle de mobilité permet à des collections d'hôtes de se regrouper selon les relations sociales entre leurs possesseurs. Ce regroupement est ensuite reflété sur un espace topographique; avec des mouvements influencés par la force des liens sociaux, et pouvant aussi changer dans le temps. Comparé à des traces réelles, ce modèle de mobilité affiche une très bonne approximation du mouvement humain.

2.4. P2P sur MANETs quelle motivation ?

Après avoir défini les applications peer to peer et les réseaux MANET, dans ce paragraphe nous regardons dans quel possibilité nous pourront combiner ces deux paradigmes pour palier au problème de la mobilité dans les applications peer to peer. Initialement posée lors du séminaire de Dagstuhl [56], cette problématique connaît un développement sur différents plan. Elle consiste à étudier les possibilités de tirer le plus grand profit en combinant les technologies p2p aux réseaux ad hoc mobiles. Parmi ces applications, le transfert efficace de données nous intéresse plus particulièrement dans cette thèse.

Comme nous l'avions vu, un MANET est une collection de nœuds mobiles qui s'auto-organisent en un réseau sans fil sans recourir à une infrastructure préexistante. Dans un MANET, les applications suivent généralement un modèle *égal-à-égal* plutôt que *client-serveur*. Un MANET est souvent construit pour une application spécifique, c'est un réseau orienté application. Pour ces raisons, le comportement d'un réseau ad hoc est souvent qualifié de peer-to-peer (P2P). Ce n'est cependant pas tout à fait compatible avec le sens général de P2P dans un contexte plus large, celui de l'Internet.

Dans l'infrastructure fixe (i.e. Internet), un réseau P2P est essentiellement un réseau overlay (sur la couche application) motivé par la nécessité d'avoir des fonctions spécifiques qui ne sont, parfois, pas possibles (ou non rentables) à réaliser dans les couches inférieures (couche IP). Ces fonctions doivent être effectuées par soit par l'application soit par un protocole d'inter-couche *middleware*. Des exemples classiques de ces solutions dédiées à Internet sont les overlays temps réel multicast

(qui permettent de surmonter le manque des routeurs pour soutenir le multicast IP), les systèmes P2P non structurés tels que Gnutella, BitTorrent, et KaZaa, et ; structurés tels que Pastry [57], Chord [58], ou CAN [59]. Ces derniers sont généralement implémentés comme des réseaux à contenu adressable et assurent une certaine efficacité quant au routage des données, qui est par ailleurs, pas possible sur la couche IP.

D'un autre coté, MANETs sont basés sur une architecture de réseau qui est tout à fait différente de celle définie pour l'Internet filaire. En fait, une architecture typique MANET nous rappelle les architectures introduites dans les années 80 pour inter-connecter des réseaux locaux, avec un routage au niveau 2 basé sur les adresses MAC (Medium Access Control). À ce niveau, le routage MANET est non-hiérarchique et est présenté donc, à plusieurs couches sous-jacentes à l'application P2P. Cela implique que, dans un MANET le concept P2P doit être intégré au-dessus de la couche 2.

2.4.1. Différences

La plus importante différence entre un réseau MANET et le P2P est leur raison d'exister. Alors qu'un MANET est créé pour inter-connecter des terminaux à des fins de communications (axées utilisateur). Ce qui motive la création d'un réseau P2P est différent. En raison de la grande capacité d'un réseau P2P, l'objectif principal de la plupart de ses utilisateurs est de rechercher et partager des données dans le réseau sans avoir nécessairement envie de communiquer avec d'autres utilisateurs. Dans les réseaux P2P, l'échange des données se fait via un lien direct à partir d'un pair à l'autre. Les pairs intermédiaires sont juste utilisés pour configurer une connexion entre ces deux et ne sont ni nécessaires ni voulu dans la transmission des données. Dans un réseau ad hoc l'action des nœuds intermédiaire est essentielle. Ceux-là participent activement dans la découverte d'itinéraire, ainsi que l'établissement de la connexion. Par conséquent, les connexions dans MANET sont indirects, alors que le P2P celles-ci sont directes. Aussi, Il faut noter, que le lien direct d'une connexion P2P est beaucoup plus facile à maintenir qu'un lien indirect multi sauts dans un réseau ad hoc. De plus, dans le cas de mobilité des nœuds intermédiaires, le nombre de ré-établissement de routes augmente considérablement.

Une autre distinction entre ces deux réseaux se trouve dans leur structure. Une couche virtuelle est créée pour connecter les ordinateurs d'un réseau P2P. Le séparant ainsi, du réseau physique. De ce fait, des ordinateurs qui sont à des milliers de kilomètres les uns des autres peuvent être voisins sur la couche P2P. En revanche, la position physique des nœuds dans un MANET peuvent être estimées approximativement, à cause de la densité de leur distribution dans une surface donnée. À côté de cela, la structure physique d'un réseau ad hoc reflète directement sa topologie logique.

Il existe aussi des différences dans l'utilisation des algorithmes de routage entre les réseaux P2P et de MANET. Plusieurs publications [60][61] montrent que la dynamique de routage ad hoc système ne fonctionne que dans les petit MANETs, avec moins de 100 nœuds. Avec un plus grand nombre de nœuds, les mises à jour doivent être envoyées plus fréquemment. Ce qui fait que le réseau soit préoccupé plus par la signalisation que par le trafic de données. À part le problème du passage à l'échelle, les algorithmes de routage proactifs sont réalisables dans les réseaux ad hoc, alors qu'ils ne fonctionnent pas du tout dans un réseau P2P. En effet, la charge générée par l'utilisation, dans de grands réseaux P2P, d'un routage basé sur l'algorithme de Bellman-Ford [62], est trop élevée. L'utilisation d'un tel algorithme de routage dynamique dans un réseau P2P, impliquerait, la transmission des listes d'adresses de tous les pairs vers tous les autres pairs, qui ne manquera pas d'échouer.

Comme mentionné précédemment, l'algorithme de routage P2P est exécuté uniquement lors d'une requête de recherche, donc il peut être déclaré comme un algorithme de routage réactif. En dépit de l'utilisation d'algorithmes de routage réactifs similaires dans les deux réseaux, l'exécution est différente. Dans un réseau ad hoc de la recherche se termine, c'est-à-dire le message de requête n'est plus retransmis, lorsque son destinataire est trouvé, ou si l'un des nœuds intermédiaire connaît un itinéraire vers cette destination. En revanche, cette requête ne s'arrête pas dans un réseau P2P, même si ce nœud qui possède le contenu fichier recherché est localisé. La requête est toujours retransmise et ne s'arrête lorsque le champ TTL est égal à zéro.

La fiabilité des canaux dans la structure physique diffère entre réseaux P2P et MANET. Si liens câblés sont de très haute fiabilité, des liens sans fil souffrent toujours de changements imprévisibles, ce qui provoque de nombreuses erreurs. Par conséquent, le taux d'erreur d'une liaison sans fil est d'une plus grande ampleur. Ainsi, la création d'une nouvelle route sur plusieurs sauts s'annonce être plus facile dans un réseau P2P.

La position, fixe pour un nœud P2P, ou mobile pour nœud ad hoc mobile également, influe sur la disponibilité des ressources (bande passante, batterie, mémoire et puissance de calcul). Si un terminal P2P a des ressources presque illimitées, un nœud ad hoc celles-ci sont très limitées.

En résumé, malgré les différences séparant les réseaux ad hoc sans fil et peer-to-peer, (résumées dans le tableau 2), nous verrons par la suite, qu'il se trouve que ces deux là possèdent beaucoup de similitudes, ce qui est décrit dans le paragraphe suivant.

Différences	Peer-to-Peer	MANET
Raison de création	Recherche et partage de ressources sur Internet	inter-connexion de terminaux pour établir une connexion
Connexion entre les nœuds	Filaire et directe sur la couche application	Sans fil impliquant plusieurs nœuds intermédiaires
Fiabilité des connexions	Haute, grâce aux liens filaires	Faible, due aux liens radio
Structure	Sur-couche virtuelle indépendante de la couche physique	La couche logique correspond à la structure physique
Taille du réseau physique	Peut couvrir le monde	Les nœuds sont densément distribués
Routage des requêtes	S'arrête quand sont TTL est nul	S'arrête si la destination est trouvée
Mobilité des nœuds	Fixe	Mobile
Disponibilité des ressources	Pratiquement illimitée	Très limité

Tableau 2 – MANETs vs.P2P: Différences

2.4.2. Similitudes

Peer-to-Peer et MANET, se basent sur le concept d'auto-organisation. Dans la plupart des cas, à l'exception des protocoles P2P hybrides [63], aucune entité centrale, de gestion et de coordination du réseau n'existe. Le réseau est créé, dès que les nœuds y participant, décident de se connecter. Ainsi, le réseau change en permanence de topologie, parce que le comportement des nœuds altère fréquemment les connexions existantes.

Ce changement fréquent de la topologie est un autre aspect commun des deux réseaux. Dans les MANETs, cela est causé par la mobilité des nœuds. Ainsi, les connexions déjà établies avec d'autres nœuds se perdent, quand le terminal mobile sort du champ de transmission de ses voisins. Dès lors, les nouveaux liens doivent être établis avec de nouveaux voisins pour maintenir la connectivité.

Dans les réseaux P2P, la mobilité n'est pas un challenge, mais les fréquents des arrivés/départs des nœuds sont problématiques. La durée moyenne de session est estimée aujourd'hui à 2 heures par jour, en moyenne [64]. De plus, la dynamique de l'affectation des adresses IP par les serveurs et les traducteurs d'adresses causent d'autres problèmes, comme l'incapacité de retrouver un nœud par une adresse IP stable. Ainsi, il est presque impossible de réunir en un court laps de temps toutes les informations sur l'ensemble du réseau, en raison de la dynamique de changements.

Une autre question cruciale qui se pose concernant l'auto-organisation, est ce que les inondations sont nécessaires dans tout les niveaux. Comme les deux réseaux sont basés sur une topologie en constante évolution, les éléments du réseau doivent être régulièrement sondés, pour vérifier la disponibilité des liens et des nœuds. Cela n'est possible que par la diffusion de messages, en l'absence d'un gestionnaire centralisé.

Un autre problème qui se produit dans les deux réseaux, est le mécanisme de bootstrap, c'est à dire la manière dont un nouveau participant peut se connecter aux réseaux P2P ou MANET. Il est nécessaire, que tout nouveau nœud trouve des pairs actifs du réseau. En se connectant à l'un d'eux, le nouveau nœud devient à son tour un membre actif. Pour trouver des membres actifs du réseau, une sorte de passerelle est utilisée, comme une balise à la disposition des nouveaux participants, à qui s'adressent aussi les membres actifs pour être trouvés. Ainsi, tout nouveau nœud doit connaître, au moins, l'adresse de ce portail, pour être en mesure de se connecter sur le réseau ad hoc. Dans un réseau P2P, l'adresse IP de cette passerelle est préconfigurée. Dans MANET, le portail consiste en une bande de fréquences, qui constitue le canal de signalisation des nœuds ad hoc. C'est sur ce canal que le nouveau terminal doit annoncer sa présence, pour devenir un membre actif du réseau.

Il n'est pas facile, pour les réseaux MANET et P2P, d'assurer des fonctionnalités de gestion de réseau, comme l'autorisation, l'authentification et la traçabilité (AAA), ou même la qualité de service. En particulier, les deux notions d'autorisation et d'authentification constituent le fondement de la confiance dans un environnement non fiable, surtout quand il n'existe pas d'autorité centrale. Concernant la sécurité, aucune approche de sécurité, comme par exemple IPSEC [65], ne peut être utilisée telle quelle dans les couches basses. En effet, dans ce type de réseau, les données utiles et de signalisation doivent transiter, même en partie, à travers d'autres nœuds du réseau. Ainsi, une sécurité de bout en bout ne peut être assurée par des tunnels IPSEC, mais seulement par des techniques de cryptage mis en œuvre sur les couches supérieures.

Comme mentionné précédemment, la qualité de service est également une question importante à résoudre dans n'importe quel réseau ad hoc ou p2p. Étant donné que les connexions sont initiées et gérées par des pairs, aucune garantie n'est assurée. Ainsi les applications critiques, en termes de latence et de bande passante, comme le streaming vidéo sur MANET ou P2P, connaissent cette

difficulté, en plus des contraintes physiques de ces réseaux. Un aperçu des similarités est donné dans le tableau 3.

Similitudes	Peer-to-Peer	MANET
Routage de base	Diffusion virtuelle et inondations	Diffusion et inondations physiques
Topologie	Plate et changeant fréquemment.	Plate et changeant fréquemment.
Fiabilité des nœuds	Faible	Faible
Connexion	Saut par saut via des liaisons filaires, portée illimitée	Saut par saut via liens radio, limités par leur portée
Accès au réseau	Via portail (Un nœud phare)	Via portail (Canal broadcast)
Passage à l'échelle	Limitée par le trafic de contrôle	Limitée par l'overhead causé par le routage
Gestion du réseau	QoS et AAA difficiles car pas d'entité centrale de contrôle	QoS et AAA difficiles car pas d'entité centrale de contrôle
Sécurité	Pas de sécurité (IPSEC) utilisable au niveau réseau	Pas de sécurité en couches basses implémentée jusqu'alors

Tableau 3 – MANETs vs.P2P:Similitudes

Afin de combiner les forces des deux approches, et pour palier à leurs limitations, nous allons voir, dans ce qui suit, comment une synergie entre les deux réseaux peut être réalisée.

2.4.3. Synergie entre les systèmes P2P et MANET

Les caractéristiques communes partagées par les overlays p2p et MANETs montre aussi que ces deux réseaux sont confrontés au même défi fondamental, celui de fournir des connexions dans un environnement complètement décentralisé. Ainsi, il existe une synergie entre les deux en termes

d'objectifs et de principes conceptuel de leurs protocoles de routage, qui doivent faire face à des topologies dynamiques en raison des changements causés par les arrivées, les départs ou de la mobilité des nœuds.

Par conséquent, une direction de recherche prometteuse que nous avons poursuivie, pour les réseaux du futur, est celle de la synergie entre les overlays p2p et les protocoles de routage MANET pour concevoir de nouveaux protocoles plus fiables en intégrant des informations issues des deux niveaux

Il existe dans la littérature [66] deux manières de déployer des systèmes p2p sur MANET, la première consiste à intégrer les protocoles réseau et applicatif en une seule application; la seconde approche fait communiquer les protocoles de différentes couches entre eux. Ces deux approches sont détaillées dans ce qui suit.

2.4.4. P2P MANETS

La motivation pour proposer des abstractions P2P dans les MANETs, vient du fait que ces deux réseaux partagent de nombreuses caractéristiques essentielles, telles que la décentralisation, qui fait des applications P2P, développées pour Internet, des candidats potentiels au déploiement sur MANET. Par exemple, les applications de partage de fichiers P2P, telles que Gnutella, sont conçues avec une philosophie dite *serverless* (littéralement, sans serveur), ce qui les rend potentiellement bien adaptées à un environnement sans infrastructure comme les MANETs.

Les applications P2P structurées, développées pour Internet, ont le rôle de fournir un substrat pour la construction d'une variété évolutive et robuste de programmes distribués destinés à Internet. Tels que les systèmes de stockage distribué [67][68], de multicast applicatif [69][70][71][72][73] et de recherche textuelle [74]. Une abstraction DHT, implémentée par ces overlays structurés, résout de nombreuses questions difficiles, y compris la tolérance aux pannes, la localisation d'objets, le passage à l'échelle, la disponibilité, de l'équilibrage de charge, et le déploiement incrémentale. La motivation de soutenir l'abstraction DHT dans MANETs est semblable à son homologue sur Internet.

En raison de sa performance en termes de nombreuses propriétés communes aux applications distribuées, une abstraction DHT, si elle est déployée dans MANETs, pourrait également lever de nombreux verrous lors de la construction d'applications réparties mobiles. Par exemple, des applications telles que le partage de fichiers et la découverte de ressources pourraient bénéficier des techniques d'indexation/de recherche fournies par DHT.

2.4.5. Approches de conception du P2P MANET

Les différentes approches de conception d'application de peer-to-peer pour MANETs sont classées en fonction de la nature de l'overlay P2P, pouvant être structurée ou non structurée. De plus, étant donné que les nœuds MANET jouent le rôle d'hôtes et de routeurs en même temps, ils sont effectivement impliqués dans le soutien des abstractions de la couche P2P. Ces abstractions peuvent être implémentées, soit sur la couche réseau ou sur la couche application. Si un protocole P2P est réalisé sur un protocole de routage MANETs, la conception est dite en couches ou stratifiée. Si un protocole P2P est intégré à un protocole de routage MANET sur la couche réseau, cette approche est alors dite intégrée. Ces deux approches sont détaillées dans les sections suivantes.

2.4.5.1. *Stratifiée et non structurée.*

Cette conception permet de superposer des protocoles P2P non structurés existants (à la Gnutella) au-dessus d'un protocole de routage MANET. Cette conception est similaire à l'approche Internet, qui place un protocole P2P au dessus de la couche transport..

Les travaux Oliveira al. [75] étudient l'exécution d'une application P2P non structurées dans un MANET. Plus précisément, les auteurs évaluent la performance relative des trois protocoles de routage DSR, AODV et DSDV pour le support d'une application P2P non structurée.

Les résultats observés montrent que les performances des protocoles sont mauvaises, et que le taux des paquets transférés est plus faible.

Ceci peut être expliqué par le fait que Gnutella, tournant sur la couche application, va contacter des voisins sur le même niveau pour résoudre une requête. Ces voisins vont faire suivre cette demande si le TTL est non nul. Cependant, en raison de la mobilité des nœuds, ces voisins ne tenant pas compte de la topologie physique du réseau ad hoc, peuvent avoir besoin de communications multi-sauts pour être atteindre la destination. En conséquence, chaque saut réalisé par Gnutella sur la couche application peut entraîner une découverte de routes par inondations, laquelle est coûteuse au niveau du protocole de routage multi-sauts.

2.4.5.2. *Intégrée et non structurée.*

Dans cette approche, l'algorithme d'un protocole P2P non structuré est intégré aux fonctions d'un protocole de routage MANET pour des interfaces P2P soutenir non structurées.

Le travail de Klemm et al. [76] propose, dans ce sens, l'intégration d'une application P2P non structurée, à la Gnutella, dans la couche réseau. Le système développé, appelé ORION, intègre les processus de requête P2P avec celui effectué par le protocole de routage AODV (voir 2.3.1.2.a).

Cette proposition souffre d'un problème de passage à l'échelle, car une augmentation de la taille du réseau provoque une grande surcharge de contrôle dans le réseau. À cause de l'inondation double effectuée sur les deux niveaux du système.

2.4.5.3. *Stratifiée et structurée.*

Celle-ci consiste en un protocole P2P structuré tels que Pastry, CAN, Chord, ou Tapestry fonctionnant comme une application au-dessus d'un protocole de routage MANET.

Crossroad [77] est un exemple de cette approche, dans laquelle une DHT consciente de la topologie (topology-aware), basée sur Pastry, est superposée sur le protocole de routage AODV, et OLSR avec une altération minimale de ces derniers. Néanmoins nous pouvons constater que ce type de proposition rend difficile l'exploitation de certaines opportunités d'optimisation, en matière d'interaction entre la DHT et les protocoles de routage multi-sauts sous-jacents.

2.4.5.4. *Intégrée et structurée.*

Cette méthode propose d'intégrer un protocole P2P structuré avec les méthodes du protocole de routage MANET afin fournir une abstraction de la table de hachage distribuée.

Dans [78] Das et al. ont proposé Ekta, une approche intégrant les fonctions du protocole DHT Pastry, dans le protocole de routage multi-sauts DSR. L'idée clé de cette intégration est de placer une DHT Pastry (opérant sur un espace de noms logique), sur le protocole de routage MANETs (agissant sur un espace de noms physique), par l'intermédiaire d'un mapping entre les identifiants des nœuds mobiles et de leurs adresses IP. De cette façon, la structure de routage d'une DHT et celle d'un protocole de routage multi-hop, sont intégrés afin d'exploiter pleinement les interactions entre les deux protocoles de routage et optimiser la performance du système résultat.

2.5. Mobile P2P Streaming

Le streaming multimédia doit respecter certaines règles pour arriver à bon port. Le délai de transmission, la variation de ce délai et les taux de perte constitue les métriques majeures à surveiller. Dans les réseaux sans fils, WLAN en particulier, la fluctuation du signal radio doit être prise en considération lors de la conception d'application wireless streaming en général et mobiles en particulier.

Il est clair qu'il y a un nombre important de verrous à lever pour arriver mettre en place un streaming de qualité sur les réseaux sans fils. Que ce soit en infrastructure ou en Ad Hoc il faut tenir compte non seulement des caractéristique du flux à émettre, mais aussi, certaines conditions de transmission. Notre approche est dite applicative, car nous souhaitons palier à ces problèmes en proposant des nouveaux mécanismes au niveau applicatif sans toucher à la couche réseau.

Enfin, nous pensons que le futur des applications multimédia, pour dispositif nomades, les sortira de leur passivité. Les systèmes de streaming communiqueront leurs besoins en qualité de service, pour s'adapter aux conditions de la transmission. Cette adaptation doit être dynamique et totalement abstraite à l'utilisateur.

En général, la plupart des applications qui impliquent la totalité (ou une grande partie) des nœuds dans un MANET, tels que la diffusion, sont susceptibles d'être plus efficace si elles sont construites directement au-dessus de la couche physique au lieu d'être placées sur un substrat P2P structuré. Ceci pour la simple raison que l'efficacité de celui-ci souffre des nombreuses inondations causées par la découverte des routes, alors que chacune de ces inondations peut déjà atteindre l'ensemble (ou un grand nombre) des nœuds dans le MANET.

Chapitre 3. SM²Stream: Diffusion multimédia dans les réseaux disruptifs

3.1. Résumé

Dans ce chapitre nous nous sommes intéressés au maintien des sessions multimédia dans la mobilité. Parmi les challenges du streaming vidéo en mobilité sur les réseaux sans fil, nous abordons, dans cette contribution, la problématique du maintien la QoS des sessions multimédia lors de la mobilité des utilisateurs. Dans des applications de streaming vidéo, le buffering coté terminal est une technique bien connue pour palier au problème de pénurie du contenu dans le buffer du client. Toutefois, en raison de la mémoire limitée du terminal et de l'imperfection du canal sans fil, le contenu multimédia dans le buffer, qui peut subir des fluctuations importantes et ne peut suffire à couvrir les périodes de blocage. Pour palier à ce problème, nous proposons une nouvelle méthode de buffering contrôlée par le terminal qui s'appuie sur la prédiction du handover afin de supporter un service de streaming de haute qualité et sans coupure transparente aux utilisateurs mobiles. Ces optimisations sont faites au niveau applicatif sans aucune intervention sur l'infrastructure de l'opérateur.

3.2. Introduction

3.2.1. Contexte

Le projet SUMO s'intéresse à la continuité du service, l'adaptation et la protection du contenu numérique appropriée quel que soit le lieu et/ou le fournisseur de service. Le projet étudie également de nouveaux services de mobilité sur une infrastructure de technologies d'accès sans fil hétérogènes (multi-technologies) avec mobilité totale entre ces accès, ce qu'on désigne communément par le modèle de communication vertical et transparent. Notre intervention dans ce projet été de fournir un mécanisme de buffering régi par une prédiction du temps restant avant la déconnexion d'un point d'accès. Cette approche permet d'assurer un service de streaming de qualité à des terminaux mobiles se déplaçant entre différentes cellules sans fil (WiFi).

3.2.2. Problématique

Lorsque un nœud mobile se déplace entre les cellules WLAN, la qualité des différents signaux, reçu des point d'accès avoisinant sa trajectoire, fluctuent. Le niveau signal reçu de l'AP (Access Point) actuel auquel le nœud est connecté commence à diminuer et terminal doit essayé de trouver un nouveau point d'accès adéquat. Cette opération se fait en deux phases:

La sélection du réseau ou Network Selection qui consiste en le choix du prochain point d'accès (Access point ou AP) auquel ce même nœud va s'associer parmi un certain nombre d'AP visibles.

Le handover est le temps pendant lequel un terminal mobile s'associe avec l'AP sélectionné.

Connaitre l'instant exact du passage d'un AP à l'autre est une information importante Dans ce travail, nous proposons un moyen de prédire ce handover; et ce, en se basant sur la puissance du signal reçue (Received Signal Strength RSS) par l'interface sans fil du terminal mobile sur un certain historique, afin de savoir combien de temps reste-t-il avant de quitté cette cellule pour une autre. Tout ce travail est motivé par la volonté de maintenir la session multimédia de l'utilisateur durant sa mobilité.

3.3. Introduction à la mobilité dans l'infrastructure sans fil:

Aujourd'hui nous voyons une omniprésence des accès sans fils. Néanmoins, la mobilité liée au nomadisme des terminaux IP, notamment en WiFi pose un problème important pour la garantie de la qualité de service. Les utilisateurs soucieux de la qualité perçue et les opérateurs inquiets pour la pérennité de leurs services coopèrent pour palier aux éventuelles coupures dans un service tel que la voix sur IP ou le streaming vidéo.

L'objectif de cette contribution, est faire de la diffusion sans coupure de flux vidéo vers un appareil mobile entre différents accès sans fils, connu plutôt sous le nom *handover*. Un événement pendant lequel les dispositifs nomades change de point de connexion. Ils sont alors obligés de puiser dans leurs ressources locales pour éviter la coupure du flux vidéo.

En effet, lorsque un nœud se déplace, il est amené à changer plusieurs cellules sans fils (i.e , WiMax, UMTS), c'est ce qui est communément appelé *handover*. Lors de cet évènement, il arrive que le nœud mobile perde complètement la connexion avec le fournisseur du service. En pratique, cette perte de connexion dans le cas de cellules qui se chevauchent ne dure en moyenne que quelque seconde (de 2 à 6 secondes pour le WiFi). C'est-à-dire, le temps pour le terminal de se réassocier au nouveau point d'accès et remette en place les différents niveaux de connexion

Les applications streaming vidéo s'appuient généralement sur le *buffer* coté client, qui permet de stocker temporairement une partie du flux. Ce buffer est généralement créé dans la

Random Access Memory (RAM) plutôt que dans le disque dur (à cause de la lenteur d'accès et de recherche de donnée [80][81]). Dans ce qui suit, le terme "mémoire" sera dénommé RAM.

La bonne qualité de service homogène coté terminal exige le maintien d'une bonne taille de buffer avec le contenu des médias pour assurer la continuité du *playout* en amortissant *la gigue* et *les délais de transmission*. Sans ce buffer, l'utilisateur subirait de nombreux arrêts dans la visualisation. Afin de faciliter la lecture en streaming, certains suggèrent une configuration d'une taille maximale pour le buffer. Le problème, est que plus le buffer est grand, moins de mémoire vive est disponible pour d'autres tâches. Par conséquent, l'utilisation efficace de cette RAM devrait être soigneusement prise en compte, dans la gestion du buffer de l'application streaming.

Lors des premières implémentations des buffers, la taille de celui-ci affectait le délai de démarrage du streaming (typiquement de l'ordre 5 secondes). Plusieurs améliorations ont été proposé depuis, telles que *advanced fast start*, qui vise à réduire ce délai [82]. L'idée principale étant d'afficher le peu de contenu déjà reçu, et en même temps, les données sont téléchargées avec un rythme accéléré, allant plus vite que la vitesse d'encodage de la vidéo, jusqu'à ce que le tampon soit rempli. La réduction du temps de buffering agit d'une manière bénéfique sur la qualité de l'expérience perçue par l'utilisateur.

Un autre problème bien connu dans le streaming vidéo dans l'infrastructure sans fil, est la fluctuation des ressources réseau qui induit pertes de paquets, retards, et blocage du service. Outre les variations de la qualité du canal sans fil, l'interruption au cours du processus du handover est également l'une des sources importantes de la discontinuité lors de la mobilité. Cette dernière problématique reste ouverte aujourd'hui et sa résolution constitue le principal objectif de ce chapitre.

Par la suite, nous allons présenter un mécanisme de buffering basé sur la prédiction du temps avant handover, pour l'ajustement de la mise en cache de la vidéo. L'idée principale est de recevoir les données avec un débit accéléré lorsque cela est possible, pour remplir le buffer. Puis, utiliser ce buffer lors des périodes de mauvaise réception durant le handover. Néanmoins, pour ne pas utiliser d'une manière permanente un buffer de taille importante, l'objectif est de tenter de remplir complètement le buffer, uniquement le cas d'un handover imminent.

3.4. Travaux relatifs

Dans un environnement compatible avec Mobile IP (MIP), plusieurs améliorations ont été proposées, ayant pour objectif de réduire la durée du handover ainsi que le taux de perte de paquets que génère le handover MIP tel que fast MIP [83], hierarchical MIP [84] ou le smooth handover scheme [85].

En effet, cette perte de paquets est due au passage du terminal par des zones de chevauchement de petites cellules sans fil; ou encore, à la latence du handover qui peut durer jusqu'à quelques secondes. Des techniques de pré-buffering ont déjà été employées pour surmonter la pénurie possible des buffers du terminal. Cependant, un choix imprécis ou conservateur des paramètres du buffer peuvent engendrer soit une perte de paquet durant le handover, ou bien un gaspillage de la mémoire du terminal. Une prédiction précise de l'instant du handover peut permettre la détermination de la quantité nécessaire de données à pré-bufferiser ainsi que l'instant approprié de déclenchement de ce dernier. En d'autres termes, *un mécanisme de prédiction du handover peut ainsi, permettre un buffering adaptatif et un streaming sans couture (seamless)*.

L'estimation de la bande passante disponible est un domaine de recherche actif. Généralement, cette estimation est basée sur l'utilisation des techniques de sondage, c'est-à-dire, en observant le comportement d'un paquet échangé entre le serveur et le client. Le débit de bout en bout disponible correspond à celui de la liaison radio, car cette liaison agit en général comme un goulot d'étranglement pour la transmission de données. Si le serveur est coopératif dans l'estimation la bande passante disponible, les techniques telles que Pathload [86] ou Pathchirp [87] peuvent être utilisées. En fait, Pathload et Pathchirp utilisent le temps de dispersion chez le client recevant les paquets de sonde via le goulot d'étranglement de la liaison radio pour déduire la bande passante disponible. Dans le cas où le serveur n'est pas coopératif, la technique Spruce [89], qui consiste à forcer le serveur à envoyer les paquets de sonde au client peut être envisagée. Ces techniques donnent des résultats satisfaisants. Nous supposons donc que cette information est disponible via une de ces méthodes et nous concentrons notre attention sur l'aspect prédiction de l'instant de handover.

Dans [90][91], des modèles Gauss-Markov ou Markov cachés ont été appliqués à des données historiques et récentes afin de prédire la localisation et la vitesse d'un nœud mobile. Ces informations sont alors combinées à la connaissance des cartes de déploiement du réseau d'accès pour déduire l'imminence des handovers. En outre, la prédiction de handover basée sur le chemin a fait l'objet d'un *draft* et les habitudes de mouvement des utilisateurs ont été présentées dans [92]. Cependant, ces solutions ne sont pas réalistes, du fait que les comportements d'utilisateurs mobiles aussi bien que les topologies des réseaux sans fils changent de façon irrégulière. Elles ne devraient pas être employées pour la prédiction de handover.

Une autre solution plus pragmatique de prédiction du handover, se base sur le Received Signal Strength (RSS). La quantité de temps avant la perte de connexion se calcule en utilisant une prédiction linéaire à partir de deux valeurs moyennes consécutives du RSS [93] cette prédiction s'est avérée imprécise car les valeurs en question fluctuent à cause des effets du fast fading et des zones d'ombre en log normal. Une approche alternative consiste à appliquer le modèle Grey de

premier ordre (The first-order Grey Model (GM)), pour mitiger le bruit dans les valeurs du RSS et prédire ainsi, les valeurs futures du RSS, ce qui facilite la prédiction du handover [94][95]. Récemment, le GM à permis de calculer les régions de probabilités (faible, moyenne et forte) pour un buffering adaptatif, afin d'éviter la coupure du streaming pendant le handover dans les réseaux WLAN [96]. Toutefois, ce système de buffering ne tient pas en compte ni du débit courant du flux vidéo, ni de la disponibilité du réseau; et encore moins de la latence du handover. Cette solution ne montre pas comment le terminal peut accomplir un buffering lui permettant surmonter les interruptions pendant les handovers. Certes, elle nous informe de la nécessité d'avoir un buffer, mais ne nous dit pas quelle quantité de données doit être mise en cache en avance.

En réalité, dans cette proposition, la gestion du buffer n'est pas vraiment adaptative car elle propose seulement trois tailles pour trois niveaux de probabilité de handover. Par ailleurs, l'estimation de la latence et les pertes de paquets a également été exploitée pour calculer la taille de buffer nécessaire dans les études présentées dans [97][98]. Toutefois, ces solutions ne permettent pas d'identifier l'instant où le buffering devrait être déclenché et exigent une coopération entre le client et le serveur de streaming.

Contrairement aux solutions existantes, notre objectif est de déterminer non seulement l'instant où le buffering doit démarrer mais également le nombre de connexions à ouvrir pour remplir le buffer complètement avant la déconnexion due au handover. Le buffer plein permettra de fournir les données nécessaires au maintien de la session. Nous supposons que (i) le serveur est capable de supporter des connexions simultanées (ii) le client est capable de dire quelles parties du flux il souhaite télécharger.

3.5. La mise en cache (buffering)

La solution que nous proposons ambitionne de ne réaliser aucune modification à l'infrastructure réseau ou à la procédure de handover. Nous supposons que le buffer du terminal est déjà calibré, à une taille appropriée, qui permet d'éviter les fluctuations du réseau et de bien appréhender les handovers. Cette taille peut être fixe, mais le contenu lui est variable. Pour permettre une alimentation continue du lecteur multimédia coté client, nous mettons en place un mécanisme qui adapte la quantité du contenu dans le buffer en fonction de la situation du terminal. La gestion du buffer est réalisée par le terminal mobile, indépendamment du le serveur de streaming. Aucune modification n'est requise sur le serveur, ce qui rend notre approche intéressante.

Lorsque le protocole UDP est utilisé pour transporter les flux multimédia, le débit moyen de téléchargement R_{max} doit être, au moins, égale au débit de lecture R_r (c'est-à-dire, la vitesse d'encodage du flux) dans les conditions normales du réseau. Pour que le buffer puisse se remplir à un rythme accéléré, le client peut alors régler sa vitesse de connexion si le serveur supporte la

négociation des débits. Sinon, le client ouvre $n=\lceil\Delta R/R_r\rceil$ connexions supplémentaire de débit R_r pour remplir le buffer dans les temps (ΔR étant l'augmentation du débit limitée par la bande passante disponible).

Par la suite, nous supposons que le serveur communique avec le client à un débit R_r stable pour chaque connexion UDP. Ceci pourrait générer sur la l'accès wifi une charge supplémentaire qui pourrait détériorer le débit global.

- Règle 1: Si la vidéo dans le buffer est inférieure à un certain seuil $b[t]\leq b_{min}$, ouvrir une connexion UDP pour remplir le buffer pendant $\frac{b_{max}-b[t]}{R_r+\Delta R}$, où b_{max} est la taille maximale du buffer, $b[t]$ est la quantité de vidéo dans le buffer à l'instant t et ΔR est la vitesse du flux de la connexion supplémentaires. Généralement, il est égal à la vitesse de codage, c'est-à-dire, $\Delta R = R_r$. En fait, ce débit de la connexion supplémentaire est limité par le maximum de la bande passante disponible sur le lien radio R_{max} , qui est $(R_r+\Delta R\leq R_{max})$. Le seuil b_{min} peut être défini comme la quantité minimum de vidéo nécessaire à l'affichage de la vidéo. Cette valeur peut être fixée à 2 - 3 secondes de lecture vidéo. Cette condition peut se réaliser lors d'une exécution du handover ou une fluctuation des paramètres du réseau sans fil.

- Règle 2: Si la bande passante disponible est élevée (par exemple $R_{max}> 2R_r$) et le buffer n'est pas plein, ouvrir une connexion UDP supplémentaire pour remplir le buffer.

- Règle 3: Si le handover prévu est imminent et le buffer n'est pas plein, déclencher le buffering. C'est le but principal de ce travail pour éviter une interruption du streaming lors du handover. Les détails relatifs aux différents scénarios de handover sont présentés plus loin.

Pour gérer efficacement le buffering avant le handover, le terminal mobile devrait prédire les paramètres tels que le temps restant avant le handover et le temps restant avant de quitter la cellule actuelle. Ensuite, cette information sera couplée avec les valeurs de la durée du handover et de la bande passante disponible sur le lien radio afin de prendre la décision sur le buffering. La durée du handover est définie par la latence dans le cas d'un handover horizontal, en utilisant la même interface radio. Elle dépend aussi de la technologie d'accès (WLAN). Elle correspond au délai pour l'établissement de la connexion vers un nouveau point d'accès et la mise à jour de l'adresse IP avec MIP. L'estimation de la durée a été étudiée dans [99]. Par souci de simplicité, une valeur prédéfinie de cette durée par technologie d'accès est utilisée. Pour le réseau WLAN, elle à été fixée à 5 secondes.

3.6. La prédiction du handover

La détection du handover joue un rôle primordial dans le mécanisme de buffering que nous avons proposé précédemment. Celle-ci consiste en la prédiction de deux métriques : le temps restant

avant que ce handover se produise et le temps restant avant que le terminal ne quitte la cellule à la quelle il est actuellement attaché. Pour cela, nous utilisons le modèle de Grey du premier ordre GM(1,1). En effet, la prédiction avec ce filtre établit un modèle passant des valeurs historiques et présentes (résultant des mesures du RSS) vers les futures. Nous avons choisi un tel filtre parce qu'il est efficace et n'est pas gourmand en calcul (donc facilement implémentable sur des équipements mobiles).

3.6.1. Handover horizontal dans WLAN

Le handover horizontal entre deux points d'accès WLAN en utilisant une seule interface radio est difficile à réaliser de façon transparente. Rappelons que la période de déconnexion dans ce cas correspond à la latence du handover, $T_{bo} = T_{bho}$. Pour réaliser un streaming sans coupure dans un tel contexte, nous proposons le système de buffering suivant:

Etape 1. Calcul du temps avant le handover: Lorsque le terminal reçoit un bon RSS d'un AP avoisinant, le temps restant avant handover T_{bho} est prévue pour chaque nouvelle paire de APs composée de l'AP auquel le nœud mobile est actuellement attaché et l'AP avoisinant. Les détails relatifs à la prédiction sont présentés dans la section suivante.

Etape 2. Calcul du temps d'avance nécessaire: T_{ad} est le temps d'avance nécessaire à pour remplir le buffer si le nœud mobile ouvre une connexion supplémentaire, à l'instant τ , avec un débit ΔR calculé comme suit:

$$T_{ad}[\tau] = T_{con} + \min(b_{max}, R_r/T_{bo}) - b[\tau] / \Delta R \quad (3.1)$$

T_{con} est le temps nécessaire pour établir une connexion UDP supplémentaires et $b[\tau]$ est la quantité de données déjà présente dans le buffer à l'instant τ . L'augmentation du débit dépend fortement de la bande passante disponible, c'est-à-dire, $R = \min(R_{max}[\tau] - R_r, R_r)$ où $R_{max}[\tau]$ est le débit maximum disponible à l'instant τ . Il convient de noter que le temps d'avance T_{ad} ne changera pas à chaque échantillon de mesure en raison de la fluctuation de la bande passante disponible sur une grande échelle de temps. Aussi, si $b_{max} < R_r \cdot T_{bo}$, le terminal peut allouer plus de mémoire pour le buffer multimédia (posons $b_{max} = R_r \cdot T_{bo}$). Cela permet d'éviter les pauses dans la lecture pendant le handover.

Etape 3. Condition de déclenchement buffering: Le temps d'avance T_{ad} est une contrainte dans la décision de buffering. D'une part, pour mener à bien le buffering de la quantité de vidéo requise, cette opération doit être déclenchée sous la condition suivante $T_{bho}[m] \geq T_{ad}[m]$. D'autre part, le buffering ne doit pas être déclenché trop tôt, car l'utilisateur peut changer de direction et le handover pourrait ne pas se produire. Si le

terminal maintient son cap (c'est-à-dire, sa direction et sa vitesse), $T_{bho}[m]$ tombe en dessous $T_{ad} + \Delta T$ (où ΔT est la période de mesure), et la prochaine valeur $T_{bho}[m+1]$ est, sans doute, inférieure ou égale à T_{ad} . À priori, la condition de déclenchement satisfait l'équation $T_{bho}[m] \leq T_{ad} + \Delta T$. Pour éviter une surestimation de la prédiction du handover, nous suggérons d'ajouter une marge d'hystérésis γ à la limite supérieure de sa condition de déclenchement. Le choix de γ est basé sur l'intervalle de confiance sur la fiabilité de la prédiction de l'instant du handover (c'est-à-dire, celles prédites lorsque le terminal se rapproche de l'instant du handover). Le buffering est alors déclenché à l'instant m si:

$$T_{bho}[m] \leq T_{ad}[m] + \Delta T + \gamma \quad (3.2)$$

La connexion supplémentaire perdurera jusqu'à ce que le buffer soit rempli ou jusqu'à ce que le handover se produise. On notera que, si $T_{bho}[m] < T_{ad}[m]$, une ouverture de connexion supplémentaire ne suffira pas pour remplir le buffer à temps. Dans ce cas, nous aurons besoin du mécanisme de recouvrement décrit ci-après.

Etape 4. Buffering de recouvrement: Durant le buffering, le terminal peut connaître une forte baisse de l'estimation des valeurs du temps avant le handover (correspondant aux instants $j > m$). Ceci peut être dû à un changement brutal de la vitesse de l'utilisateur mobile. Par conséquent, lors du buffering, nous veillons à ce que la condition suivante soit vérifiée:

$$\min\{b_{max}, R_r \cdot T_{bo}\} b[j] > \Delta R \cdot T_{bho}[j] \quad (3.3)$$

Si (3.3) est vraie, cela signifie que la durée du buffer est supérieure à T_{bho} . Si $T_{bho}[j] > T_{con}$ (c'est-à-dire, le temps restant avant le handover est assez grand pour ouvrir une nouvelle connexion), initier des connexions UDP supplémentaires à une vitesse $\Delta R'$. La valeur de $\Delta R'$ est calculée comme suit:

$$\Delta R' = \frac{\min\{b_{max}, R_r b\} - b[j] + R t_{con}}{T_{bho}[j] - T_{con}} - \Delta R \quad (3.4)$$

En d'autres termes, le nombre de nouvelles connexions UDP supplémentaires ($n' = |\Delta R' / R_r|$) pour compenser un changement brusque de la valeur T_{bho} lors du buffering. Donc, le nombre total de connexions supplémentaires devient $(1+n')$.

3.6.2. Présentation du filtre grey GM(1,1)

La spécificité du filtre Grey c'est qu'il construit une équation différentielle ordinaire du premier ordre à partir d'une séquence de valeur de RSS mesurées. Etant donné un ensemble de valeurs de RSS d'un AP avoisinant le terminal $X^{(0)} = \{X^{(0)}[1]; X^{(0)}[2]; \dots; X^{(0)}[n]\}$ où $X^{(0)}[i]$ est la valeur discrète du

RSS à un temps i , la séquence génératrice accumulée $X^{(1)} = \{X^{(1)}[1]; X^{(1)}[2]; \dots; X^{(1)}[n]\}$ est donnée par $X^{(1)}[k] = \sum_{i=1}^k X^{(0)}[i]$. Le modèle Grey différentiel ordinaire du premier ordre est formulé comme [100]:

$$(dX^{(1)}[k]/dk) + aX^{(1)}[k] = b \quad (3.5)$$

où $[a, b]$ sont les paramètres à déterminer. En utilisant la méthode des moindres carrés des erreurs, $[a \ b]^T = (B^T B)^{-1} B^T Y_n$ où

$$B = \begin{bmatrix} -0.5 (X^{(1)}[0] + X^{(1)} [1]) & 1 \\ -0.5 (X^{(1)}[1] + X^{(1)} [2]) & 1 \\ -0.5 (X^{(1)}[2] + X^{(1)} [3]) & 1 \\ \vdots & \vdots \\ -0.5 (X^{(1)}[n-1] + X^{(1)}[n]) & 1 \end{bmatrix}$$

$$\text{et } Y_n = [X^{(0)}[2], X^{(0)}[3]; \dots; X^{(0)}[n]]^T$$

Ensuite, les valeurs filtrées et prédites du RSS $X^{(0)}[k]$ sont données par:

$$\hat{X}^{(0)}[k+1] = (X^{(0)}[1] - \frac{b}{a})(1 - e^a)e^{-ak} \quad \text{pour tout } k \geq 1 \quad (3.6)$$

3.6.3. Prédiction du temps avant handover

Le temps avant handover est un indicateur clé pour le buffering adaptatif afin d'éviter les coupures du au handover horizontal. La décision du handover horizontal entre deux AP WLAN est prise sur la base des valeurs RSS. Pendant la communication avec la cellule initiale, le terminal mesure à la fois le RSS et le potentiel offert par celle-ci et les cellules voisines.

Le handover sera donc déclenché si $\hat{P}_s + \delta h \leq \hat{P}_n$. \hat{P}_s et \hat{P}_n étant les RSS filtrés des cellules actuelles et voisines respectivement. δh est une marge d'hystérésis permettant d'éviter un effet ping-pong du handover. Après le déclenchement du handover, la communication peut être interrompue. Si l'instant du handover (ou le temps restant avant celui-ci) peut être prédit, le mécanisme de buffering peut être maintenu efficacement pour assurer un streaming sans coupure. Lorsque le terminal mobile reçoit le bon signal à partir d'un nœud d'accès voisins, il emploie le filtre GM(1,1) afin de prédire les futures RSS des points d'accès actuels et voisins.

Notons par $P_s[i]$ et $P_n[i]$, les RSS provenant des points d'accès actuel et voisins ($i < m$ ou m est l'indice de la dernière mesure du RSS). Tant que le handover ne se produit pas, calculer \hat{P}_s et \hat{P}_n , les valeurs RSS prédites de P_s et P_n . Pour prédire le future instant de handover $k > m$ tel que $\hat{P}_s[k] + \delta h \leq \hat{P}_n[k]$. Par conséquent, l'instant de déclenchement du handover est donné par:

$$T_{bho} = (k - m)T_{meas} \quad (3.7)$$

Tel que T_{meas} est la période de la mesure du RSS. Dans le cas des réseaux WLAN, par exemple, T_{meas} est égale à la valeur de la période l'émission du beacon qui est, par défaut, de 100 millisecondes.

La précision de la prédiction du handover dépend du nombre d'échantillons de RSS ($P[m - N]$, $P[m - N + 1]$, . . . , $P[m]$) correspondant aux entrées du filtre GM (1,1). Généralement, la séquence de valeur de taille N est assez petite, car seulement deux paramètres doivent être identifiés dans (3.5). Toutefois, en raison de la fluctuation du RSS plus cette séquence est grande, plus les valeurs RSS prédites \hat{P} sont régulières.

Comme les valeurs prédites $\hat{P}_s[k]$ ($k > m$) sont calculées à partir de valeurs mesurées jusqu'à l'instant m, la précision de $\hat{P}_s[k]$, d'où la précision du T_{bho} , est une fonction décroissante de $(k - m)$. Les valeurs prédites T_{bho} sont obtenus avec l'hypothèse que les futurs RSS suivent l'évolution de l'ajustement de courbe (3.7). En d'autres termes, la direction et de la vitesse du nœud mobile, sont supposées être inchangées entre l'instant $(m - N)$ et l'instant k. Par conséquent, nous ne pouvons pas prédire le temps avant handover trop en avance.

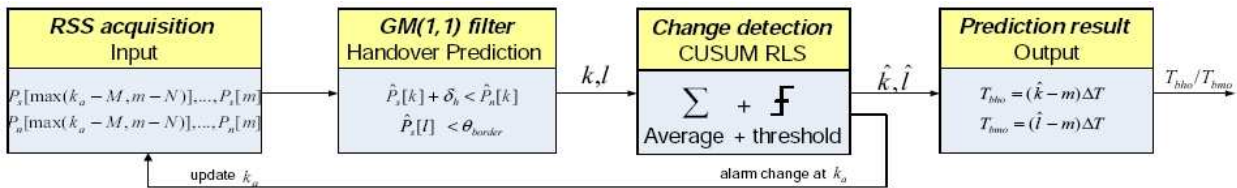


Figure 9: Algorithme de prédiction du T_{bho}

Il serait intéressant d'avoir une idée approximative de l'ordre de grandeur du temps avant handover. Selon (3.2), le handover doit être prédit au moins avec T_{ad} secondes d'avance. Si l'on remplace l'augmentation du débit par $\Delta R = Rr$ dans (3.1), nous obtenons $T_{ad} = T_{con} + T_{bho}$. Le temps d'établissement de la connexion T_{con} est donc égal à plusieurs Round-Trip Time (RTT) pour l'échange de messages de contrôle entre le client et le serveur. Ainsi, l'ordre de grandeur de T_{con} est de l'ordre de plusieurs centaines de millisecondes. La durée du handover T_{bo} entre deux points d'accès WLAN peut atteindre quelques secondes. Par conséquent, dans notre système de prédiction, il est nécessaire de prévoir l'instant du handover de quelques secondes en avance pour être en mesure d'exécuter le mécanisme de buffering.

Comme évoqué précédemment, la précision de la prédiction du handover dépend de la régularité des RSS mesurés. Un changement brutal de trajectoire ou de vitesse de l'utilisateur peut provoquer des erreurs de prédiction du RSS. Normalement, l'effet d'une erreur de prédiction en raison d'un changement de mouvement s'estompe complètement au bout de N mesures. Néanmoins, il serait intéressant pour nous d'atténuer cette erreur plus rapidement. Pour cela, nous

suggérons d'utiliser un mécanisme de détection des changements de direction/vitesse à l'intérieur du système de prédiction du handover. Cette méthode est illustrée dans la figure. Le principe de base de ce mécanisme est de détecter le plus tôt possible un changement de mouvement (direction et / de vitesse) afin d'ajuster les entrées du filtre GM (1,1). Par la suite, nous montrons qu'un changement mineur de vitesse n'a pas beaucoup d'incidence sur les résultats de la prédiction.

Un changement de direction et/ou de la vitesse n'implique pas une altération brutale dans l'évolution du RSS. En effet, cette dernière est fonction de la distance entre l'utilisateur mobile et la station de base qui ne peut pas changer brusquement. Par conséquent, nous ne pouvons pas détecter immédiatement un tel changement. Il est également difficile d'utiliser les RSS afin de détecter un tel changement parce que la séquence de RSS ne contient pas de composante déterministe explicite. Cela explique pourquoi nous utilisons la détection de changement après l'application du filtre GM (1,1) plutôt que juste après la mesure des RSS. Le mécanisme de détection de changement observe le comportement de l'instant du handover prédit $y_t = k$. Une brusque variation de y_t implique un changement dans le rythme du mouvement de l'utilisateur.

L'algorithme de détection des changements séquentiels est celui de la somme cumulée (CUSUM) proposé par Page dans [107]. Nous utilisons le filtre CUSUM Recursive Least Square (RLS), qui combine des filtres adaptatifs avec le test CUSUM comme détecteur de changement, pour prévenir les changements dans la séquence d'yt. L'algorithme CUSUM RLS est décrit comme suit:

$$\hat{\theta}_t = \lambda \hat{\theta}_{t-1} + (1 - \lambda)y_t \quad (3.8)$$

$$\epsilon_t = y_t - \hat{\theta}_{t-1} \quad (3.9)$$

$$g_t^1 = \max(g_{t-1}^1 + \epsilon_t - v, 0) \quad (3.10)$$

$$g_t^2 = \max(g_{t-1}^2 - \epsilon_t - v, 0) \quad (3.11)$$

Par la formule (3.8), on obtient l'estimation RLS où λ est considéré comme un facteur d'oubli. Une grande valeur de ce facteur d'oubli ($\lambda = 0,9$) est choisie de manière volontaire pour donner plus de poids pour la dernière valeur calculée. Dans (3.10), t est l'erreur de prédiction utilisée comme distance pour détecter un changement. Le test statistique g_t^1 (resp. g_t^2), par addition d'entrées ϵ_t , avec l'idée de déclencher une sorte d'alarme lorsque la somme dépasse un certain seuil h . Si λ est une séquence de bruit blanc (pas de changement), le test statistique s'éloigne et se comporte telle une marche aléatoire. La soustraction d'une petite déviation v et une remise à zéro (une fois g_t devient négatif) sont utilisés pour prévenir des fausses alarmes et éviter une longue attente, après la détection d'un changement. Sachant qu'un changement peut entraîner une augmentation ou une diminution de l'instant du handover y_t , nous utilisons deux tests parallèles g_t^1 et g_t^2 pour détecter l'augmentation ou la diminution de la valeur moyenne. Quand un

changement est détecté à l'instant $k_a = t$ (c'est-à-dire, $g_t^1 > h$ ou $g_t^2 > h$), il faut réinitialiser $g_t^1 = 0$, $g_t^2 = 0$ et $\hat{\theta}_t = y_t$.

Les deux paramètres conceptuels du test CUSUM sont la déviation v et le seuil h . Selon [138], la déviation est généralement fixée à la moitié du changement attendu $v = 0.5|\theta_{k_a} - \theta_{k_a-1}|$ quant au seuil h , il dépend des caractéristiques de la séquence du signal et est souvent déterminé par apprentissage à l'aide d'un ensemble de données [139]. Dans la littérature, différentes méthodes pour le choix de h basées sur la probabilité de fausses alertes et le délai approprié pour la détection d'un changement (le délai de détection des changements) ont été proposées dans [101][102]. Cependant, la formulation est asymptotique et reste difficile à appliquer dans la pratique.

Les résultats de l'algorithme de détection de changement sont l'instant du handover $\hat{k} = [\hat{\theta}]$ ($[x]$ désigne le plus proche entier de x) et l'instant de changement k_a . Par conséquent, nous obtenons $T_{bho} = (k-m)\Delta T$. Au début de la prédiction, k_a est mis à 0. Une fois qu'un changement est détecté, une nouvelle valeur est mise à jour et reporté dans la séquence d'entrées RSS. Si aucun changement n'est détecté, le filtre GM (1,1) utilise les N derniers RSS en entrée. Sinon, seules les dernières valeurs RSS associées à un nouveau mouvement seront utilisées plutôt que toutes les N dernières valeurs RSS. Une fois le changement détecté à l'instant k_a , celui-ci s'est probablement produit à l'instant $(k_a - M)$ où M est le délai de détection de changement. En général, M est inférieur à la moitié de la taille de la séquence de données d'entrée ($M \leq N / 2$).

Le choix de la taille d'entrée N est un compromis à faire entre la régularité de la prédiction et le délai dans la détection de changements. Si N est grand, la détection du changement est trop lente mais un système stable. Mais si on choisit une petite valeur de N alors les fluctuations du T_{bho} peuvent être importantes. Comme la force du signal radio est spatialement corrélée [103], il est préférable d'utiliser une séquence de RSS mesuré au sein d'une corrélation de la distance. En milieu urbain, une corrélation de distance de 20m est largement adoptée [104]. Par conséquent, les valeurs de M et N sont déterminées à partir de l'intervalle de mesure ainsi que la vitesse du terminal.

3.6.4. Prédiction du temps avant l'abandon de la cellule

Contrairement à la prédiction de la T_{bho} , la prédiction de T_{bmo} est uniquement basée sur le RSS du point d'accès actuel $P_s [i]$. L'idée principale est de prévoir l'instant où l'utilisateur quitte la cellule actuelle (ci-après le l'instant d'abandon de la cellule). À cet instant, $\hat{P}_s [i]$ est inférieur à θ_{limite} (c'est-à-dire, $P_s [i] < \theta_{limite}$). C'est le seuil de la force du signal au-dessous duquel la communication ne peut plus être maintenue.

Le schéma de prédiction du T_{bmo} est illustré dans la figure 24. Le système de prévision utilise les séquences de données disponibles P_s ($[m-N]$, P $[m-N +1]$, . . . , P $[m]$). De même que pour la prédiction du T_{bho} décrite ci-dessus, afin d'atténuer les effets dus aux changements de mouvement, nous utilisons un filtre CUSUM RLS pour détecter le changement de mouvement et lisser la valeur du temps avant l'abandon de la cellule. Le mécanisme de détection de changement contrôle en permanence l'évolution de l'instant prédit $y_t = l$. Si un changement est détecté à l'instant k_a , l'entrée du filtre GM(1,1) sera mise à jour en conséquence. Les choix de la taille de la séquence de données N ainsi que les paramètres conceptuels du CUSUM (h, v) sont identiques à ceux utilisés dans la prédiction du T_{bho} . L'instant d'abandon de cellule prédit $\hat{l} = [\hat{\theta}_t]$ ($\hat{\theta}_t$ étant la valeur filtrée de l'instant d'abandon de cellule prédit y_t) est utilisé pour calculer T_{bmo} :

$$T_{bmo} = (\hat{l} - m)T_{meas} \quad (3.12)$$

où m est l'instant de la dernière mesure et T_{meas} est la durée entre deux échantillons consécutifs RSS dans le système de prédiction.

3.7. Evaluations

3.7.1. Simulations

L'objectif de ce paragraphe est de valider l'efficacité du système de prédiction du handover proposé, selon différents scénarios de mouvement de l'utilisateur. Nous avons simulé le mouvement d'un utilisateur entre deux WLAN AP avoisinant sa trajectoire. Dans les simulations, les valeurs du RSS à une distance d de l'AP sont obtenues en utilisant le modèle COST-231 [105] :

$$Pr[dB] = K_1 - K_2 \cdot \log(d) + X_\sigma \quad (3.13)$$

où K_1 dépend de la transmission et la réception des gains d'antenne, la hauteur de l'antenne, la fréquence centrale et de la longueur d'onde du signal, et K_2 représente le facteur de l'affaiblissement de propagation. X_σ étant la moyenne indépendante et identiquement distribuée du processus aléatoire Gaussien modélisant l'évanouissement d'une zone d'ombre (shadow fading). Dans le cas du handover sur WLAN, des utilisateurs mobiles piétons ont une vitesse maximale de 2 m/s, dans un intervalle de mesure de 100 ms, nous posons $N=100$ (c'est-à-dire, les valeurs du RSS mesurées sur une distance de corrélation de 20m).

Après avoir examiné les différents scénarios de changements de directions, nous choisissons $v = 15$ et $h = 2000$ comme paramètres pour la détection des changements. Pour se faire, un changement interviendrait toujours après un délai maximum de 40 échantillons. Comme résultats, nous avons mis $M = 40$ pour l'ensemble des simulations.

Dans la première simulation, l'utilisateur se déplace de l'AP1 vers l'AP2 sans changement de direction et de vitesse. L'évolution du RSS depuis les deux APs et le T_{bho} prédit selon $A \pm h = 0$, sont représentés dans la Figure 10.

Nous pouvons constater que la prédiction de résultat est plus fluctuante lorsque l'utilisateur est loin de la zone de handover et coïncide avec des valeurs réelles lorsque celui-ci s'y rapproche. Le résultat montre que la méthode proposée, peut prédire l'instant du handover plus de 10s d'avance. Nous réitérons cette simulation de nombreuses fois et observons les fluctuations de T_{bho} à 10 et 5 secondes avant le handover. L'intervalle de confiance [106] de T_{bho} est calculé comme suit:

$$P(|T_{\text{bho}} - \underline{T_{\text{bho}}}| \leq t_{n,\alpha/2} \cdot \sigma_m / \sqrt{S} = 1 - \alpha \text{ tel que } n = S-1 \quad (3.14)$$

(3.14) où S est le nombre de simulations, T_{bho} est la moyenne T_{bho} sur S simulations et m est son écart-type. Dans l'équation ci-dessus, $(1 - \alpha)$ est l'intervalle de confiance et $t_{n,\alpha/2}$; est le pourcentage de distribution de Student tels que $P(|t_n| > t_{n,\alpha/2}) = \alpha$. Cela signifie que nous sommes sûr à $(1-\alpha)$ 100% que les prédictions de temps avant handover varie entre $T_{\text{lower}} = (T_{\text{bho}} - t_{n,\alpha/2} \cdot \sigma_m / \sqrt{S})$ et $T_{\text{upper}} = (T_{\text{bho}} + t_{n,\alpha/2} \cdot \sigma_m / \sqrt{S})$. Selon nos résultats de simulations, avec le niveau de confiance de 90%, l'intervalle de confiance de T_{bho} à 10s et 5s avant le handover, sur 100 simulations est de $(10,57 \pm 0,33)$ s et $(5,24 \pm 0,19)$ s respectivement.

Les résultats confirment la fiabilité de nos T_{bho} prédit et la surestimation de la tendance des valeurs prédites par rapport aux valeurs réelles. Ce dernier est dû à la nature du filtre GM(1,1) a été bien connu dans la littérature. Néanmoins, ces résultats montrent la nécessité d'ajouter un hystérésis, à la condition du déclenchement du buffering (3.2), pour éviter les fluctuations imprévues et la surestimation. Dans ce cas, un γ autour de 0.4 s devrait être adéquat. En supposant que $T_{\text{bho}} = 2$ s, $R_r = 500$ Ko/s, $T_{\text{con}} = 0.1$ s et $\gamma = 0.4$ s et la taille du buffer du terminal dans des conditions normales du réseau est maintenu autour de 600ko, en appliquant le schéma de buffering proposé, l'évolution du buffer est illustrée dans le bas de la Figure 10. Nous pouvons voir clairement que le handover s'est réalisé d'une manière transparente pour l'application (sans coupure dans le playout) en améliorant, ainsi, la qualité d'expérience perçue par l'utilisateur finale.

3.7.2. Résultats et Discussions

Nous comparons notre approche à deux autres: une sans mécanisme de détection de changement dans le mouvement et un autre qui suppose que la détection d'un changement est immédiate. Cependant, nous noterons que dans la réalité une détection immédiate est impossible. Les résultats de prédiction du temps avant handover utilisant les trois programmes le montrent bien dans la Figure 10. Nous pouvons observer que le schéma de prédiction que nous proposons donne un meilleur résultat par rapport à ceux obtenus par les deux autres. D'une part, si la détection du changement n'est pas utilisée, l'erreur du T_{bho} prédit est importante et ne peut être réduite

qu'après N étapes. D'autre part, si un changement est détecté immédiatement, les valeurs T_{bho} prédites immédiatement après le changement est fluctuant beaucoup en raison de la petite taille des entrées RSS utilisées dans le filtre GM (1,1). Les résultats démontrent la rationalité et l'efficacité de notre système de prédiction du handover. Là aussi, on peut constater à partir de la Figure 10 (bas) que la taille du buffer ne fait qu'augmenter en prévision d'un handover intempestif, et n'atteint jamais une valeur nulle.

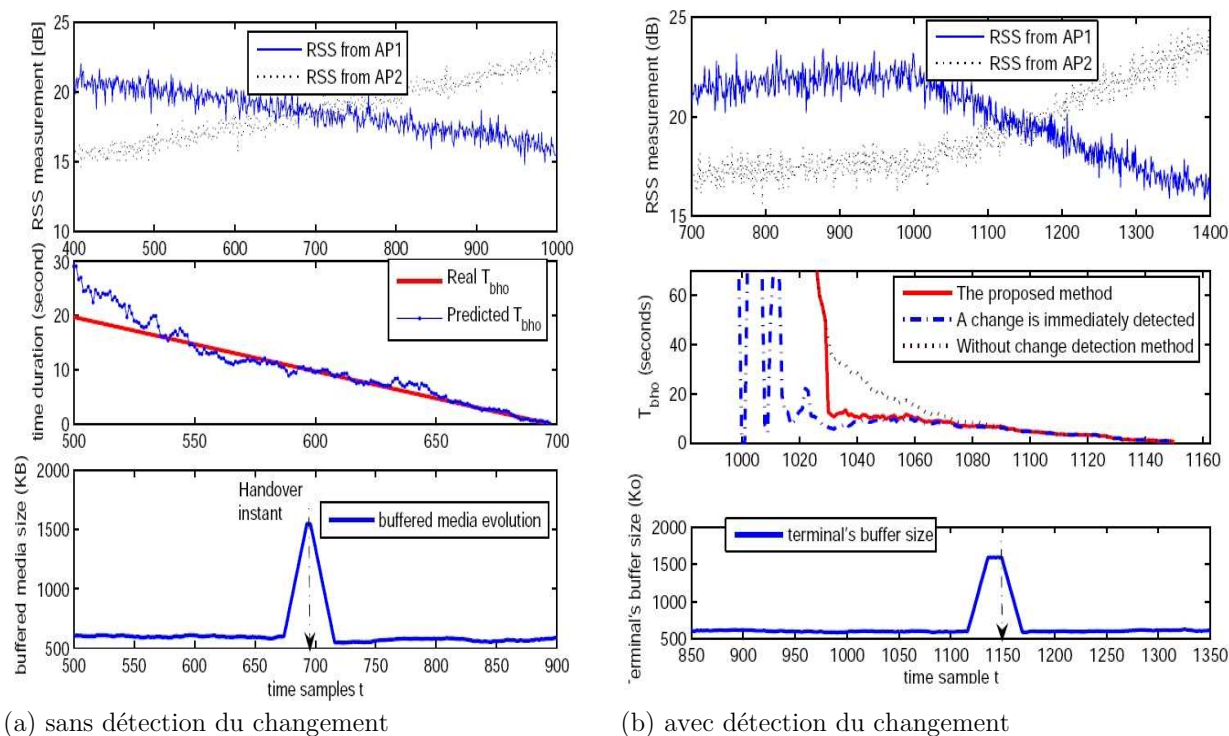


Figure 10: Prédiction du T_{bho} et évolution du buffer

3.7.3. Expérimentation

Dans cette phase, nous nous sommes intéressés à la méthode expérimentale afin de valider les résultats analytiques et simulés.

3.7.3.1. Description du banc d'essai

Notre banc d'essai est composé de trois points d'accès WLAN AP_1 , AP_2 et AP_3 dans une zone plate sans obstacles comme illustrée dans la Figure 11, La distance entre l' AP_1 et AP_2 est d'environ 50 mètres. Un MN (laptop) et déplacé entre AP_1 et AP_2 avec une vitesse d'environ 1,5 m/s (piéton). Une session de streaming multimédia à partir d'un serveur directement relié à ces APs via un réseau filaire. Les signaux de toutes les APs sont mesurés toutes les 100 millisecondes. Nous avons intégré l'algorithme de prédiction basé sur le filtre GM(1,1) dans un logiciel utilitaire de collecte d'information sans fil afin de prédire le temps restant avant le handover T_{bho} .

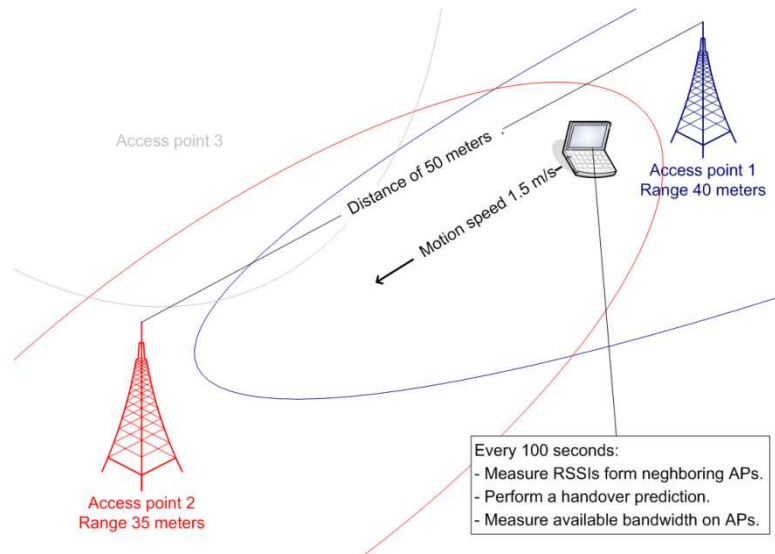


Figure 11: Schéma du banc d'essai

Le client et le serveur utilisent le logiciel VideoLAN¹³ pour le streaming sur UDP. Le mécanisme de buffering a été intégré directement dans VLC. Le code a été modifié pour ouvrir automatiquement des sessions parallèles avec le serveur pour accélérer le remplissage des buffers lorsque nécessaire. Les détails des algorithmes se trouvent dans le paragraphe suivant. Le serveur reste inchangé. Par ailleurs, et afin de surveiller la bande passante disponible de bout en bout entre le client le serveur de streaming, nous utilisons l'outil Stab qui est un utilitaire de mesure basé sur la technique Pathchirp.

3.7.3.2. Algorithmes réalisés dans VLC

Le remplissage se fait grâce à la création de plusieurs buffers associés chacun à une connexion. Plusieurs connexions en parallèles peuvent être déclenchées pour la lecture du flux en cours. Comme expliqué ci-dessus, chaque buffer peut stocker une séquence future du flux multimédia (Figure 12). Mis bout à bout ces buffers vont contenir la totalité de la séquence nécessaire en local pour un playout durant la déconnexion du handover. Le nombre de buffers ainsi que leurs tailles seront fixés en fonction du bit rate de la vidéo, la taille de la vidéo restante, le temps devant le handover ainsi que sa durée.

¹³ The VideLan Client <http://www.vidéolan.org>

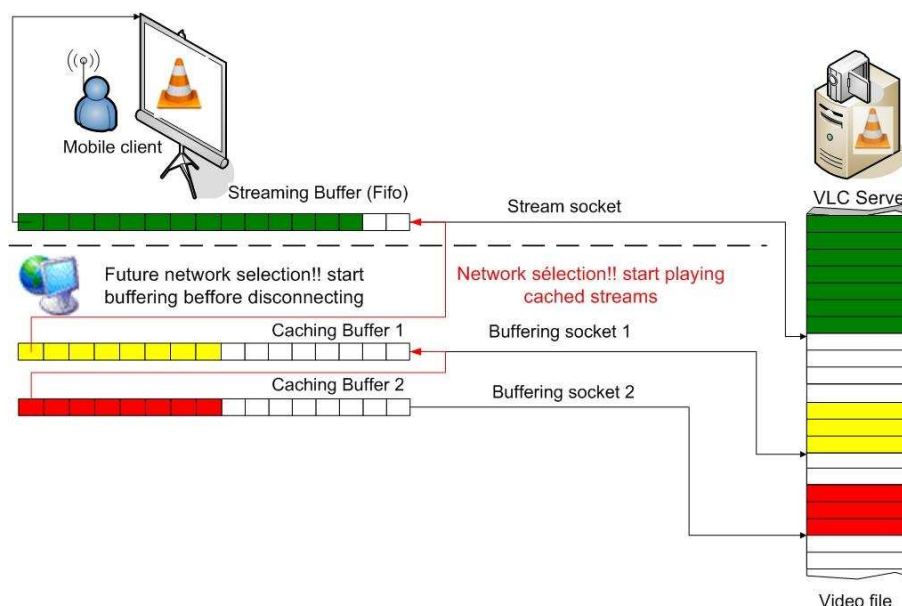


Figure 12: Schéma du buffering

Après avoir étudié de près l'architecture logicielle de l'application VLC, nous avons effectué les changements nécessaires pour mettre en place ces mécanismes et assurer une lecture sans coupure de la vidéo même lorsque la connexion physique est coupée. Les points qui suivent détaillent les algorithmes utilisés dans le but de la réalisation de notre expérimentation.

Etape 1. *Estimation du T_{bho}* : L'algorithme 1, mesure chaque 100 ms les RSSs reçu des APs qui se trouvent à proximité. En effet, nous avons adapté la méthode de recensement des réseaux IWLIST faisant partie des outils linux Wireless Tools. Que nous avons intégré à notre client de prédiction et d'exécution du handover.

Algorithme 1: Collecte d'information sur les réseaux avoisinants

```

for each beacon_period do
  for each  $AP_i$  do
     $AP_i^{j=1..20} = Get\_RSSvalue(AP_i)$ 
    if  $AP_i == Max(BestAP_i, AP_i)$  and  $(ActualAP \neq AP)$  then  $BestAP = AP_i$ 
  end for
end for
if  $(ActualAP^{j=1..20} \neq Threshold)$  then  $predict\_T_{bho}(ActualAP, BestAp)$ 
else  $trigger\_handover(BestAP, T_{bho}, T_{bmo})$ 

```

Etape 2. Les valeurs mesurées sont ensuite enregistrées sur un historique de 2 secondes (équivalent à 20 valeurs). Chaque seconde, l'algorithme 2 prédit les valeurs des temps avant handover et temps avant de quitter la cellule actuelle, en appliquant le filtre

Grey et renvoie la décision de faire un handover si la valeur du RSS de l'AP actuel atteint un certain seuil est atteint.

Algorithme 2 : prédiction des T_{bho} et T_{bmo}

```

for each  $k, l$  do
  if ( $ActualAP_k + ho\_threshold > BestAP_k$ )
    and ( $ActualAP_l > border\_threshold$ ) then  $T_{bho} = k, T_{bmo} = l$ 
  return  $T_{bho}, T_{bmo}$ 
end for

```

Etape 3. Mécanisme de mise en cache est décrit dans l'algorithme 3, celui-ci prend en compte les temps T_{bho} , T_{bmo} , le bit rate de la vidéo et la connexion disponible pour ouvrir un nombre précis de connexions (suivant la relation) pour alimenter les buffers correspondant. Ce remplissage se fait bien évidemment à partir de positions futures dans la vidéo.

Algorithme 3 : mise en cache

```

 $n\_connections = bandwidth\_stream\_rate / stream\_rate$ 
 $open\_rtsp(n\_connections, n\_buffers)$ 
while  $T_{bho} \neq 0$  and  $not\_connected$  do
  for each  $i_{con} \leq n\_connections$  do
     $get\_rtsp\_packets(pos_{i_{con}}, buffer_{i_{con}})$ 
  end for
   $T_{bho} - -$ 
end while

```

L'exécution du handover (Algorithme 4) commence lorsque le mobile est à la frontière de la cellule actuelle. C'est à ce moment précis ou sont lancées deux méthodes, la sélection du nouvel AP et le rétablissement de la connexion avec mobile IP, avec le filtrage du flux MPEG2-TS à la reprise du streaming via le nouvel AP, une étape de calibrage des trames TS qui encapsulent les flux MPEG2 est nécessaires. Elle permet de juxtaposer les flux venant des buffers avec celui que le client reçoit une fois la connexion au réseau est rétablie.

Algorithme 4: exécution du handover

```

if new_ap_selected then
  for each icon and (buffericon ≠ empty_buffer) do
    play_from_buffer(buffericon)
    buffericon = empty_buffer
  end for
  associate(BestAp)
  retrieve_mobile_ip(new_address)
  rtsp_get(server_ip, last_buffer_pos)
  calibrate_stream(buffermax-icon, playback_buffer)
  gather_info(ActualAP, BestAp)
end if

```

La figure suivante résume l'essentiel des composantes logicielles intervenant dans notre proposition expérimentale.

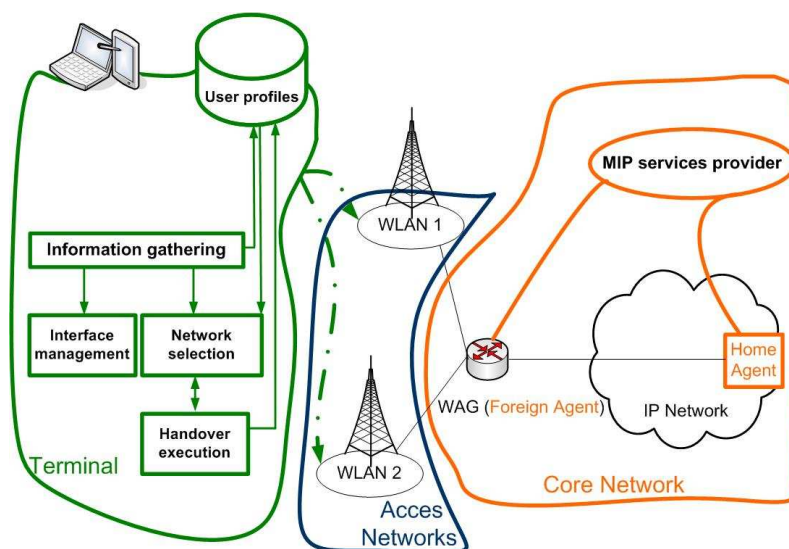


Figure 13 Composantes logicielles de SM²Stream

3.7.3.3. Résultats

En s'éloignant de l'AP₁, les valeurs mesurées des RSS des trois AP sont présentées dans la Figure 14. Après avoir recueilli 20 échantillons RSS mesurés à partir de l'AP₁ et l'AP₂, le système de prédiction de handover est là pour estimer le T_{bho} . Ces valeurs sont également décrites sur la Figure 15. Ce résultat est obtenu dans le cas d'un $\delta_n = 0$. La figure montre clairement que la différence entre les T_{bho} réel et T_{bho} prédit est très faible. Ceci confirme la faisabilité et, une fois de plus, l'efficacité de la méthode de prédiction proposée.

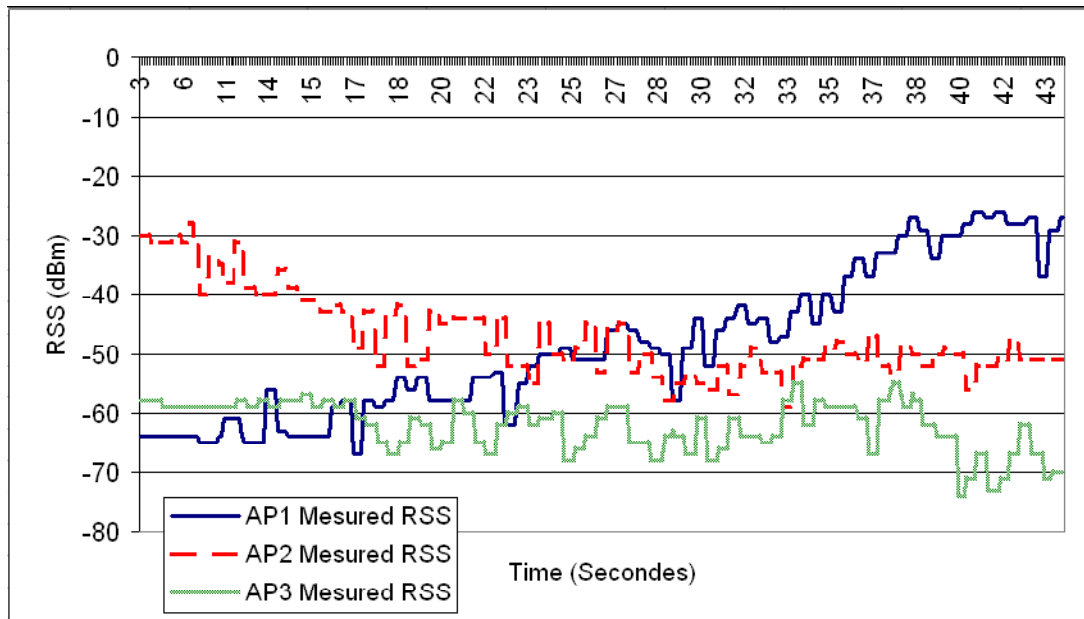


Figure 14 Evolution des RSS reçus

Ensuite, nous évaluons la performance du mécanisme de buffering. Premièrement, nous provoquons plusieurs handovers entre AP₁ et AP₂ et nous collectons suffisamment d'information sur le délai du handover. Notons que le délai maximum de handover peut atteindre 8 secondes. Cette valeur est ensuite mise dans le schéma de buffering. En outre, la bande passante disponible est régulièrement contrôlée par Stab et le délai de connexion T_{con} est fixé à 2 RTT. Dans un mode de fonctionnement normal, le client stocke environ 2 secondes de la vidéo dans le buffer.

Dans notre expérience, la tâche de buffering décide d'ouvrir une seconde connexion à la $t=19s$ et le handover se produit à $t=22$. L'évolution de la taille du buffer est présentée dans la Figure 15. Notons que la taille de la vidéo est ici exprimée en terme de son temps de lecture pas en octets. Nous constatons que la taille du buffer n'est jamais nulle, même au cours de la période du handover. Et donc aucune interruption dans la visualisation n'a lieu. Aussi, la taille du buffer n'est jamais trop grande. Celle-ci augmente uniquement lors de la préparation du handover. Elle implique l'efficacité de l'utilisation de la mémoire. Ce résultat expérimental confirme la faisabilité de notre solution pour le streaming multimédia pour les utilisateurs mobiles à travers les réseaux sans fils.

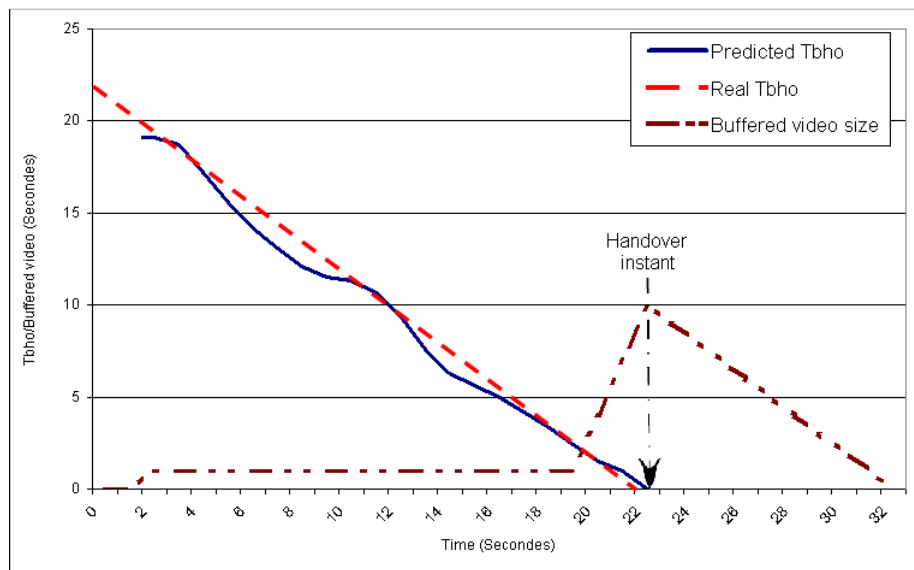


Figure 15: *Tbho prédit et évolution du buffer*

3.8. Conclusion

Dans ce chapitre nous avons adressé le streaming de la vidéo en mobilité dans un réseau sans fils multi-cellules. Plus précisément, nous résolvons les problèmes liés à l'impact défavorable de la mobilité d'un terminal, entre différents points d'accès sans fil, sur la qualité du streaming vidéo. Ce travail a permis de proposer un mécanisme de buffering réactif basé sur la prédiction du temps avant le handover. Grâce à une implémentation réelle de ce mécanisme de buffering associé à une anticipation du handover, nous avons pu, ainsi, valider des modèles théoriques proposés pour l'optimisation du buffering.

Chapitre 4. MADP2PStream: Modèle de streaming peer to peer sur réseaux ad hoc mobiles

4.1. Introduction

Un concepteur doit évaluer, en plus de l'intégrité, un certain nombre d'indicateurs de performance d'un protocole. Il existe essentiellement trois outils disponibles pour réaliser une telle évaluation: la modélisation analytique, la simulation et le prototypage. Même si l'implémentation et l'essai direct (à l'aide de simulation ou d'un prototype) semblent être la façon la plus simple pour évaluer les réseaux P2P, il n'est pas toujours aisé de suivre cette approche. Les simulations ou Les mesures directes sur des prototypes présentent des problèmes de passage à l'échelle. En effet, celles-ci sont généralement réalisées dans des systèmes de petite taille qui ne reflètent pas forcément le comportement du système à large échelle. Les réseaux P2P représentent en effet l'opposé de cette situation: la taille typique d'une communauté P2P commence à partir de milliers d'utilisateurs (pour la diffusion) à des millions d'utilisateurs (pour le partage de fichiers). Dès lors, se pose le problème de la simulation de réseaux d'une telle taille.

Si, d'une part, les outils analytiques ne présentent pas ce problème de passage à l'échelle, ils ne permettent en général de modéliser qu'un comportement simplifié du système. En effet, construire un modèle analytique complet d'un système complexe comme les réseaux P2P est une tâche difficile voire impossible: car celui-ci doit représenter toutes les caractéristiques essentielles du système étudié tout en supposant que le comportement de certains éléments du système ont un faible impact sur sa performance globale. Le challenge de la modélisation est effectivement de trouver un outil d'analyse qui soit capable prendre en compte ces simplifications sans compromettre la fidélité de la modélisation du comportement global du système.

Comme nous l'avons souligné, le streaming vidéo en p2p a été proposé comme solution au déploiement coûteux de serveurs vidéo tirant parti des ressources de chacun des pairs. Cette utilisation des ressources de chaque paire permet d'équilibrer la charge induite par le transfert de contenus vidéo entre les différents pairs. L'utilisation de cette même technique initialement développée pour les réseaux infrastructure vers les réseaux ad hoc mobiles introduit de nombreuses

difficultés telles que la la les problèmes de connectivité et , la monopolisation de la puissance de calcul, la mémoire vive ainsi que la capacité de stockage des paires par les tâches en rapport avec le réseau p2p. Tous ces aspects doivent être étudiés pour proposer ou valider de nouvelles solutions de streaming sur des réseaux ad hoc.

Dans ce chapitre, nous proposons un nouveau modèle analytique appelé MADP2PStream (Mobile AD hoc Peer TO Peer STREAMing) dont le but est de modéliser de manière adéquate le service peer to peer streaming sur MANET. La contribution principale est l'introduction de la mobilité dans un modèle de streaming peer to peer existant afin de pouvoir identifier les paramètres essentiels à prendre en compte pour développer des protocoles de streaming p2p qui soient efficaces sur des réseaux ad hoc mobiles. Il s'agit en effet de cerner, grâce à ce modèle, les perturbations qui peuvent survenir dans le système en raison de la dynamique additionnelle engendrée par la mobilité des nœuds.

4.1.1. Objectif de cette contribution

Le récent succès des applications de streaming p2p dépasse celui des protocoles traditionnels, les protocoles clients/serveurs ou multicast. Les architectures basées sur les arbres applicatifs proposées auparavant, cohabitent maintenant avec des structures maillées plus résistantes à la variation des débits dans les environnements sans fils et à la dynamique des nœuds mobiles. L'utilisation de la technologie p2p dans les environnements MANET est étape naturelle de ce développement. Néanmoins, la migration des applications p2p streaming d'un environnement fixe vers les réseaux MANET ne se fait pas sans problèmes et nécessite de lever un ensemble de verrous. Dans ce chapitre nous proposons un modèle qui permet, lors du port du streaming p2p classique (fixe) sur MANET, de capturer les paramètres essentiels qui affectent le système; c'est-à-dire le comportement du système lorsque les paires sont mobiles. Ces métriques sont principalement reliées à la qualité du streaming; c'est-à-dire d'une part les délais de la distribution des éléments du stream aux différents nœuds et d'autre part la connectivité d'un nœud. En effet, dans le cas d'une forte mobilité, la connectivité d'un nœud peut avoir un impact important sur les pertes des paquets et donc sur la qualité du stream.

Afin de prendre en compte ces différents aspects, cette contribution vise à concevoir un modèle, qui se veut le plus exhaustif, pour représenter le comportement du streaming p2P multimédia sur les réseaux ad hoc mobiles. Nous prenons en compte trois aspects importants que sont: (i) les technique de streaming p2p présentés dans le modèle de Carra et al et introduit dans la section 4.3 (ii) Le modèle de mobilité basé sur les communautés sociales, proposé par Musolesi et al., et décrit dans le point 4.4 ; et enfin, (iii) la prise en compte de l'overhead des protocoles de routage proposés par Viénot et al. est détaillées dans la section 4.5.

Une fois construit, ce modèle est évalué numériquement (en section 4.7), grâce aux modifications de Groover¹⁴, un solveur numérique proposé par Carra *et al.*, dans lequel nous avons intégré une fonction de génération de matrice de connectivité basée sur les interactions sociales des nœuds mobiles. Dans le calcul du délai nous avons aussi soustrait des débits offerts par chaque nœud, l'overhead provoqué par les messages de maintenance des protocoles de routage. Ce modèle et son évaluation permettent de mieux comprendre le fonctionnement du streaming p2p sur des ad hoc mobiles.

4.2. Travaux existants

Plusieurs travaux en rapport avec cette problématique existent. Carra et al. dans [108][109] proposent un modèle analytique complet qui régit les protocoles de distribution de contenus résolu par la méthode Monte-Carlo. Le modèle intègre un ensemble d'équations différentielles qui déterminent les métriques les plus importantes du streaming; telles que les délais de téléchargement et le nombre de sauts qui séparent le nœud récepteur de la source. Plusieurs architectures de distribution basées sur les arbres et les mailles (dite mesh) ont été modélisées et résolues numériquement dans [110][111]. Ce modèle a l'avantage de bien être adapté à des scénarios présentant un très grand nombre de nœuds. Il permet de quantifier avec précision les différences entre ces deux architectures de distribution de contenu. Dans cette contribution, notre intérêt est porté plus particulièrement sur les topologies mesh, car elles sont plus efficaces que celles à base d'arbres. Cette proposition nous semble plus pertinente pour supporter la mobilité dans le modèle du streaming p2p sur MANET et serait donc intéressant d'observer alors son impact sur la qualité du streaming.

Biskupski et al. [112] ont étudié la topologie en mesh pour le streaming p2p. Ayant assimilé une topologie mesh à plusieurs arbres applicatifs de diffusion, les auteurs soulignent les propriétés des arbres de diffusion qui apparaissent dans ce mesh. Ils recensent les différentes caractéristiques pour enfin les comparer aux arbres à diffusion optimale. Les auteurs montrent ainsi que les arbres qui émergent sont de hauteur sous-optimale et déséquilibrés, ce qui se traduit par une augmentation du retard de la mise en cache dans ce type de système p2p et en particulier dans des environnements hétérogènes. Cependant, pour palier à ce problème, ils proposent un algorithme d'optimisation permet d'adapter la topologie du mesh afin de raccourcir la hauteur des arbres de diffusion et réduire ainsi les délais du buffering. Néanmoins, dans le cas de mobilité, comme nous l'avons souligné, la topologie mesh reste la structure la plus optimale pour supporter la dynamique des nœuds mobiles.

¹⁴Stochastic graph process solver, disponible en ligne sur <http://dit.unitn.it/networking/groover-main.html>

Dans [112], les auteurs présentent une étude analytique quantitative des caractéristiques d'un modèle de streaming hybride. Sur cette base, ils dérivent une équation qui décrit l'évolution de la capacité du système de streaming d'un seul fichier. Ils étendent ensuite l'étude à un scénario de streaming de fichiers multiples. Ils montrent aussi comment un tel système peut répartir de manière optimale la bande passante du serveur entre les différents fichiers existant. Le départ ou la panne imprévisible des pairs étant un facteur critique qui affecte les performances d'un système P2P, les auteurs proposent alors la notion de durée de vie des pairs pour modéliser l'échec d'accès à un pair (générée par une distribution exponentielle) et introduisent une équation initiale d'évolution de la capacité du système. Dans notre contribution, en plus de la dynamique générée par les départs des nœuds, nous nous intéressons également à celle provoquée par leur mobilité. En réalité, la notion de durée de vie change sensiblement dans le cas de nœuds mobiles car ils peuvent toujours être présents dans le réseau mais pas dans le même voisinage.

La modélisation du streaming p2p sur MANET a fait l'objet de très peu de travaux. Néanmoins, un travail intéressant dans ce domaine a fait l'objet de notre attention et est présenté dans la section suivante, c'est celui de Carra et al. Ce modèle

4.3. Modélisation du p2p streaming basé sur le modèle de Carra *et al.*

Les Processus de Graphe Stochastique avec Contraintes (CSGP : Constrained Stochastic Graph Processes), présentés dans [115], sont connus pour leur puissance d'analyse des systèmes p2p de distribution de contenu. En effet, ils permettent d'étudier d'autres contraintes que le temps de téléchargement qui n'est plus suffisant qu'on il s'agit d'applications de streaming dans lesquels le délai, représenté par la distance en nombre de sauts entre la source du stream et la destination, est une métrique essentielle

Carra et al, ont en effet utilisé les CSGP pour construire un modèle analytique du streaming p2p. Ce modèle nous a inspiré pour construire notre propre modèle.

Le modèle de Carra et al capture, en effet, parfaitement les éléments clé d'une application de streaming, à savoir le nombre des liens actifs, et les délais de lecture à la réception de la vidéo, ainsi que la probabilité de ne pas recevoir la vidéo à cause de la dynamique du réseau.

4.3.1. Principes de base

Le modèle considère le système à un instant t , chaque nœud à deux types de relations: des parents (ou des enfants) et des voisins. Pour les parents, un identifiant de la bande échangées est associé - différents identifiants sont sélectionnés selon l'état de la bande, qui peut être actif ou en veille.

Une bande étant une proportion d'un flux multimédia et plusieurs bandes sont nécessaires pour reconstruire un stream.

Soit α_k et σ_k , respectivement, les identifiants actifs et en veille d'une bande k . η est l'identifiant qui indique que deux nœuds sont voisins dans l'overlay. Le graphe représentant les réseaux de distribution peut donc être décrit par la matrice de connectivité S , où chaque élément s_{ij} décrit la relation entre les nœuds i et j :

n'échangent pas de bandes si les nœuds i et j sont voisins, et i est un parent actif de j avec une bande k si les nœuds i et j sont voisins et i est un parent en veille de j avec une bande k

Étant donné que chaque parent peut envoyer une seule bande à chaque enfant, s_{ij} peut prendre seulement les valeurs définies ci-dessus. En plus de la matrice d'adjacence, chaque nœud a un débit sortant qui peut être représenté, pour un débit par bande connu, par le nombre maximum d'enfants qu'un parent actif peut supporter.

Les transitions d'état sont déterminées par les événements tels l'arrivée, le départ, et la mise à jour. Nous supposons que les taux d'apparition de ces événements sont distribués exponentiellement avec respectivement les paramètres λ , μ et λ_{up} . Pour chaque événement, il est possible de calculer les probabilités de transition d'un état S à un état S' défini par une nouvelle matrice de connectivité. Chaque événement correspond à une série d'opérations. Par conséquent, la transition correspondante n'est pas simple et de plus d'un élément dans la matrice de connectivité pourrait changer.

Contrairement aux différents modèles analytiques proposés et présentés dans la littérature lesquels ne permettent d'analyser que des graphes statiques[113], le modèle de Carra et al basé sur les graphes stochastiques permet de réaliser les mêmes analyses dans des graphes. Ce formalisme mathématique est utilisé pour étudier les problèmes fondamentaux de performance des systèmes de streaming peer to peer. Le niveau d'abstraction qu'il offre a l'avantage de capturer les comportements des nœuds dans différentes conditions de réseau en gardant une complexité limitée. Le modèle de Carra et al est très important dans la réalisation de notre objectif de modélisation du streaming p2p sur MANET par sa prise en compte de la dynamique du réseau, c'est-à-dire l'évolution du graphe dans le temps.

4.3.2. Analyse des applications streaming fondées sur le mesh

F Dans ce modèle de Carra et al, le réseau p2p est représenté par un graphe dont les sommets sont les nœuds et les arêtes reflètent leurs relations réciproques. Quand un utilisateur commence à partager de la vidéo, il reçoit et retransmet les paquets vidéo, il utilise un sous ensemble de ses connexions actives (sortantes et entrantes). Le nombre de parents actifs constitue une métrique

qui mesure la qualité de réception de la vidéo. Notre objectif est d'analyser les caractéristiques de ce graphe de distribution et plus particulièrement le sous graphe, construit par l'application de streaming exploité par les nœuds mobiles, comme illustrée dans la Figure 16.

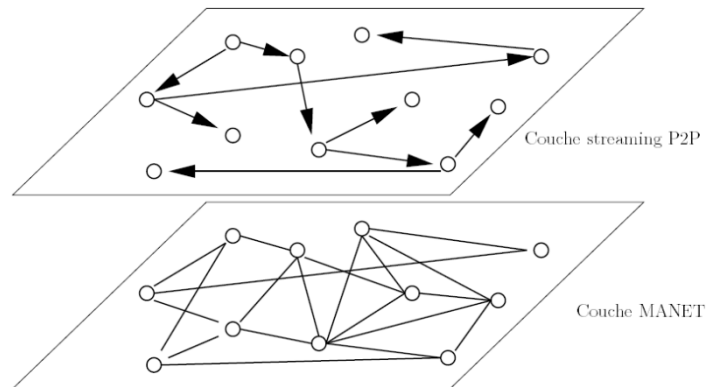


Figure 16 Couches du modèle

En général, le graphe du streaming p2p varie dans le temps. A tout moment, des nœuds et des liens sont amenés à réagir ou disparaître. Cette évolution peut alors être perçue comme un processus stochastique aux propriétés markoviennes, parce qu'à un instant $t + \Delta t$, le graphe dépend étroitement de son état précédent, soit l'instant t ; ainsi que de tous les événements qui surviennent durant Δt , tels que l'arrivée, le départ ou, dans notre cas, la mobilité des nœuds.

Le modèle s'applique à n'importe quelle application de streaming p2p, car il en capture les caractéristiques de base, communes à tous les protocoles de type hybride tels que présentés dans [136][137], où le réseau mesh est construit par la superposition de plusieurs arbres. Considérons alors une application de streaming p2p qui construit un réseau overlay. Le contenu est distribué en utilisant différentes bandes. Une bande étant une proportion d'un flux multimédia.

En résumé, et à partir de la comparaison dans le paragraphe 2.2.2.4.c, il apparaît évident que les topologies mesh sont les plus adéquates pour le support de la mobilité, car les arbres sont très difficiles à maintenir en cas de mobilité et de dynamique forte dans les arbres. Nous avons donc adopté dans notre étude une topologie de type mesh, ce qui rejoint également les principes même du peer to peer streaming.

4.3.3. Comportement du système ciblé

4.3.3.1. *Le streaming*

Le contenu est distribué en utilisant R différentes bandes. Chaque bande contient une partie du flux (codée en utilisant une technique telle que le MDC : Multiple Description Coding [119]). Un nœud a besoin de $R' < R$, sur les R bandes qui composent le flux, pour parvenir à reconstituer le stream avec une bonne qualité, tandis que le reste, c'est-à-dire les $R - R'$ bandes, contiennent des informations redondantes qui servent, par des mécanismes de correction d'erreurs (tels que le FEC : Forward Error Correction [121]), à corriger les éventuelles pertes de paquets vidéo.

Dans le cas de topologies fixe, l'évolution du réseau est soumise à deux principaux événements que sont : l'arrivée et le départ des nœuds. Nous supposons que les arrivées et les départs sont distribués exponentiellement avec des taux respectifs $\lambda(t)$ et $\mu(t)$. La dépendance du modèle du temps, rend le modèle plus souple et permet de décrire divers modes d'arrivée, comme le *flash crowd* (arrivées massives), phénomène très connu dans la communauté p2p. Le modèle définit la durée de la vidéo par t_{str} et le nombre moyen de nœuds récepteurs par N . Une fraction des nœuds rejoint le réseau à $t=0$, et il y a un délai de temps pendant lequel $\lambda(t)$ reste supérieur à $\mu(t)$ et cela jusqu'à ce que l'état stationnaire soit atteint. Le cas d'un réseau ad hoc mobile, l'essentiel de notre contribution, sera explicité dans la section 4.6.

Le taux de départ $\mu(t)$ correspond à l'inverse de la moyenne du temps passé par un nœud dans le système (temps de séjour). À l'état stationnaire, les départs et les arrivées de nœuds se compensent. Durant un intervalle de temps T , le rapport entre le nombre total de départs et le nombre moyen de nœuds actifs définit la dynamique du système. Par exemple, si ce paramètre est égal à 1, cela veut dire qu'au cours de l'intervalle T le nombre de nœuds ayant quitté le réseau est égal à la moyenne des nœuds du système, en d'autres termes, il y a un changement stochastique complet des nœuds durant cette période.

Les nœuds sont répartis en différentes classes en fonction de leur bande passante. Chaque classe j a des débits montant $b_u^{(j)}$ et descendant $b_d^{(j)}$, qui peuvent être symétriques, asymétriques ou corrélés, par exemple, $b_d^i + b_u^i$ constant, comme dans un cas où le médium d'accès est partagé (cas des réseaux sans fils). Les bandes passantes sont des variables aléatoires décrites par une fonction de densité de probabilité connue.

Nous supposons que tous les nœuds ont un débit de téléchargement au moins égal à la vitesse de la vidéo est égale à r_{str} . Dès lors, à chaque bande peut être téléchargée à une vitesse égale à r_{str}/R' . Nous supposons que le serveur est capable d'envoyer toutes les bandes R , c'est-à-dire, qu'il a une bande passante supérieure à $R \cdot r_{str}/R'$. Par ailleurs, chaque nœud a une contrainte, sur les nombres

maximum k_{max} et minimum k_{min} de téléchargements, qui limitent la possibilité de degrés sortants du nœud. Parmi ses B voisins, un nœud choisit ses parents, c'est-à-dire, ceux auprès desquels il obtient la vidéo. Seuls R' parents sont actifs à la fois tandis que les autres restent en veille, dans la mesure où ils ne seront utilisés en backup dans le cas de la défaillance d'un ou plusieurs parents actifs.

4.3.3.2. Arrivée, départ & mise à jour

A la création de réseau, les nœuds construisent un arbre de diffusion par bande. Le nombre de nœuds dans chaque arbre dépend des caractéristiques de la diffusion, par exemple, les débits de sortie des nœuds parents, qui limitent le nombre de fils qu'ils peuvent avoir. Chacun des nœuds collabore avec plusieurs arbres de diffusion à la fois. Quand un nouveau nœud arrive, il choisit au hasard un premier nœud de contact, grâce auquel il construit sa liste de voisins. Afin d'augmenter son degré de connexion, un nœud choisit périodiquement de nouveaux liens parmi ses voisins avec une fréquence λ_{up} .

Lorsqu'un nœud quitte le réseau p2p, toutes ses connexions entrantes et sortantes sont annulées. Ceci met des nœuds dans un état *orphelin* lesquels vont tenter de trouver un nouveau nœud parent dans leur voisinage. Si le parent disparu s'est mis au préalable en veille, le nœud ne réagit pas (il perd tout simplement un parent de secours). Si le parent disparu était actif, le nœud tente alors de solliciter d'abord l'un de ses parents de secours, c'est-à-dire, qu'il va d'abord essayer de télécharger le stream à partir du parent en veille si celui-ci dispose d'un débit disponible. Si le nœud n'a pas de parents de secours, il perd temporairement la réception du stream et donc de la qualité pendant toute la période de recherche d'un nouveau parent.

4.3.4. Modèle mathématique

L'analyse du graphe peut être réalisée à travers l'étude des propriétés de distributions de degré et de délai. Etant donné un processus *Markovien*, le comportement temporel peut être décrit en utilisant une équation différentielle de type *Chapman-Kolmogorov*, connue sous le nom d'*Equation Maitresse* (EM) [115]. Pour un nœud s , la variation de la probabilité de trouver la valeur α (α étant le degré de connexion ou le délai) au temps t , peut s'exprimer par :

$$\frac{\partial p(\alpha, t)}{\partial t} = \sum_{\beta} w_{\beta, \alpha}(t) p_s(\beta, t) \quad (4.1)$$

où $w(t)$ est le taux de transition de la valeur β à la valeur α au moment t . Les taux de transition sont étroitement liés aux politiques du protocole de streaming. La formulation générale de l'EM doit être spécifique à notre problème, c'est-à-dire, le modèle doit définir toutes les transitions possibles. L'EM détermine l'évolution dans le temps du système stochastique pour chaque nœud s .

Il est également intéressant d'avoir des représentations de la valeur moyenne (degré k ou délai l). Le modèle de Carra prévoit des équations dites de taux (Rate Equations : RE):

$$\frac{\partial \bar{\alpha}}{\partial t} = \frac{\partial}{\partial t} \sum_{\alpha} \alpha p_s(\alpha, t) \quad (4.2)$$

Celles-ci décrivent la moyenne des quantités et expriment le comportement déterministe du système: en fait, les REs sont un ensemble d'équations différentielles qui décrivent l'évolution temporelle de la moyenne des propriétés du système. La Figure 17 montre la relation entre les résultats de l'EM et le résultat de la RE pour une variable aléatoire donnée par exemple, la connectivité (degré) et le délai (nombre de sauts) d'un nœud. L'équation maîtresse donne un aperçu clair du système, car elle caractérise ses propriétés dans le temps. Tandis que, l'équation de taux donne une valeur moyenne qui est équivalente à l'approximation du système.

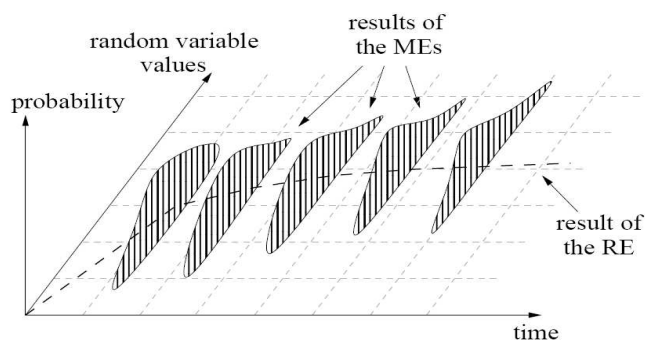


Figure 17 : Comparaison des résultats des équations maîtresse et de taux [109]

A partir de ces équation Carra et al. déduisent celles représentant les caractéristiques principales du graphe de distribution du streaming : à savoir la distribution de probabilités du degré et du délai. Dès lors, à partir de ces formules, l'analyse du graphe peut être envisagée.

La distribution du degré de connexion $p_s(k, t)$ est la probabilité que le nœud s ait k connexions à l'instant t . En ayant la distribution du degré de chaque nœud dans le graphe, nous pouvons déduire degré total de connexion du réseau :

$$P(k, t) = \frac{1}{N(t)} \sum_{s=1}^{N(t)} p_s(k, t) \quad (4.3)$$

$N(t)$ étant le nombre de nœuds attachés au réseau à un instant t .

Nous pouvons facilement distinguer entre les degrés sortant et entrant en définissant une équation pour chaque type de connexion.

La distribution du délai représente la distance entre le nœud source du flux et le nœud de destination selon l’algorithme du plus court chemin. Soit $p_s(l, t)$ la probabilité que le nœud s soit à l sauts de la source à un instant t .

La distribution globale du délai est dérivée de la même manière que la formule précédente et peut s’exprimer comme suit:

$$P(l, t) = \frac{1}{N(t)} \sum_{s=1}^{N(t)} p_s(l, t) \quad (4.4)$$

L’approche proposée par Carra *et al.* Permet de fournir la solution de l’EM, et donc, la caractérisation complète du système.

Lorsque la complexité du graphe augmente les ressources nécessaires pour le résoudre deviennent coûteuses, pour cela les auteurs proposent de se concentrer sur les RE afin d’obtenir une analyse de la valeur moyenne. Ainsi, leur méthode offre une grande souplesse qui nous permet de décider du niveau de détail souhaité dans l’analyse de notre système. Pour de plus amples informations quant à la description détaillée de ce modèle, le lecteur peut se référer aux travaux de Carra et al présentés dans [108][109].

Dans la section 4.6, nous mettons en œuvre cette représentation analytique pour finir l’étude que nous menons sur les streaming p2p sur MANET.

4.4. Modélisation de la mobilité par Musolesi *et al.*

Le modèle de mobilité est très important dans l’étude des réseaux. Ce domaine a connu un intérêt croissant ses dernières années pour essayer de comprendre comment la mobilité impact sur les performances du réseau. Parmi les modèles de mobilité que nous avons présentés dans la section 2.3.2, nous utiliseront le modèle de mobilité basé sur la notion de communauté issu de la théorie des réseaux sociaux [121]. Un des principaux points forts de ce modèle est le réseau social, qui relie les utilisateurs mobiles entre eux afin de générer synthétiquement des structures de réseau réalistes. Ce modèle permet à une collection d’hôtes d’être regroupés d’une manière représentative des relations sociales entre les individus *ou les affinités* qu’ils entretiennent. Ce regroupement est alors seulement reflété sur un espace topologique, avec une topographie biaisée par la force des liens sociaux. Les mouvements des nœuds sont également régis par les relations entre eux. Le modèle permet définir différents types de relations sur une période de temps définie (journée, semaine, etc.). Il serait intéressant, par exemple, de pouvoir décrire des relations professionnelles (câd sur le lieu de travail pendant les heures de travail, les jours de semaine) avec un poids plus important que les relations familiales ou amicales (câd les soirs et week-ends).

Ce modèle a été évalué au moyen de traces réelles de mobilité fournies par le Laboratoire de recherche Intel dans [128], où les auteurs montrent que le modèle fournit une bonne approximation

de mouvements réels vis à vis de plusieurs paramètres fondamentaux, telles que la répartition de la durée des contacts et des temps d'inter-contacts. En particulier, les données montrent qu'une loi de puissance approximative régit une grande gamme des temps d'inter-contacts contrairement aux durées de contacts qui suivent la même loi mais sur un éventail plus limité de valeurs. Ces caractéristiques statistiques sont également très similaires à celles observées par les chercheurs de l'université de Californie à San Diego et du Dartmouth College [129].

Un exemple de réseau social est représenté dans la Figure 18. Le réseau social est représenté par graphes pondéré ou chaque lien définit le degré d'affinité entre deux individus.

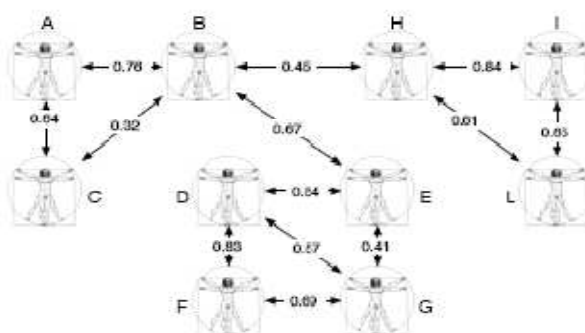


Figure 18: Un exemple de réseau social [121]

Nous allons maintenant, à partir de ce graphe social, décrire plus en détails les aspects essentiels du modèle de mobilité basé sur la communauté.

4.4.1. Matrice d'affinité (affinity)

Dans le graphe social, chaque nœud représente une personne. Le poids, associés à chaque lien du réseau, sont utilisés pour modéliser la force des interactions entre les individus (affinité). Nous supposons que ces poids, qui sont exprimés en tant que mesure de l'affinité des liens sociaux, peuvent aussi être vus comme une mesure de la probabilité de co-localisation géographique. Nous modélisons le degré d'affinité sociale entre deux personnes par une variable dont les valeurs sont comprises dans l'intervalle $[0, 1]$. 0 indique que l'affinité est quasi absente ; alors que 1 indique une affinité sociale absolue. Différents réseaux sociaux peuvent être valables pour les différents moments de la journée ou de la semaine.

Ainsi, le réseau présenté dans la Figure 18 peut être représenté par une matrice symétrique Aff de 10×10 illustrée sur la Figure 19. Les lignes et colonnes correspondent à des noms de nœuds. Dans ce qui suit, cette matrice sera appelée matrice d'affinité. L'élément générique $Aff(i,j)$ qui représente l'affinité entre deux individus i et j , sera appelé indicateur d'affinité. Les valeurs diagonales représentent les relations de l'individu avec lui-même et sont, par convention, à 1..

$$\text{Aff} = \begin{bmatrix} 1 & 0.76 & 0.64 & 0.11 & 0.05 & 0 & 0 & 0.12 & 0.15 & 0 \\ 0.76 & 1 & 0.32 & 0 & 0.67 & 0.13 & 0.23 & 0.43 & 0 & 0.05 \\ 0.64 & 0.32 & 1 & 0.13 & 0.24 & 0 & 0 & 0.15 & 0 & 0 \\ 0.11 & 0 & 0.13 & 1 & 0.54 & 0.83 & 0.57 & 0 & 0 & 0 \\ 0.05 & 0.67 & 0.24 & 0.54 & 1 & 0.2 & 0.41 & 0.2 & 0.23 & 0 \\ 0 & 0.13 & 0 & 0.83 & 0.2 & 1 & 0.69 & 0.15 & 0 & 0 \\ 0 & 0.23 & 0 & 0.57 & 0.41 & 0.69 & 1 & 0.18 & 0 & 0.12 \\ 0.12 & 0.43 & 0.15 & 0 & 0.2 & 0.15 & 0.18 & 1 & 0.84 & 0.61 \\ 0.15 & 0 & 0 & 0 & 0.23 & 0 & 0 & 0.84 & 1 & 0.65 \\ 0 & 0.05 & 0 & 0 & 0 & 0 & 0.12 & 0.61 & 0.65 & 1 \end{bmatrix}$$

Figure 19: Exemple d'une Matrice d'affinité représentant un réseau social.

La définition de cette matrice d'affinité est une étape clé de ce modèle. Il s'agit clairement d'un modèle simplifié de relations humaines. La définition de ces poids est un domaine de recherche à part entière en sociologie [128]. La matrice d'affinité est également utilisée pour générer la matrice d'adjacence ou de connectivité.

En partant de la matrice Aff , nous générons une matrice C des donnée en entrée $C_{i,j}$, si et seulement si $Aff_{i,j}$ est supérieure à un certain seuil. La connectivité générée par la matrice d'affinité, est illustrée sur la Figure 20. L'idée cardinale, est d'avoir un seuil d'affinité au-dessus duquel deux personnes sont considérées être en relation.

$$\text{Adj} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Figure 20: Exemple de matrice d'adjacence représentant le réseau social.

4.4.2. Modèle social de la communauté

La matrice d'affinité, et par conséquent, la matrice de connectivité, peuvent être obtenues, soit à partir de données disponibles (par exemple, une enquête sociologique), ou bien, en utilisant des modèles mathématiques reproduisant des caractéristiques particulières des réseaux sociaux réels. Le *community based mobility model* utilise par défaut le modèle social *Caveman* [122] pour la génération de réseaux sociaux réalistes, basée sur la notion de *clustering* et une faible longueur moyenne du chemin (diamètre du graphe). Cependant, il s'agit d'un aspect personnalisable, c'est à dire, si le type de scénario à tester est clair, une matrice définie par l'utilisateur peut aussi être utilisée.

Le scénario de simulation est établi par la transposition des groupes d'hôtes sur certaines zones de l'espace géographique. Après la définition du graphe social décrit ci-dessus, les groupes, un

ensemble de nœuds fortement connectés dans le graphe, doivent être isolés. Les auteurs utilisent l'algorithme proposé dans [127] pour détecter la présence de structures de communautés dans les réseaux sociaux représentés par des matrices, telles que la matrice de connectivité définie précédemment. Cet algorithme est basé sur le calcul de ce que l'on appelle *betweenness* d'une arête. Celle-ci donne une mesure de la *centralité* des nœuds

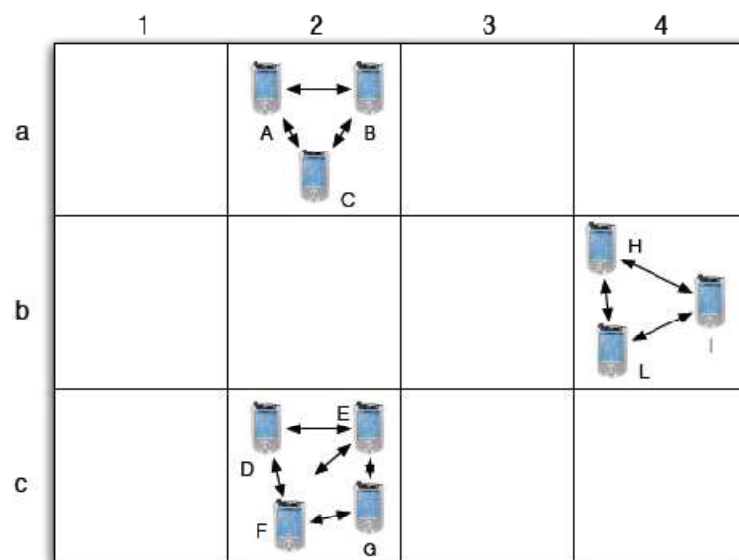


Figure 21: Exemple d'une communauté mobile [121].

Afin d'illustrer ce processus, nous devons maintenant examiner le réseau social dans la Figure 18. Trois communautés sont détectés: $C_1 = (A, B, C)$, $C_2 = (D, E, F, G)$ et $C_3 = (H, I, L)$. Maintenant que les communautés sont identifiées compte tenu de la matrice, celles-ci doivent être associées à leurs positions.

Une fois les communautés identifiées, chacune d'elles est associée de manière aléatoire à une position géographique donnée sur une grille (c'est-à-dire, une case). Nous utilisons le symbole $S_{p,q}$ pour noter une case de coordonnées p,q . Le nombre de lignes et de colonnes sont des entrées du modèle de mobilité.

En revenant à l'exemple, nous pouvons voir comment ces groupes peuvent être placés sur une grille de 3x4 (la dimension de la grille est configurable par l'utilisateur et agit sur la densité des nœuds dans chaque case). Les trois communautés C_1 , C_2 , C_3 sont placées respectivement dans la grille dans les cases $S_{a,2}$, $S_{c,2}$, $S_{b,4}$. Chaque nœud d'une communauté est placé dans des positions choisies au hasard à l'intérieur de la case attribuée.

Comme décrit dans le paragraphe précédent, un mobile est d'abord placé dans une certaine position dans la grille. Puis, afin de dicter son mouvement, un objectif lui est assigné. Plus formellement, nous disons que tout mobile m est associé à une case $S_{p,q}$ si la réalisation de son objectif peut se faire à l'intérieur de cette case. Notons que le nœud m n'est pas toujours nécessairement placé à l'intérieur de la case $S_{p,q}$, en dépit de cette association (voir ci-dessous).

L'objectif d'un nœud est simplement un point représentant sa destination finale (même principe que celui du modèle Random Way-Point), sauf que la sélection de l'objectif n'est pas aléatoire. Quand elle est initialement établie, la destination de chaque mobile est choisie au hasard à l'intérieur du carré associé à sa communauté (c'est-à-dire, les premiers objectifs de tous les membres de la communauté C_l seront choisis à l'intérieur de la case $S_{a,l}$).

Quand un objectif est atteint, un nouvel objectif est choisi de la manière suivante: Un certain nombre d'hôtes (zéro ou plus) sont associés à chaque carré $S_{p,q}$ à un instant t . Pour un nœud donné, chaque carré exerce une certaine attractivité sociale. Celle-ci représente une mesure de son importance en termes de relation sociale pour les nœuds mobiles. L'importance sociale est calculée en évaluant la force des relations avec les hôtes qui se dirigent vers cette même case (c'est-à-dire, avec ceux qui ont pour objectif actuel cette case particulière).

Plus formellement, étant donné $CS_{p,q}$ (c'est-à-dire, l'ensemble des hôtes associée à la case $S_{p,q}$), SA_{p,q_i} qui définit l'attractivité sociale de cette case pour le mobile peut être calculé de la manière suivante :

$$SA_{p,q_i} = \frac{\sum_{j \in CS_{p,q}} m_{i,j}}{w} \quad (4.5)$$

Où w représente le nombre de mobiles associés à la case $S_{p,q}$.

Ici, l'attractivité sociale de la case située à la position (p, q) pour un hôte i est définie par la somme des indicateurs d'affinité que représentent ses relations avec les autres hôtes situés dans cette même case, normalisé par le nombre total d'hôtes associés à cette case. Si $w=0$ (c'est-à-dire, la case est vide), la valeur du SA_{p,q_i} devient nulle.

Le modèle de mobilité choisi propose deux mécanismes de sélection du prochain objectif ; le premier, déterministe, basé sur la sélection de la case de plus forte attractivité ; et le deuxième, probabiliste, basé sur une probabilité de sélection d'un but dans un carré proportionnellement à son attractivité. Dans la première approche, les objectifs sont choisis uniquement à l'intérieur des carrés associés à la communauté, tandis que dans la seconde, les mobiles peuvent également choisir, au hasard, leurs objectifs dans d'autres endroits de l'espace de simulation, avec une certaine probabilité.

Plus précisément, le deuxième mécanisme permet une sélection des destinations non pas uniquement sur la base des relations sociales pour ajouter plus de réalisme au modèle. Grâce à ce mécanisme, le nouvel objectif est choisi, au hasard, à l'intérieur du carré caractérisé par la plus forte attractivité sociale; cela peut être à l'intérieur de la même case ou dans une autre. De nouveaux objectifs sont choisis à l'intérieur de la même zone lorsque le réseau social, en entrée, se compose de communautés faiblement connectées. Dans ce cas, les hôtes associés à différentes communautés ont, en moyenne, de faibles relations entre eux. D'autre part, quand un mobile a de solides relations avec les deux groupes, il peut être attiré par une autre case. D'un point de vue théorie des graphes, cela signifie que l'hôte est situé entre deux (ou plusieurs) clusters de nœuds dans le réseau social.

Une solution alternative choisit le prochain objectif selon l'attractivité de chaque zone de l'espace. C'est-à-dire, le modèle attribue une probabilité $P(s = S_{p,q_i})$, de sélectionner la case S_{p,q_i} définie comme suit:

$$P(s = S_{p,q_i}) = \frac{SA_{p,q_i} + d}{\sum_{j=1}^{p \times q} (SA_{p,q_j} + d)} \quad (4.6)$$

où $d > 1$ est une valeur aléatoire qui assure que la probabilité de choisir une destination dans un carré soit toujours non nulle. Le paramètre d peut être utilisé pour augmenter le caractère aléatoire du modèle lors de la sélection du nouvel objectif. Ceci peut être utilisé pour augmenter le réalisme du scénario de génération, car, dans des situations réelles, les humains se déplacent également vers des zones où il n'y a personne pour des raisons non liées à leur environnement social.

4.5. Prise en compte de l'overhead dans le calcul des délais

Les travaux de Viénot et al. [128] évaluent, d'une manière substantielle, le débit consommé par la circulation des messages de contrôle générés par les protocoles de routage, réactifs et proactifs. En effet, le modèle qu'ils proposent permet de prédire l'overhead en considérant la mobilité et le type de routage. Il prend en compte différents paramètres du réseau tels que la densité des nœuds δ , le taux de mobilité μ .

Les paramètres utilisés pour modéliser le trafic réseau sont: le nombre de nœuds mobiles N , la durée de vie moyenne d'un lien T_{vie} , sachant que $T_{vie} = 1/\mu$ est le taux de perte de liens). En ce qui concerne le modèle de mobilité, la longueur moyenne d'un chemin L , dépend principalement de la topologie du réseau. Les auteurs supposent que μ et L restent constants dans un réseau toujours connecté. Selon le type de protocoles de routages ad hoc, réactif ou proactif, certains paramètres viennent compléter le modèle.

4.5.1. Cas de protocole réactif

Kj Les paramètres suivants sont utilisés pour modéliser un protocole réactif: le nombre moyen σ de routes créées par nœud (Route Request), et le nombre λ de routes actives par nœud. Une route active correspond à un chemin par lequel la source envoie des paquets en continue vers une destination. Les protocoles réactifs peuvent inclure des messages *hello* pour prévenir la perte de lien (Cf. 2.3.1.1). Si c'est le cas, h_r dénote leur fréquence et H_r leur taille.

Nous devons aussi considérer des métriques spécifiques au protocole [128]: le nombre moyen d'émission de Route Request (peut aussi inclure les nombre de Route Reply de la destination) B_r , ainsi que le facteur d'optimisation de la requête de route, qui dépend des paramètres du réseau et du trafic ($o_r = B_r/N$). Pour un protocole d'inondation pure, si nous considérons le Route Reply, nous obtenons $o_r = 1 + R/N$, R étant le nombre des messages de réponse. La surcharge générée par l'envoi de messages hello est donc $h_r \cdot N$ paquets par seconde consommant ainsi un débit de $h_r \cdot H_r \cdot N$.

Dans le cas d'un réseau ad hoc fixe, un protocole de routage réactif produit des requêtes de route à chaque seconde. Ceci va générer $\sigma \cdot o_r \cdot N^2$ paquets de contrôle dans le réseau; Donc la surcharge totale provoquée par le protocole de routage est:

$$h_r \cdot N + \sigma \cdot o_r \cdot N^2 \quad (4.7)$$

si RQ_r désigne la taille d'un paquet de requête, donc cet overhead occupe une bande passante de:

$$h_r \cdot N + \sigma \cdot o_r \cdot R \cdot Q_r \cdot N^2 \quad (4.8)$$

Sachant que la mobilité dans un MANET est considérée comme une succession de perte et de création de lien. La perte de lien est la représentation la plus significative de la réaction du protocole de routage à la mobilité des nœuds, car il doit réagir rapidement à cette perte surtout quand le lien fait partie d'une liaison active.

Toutefois, quand une perte de lien est détectée, le protocole génère une nouvelle requête de route afin de réparer toutes les routes qui passent par ce lien. S'il y a $\lambda \cdot N$ routes actives, il y aurait alors $\lambda \cdot N \cdot L$ liens actifs. Quand un lien est perdu, soit la source, soit le nœud qui détecte cette perte, initie une demande de découverte de route pour chacune des destinations atteintes via ce lien. Dans le cas ou ce n'est pas la source qui agit, un gain est rajouté au paramètre o_r , ce qui porte la surcharge totale de

$$\mu \cdot o_r \cdot (\lambda NL) \cdot N = \mu \cdot o_r \cdot \lambda \cdot L \cdot N^2 \quad (4.9)$$

Paquets, correspondant à un débit consommé de:

$$\mu \cdot o_r \cdot \lambda \cdot L \cdot RQ_r \cdot N^2 \quad (4.10)$$

4.5.2. Cas de protocole proactif

Les protocoles proactifs (présentés dans 2.3.1.1) utilisent principalement une émission régulière de messages de contrôle ; ils ne génèrent donc aucune surcharge lors de la création de route.

Les paramètres qui interviennent dans ce type de protocoles sont : la fréquence h_p d'émission des messages hello pour découvrir préalablement la topologie locale. Ces mobiles diffusent aussi, avec une fréquence t_p , un autre type de paquets pour avoir une connaissance globale de la topologie (au delà de leurs voisinages). Ils définissent aussi AN_p , le prochain saut actif (*the active next hop*); qui correspond au nombre moyen de liens actifs par nœud, utilisé pour connaître quel type de changement de topologie peut enclencher un contrôle de trafic additionnel. Sachant que les protocoles proactifs peuvent profiter de leur connaissance de la topologie pour optimiser la diffusion, un facteur d'optimisation $o_p = B_p/N$ est alors défini, B_p étant le nombre d'émissions nécessaires pour faire une diffusion topologique (*Topology Broadcast*).

Un protocole proactif produit $h_p \cdot N$ messages hello en une seconde pour tout le réseau. Et $t_p \cdot N$ messages de contrôle de topologie par seconde, produisant $o_p \cdot t_p \cdot N^2$ paquets. La surcharge dans un réseau ad hoc fixe est alors:

$$h_p \cdot N + o_p \cdot t_p \cdot N^2 \quad (4.11)$$

L'utilisation de la bande passante est de

$$h_p \cdot H_p \cdot N + o_p \cdot t_p \cdot T_p \cdot N^2 \quad (4.12)$$

Où T_p est la taille d'un paquet de contrôle de topologie.

Pour évaluer la surcharge supplémentaire générée en réaction à la mobilité des nœuds, celle-ci est similaire au cas du routage réactif. Le nœud qui détecte la rupture de lien émet un paquet de contrôle de topologie supplémentaire. En moyenne, un nœud peut faire partie de λN routes actives, et différentes routes peuvent avoir un lien sortant commun. Cependant, la probabilité que le prochain saut pour ces routes soit le même est, certainement, plus grande que la probabilité qu'ils aient la même destination. Avec l'introduction du nombre moyen de liens actifs AN_p par nœud, la surcharge totale dans le cas mobile est égale à:

$$\mu \cdot o_p \cdot AN_p \cdot N^2 \quad (4.13)$$

La bande passante consommée est elle de:

$$\mu \cdot o_p \cdot AN_p \cdot T_p \cdot N^2 \quad (4.14)$$

Dans notre modélisation du streaming p2p sur MANET, nous nous intéressons au cas mobile. Donc, nous refléterons la surcharge provoquée dans chaque nœud sur la distribution du délai et cela en déduisant du débit offert par chaque nœud la valeur de la bande passante consommée par les messages de contrôle.

Dans ce qui suit, nous expliquons comment prendre en compte cette surcharge et la transformer en délai supplémentaire pour l'intégrer enfin dans le modèle MADP2PStream.

Les distributions de probabilités de degré et de délai sur les nœuds en réaction à la mobilité sont fortement influencées par la dynamique causées par le mouvement des nœuds. Pour étudier cette mobilité dans le modèle du streaming p2p, nous prenons en compte deux aspects: l'interaction sociale des utilisateurs représentée par la matrice d'affinité et la perte des liens sans fils représentée soit par les nouvelles routes générées par le protocole de routage. Cette étude est réalisée dans la section suivante.

4.6. Représentation de la mobilité dans le graphe de streaming p2p

L'objectif de cette partie est définir un nouveau modèle qui intègre toutes les contributions présentées précédemment et qui permette de modéliser le streaming p2p dans un réseau MANET. L'approche consiste à étendre le modèle du streaming p2p de Carra *et al.*, pour prendre en compte la mobilité des nœuds dans un réseau MANET. Pour cela, on utilise le modèle de mobilité introduit par Musolesi *et al.* Augmenté des proposition de Viénot *et al.*, pour prendre en compte l'overhead des protocoles de routage.

En effet, la mobilité dans MANETs peut être vue comme une série de ruptures et de créations de liens. En d'autres termes, un nœud qui se déplace perd les liens avec ses voisins de départ et crée d'autres connexions dans la zone d'arrivée. C'est de cette façon que nous représentons, dans MADP2PStream, le modèle de mobilité de la communauté social sur la base de la matrice d'affinité *Aff*.

Maintenant, nous allons extraire de modèle de mobilité, les paramètres qui nous intéressent et les traduire dans le modèle de streaming p2p afin de prendre en compte le mouvement des nœuds. Si nous reprenons le graphe du modèle de streaming p2p et ses deux équations principales (4.3) et (4.4) auxquelles nous rajoutons la notion d'affinité entre les nœuds dans le temps, cela nous donne deux nouvelles formules représentant la distribution totale du degré

$$P(k, t) = \frac{1}{N(t)} \sum_{s=1}^N p_s(k, t) \cdot [\sum_{v=neihb1}^k Aff_{s,v}(t)] \quad (4.15)$$

$p_s(k, t)$ La probabilité pour un nœud s d'avoir k connexions actives et donc k voisins

$Aff_{s,v}$ L'affinité entre un nœud s et ses v voisins
 $N(t)$ Nombre de nœuds dans le réseau à l'instant t

La même approche est utilisée pour le délai, ce qui nous donne

$$P(l, t) = \frac{1}{N(t)} \sum_{s=1}^{N(t)} p_s(l, t) \cdot \left[\sum_{strip=stripe1}^{stripen(t)} \left(\prod_{prnt=1}^{parnten(t)} Aff_{s,prnt}(t) \right) \right] \quad (4.16)$$

$p_s(l, t)$ La probabilité pour un nœud s de recevoir le flux avec un délai l
 $Aff_{s,prnt}$ L'affinité deux à deux des nœuds constituant le chemin entre un nœud s et le parent $prnt$ qui fourni une bande $strip$
 $N(t)$ Nombre de nœuds dans le réseau à l'instant t

D'un autre coté, et concernant le routage, lorsque la rupture d'un lien est détectée, le protocole de routage MANET, qu'il soit réactif ou proactif, va générer un nouvel itinéraire afin de réparer le lien brisé. Cette opération de maintenance consomme une partie de la bande passante des nœuds mobiles, à cause des messages de maintien de la topologie (Cf. 4.5). Ceci nous donne des délais plus importants pour l'acheminement des paquets vidéo.

$$P(l, t) = \frac{\left[\sum_{s=1}^{N(t)} p_s(l, t) \cdot \left[\sum_{strip=stripe1}^{stripen(t)} \left(\prod_{prnt=1}^{parnten(t)} Aff_{s,prnt}(t) \right) \right] \right]}{N(t)} \cdot \left(1 + \frac{\mu o_r \cdot \lambda L R Q_r N^2(t)}{b_u^s} \right) \quad (4.17)$$

Pour un routage réactif ayant comme paramètres :

μ le taux de mobilité
 o_r facteur d'optimisation de la requête de route
 λ densité de trafic
 L Longueur moyenne d'une route
 RQ_r taille d'un paquet de requête de route
 b_u^i débit montant par nœud

et la formule suivante pour un routage proactif

$$P(l, t) = \frac{\left[\sum_{s=1}^{N(t)} p_s(l, t) \cdot \left[\sum_{strip=stripe1}^{stripen(t)} \left(\prod_{prnt=1}^{parnten(t)} Aff_{s,prnt}(t) \right) \right] \right]}{N(t)} \cdot \left(1 + \frac{\mu \cdot o_p \cdot A N_p \cdot T_p \cdot N^2(t)}{b_u^s} \right) \quad (4.18)$$

Avec :

μ le taux de mobilité
 o_p facteur d'optimisation de la requête de route

AN_p nombre moyen de liens actifs
 T_p taille de message de contrôle de topologie
 b_u^i débit montant par nœud

4.7. Evaluations

4.7.1. Groover: solveur d'équations différentielles

Le solveur de processus de graphe stochastique (GROOVER) est un outil d'intégration basé sur la méthode de Monte Carlo. Il explore l'espace d'état d'une chaîne de Markov dont les états sont représentés par des graphes: chaque graphe contient les nœuds qui ont été atteints par le contenu ainsi que les liens utilisés pour sa diffusion. Il a été développé à l'Université de Trento et son objectif est d'analyser les systèmes de distribution de contenu. Ce solveur supporte les deux topologies mesh et arbre.

Nous avons repris le code de Groover, dans lequel nous avons intégré l'évolution dynamique de la connectivité représentée par la matrice du modèle de mobilité de Musolesi et la contrainte sur l'overhead calculée par la formule de Viénot. Le système prend en compte d'une part la taille du paquet de maintenance du protocole de routage MANET et d'autre part les débits offerts par chaque nœud pour déduire les délais d'acheminement. L'algorithme suivant reprend l'essentiel de cette adaptation :

Algorithme 5: modification de l'algorithme principal de Groover

```
for each Time_Sim do  
  ...  
  ApplyCommunity(Adj,Aff)  
  for each Adjij do  
    ComputeDelay(Adjij,Overhead)  
    ComputeDegree(Adjij)  
    ComputeQuality(Adjij)  
  end for  
  ...  
end for
```

4.7.2. Application du modèle MADP2PStream

Dans cette expérimentation, nous utilisons une configuration de $N = 100$ nœuds. La distribution de la bande passante en entrée est composée de trois catégories: des nœuds lents, des moyens et des rapides avec un débit symétrique respectivement égal à r_{str} , $2r_{str}$ et $5r_{str}$ avec des probabilités égales à 0.2, 0.4 et 0.4. Le taux de flux est divisé en R' bandes et la source en génère R . Les résultats sont obtenus pour $R = 4$ et $R' = 3$.

Le temps de l'observation du temps, égale à T_{str} . Nous considérons un schéma d'arrivée des nœuds, avec un nombre initial égal à $0.5N$, respectivement, les autres nœuds arrivent dans les $0.2 T_{str}$. La moyenne des temps de séjour est fixée à $2 T_{str}$. Chaque nœud peut avoir jusqu'à 30 voisins dans le graphe. Ce nombre reste variable selon la dynamique des nœuds.

Les flux vidéo est décomposé en plusieurs morceaux, de l'ordre de quelques paquets vidéo, et nous fixons la taille du bloc à S_b , de telle sorte que $S_b/r_{str} = 1$ unité (c'est-à-dire un bloc par unité temporelle). Un nœud devient éligible pour le téléchargement du contenu, après un délai égal à la durée du téléchargement d'un seul morceau. Ainsi, le retard peut être considéré comme le retard relatif à partir du nœud source. T_{str} est alors fixé à une longueur de 1000 $S_b/r_{str} = 1000$ unités.

Mis à part les propriétés degré et délai, nous considérons également la qualité du réseau. Au départ d'un parent, le nœud orphelin active l'un de ses parents en veille. Si ce père possède suffisamment de bande passante pour l'aider, il ne verra aucune interruption de service. Si aucun parent n'est en mesure d'aider, le nœud en question recherche un nouveau parent, avec une éventuelle interruption de service. Suivant ces règles nous mesurons alors la qualité de l'expérience par nœud, c'est-à-dire le pourcentage de nœuds qui ont réussi à trouver un parent de secours.

nombre d'itérations	400
nombre maximum de voisins	30
nombre min. de fils	1
nombre max. de fils	15
nombre de bandes	4
nombre de bandes nécessaires	3
nombre de nœuds	1000
nombre de nœuds actif au départ	500
période d'arrivée massive	0.2
durée séjour moyenne	5000
durée d'une vidéo	1000
taux de mobilité	0.5

Tableau 4 autres paramètres du modèle

4.7.3. Analyse du degré

En analysant l'évolution du degré de connectivité, nous vérifions si la subdivision en bandes améliore le processus de distribution ou pas. D'une part, un nombre plus important plus de bandes implique un taux par bande plus faible et donc la perte d'une seule bande doit avoir un impact faible sur la qualité d'expérience du nœud. D'autre part, chaque nœud doit maintenir plus de connexions actives ce qui augmente la probabilité que l'un de ces liens se rompe.

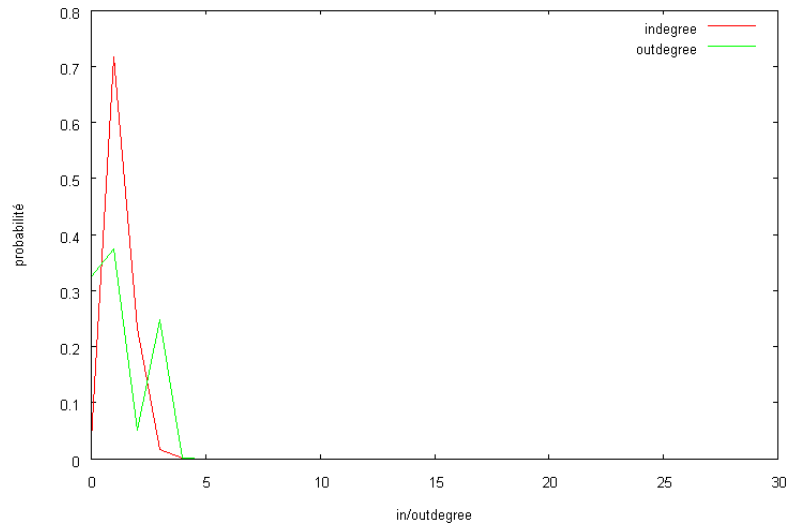


Figure 22: Distribution des degrés entrant et sortant.

Nous allons analyser la moyenne temporelle du degré en regardant les résultats de l'équation de taux (Figure 22 (b)) obtenus par l'équation (4.3).

Une valeur stable est atteinte au bout de quelques unités de temps: cela signifie que la structure, même en présence de mobilité est en mesure de maintenir un niveau élevé de qualité du streaming

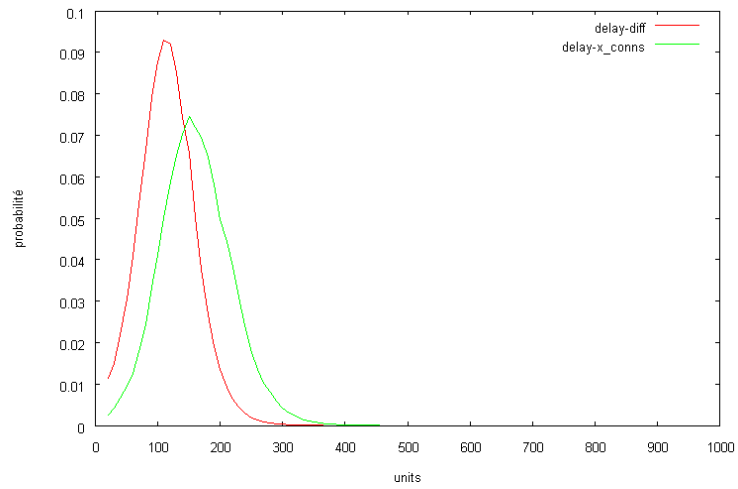
La Figure 22 indique la distribution du degré entrant à l'instant $t=T_{str}$, calculée avec l'équation. (4.3). Dans ce cas, nous avons le nombre initial de nœuds égal à $0.1N$ et le temps moyen de séjour T_{str} .

Quand R' tend vers R , la distribution atteint son maximum, cela signifie que tous les nœuds du réseau sont en mesure de recevoir la totalité des bandes, car le degré reste toujours supérieur ou égal à R' .

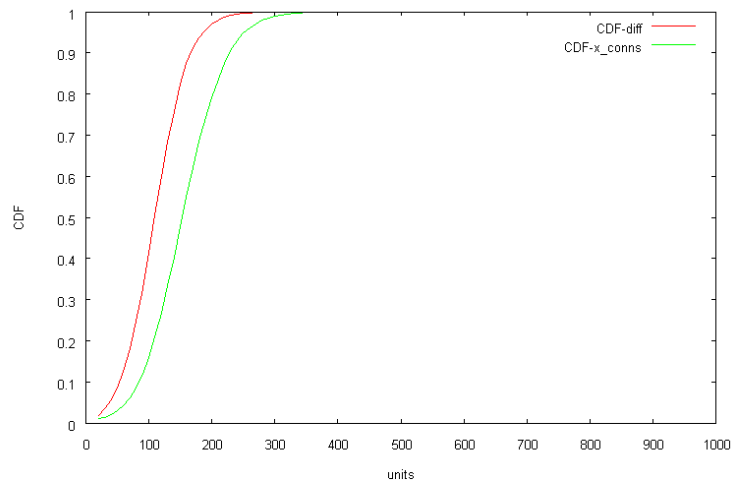
Avec $R' = 4$, chaque nœuds va avoir exactement 4 parents: cela signifie que, dans le cas où un père disparaît, la qualité perçue par le nœud, peut être temporairement affectée.

4.7.4. Analyse du délai

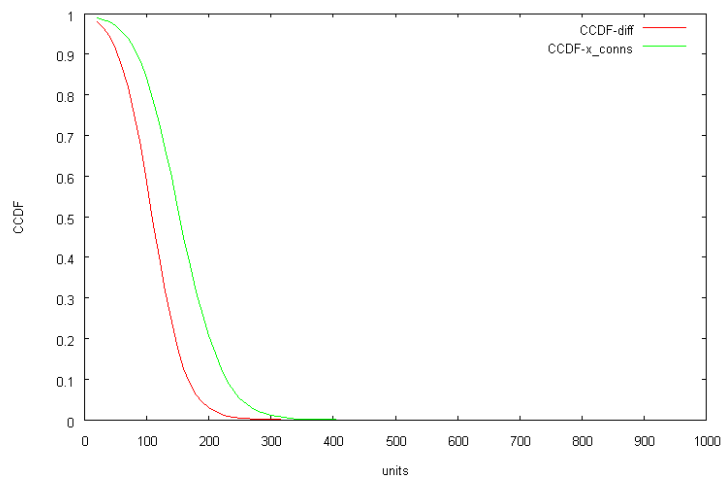
Le délai, exprimé en unités de temps, représente le nombre de sauts séparant un mobile de la source du stream. Grace aux Complementary Cumulative Distribution Function (CCDF, définie comme le complément à 1 de la Fonction Cumulative de Répartition) afin d'étudier la traine de cette distribution, nous considérons $R' = 4$.



(a) Distribution des probabilités



(b) CDF du délai à $t=T_{str}$

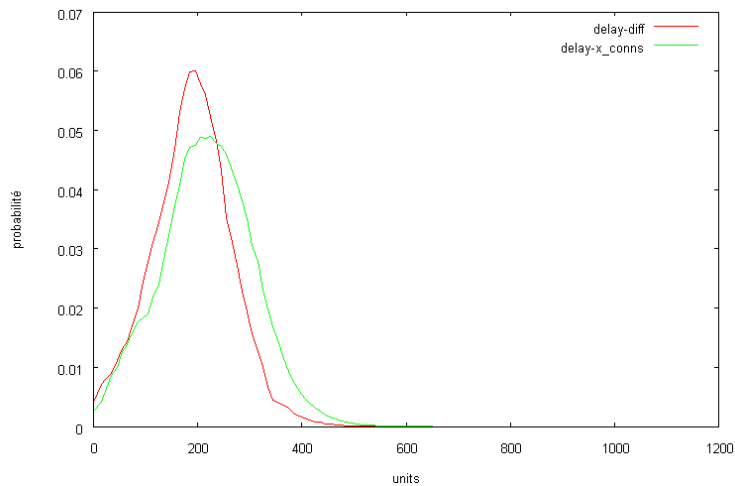


(c) *CCDF* à $t=T_{str}$

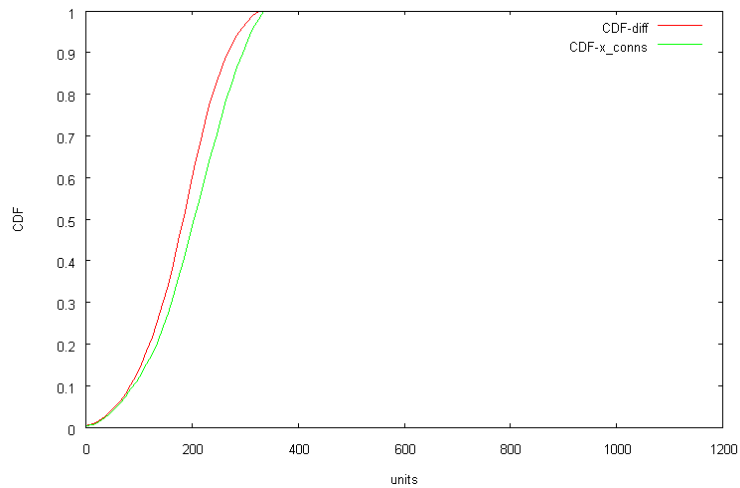
Figure 23: Evaluations du délai (cas fixe).

La Figure 23 présente les distributions des délais dans le cas fixe, nous observant un délai de diffusion moyen de 100 avec une probabilité de 90%. Ce délai avoisine les 200 avec une probabilité

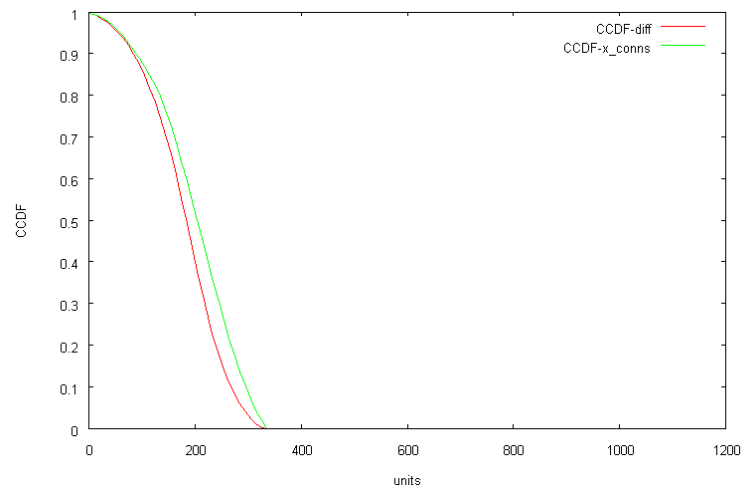
Les figures ci-dessous, illustrent l'impact de la mobilité sur le délai (formule 4.16). Celle-ci a un effet secondaire: étant donné que chaque nœud a besoin de tous ses liens actifs afin de lire correctement le flux, l'introduction de la mobilité donne des valeurs plus élevées de la probabilité des délais, puisque ce maximum dépend étroitement de la dynamique du réseau.



(a) Distribution des probabilités



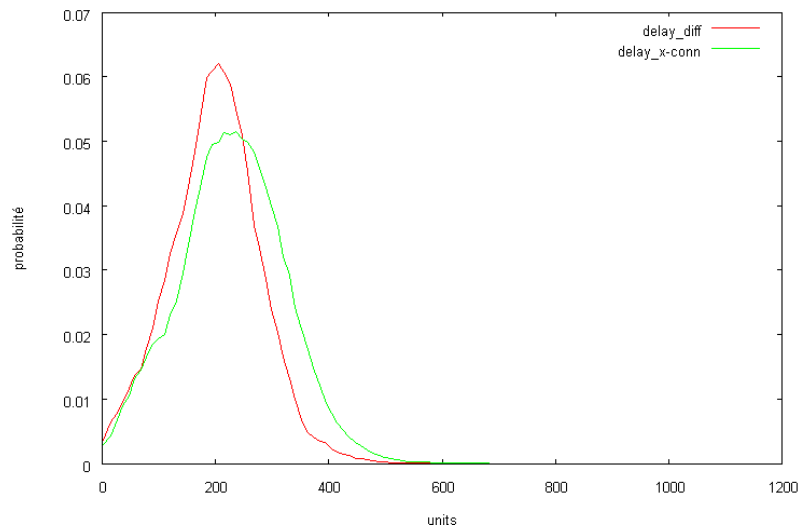
(b) CDF du délai à $t=T_{str}$



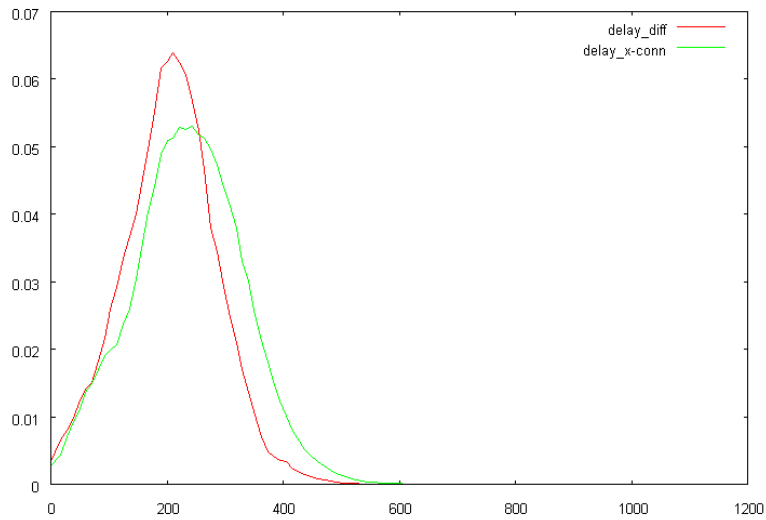
(c) *CCDF* à $t=T_{str}$

Figure 24: Evaluations du délai (cas mobile sans considération de l'overhead).

Avec la considération de l'overhead, selon les formules 4.17 et 4.18 nous recalculons la nouvelle distribution des probabilités de délai. Nous remarquons (dans la Figure 25) que la distribution moyenne du délai augmente comparé au résultat de la figure 24.



(a) cas proactif



(b) cas réactif

Figure 25: Distribution du délai (cas mobile avec considération de l'overhead)

4.7.5. Analyse de la qualité

L'étude de l'évolution du degré et du délai n'est pas en mesure de capter tous les aspects liés à la qualité des flux reçus par un nœud mobile i . Dans le Tableau 5 nous récapitulons des résultats complémentaires obtenus à partir de la solution de l'EM. La valeur de la dynamique est calculée selon la fréquence des arrivées:

Tableau 5. Statistiques sur les nœuds

paramètre	Cas fixe	Cas mobile
nombre moyen d'arrivées	1173.481000	1136.915000
nombre moyen de départ	1824.136000	550.127500
nombre moyen d'erreur	28.700000	431.305000
pourcent. moyen de départ	15.5452	48.4256
pourcent. moyen d'erreur	0.2446	37.9739
nombre moyen de nœuds actifs	910.345000	586.787500
arrivées ratées	0.000000	0.000000
moyenne d'orphelin	370.366000	646.840000
pourcent. moyen recouv. rapide	88.2595	00.0525
pourcent. moyen recouv. Après recherche	11.6802	58.7468
pourcent. moyen erreur	0.0603	41.2007

Sur toute la durée de vie du nœud, il y a une probabilité non nulle que l'ensemble des parents d'un nœud i le quittent sans qu'il soit en mesure d'en trouver d'autres. Cette situation provoque une erreur dans le nœud i et laisse le système dans un état instable. La probabilité d'avoir à tout instant t , $p(0, t)$ un degré sortant nul, en intégrant sur le temps t , nous pouvons

calculer la probabilité qu'un nœud quitte avec un message d'erreur. Ce résultat est indiqué dans la ligne pourcentage d'erreur du tableau 5 pour différentes cas de figure.

4.8. Discussion et conclusions

Dans ce chapitre, nous avons proposé MadP2PStrem : un modèle analytique, qui se veut le plus exhaustif possible, pour représenter et donc mieux appréhender le streaming peer to peer dans les réseaux ad hoc mobiles. Cette étude a consisté à regrouper les différents aspects intervenant dans cette modélisation, à savoir en premier, le modèle du streaming peer to peer pour réseau fixe que nous avons étendu par un modèle de mobilité basé sur les interaction sociales des nœuds et enfin la prise en compte du débit consommé par les messages de contrôle des protocoles de routage MANET dans deux cas le proactif et le réactif appliqués dans un scénario dédié à la mobilité piétonne dans le but de partager des expériences vidéo en communauté.

Quant il s'agit d'une mobilité restreinte dans un espace de simulation limité et dense en matière de nœuds, les protocoles de routage les plus adéquats restent les proactifs. Toutefois, dans des cas plus clairsemé, la solution réactive reste la plus appropriée. Les résultats de l'étude de ces modèles nous incite à aller plus loin, pour proposer une preuve de concept, par la simulation et le prototypage. Ceci fait l'objet du chapitre suivant.

Il reste néanmoins plusieurs améliorations à faire en ce qui concerne MADP2PStream, car dans un scénario de mobilité importante, les délais de réception de la vidéo deviennent très importants car l'overhead consomme plus de débits sortant des parents. Mais dans un cas de mobilité piétonne nous pensons que cette proposition est facilement implémentable.

Chapitre 5. : *MadTorStream - un protocole pour le streaming p2p mobile*

5.1. Introduction

La production de contenu vidéo engendre, de nos jours, un grand volume de trafic sur Internet. Les réseaux sociaux et professionnels émergeant (ex. Facebook, LinkedIn...) et les portails vidéo agissent comme un véritable accélérateur de ce phénomène. En plus des applications IPTV et de la vidéo à la demande VoD, le partage de vidéos personnelles est également dans la course.

Après le partage de fichiers et la téléphonie sur IP, les applications de streaming peer-to-peer suscite un grand intérêt, autant chez les académiques que chez les industriels. L'objectif principal de cette technologie est de faire coopérer des nœuds en vue de répartir la charge entre eux leur permettant ainsi, d'envoyer et de recevoir des parties du contenu partagé. Un grand nombre d'applications de streaming peer-to-peer sont actuellement utilisées sur Internet, tant pour la vidéo à la demande et en temps réel tels que des services IPTV. On peut citer Joost, Pplive, SopCast, UUSec.

Les utilisateurs sont aussi fascinés par les nouvelles technologies de télécommunication. En particulier, les équipements nomades, devenues indispensable dans leur quotidien. Un ensemble d'interfaces de communication (WiFi, Bluetooth, GPS...) est désormais intégré dans des appareils mobiles comme les téléphones, PDA, lecteurs multimédia. Le réseau ad hoc mobile (MANET) est une application de ces technologies sans fil qui ne nécessitent pas d'infrastructure.

Partant du fait que les applications multimédia sont de plus en plus présentes dans la vie nomade, nous pouvons facilement imaginer un scénario où les utilisateurs mobiles construisent leur propre réseau, spontanément, sans recourir à l'infrastructure, dans le but d'échanger des vidéos entre eux. Juste en s'appuyant sur les interfaces sans-fil incorporées dans leurs mobiles. Comme ces appareils sont limitées en termes de mémoire, ils ne peuvent pas stocker l'ensemble des flux vidéo, et seraient, sans doute, amenés à ne garder qu'une partie de celui-ci. Autrement, les performances de leurs mémoires diminuent.

Considéré comme trop semblable aux topologies peer-to-peer, par un comportement similaire: les arrivées et départs de nœuds, les réseaux ad hoc mobiles sont construits dans la plupart du temps par des appareils nomades avec un protocole spécifique (proactif, réactif ou hybride) pour router les données entre eux.

Indépendamment du protocole de routage, les nœuds mobiles ont besoin d'une certaine quantité de mémoire de traitement comme un buffer pour faire face aux problèmes de rupture du flux vidéo en raison de leur motion.

5.1.1. Persistance et réplication pour palier à la mobilité

Le développement du trafic peer to peer sur Internet à commencé par l'échange de fichiers musicaux. Le streaming vidéo prend à lui aussi un part plus importante. La mobilité est aussi de plus en plus présente dans le quotidien des utilisateurs souciant de partager des contenus souvent issu de leurs créations (photo et vidéo prise spontanément. Le peer to peer représente une solution pertinente au problème du coût du déploiement de serveur dédié, offrant une répartition de la charge sur les nœuds participant dans ce partage en utilisant leurs ressources respectives.

Quand on porte le streaming peer to peer sur les réseaux ad hoc mobiles, nous devons faire face à quelques problèmes relatifs à la puissance de calcul limitée (i.e décodage), la limite de la mémoire ainsi que la capacité de stockage. Dans ce chapitre nous allons proposer une technique de clustering pour les MANETs permettant le streaming peer to peer. Cette proposition vise à alléger de la discontinuité (disruption) causé au churn additionnel engendré par la mobilité des nœuds.

5.1.2. Mobilité augmente la diversité

Dans notre proposition, comparée à BitTorrent, il existe une diversité des blocs entre les localités (voisinages) visitées par le nœud mobile. En d'autres termes, les blocs disponibles à un moment sont différents d'un nœud à l'autre dans lesquels ce nœud séjourne. Nous nous appuyons sur cette hypothèse pour modifier le protocole BitTorrent (surtout sa stratégie de sélection de bloc voir paragraphe précédent)

5.2. Travail effectué

Notre objectif, dans le présent travail, est de proposer MadTorStream : une combinaison de p2p streaming, basée sur BitTorrent avec MANETs. Afin de permettre le partage de vidéos dans une communauté mobile. Nous prenons en compte les contraintes liées aux canaux sans fil, les protocoles de routage sous-jacent ainsi que la mobilité protocoles réseau.

Ce travail est proposé dans le cadre du projet européen ITEA ExpeShare. Un projet collaboratif européen de recherche dans lequel participe notre laboratoire. En effet ExpeShare vise à réaliser un

partage d'expériences sur les communautés mobiles via des technologies de partage de contenu adapté sur des réseaux sans fils, avec ou sans infrastructure. Dans notre étude nous traiterions du cas sans infrastructure.

Le reste de ce chapitre est organisé comme suit: dans la section suivante 5.2.1, nous positionnons notre approche par rapport à d'autres tentatives. La section 5.3.2 présente les points d'action pour permettre le mobile p2p streaming. Dans la section 5.3, nous donnons une vue d'ensemble de MadTorStream. L'évaluation de la performance vient en section 5.4. Et enfin, la section 5.6 conclut la contribution.

5.2.1. Travaux relatifs

Le streaming peer-to-peer et les MANETs sont des domaines de recherche bien étudiés. A notre connaissance, aucune étude ne propose des mécanismes de streaming p2p pour les réseaux mobiles ad hoc. De nombreuses applications p2p ont été proposées pour MANETs, telles le partage de fichiers [129]-[135], la recherche et l'indexation de contenu[133][134], la téléphonie p2p [135]. Récemment, une proposition de partage de fichiers pour MANETs, basée sur BitTorrent, a été présentée dans [129], tandis que les études [130]-[132] n'ont pas dépassé l'étape de recherche de contenu.

Les applications peer-to-peer de streaming vidéo sur Internet ont connu une croissance exponentielle au cours des deux dernières années. Les utilisateurs partagent, et même produisent, de plus en plus de contenu sur le Web à l'aide de plates-formes novatrices comme Joost, l'un des nombreux clients p2p streaming sur Internet.

Après recherche, le partage de fichiers et la VoIP de recherche se concentre désormais sur les applications de vidéo streaming peer-to-peer qui sont de plus en plus étudié, des études de mesure comme [112][116][136] tentent d'apprécier, le plus profondément possible, les différentes caractéristiques de ces systèmes. Plusieurs modèles analytiques sont proposés pour capturer les paramètres essentiels des applications de distribution de contenu vidéo sur Internet. Parmi ces travaux, une série de proposition d'adaptation de BitTorrent pour le streaming a été donnée dans [138][139]. Dans lesquelles les auteurs ont proposé de changer la stratégie de sélection de blocs. Ces mêmes études montrent qu'un voisinage de 15-20 pairs est suffisant, pour atteindre un streaming de qualité. Il ya une dualité entre différence la capacité du nœud source, nécessaire pour garantir un certain débit à un nombre d'utilisateurs, et le nombre de pièces disponibles au partage.

5.2.2. Manet modèles de routage et de la mobilité

Depuis l'apparition du MANETs où de nombreux modèles de mobilité proposées par la communauté. Entre autres, Le Boudec et al. ont proposé le Random Walk [41] ils ont aussi montré sa performance, comme évoqué dans le paragraphe 4.2 notre intérêt est plus dans un modèle de mobilité sociale qui est plus représentatif de notre scénario de communauté mobile.

Après une longue recherche dans la littérature concernant les modèles de mobilité MANETs, en comparant les avantages et les inconvénients de chacun. Dans ce travail nous utilisons le modèle de mobilité fondée sur le réseau social de la théorie, proposée par Musolesi et al. dans [121].

En utilisant des propriétés de réseaux sociaux, ce modèle se base sur l'observation suivante: dans un MANET, le mouvement du dispositif portable est basé sur les décisions de l'utilisateur et son comportement social. Pour considérer ce type de comportement, le modèle prend en compte le réseau social qui relie ces personnes. En d'autres termes, les mouvements des groupes ainsi que des nœuds sont régis par leurs relations sociales. Le fonctionnement de ce model est détaillé dans la section 4.4.

Ce modèle de mobilité permet à des collections d'hôtes de se regrouper selon les relations sociales entre leurs possesseurs. Ce regroupement est ensuite reflété sur un espace topographique, avec des mouvements influencés par la force des liens sociaux, et pouvant aussi changer dans le temps. Cela correspond bien à notre scénario étudié, lié à la communication de groupe dans une zone d'exposition, où des utilisateurs partagent leur expérience en utilisant le streaming peer-to-peer. Ce travail est proposé dans le cadre du projet européen ITEA ExpeShare.

5.2.3. Mobile peer-to-peer streaming

De la littérature il y a peu de travaux qui traitent le streaming sur MANETs, et encore moins, le streaming peer-to-peer dans ces réseaux sans infrastructure. Dans cette étude, nous avons pour but d'aller d'un pas en avant pour comprendre cette problématique, à une époque, où la vie nomade, combinée aux réseaux sociaux et aux applications multimédia sont à la page.

5.2.4. Difference entre file sharing et streaming p2p

Le défi du streaming P2P est de fournir un débit soutenu à tous les participants du réseau. Alors que dans le partage de fichiers P2P le contenu peut être distribué en best effort, dans le streaming vidéo, une bande passante insuffisante conduit à une mauvaise qualité de service, telles que le décrochage ou le gel, ce qui est très ennuyeux pour les utilisateurs mobiles.

Contrairement aux protocoles de partage de fichiers, et dans un souci de supporter une diffusion basée sur les topologies maillées (Mesh Based), le nœud cible lit la vidéo avec un certain décalage comparé à la source. Les pièces de la vidéo sont reçues et reconstruites dans un buffer d'affichage (display buffer). Chacun des nœuds maintient une fenêtre glissante qui reflète ce buffer en notifiant, à ses voisins, chaque pièce déjà reçue et marquant celle qui lui manque. Cette fenêtre se défile avec la vitesse de lecture de la vidéo (BitRate). Le début du buffer pointe sur le bloc actuellement en lecture. Les blocs manquant à leur date d'affichage sont ignorés ce qui cause une dégradation dans la qualité perçue de la vidéo.

Toutes les études citées auparavant se sont concentrées sur la file sharing utilisant la présence des blocs en tant que métrique principale pour mesurer la performance. Cependant, quand il s'agit de diffusion vidéo, l'étude de ce paramètre devient alors insuffisante, car les blocs doivent, en plus, être présentés à temps. Autrement, l'utilisateur percevra des méfais sur la qualité de la vidéo (discontinuité, perte de qualité, problèmes...).

5.3. Vue d'ensemble de MadTorStream

Dans ce paragraphe, nous présentons les différentes composantes de l'architecture proposée. Nous commençons par le réseau sous-jacent dans une approche bottom-up jusqu'à la couche application. La cinétique du système proposé se décrit comme suit: Imaginons le cas d'une conférence en se déroulant dans un espace donné. Où des participants, de différents organisations et centres d'intérêt, vont se retrouver pour une certaine période. Ces utilisateurs sont équipés de téléphones mobiles avec WiFi intégré. Quand un participant entre dans la zone d'exposition, il rejoint la communauté (un réseau ad hoc spontané créé à cet effet). Il aura un profil, une présence ainsi qu'une liste d'objets (vidéo ou autre) qu'il veut mettre en partage avec d'autres participants.

Notre intérêt est de réaliser l'échange de vidéo, essentiellement l'adaptation d'un algorithme classique de partage de fichiers (qui est la stratégie rarest first de BitTorrent). Ceci consiste à choisir non pas le bloc le plus rare mais celui avec l'estampille d'affichage la plus imminente. La phase de recherche du contenu vidéo ne sera pas traitée dans cette thèse. Celle-ci, peut se faire grâce à des mécanismes distribués (tel que des DHT ou des Trackers Distribués). Nous supposons donc, que la vidéo est déjà localisée, nous nous concentrons seulement sur les aspects streaming.

Le nœud demandeur obtient une liste de voisins partageant le même flux vidéo. Parmi lesquels, un système de sélection de pairs permettra d'élire les pairs (demandant des blocs de la vidéo) à qui il va donner des bouts de cette vidéo. Nous gardons la même stratégie de sélection de pairs que celle employée par le protocole BitTorrent [12] et modifiée dans BitHoc, plus de détail quant à cet algorithme sont disponibles dans [129].

Notre principale modification, apportée au protocole BitHoc, porte sur la politique de sélection de blocs. Nous avons pris en compte les recommandations données dans [138] et [139] pour changer l'ordre dans lequel les morceaux de la vidéo sont choisis en utilisant la politique du Closest Deadline First. La mobilité des pairs est régie par le modèle de mobilité. Les paramètres comme la vitesse, temps de pause sont donnés dans la section suivante.

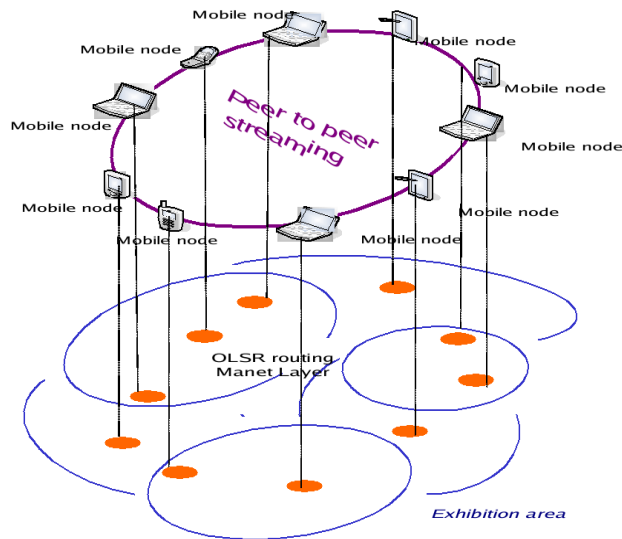


Figure 26: *Vue d'ensemble de MadTorStream.*

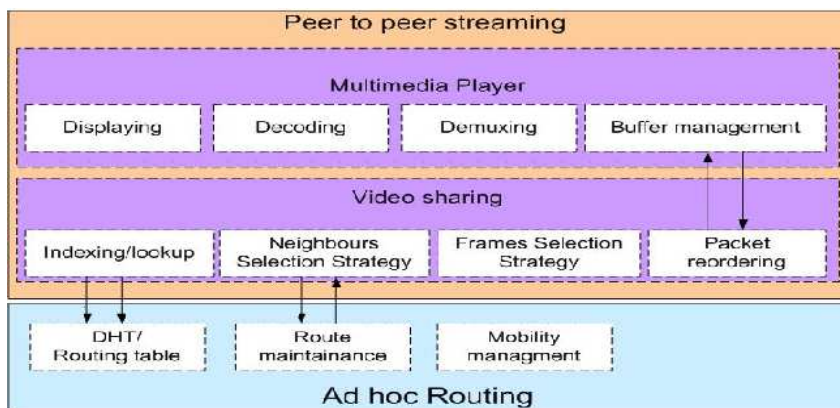


Figure 27: *Vue en couches de MadTorStream.*

5.3.1. Manet modèles de routage et de la mobilité

Le protocole de routage sera le même utilisé dans BitHoc, à savoir DSDV [140] (Dynamic Destination-Sequenced Distance-Vector Routing) est utilisé dans ce cas, nous avons rajouté une extension par rapport à la mobilité du nœud. Nous sommes passés du cas fixe au mobile, sur la base du modèle de mobilité définie dans les paragraphes suivants.

Selon DSDV, chaque nœud mobile maintient une table de routage, qui est exploitée par le BitHoc tracker BitTorrent. Les voisins sont gérés en fonction de leur proximité du pair (en nombre de sauts).

5.3.2. Au niveau peer-to-peer (couche de partage vidéo)

Sur la couche p2p, une adaptation du protocole de partage de fichiers sur MANETs, BitHoc, tenant compte des changements proposés.

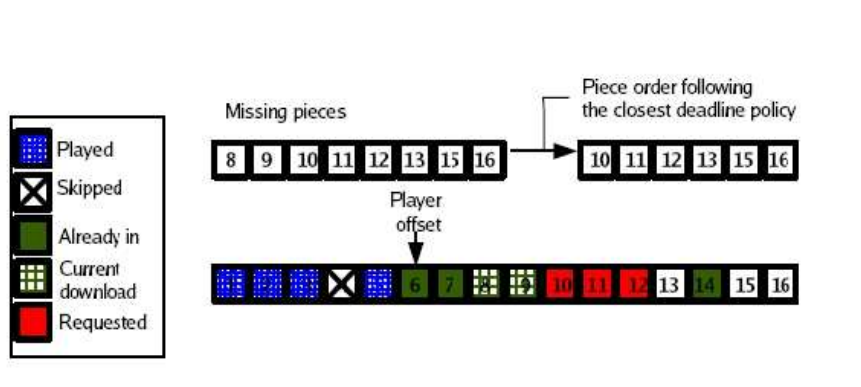


Figure 28: La stratégie Closest Deadline First.

A ses voisins, un nœud mobile va demander les morceaux de vidéo avec la date la plus proche (pièce n ° 10 dans le cas de la Figure 28). Comme dans BitTorrent [12], chaque pair mobile satisfait les demandes de ses voisins dans l'ordre de mérite (il donne un bloc au voisin qui m'a fourni avec le meilleur upload dans le passé).

5.3.3. Application du Modèle de mobilité

Comme nous l'avons vu dans la section 4.4, le community based mobility modèle consiste à appliquer une matrice d'affinité sociale sur la base des relations entre les participants d'un meeting. Dans de tels événements des personnes se déplacent généralement en groupes. Ces groupes sont formés selon les affinités sociales représentées dans une matrice d'affinité composée par les probabilités que deux utilisateurs mobiles se parlent les uns aux autres quand ils sont l'un dans la portée de l'autre. Et lorsque cette probabilité supérieure à un seuil donné. Suivant ce modèle, nous avons généré les traces correspondant à notre scénario de meeting. Dans ce qui suit, nous détaillons les paramètres considérés dans nos simulations.

5.4. Simulations

Les simulations ont été menées en utilisant NS-2¹⁵ (Network-Simulator2), avec une adaptation du protocole streaming p2p basé sur BitHoc pour NS-2 proposée dans [129]. Nous avons modifié la stratégie de sélection des pièces pour supporter le streaming. Nous lançons cette simulation avec les paramètres résumés dans le tableau II. Dans ce scénario, nous ne considérons pas les départs et les arrivées, seule la mobilité nous intéresse. La dynamique causée par les départ/arrivée sera abordée à l'avenir.

Afin de considérer la mobilité des nœuds, nous avons généré un les traces du mouvement des pairs mobiles en utilisant l'outil proposé dans [55]. Il s'agit d'un générateur de mobilité écrit en C++ et disponible en ligne¹⁶. Cette trace est ensuite injectée dans le simulateur MadTorStream, changeant le comportement des nœuds de fixe à mobile avec un certain nombre de paramètres (voir Tableau 6).

Paramètre		Définition	Valeur
MANET	S	Vitesse	0 - 2 m/s
	MAC	Modèle de la couche MAC	802.11b
	area	Surface de simulation	600x600 m ²
	node_range	Portée du nœud transmission	70 m
P2P	data_rate	Débit du nœud up/dl	1Mbps/1Mbps
	max_num	Nombre de nœuds mobiles	30
	piece_num	Nombre de blocs	100
	piece_size	Taille d'un bloc	100 kb
	num_neighbors	Nombre voisins par nœud	15-20
	upload_history	Période de sélection des voisins	60"

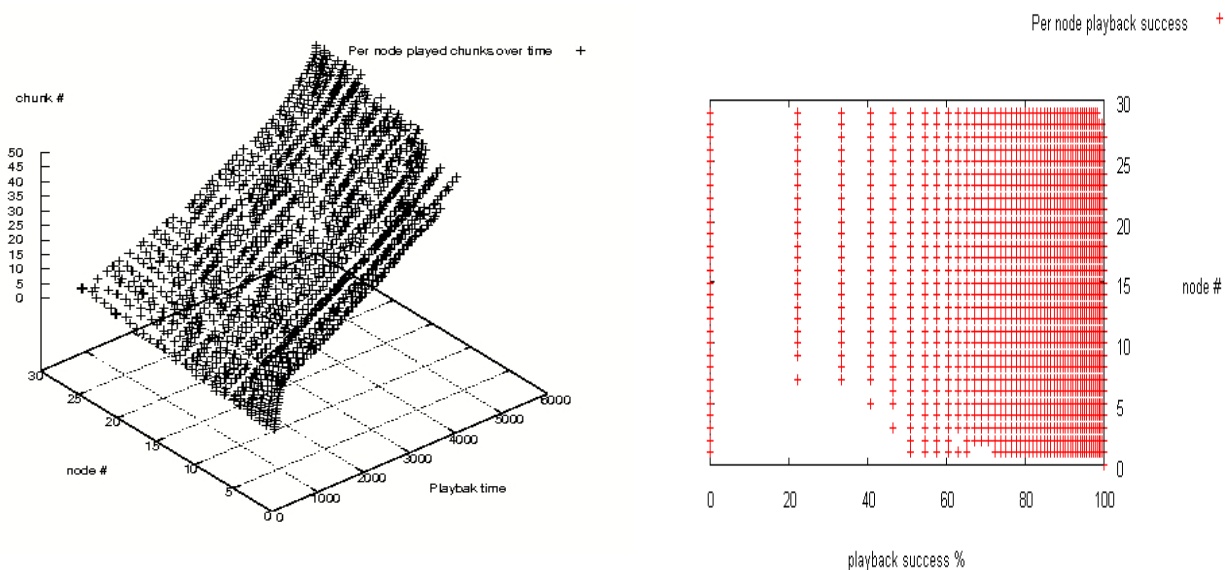
Tableau 6 – Paramètres de simulation

5.5. Discussion des résultats

Afin d'apprécier l'efficacité du système proposé ci-dessus, nous mesurons le taux de succès de la lecture. La distribution dans le temps de la présence des blocs sur les nœuds pour voir l'effet de la modification de la stratégie de sélection des pièces (Closest Deadline First) pour le streaming sur la qualité du perçues par le nœud.

¹⁵ "NS: The Network Simulator", <http://www.isi.edu/nsnam/ns/>.

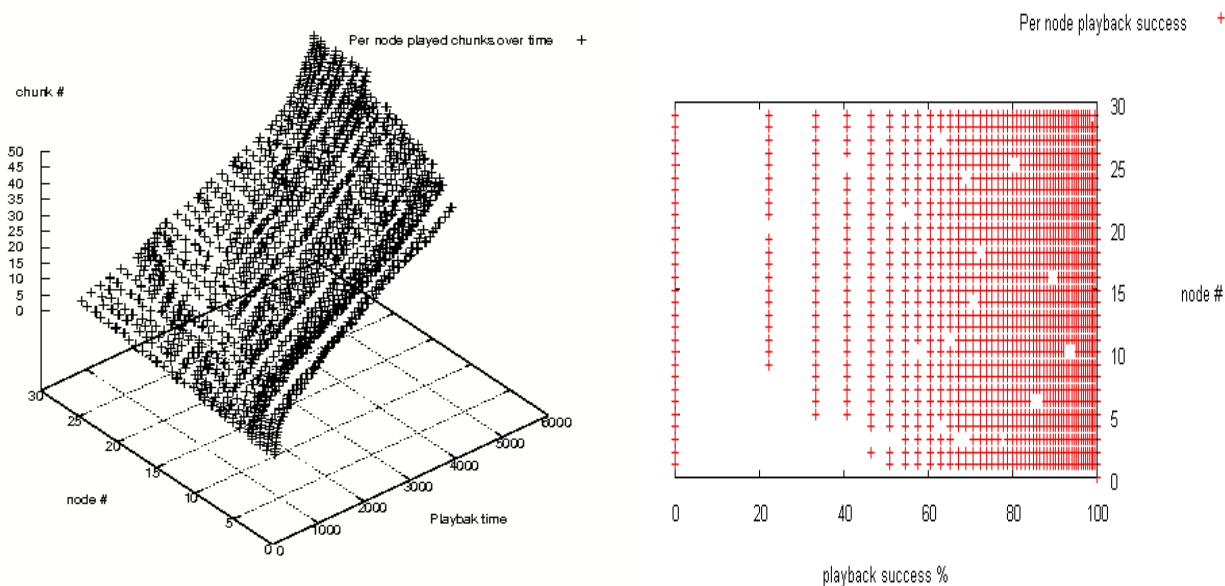
¹⁶ M. Musolesi and C. Mascolo. A Social Network Founded Mobility Models for Ad Hoc Network Research, <http://www.cs.ucl.ac.uk/research/mobile/mobilitymodels/>



(a) distribution temporelle des blocs sur les nœuds

(b) Taux de succès de lecture par nœud

Figure 29: Résultats avec Closest Deadline First.



(a) distribution temporelle des blocs sur les nœuds

(b) Taux de succès de lecture par nœud

Figure 30: Résultats avec Rarest First.

Les résultats de cette distribution sont représentés dans les Figure 29 pour la politique Closest Deadline First et Figure 30 pour l'originale de BitTorrent, Rarest First. Nous observons que le temps entre le premier et le dernier morceau est légèrement réduit avec notre stratégie. Cependant, nous observons une *séquentialité* dans les deux stratégies Rarest First et Closest Deadline First. Cela est dû au fait que tous les pairs sont arrivés en même temps et ont demandé les mêmes blocs dans le même ordre, ce qui est problématique par rapport au rarest first.

Le principal intérêt de faire cette étude est de voir si le streaming est de qualité acceptable pour chacun des pairs. Pour ce faire, nous utilisons une mesure déjà introduite par WF Poon et al. dans [140]: la probabilité de succès de la lecture, qui est définie comme la proportion des blocs demandés, que le nœud a réussi à afficher, de l'ensemble de la vidéo. Dans les Figure 29 (b) et Figure 30 (b), nous montrons la distribution, par nœud, du taux de succès de la lecture. Ce gain est très prometteur pour les nœuds mobiles qui se déplacent à une vitesse entre 0 et 2 m/s. Cela correspond bien à notre scénario de mobilité piétonne.

5.6. Conclusions

Cette partie de notre contribution, s'est axée sur la question du streaming sur MANETs en utilisant une adaptation de BitHoc, un BitTorrent pour Ad Hoc, que nous avons évaluée dans des simulations NS-2, tenant compte des contraintes liées aux canaux sans fil, les protocoles de routage MANET sachant que l'application streaming p2p est totalement indépendante du réseau sous-jacent. Nous avons la conviction que la proposition d'un tel système de streaming vidéo sur MANETs est un domaine de recherche important à explorer.

Nous avons présenté une architecture de p2p streaming ad hoc pour les communautés mobiles. Sur la base d'un modèle de mobilité sociale, nous avons montré que, par rapport au partage de fichiers, l'amélioration qui doit être apportée lors de la conception d'applications de streaming sur MANETs, à la BitTorrent, est presque la même que dans les scénarios fixes: La stratégie de sélection de blocs vidéo doit tenir compte de leurs dates limites de présentation, sans quoi, la performance du système s'écroule. Nous avons mis en évidence, dans cette étude, l'importance de la « *séquentialité* » dans le renforcement de la diversité des blocs présents dans la fenêtre glissante, offrant ainsi, plus de persistance pour un streaming de qualité.

Les résultats de cette étude montrent la faisabilité du streaming mobile basée sur BitTorrent utilisant un protocole de routage proactif. Suivant un modèle de mobilité fondé sur le comportement d'une communauté sociale, nous avons pu prouver par simulation que le streaming p2p peut être envisagé sur les MANETs. Les investigations restent ouvertes quant au choix du protocole de routage et des modèles de mobilité suivant le scénario dans lequel de telle architecture seront déployé. Pour cela une prochaine étape consiste à comparé deux scénarios de mobilité Le premier, en mobilité réduite avec un protocole proactif OLSR, le second en haute mobilité avec le protocole réactif AODV. Pour voir la concordance des résultats avec ceux obtenu dans la modélisation avec MADP2PStream.

Chapitre 6. Conclusions et perspectives

Nous pensons que la combinaison du partage de vidéo et des communautés sociales mobile qui semble embrasser un formidable développement décollera bientôt. Offrant ainsi, aux utilisateurs nomades, une large autonomie et une grande liberté de partage de leurs contenus, sans pour autant avoir à recourir à l'infrastructure souvent très coûteuse (tels que gprs/3G). En effet cette liberté devient possible grâce à des réseaux spontanés formés par des appareils mobiles.

Le principal objectif poursuivi visé dans cette contribution est d'éclairer les zones d'ombres couvrant des domaines, divers et complémentaires, impliqués dans l'aboutissement des futures applications multimédia mobiles. En effet, la proposition de nouveaux protocoles de streaming multimédia sur réseaux MANET est le fruit de la convergence de mécanismes sur les couches réseau et applicatif. Des compromis sont alors proposés pour tirer profit des avantages respectifs des deux niveaux, en palliant à des problèmes tels que la dynamique des nœuds mobiles.

Dans cette conclusion, sont rappelées les principales fonctionnalités des systèmes de streaming en mobilité qui sont proposés dans ce travail de thèse. Aussi y sont synthétisés les apports de SM²Stream, MADP2PStream et MadTorStream, en rapport aux contraintes posées par la mobilité, au streaming. Enfin, quelques perspectives pertinentes sont avancées pour l'amélioration de nos propositions, présentées, elles expriment manifestement dans le domaine en général, des axes de recherche potentiels.

6.1. Synthèse du travail réalisé

Durant ces trois dernières années, notre recherche a permis d'aborder toutes les notions relatives au streaming multimédia sur les réseaux sans fil. Nous avons alors tenté de comprendre les différents aspects liés à cette problématique, notamment dans le cas de mobilité. C'est à cette étape que nous nous sommes intéressés aux processus et mécanismes de prédiction du handover dans le but de réaliser un service de streaming sans coupure dans la mobilité. Nous avons pu relever que, dans la pratique, la majorité des contraintes que nous pensions avoir bien étudiées, par des formules mathématiques ou par des simulations, s'avèrent incomplètes eu égard à l'hostilité des conditions réelles dans lesquels nos mesures ont été effectuées.

Nous avons pu, par la suite étudier le streaming dans des réseaux ad hoc mobiles, l’alliance de ces deux domaines existait déjà, mais pour des applications moins exigeantes, en terme de ponctualité, que les streaming multimédia. Pour relever ce déficit, nous avons commencé par l’analyse de plusieurs modèles relatifs aux différentes facettes de notre proposition. Une fois les choix faits, nous avons proposé un modèle en ramenant tous ces aspects à la même échelle : celle de la qualité du streaming mobile.

Suite à cela, nous avons proposé une architecture

6.2. Résumé des contributions

6.2.1. Streaming sans coupure en mobilité sur WLAN

« SM²Stream »

Dans notre première contribution, nous avons proposé SM²Stream un système de mise en cache adaptative, guidée par la prédiction du handover. Lors de la mobilité d’un terminal, ce système vise à assurer un service de streaming vidéo sans coupures, *seamless*. Cette proposition permet de prédire le temps restant avant handover, grâce à un filtre mathématique capable de prédire à quel instant les valeurs de force du signal tombent en dessous d’un certain seuil. Il ordonne alors au terminal de prévoir une déconnexion dans les secondes qui suivent. Cette prévision consiste à télécharger, en ouvrant des connexions multiples, la quantité de vidéo nécessaire pour couvrir la période de déconnexion, lors du passage entre les deux cellules sans fil.

Après avoir validé cette proposition par simulation, nous avons implémenté SM²Stream dans un lecteur multimédia populaire. Cette réalisation a consisté en la modification du schéma classique de mise en cache de la vidéo, en ouvrant plusieurs connexions en parallèle, avant le handover, et de faire les calculs nécessaires, via un module de prédiction basé sur le filtre Grey, pour ne pas avoir d’interruption de la vidéo au moment opportun, c’est-à-dire lors du handover entre les cellules sans fils. Cette étude nous a principalement permis de réaliser à quel point les bancs d’essai réels restent le meilleur moyen d’évaluation de ce type de propositions. Néanmoins, les résultats obtenus se sont révélés très satisfaisants.

6.2.2. Modèle de streaming p2p sur MANET « MADP2PStream »

Les protocoles peer to peer ont révolutionné le partage de l’information sur internet. Ils fonctionnent sur des principes de réciprocité et de récompense, incitant les nœuds à donner le meilleur de leurs ressources. Le streaming vidéo a adopté ce paradigme distribué pour la dissémination des paquets vidéo entre les utilisateurs.

Les MANETs sont des réseaux spontanés auto-organisés, dans lesquels des nœuds mobiles communiquent via des liens sans fils, sans recourir au soutien de l’infrastructure. Ces mobiles sont

caractérisés par une portée limitée et échangent donc sur des chemins multi sauts, chacun assurant le rôle de relai pour ses voisins. La mobilité est l'inconvénient majeur des ces réseaux. En effet, le mouvement des nœuds génère des pertes de liens menant à une dégradation du service partagé.

Nous avons cherché à proposer un modèle mathématique afin d'étudier la réaction des protocoles de streaming vidéo en P2P à la mobilité des nœuds d'un réseau ad hoc mobile. MADP2PStream s'inspire d'un modèle de référence pour le streaming P2P, dans lequel nous avons introduit deux notions primordiales des réseaux MANET : d'une part la mobilité, représentée par un modèle synthétique mais assez réaliste, basé sur les réseaux sociaux et d'une autre part, un modèle, représentant la surconsommation supplémentaire des débits provoqués par cette « *nomadisme* » des nœuds.

Les résultats exposés et commentés dans le Chapitre 4 , sont très prometteurs ; ce qui de fait a renforcé notre conviction, quant à la faisabilité des futures applications vidéo, sur les dispositifs nomades. Par le biais d'un réseau spontané coopératif, basé sur des paradigmes de MANET et des principes de streaming P2P, les nœuds mobiles échangeront des contenus vidéo sans avoir recours à l'infrastructure.

6.2.3. Streaming bittorrent sur MANET « MadTorStream »

Le dernier volet de cette thèse, consacré à la validation de simulation de MadTorStream , porte principalement sur une proposition adaptant un protocole de partage de fichiers sur MANET, à la BitTorrent, pour le streaming vidéo. Nous avons repris des indications parues dans plusieurs travaux de recherche, relatives aux étapes de cette transformation de BitTorrent, pour la diffusion vidéo. Les résultats ont été validés sur un des plus grands simulateurs de réseaux NS-2 avec un scénario dédié aux communautés mobiles et en appliquant un modèle de mobilité basé sur les interactions sociales des utilisateurs. Nous avons également pris en considération un protocole de routage proactif, avec un modèle réaliste de la couche physique, implémenté dans le simulateur.

La simulation nous a essentiellement permis de comparer et de décider de la stratégie de sélection de bloc à adopter afin d'en avoir le moins possible, qui ratent leur heure d'affichage, tout en comptant bien sur la diversité supplémentaire apportée par le mouvement des nœuds.

6.3. Perspectives

L'analyse rétrospective montre clairement que, dans ce domaine du streaming multimédia en mobilité et sans infrastructure, de nombreuses difficultés devront être surmontées ; de ce fait, de multiples voies sont à explorer tant sur le plan applicatif que sur le réseau. Dans ce qui suit nous donnons une liste, non exhaustive, de points qui restent à élucider.

Dans le futur, nous proposerons des améliorations du modèle MADP2PStream en introduisant des techniques de clustering sur le MANET ainsi que des mécanismes de réplication de paquets vidéo, visant à réduire la consommation en bande passante des nœuds. Actuellement, la mise en œuvre d'une application de streaming BitTorrent sur MANET, est en cours dans notre laboratoire, sur la base de l'architecture définie dans le Chapitre 5. Ce qui nous permettra d'approfondir l'étude des contraintes liées aux propriétés du canal physique ainsi que leur impact sur la qualité de l'expérience acquise par l'utilisateur.

6.3.1. Adaptation de paquets vidéo pour handover

Dans un souci d'optimisation de l'exécution de streaming durant les handovers, il serait intéressant de voir comment pourrions-nous jouer sur la vitesse d'encodage de la vidéo pour réduire la taille des paquets afin d'en « *bufferiser* » plus, en moins de temps. (Ce qui permet de se passer de l'usage des mécanismes d'adaptation classique, qui changent le format d'encodage, pour adapter le flux dans sa totalité, aux conditions du réseau actuel.) En d'autres termes, au lieu d'utiliser les mécanismes d'adaptation classique qui changent le format d'encodage pour adapter le flux, dans sa totalité, aux conditions du réseau actuel. Ce que nous préconisons dans l'avenir, ça serait, non pas, de faire une adaptation de toute la vidéo, mais de l'établir au fur et à mesure que les conditions du réseau s'améliorent. Ce mécanisme nécessite des serveurs de streaming coopératifs, dans le cas où il serait réalisé sur la couche application.

6.3.2. Mécanisme de réplication pour streaming P2P sur MANET

Les mécanismes de réplication, surtout connus dans les réseaux peer to peer, peuvent apporter une amélioration majeure lorsqu'elles s'appliquent à des réseaux mobiles ad hoc, en particulier pour les applications critiques comme la vidéo en streaming. Dans cette section, nous présentons un nouveau système de réplication de paquets vidéo pour éviter leurs pertes, qui provoqueraient une mauvaise expérience de l'utilisateur. Nous donnons une preuve de concept sur la base d'un début de simulation de la solution proposée.

Nous parlons de réplication, parce que c'est un principe fondamental dans le partage de contenu p2p. Celle ci propose, par des mécanismes, de dupliquer avec des paramètres bien précis, les morceaux de la vidéo, pour palier au risque de perte en cas de départ de sa source initiale. Il existe deux types de réplication. La première, instinctive, est faite par redondance classique, c'est à dire que chaque nœud offre plusieurs copies du même bloc à ses voisins selon un schéma donné. Puis il y a une autre technique, plus élaborée, qui consiste à rajouter des blocs de redondance à l'information utile. Ces blocs de données permettent de reconstruire l'envoi initial en utilisant des algorithmes déjà bien éprouvés dans des applications client/serveur existantes.

Pour les réseaux ad hoc mobile, il existe une dualité dans le principe de réplication, car il y a un compromis à faire, soit par une stratégie de redondance, qui va encombrer le canal radio, ou bien alourdir l'application mobile, en lui intégrant des codes correcteurs pour extraire l'information à partir du jeux de blocs reçus.

6.3.3. Clustering basé sur la session de lecture

Nous proposons un schéma de regroupement, sur la base de session vidéo pour le streaming p2p mobiles. Cette approche réduit les perturbations pouvant survenir à cause de la dynamique supplémentaire causée par les nœuds en mouvement.

Depuis l'émergence de réseaux pairs à pair, de nombreux travaux ont été menés dans le domaine du clustering. En proposant des moyens de division du contrôle distribué dans les réseaux fixes ou mobiles, la combinaison de cette philosophie avec la construction d'arbres multicast applicatifs est à même d'alléger la charge sur l'ensemble du MANET.

Nous proposons, que chaque mobile maintienne un buffer pour obtenir une partie de la vidéo avant de la lire, Ce qui lui permettrait, non seulement de recueillir suffisamment de paquets vidéo pour qu'il n'y ait pas de coupure, mais également d'utiliser moins de mémoire puisque seule une partie de la vidéo, et non plus la totalité est désormais stockée dans le mobile..

Pour cela, nous construisons un cluster sur chacune des sous sessions de lecture, dans laquelle tous les nœuds regardant la même partie d'une vidéo se regroupent pour former un arbre de multicast applicatif. Lorsqu'un nouveau nœud mobile se connecte au réseau, il rejoint le cluster correspondant à la première sous-session. Et ainsi de suite jusqu'à la fin de la lecture. Ce principe a déjà été appliqué dans des protocoles streaming p2p sur Internet tel que Joost, mais il n'a jamais été envisagé pour les réseaux ad hoc mobiles ; alors qu'il présente des avantages potentiels.

6.3.4. Localisation P2P sémantique de contenu vidéo (DHT, MPEG-7 et Ontologies)

L'une des problématiques qui nous ont toujours préoccupé, est l'ajout de la sémantique dans la recherche du contenu vidéo. En combinant cela aux techniques peer to peer telle les DHT, nous sommes convaincus de pouvoir faire de la recherche distribuée de scènes vidéo directement en utilisant des mots clés. Ceci évitera à l'utilisateur de devoir télécharger toute la vidéo pour lire que la seule séquence qui l'intéresse. Un schéma d'indexation basé sur les ontologies faciliterait certainement l'enrichissement sémantique des vidéos. En se basant sur des propositions existantes d'allier les ontologies au DHT, nous en ferons l'application sur le peer to peer streaming, notamment avec l'avènement des nouvelles technologies de compression vidéo, qui incluraient des mécanismes de marquage de l'image. et axe nous semble porteur de perspectives ouvrant un vaste

champ dédié à l'approfondissement de la recherche dans le domaine des technologies des communications virtuelles et notamment la contribution à jeter les jalons des investigations dans la compréhension des systèmes de communication mobiles de demain telle est notre recommandation la plus souhaitable en guise de conclusion de notre thèse plutôt dédiée contributive à cet exaltant domaine de la cybernétique.

Cette thèse présente l'amélioration faisable sur les différents aspects du streaming mobile. Cette investigation nous avance d'un pas en avant vers la compréhension des systèmes de communication mobiles de demain.

Références

- [1]. The IEEE 802.11 Standards, see <http://grouper.ieee.org/groups/802/11/>
- [2]. Ripeanu M Peer-to-peer architecture case study: Gnutella network. Internet2 workshop: collaborative computing in higher education: peer-to-peer and beyond. Tempe, Arizona, January 30–31 2002
- [3]. Maymounkov P, Mazières D, Kademlia: a peer-to-peer information system based on the XOR metric. In: 1st international workshop on peer-to-peer systems (IPTPS 2002). Cambridge, MA, Mar (2002)
- [4]. B. Wiley. Distributed Hash Tables, Part I. *Linux Journal- October 2003*.
- [5]. S Saroiu, KP Gummadi, SD Gribble, Measuring and analyzing the characteristics of Napster and Gnutella hosts, - Multimedia Systems, 2003 - Springer
- [6]. Ferguson D, Trends and statistics in peer-to-peer. Workshop on technical and legal aspects of peer-to-peer television. Amsterdam, Netherlands, Mar. 17, 2006
- [7]. Llie D, Gnutella network traffic measurements and Characteristics. Licentiate Dissertation Series No. 2006:05, ISBN: 91-7295-084-6, April 2006
- [8]. Ritter J. Why Gnutella can't scale. No, really. URL: <http://www.darkridge.com/~jpr5/doc/gnutella.html>
- [9]. A. Wierzbicki, N. Leibowitz, M. Ripeanu and R. Wozniak. Cache Replacement Policies Revisited: The Case of P2P Traffic. 4th GP2P Workshop, Chicago, IL, April, 2004.
- [10]. Liang J, Kumar R, Xi Y, Ross KW, Pollution in P2P file sharing systems. Proc. of INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13–17 March, Miami, FL 2005

-
- [11]. Hoßfeld T, Leibnitz K, Pries R, Tutschku K, Tran-Gia P, Pawlikowski K (2004) Information diffusion in eDonkey file sharing networks. University of Würzburg, Research Report No. 341, Sept
 - [12]. Bram Cohen. Incentives Build Robustness in BitTorrent. Workshop on Economics of Peer-to-Peer Systems, 2003
 - [13]. Guha S, Daswani N, Jain R, 2006 An experimental study of the Skype peer-to-peer VoIP system. In: Proceedings of the IPTPS'06. Santa Barbara, CA, Feb 2006
 - [14]. J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN: simple traversal of user datagram protocol (UDP) through network address translators (NATs). RFC 3489, IETF, Mar. 2003.
 - [15]. Huang G, PPLive – a practical P2P live system with huge amount of users. keynote speech at P2PTV workshop, Kyoto, Japan 2007
 - [16]. Hei X, Liang C, Liang J, Liu Y, Ross KW, Insights into PPLive: a measurement study of a large-scale P2P IPTV system. In: Workshop on Internet Protocol TV (IPTV) services over World Wide Web in conjunction with WWW2006, Edinburgh, Scotland, May 2006
 - [17]. M. Nafaa and N. Agoulmine, "Analysing Joost peer to peer IPTV protocol ", 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), 2009.
 - [18]. Chu Y, Rao SG, Seshan S, Zhang H, A case for end system multicast. IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast, 2002
 - [19]. Chawathe Y, Scattercast: an architecture for internet broadcast distribution as an infrastructure service. PhD thesis, University of California, Berkeley, August 2000
 - [20]. Jannotti J, Gifford DK, Johnson KL, Kaashoek MF, O'Toole Jr. JW, Overcast: reliable multicasting with an overlay network. In Proc. of the Fourth Symposium on Operating System Design and Implementation (OSDI), October 2000
 - [21]. Padmanabhan VN, Wang HJ, Chou PA, Sripanidkulchai K, Distributing streaming media content using cooperative networking. ACM NOSSDAV, Miami Beach, FL, USA, May 2002

-
- [22]. Castro M, Druschel P, Kermarrec A-M, Nandi A, Rowstron A, Singh A (2003) SplitStream: high-bandwidth multicast in a cooperative environment. SOSP'03, Lake Bolton, New York, October 2003
- [23]. Li J, Chou PA, Zhang C, Mutualcast: an efficient mechanism for content distribution in a P2P network. Proc. Acm Sigcomm Asia Workshop. Beijing, China, Apr. 10–12 2005
- [24]. Cherkasova L, Lee J, FastReplica: efficient large file distribution within content delivery networks. In: Proc. of the 4-th USENIX Symposium on Internet Technologies and Systems. Seattle, Washington, March 26–28 2003
- [25]. Li J, Cui Y, PeerStreaming: design and implementation of an on-demand distributed streaming system. To be published in ACM Multimedia Systems Journal 2007
- [26]. Huang C, Li J DISCOVER: distributed collaborative vidéo recorder. International Conference on Multimedia & Expo (ICME'2006), Toronto, Canada, Jul. 9–12 2006
- [27]. Kostic D, Braud R, Killian C, Vandekieft E, Anderson JW, Snoeren AC, Vahdat A, Maintaining high-bandwidth under dynamic network conditions. Proceedings of 2005 USENIX Annual Technical Conference (USENIX 2005). April 2005
- [28]. Gkantsidis C, Rodriguez P Network Coding for Large Scale Content Distribution. Proc. of INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, 13–17 March 2005
- [29]. Zhang M, Luo JG, Zhao L, Yang SQ, A peer-to-peer network for live media streaming using a push-pull approach. Proc. ACM Multimedia 2005, Singapore, pp 287–290, Sept 2005
- [30]. P. Jacquet T. Clausen. RFC 3626: Optimized Link State Routing Protocol (OLSR), Oct 2003.
- [31]. C. E. Perkins, E. M. Belding-Royer, and S. Das, “Ad hoc on demand distance vector (AODV) routing,” Internet Engineering Task Force, RFC 3561, July 2003. [Online]. Available: <http://rfc.net/rfc3561.txt>
- [32]. Haas, Z.J. and Pearlman, M.R., "The Zone Routing Protocol: A Hybrid Framework for Routing in Ad Hoc Networks," in Ad Hoc Networks, edited by C.E. Perkins, Addison-Wesley, 2000.

-
- [33]. T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.
- [34]. F. Bai and A. Helmy, A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks, Book Chapter, *Wireless Ad Hoc and Sensor Networks*. Kluwer Academic Publishers, June 2004.
- [35]. A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.
- [36]. P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pockets Switched Networks and Human Mobility in Conference Environments. In *Proceedings of ACM SIGCOMM'05 Workshops*, pages 244–251, August 2005.
- [37]. A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. Real world Environment Models for Mobile Ad hoc Networks. *IEEE Journal on Special Areas in Communications - Special Issue on Wireless Ad hoc Networks*, 23(3), March 2005.
- [38]. M. McNett and G. M. Voelker. Access and Mobility of Wireless PDA User. *Mobile Computing Communications Review*, 9(2):40–55, April 2005.
- [39]. T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of MobiCom'04*, pages 187–201, New York, NY, USA, 2004. ACM Press.
- [40]. A. Einstein. *Investigations on the Theory of the Brownian Movement*. Dover Publications, 1956.
- [41]. J.-Y. Le Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proceedings of IEEE INFOCOM'05*, pages 72–79, March 2005.
- [42]. J.-Y. Le Boudec and M. Vojnovic. The random trip model: stability, stationary regime, and perfect simulation. *IEEE/ACM Transactions on Networking*, 14(6):1153–1166, 2006.
- [43]. P. Nain, D. Towsley, B. Liu, and Z. Liu. Properties of random direction models. In *Proceedings of INFOCOM'05*, March 2005.
- [44]. J.-Y. Le Boudec. Understanding the simulation of mobility models with Palm calculus. *Performance Evaluation*, 64(2):126–147, 2007.

-
- [45]. W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.
- [46]. C. Bettstetter, H. Hartenstein, and X. P´erez-Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks (Special Issue on Modeling and Analysis of Mobile Networks)*, 10(5):555–567, 2004.
- [47]. C. Bettstetter, G. Resta, and P. Santi. The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003.
- [48]. G. Resta and P. Santi. An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks. In *Proceedings of POMC’02*, pages 44–50, New York, NY, USA, 2002. ACM.
- [49]. M. Garetto and E. Leonardi. Analysis of random mobility models with PDE’s. In *Proceedings of MobiHoc’06*, pages 73–84, New York, NY, USA, 2006. ACM.
- [50]. X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc networks. In *Proceedings of the 2nd International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM’99)*, pages 53–60, 1999.
- [51]. K. Wang and B. Li. Group mobility and partition prediction in wireless ad-hoc networks. *Proceedings of ICC’02*, 2:1017–1021, 2002.
- [52]. K. Blakely and B. Lowekamp. A structured group mobility model for the simulation of mobile ad hoc networks. In *Proceedings of MobiWac’04*, pages 111–118, New York, NY, USA, 2004. ACM.
- [53]. M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. On Clustering Phenomenon in Partitioned Mobile Networks. In *Proceedings of First ACM SIGMOBILE International Workshop on Mobility Models for Networking Research (MobilityModels’08)*, Hong Kong S. A. R., China, May 2008.
- [54]. M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proceedings of the 2nd ACM/SIGMOBILE International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN’06)*. ACM Press, May 2006.
- [55]. M. Musolesi and C. Mascolo. Designing mobility models based on social network theory. *ACM SIGMOBILE Mobile Computing and Communication Review*, 11(3), July 2007.

-
- [56]. M. Gerla, C. Lindemann, and A. Rowstron. P2P MANET's - new research issues. In Dagstuhl Seminar Proceedings, number 05152, Dagstuhl, Germany, Apr. 2005.
- [57]. A. Rowstron and P. Druschel, Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, Proc. IFIP/ACM Int. Conf. on Distributed Systems Platforms, LNCS 2218, Heidelberg Germany, Springer 2001.
- [58]. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, Proc. Int. Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (ACM SIGCOMM), San Diego, CA, 2001.
- [59]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network, Proc. Int. Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (ACM SIGCOMM), San Diego, CA., 2001.
- [60]. P.Johansson et al. "Scenario Based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks", Mobicom '99, 1999 Seattle, USA
- [61]. J. Broch et al., "A Performance Comparison of Multi-Hop Wireless Ad hoc Network Routing Protocols", MobiCom '98, 1998, Dallas, USA
- [62]. O. Bertsekas, R. Gallager, Data Networks, 2nd Edition, Prentice Hall Inc., 1992
- [63]. R. Schollmeier. "A definition of Peer-to-Peer Networking towards a Delimitation against Classical Client-Server Concepts". Proceedings of WATM-Eunice 2001. pp. 131-138. 2001.
- [64]. S. Saroiu, P. K. Gummadi, S. D. Gribble. "A Measurement Study of Peer-to-Peer File Sharing Systems" Technical Report UW-CSE-01-06-02. July 2001
- [65]. R. Atkinson, S. Kent. "Security Architecture for the Internet Protocol". RFC2401. November 1998
- [66]. Y. C. Hu, S. M. Das, and H. Pucha. Exploiting the synergy between peer-to-peer and mobile ad hoc networks. In Proc. of HotOS-IX, May 2003.
- [67]. F. Dabek, M.F. Kaashoek, D. Karger, R.Morris, and I. Stoica. Wide-area cooperative storage with CFS. In Proc. of ACM SOSP, October 2001.
- [68]. A. Rowstron and P. Druschel. PAST: a large-scale, persistent peer-to-peer storage utility. In Proc. of ACM SOSP, October 2001.

-
- [69]. M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: highbandwidth multicast in a cooperative environment. In Proc. of ACM SOSIP, October 2003.
- [70]. M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. Scribe: a large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in Communication (JSAC), 20(8), October 2002.
- [71]. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using contentaddressable networks. In Proc. of the Third International Workshop on Networked Group Communication, November 2001.
- [72]. R. Zhang and Y.C. Hu. Borg: a hybrid protocol for scalable application-level multicast in peer-to-peer systems. In Proc. of NOSSDAV, June 2003.
- [73]. S.Q. Zhuang, B.Y. Zhao, A.D. Joseph, R.H. Katz, and J. Kubiatowicz. Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination. In Proc. of NOSSDAV, June 2001.
- [74]. C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In Proc. of ACM SIGCOMM, August 2004.
- [75]. L.B. Oliveira, I.G. Siqueira, and A.A. Loureiro. Evaluation of ad-hoc routing protocols under a peer-to-peer application. In Proc. of IEEE WCNC, March 2003.
- [76]. A. Klemm, C. Lindemann, and O. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In Proc. of IEEE VTC, October 2003.
- [77]. F. Delmastro. "From Pastry to CrossROAD: CROSS-layer Ring Overlay for AD hoc networks". In *Proc. of PerCom*, March 2005.
- [78]. S.M. Das, H. Pucha, and Y.C. Hu. Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks. Technical Report TR-ECE-04-04, Purdue University, January 2004.
- [79]. Q-T Vuong-Nguyen, Mobility Management in 4G Wireless Heterogeneous Networks, Thèse de doctorat, University of Evry Val d'Essonne, Evry, July 2007
- [80]. Y. Birk and Y. Wiener, A bucket-interleaving multiplexer for efficient near-on-demand streaming to resource-constrained clients, in Proc. of IEEE International Conference on Multimedia and Expo ICME'02, vol. 1, 2002.

-
- [81]. L. Cai and Y. Lu, Energy management using buffer memory for streaming data, in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 2, pp. 141-152, 2005.
- [82]. A. Ferreira, "Optimizing Microsoft Windows Media Services 9 Series", Technical Brief, Microsoft Windows Digital Media Division, March, 2005.
- [83]. R. Koodli, "Fast handovers for Mobile IPv6, IETF RFC 0468," 2005.
- [84]. H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, "Hierarchical mobile IPv6 mobility management (HMIPv6), IETF RFC 4140," 2005.
- [85]. C. Blondia, N. Van den Wijngaert, G. Willems, and O. Casals, "Performance Analysis of Optimized Smooth Handoff in Mobile IP," in *Proc. of 5th ACM Int. Workshop on Modeling analysis and simulation of wireless and mobile systems*, Atlanta, USA, 2002.
- [86]. M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth", in *Proc. of Passive and Active Measurements Workshop*, 2002.
- [87]. V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", in *Proc. of Passive and Active Measurement Workshop*, 2003.
- [88]. C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *Proc. of IEEE INFOCOM*, vol. 2, 2001.
- [89]. J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools", in *Proc. of the 2003 ACM SIGCOMM conference on Internet measurement*, USA, 2003, pp. 39-44.
- [90]. B. Liang and Z. Haas, "Predictive Distance-Based Mobility Management for Multidimensional PCS Network," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, 2003.
- [91]. J. Francois, G. Leduc, and S. Martin, "Learning movement patterns in mobile networks: a generic method," in *Proc. of European Wireless*, 2004, pp. 128-134.
- [92]. H. Karimi and X. Liu, "A Predictive Location Model for Locationbased Services," in *Proc. of ACM Int. Workshop Advances in Geographic Information Systems (GIS)*, 2003.

-
- [93]. H. Ebersnan and O. Tonguz, "Handoff Ordering using Signal Prediction Priority Queuing in Personal Communications Systems," in *IEEE Trans. Veh. Technology*, vol. 48, no. 1, 1999, pp. 20–35.
- [94]. S. Sheu and C. Wu, "Using Grey Prediction Theory to Reduce Handoff Overhead in Cellular Communication Systems," in *Proc. of IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications*, 2000.
- [95]. S. Rezaei and B. Khalaj, "Grey Prediction Based Handoff Algorithm," *Transactions on Engineering, Computing and Technology*, vol. 2, 2004.
- [96]. P. Bellavista, A. Corradi, and C. Giannelli, "Adaptive Buffering based on Handoff Prediction for Wireless Internet Continuous Services," in *Proc. of Int. Conf. on High Performance Computing and Communications (HPCC05)*, 2005.
- [97]. D. Lee, C. Lee, and J. Kim, "Seamless media streaming over mobile IP-enabled wireless LAN," in *Proc. of IEEE Consumer Communications and Networking Conference (CCNC)*, 2005, pp. 116–121.
- [98]. D. Lee, J. Kim, and P. Sinha, "Handoff-aware Adaptive Media Streaming in Mobile IP Networks," in *Proc. of Int. Conf. on Communications, France*, 2004.
- [99]. D. Lee, C. Lee, and J. Kim, "Seamless media streaming over mobile IP-enabled wireless LAN", in *Proc. of IEEE Consumer Communications and Networking Conference (CCNC)*, 2005, pp. 116-121.
- [100]. J. Deng, "Introduction to grey theory", *The Journal of Grey System*, vol. 1, no. 1.
- [101]. M. Khalil, J. El Falou, W. Duchêne, and D. Hewson, "Automatic threshold determination for a local approach of change detection in long term signal recordings," *EURASIP journal on Advances in Signal Processing*, 2007.
- [102]. M. Basseville and I. Nikoiforov, *Detection of Abrupt changes: Theory and Application*. Prentice Hall, NJ, 1993.
- [103]. M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *IEEE Electronics Letters*, vol. 27, pp. 21452146, Nov. 1991.
- [104]. ETSI, "Selection procedure for the choice of radio transmission technologies of the UMTS," *Tech. Rep. TR 101.112 v3.2.0*, April 1998.

-
- [105]. J. Doble, Introduction to radio propagation for fixed and mobile communications. Artech House, Boston, 1996.
- [106]. E. J. Kempf, Goals for Network-Based Localized Mobility Management (NETLMM), IETF RFC 4831, April 2007.
- [107]. F. Gustafsson, Adaptive Filtering and Change Detection. Wiley New York, 2000.
- [108]. Carra, Damiano;Lo Cigno, Renato;Biersack, Ernst W, Graph based analysis of mesh overlay streaming systems, IEEE Journal of selected areas in communications Vol. 25 N° 9 p.1667-1677, December 2007
- [109]. Carra, Damiano;Lo Cigno, Renato;Biersack, Ernst W, Graph based modeling of P2P streaming systems, Networking 2007, 6th IFIP international conference on Networking, May 14 -18, 2007, Atlanta, USA
- [110]. Carra, Damiano;Lo Cigno, Renato;Biersack, Ernst W, Fast stochastic exploration of P2P file distribution architectures, Globecom 2006, November 27, December 1st 2006, San Francisco, USA
- [111]. Carra, Damiano;Lo Cigno, Renato;Biersack, Ernst W, Content delivery in overlay networks: a stochastic graph processes perspective, GLOBECOM 2006, November 27th - December 1st, 2006, San Francisco, USA
- [112]. B Biskupski, M Schiely, P Felber, R Meier, Tree-Based Analysis of Mesh Overlays for Peer-to-Peer Streaming, Lecture Notes in Computer Science, Springer, 2008.
- [113]. Y. C. Tu, J. Sun, M. Hefeeda, and S. Prabhakar, "An Analytical Study of Peer-to-Peer Media Streaming Systems," ACM TOMCCAP, 1(4), 2005.
- [114]. S. Nikolettseas, J. Reif, P. Spirakis, and M. Young, "Stochastic Graphs Have Short Memory: Fully Dynamic Connectivity in Poly-Log Expected Time," Proc. 22nd Int'l Colloquium Automata, Languages, and Programming (ICALP '95), pp. 159-170, 1995.
- [115]. S. N. Dorogovtsev, and J. F. F. Mendes, "Evolution of Networks: From Biological Nets to the Internet and WWW," Oxford University Press, Oxford, January 2003.
- [116]. N. Magharei, R. Rejaie, "PRIME: Peer-to-Peer Receiver-driven MESH-based Streaming," in Proc. IEEE INFOCOM 2007, Anchorage, Alaska, May 2007.
- [117]. Y. Sung, M. Bishop, and S. Rao, "Enabling Contribution Awareness in an Overlay Broadcasting System," in Proc. SIGCOMM 2006, Sept. 2006.

-
- [118]. X. Zhang, J. Liu, B. Li, and T. S. P. Yum, "DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming," in Proc. IEEE INFOCOM 2005, Miami, Mar. 2005.
- [119]. V. K. Goyal, "Multiple Description Coding: Compression Meets the Network," in IEEE Signal Processing Magazine, pp. 7493, Sept. 2001.
- [120]. Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "The Use of Forward Error Correction (FEC) in Reliable Multicast", RFC 3453, December 2002.
- [121]. M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In Proceedings of the 2nd ACM/SIGMOBILE International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN'06). ACM Press, May 2006.
- [122]. D. J. Watts. Small Worlds The Dynamics of Networks between Order and Randomness. Princeton Studies on Complexity. Princeton University Press, 1999.
- [123]. J.Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2006-01-31). Downloaded from <http://crawdad.cs.dartmouth.edu/cambridge/haggle>, Jan. 2006.
- [124]. A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.
- [125]. J. Scott. Social Networks Analysis: A Handbook. Sage Publications, London, United Kingdom, second edition, 2000.
- [126]. S. Wasserman, K. Faust, and D. Iacobucci. Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences). Cambridge University Press, November 1994.
- [127]. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E, 69, February 2004.
- [128]. L. Viennot, P. Jacquet, T. H. Clausen, "Analyzing control traffic overhead versus mobility and data traffic activity in mobile Ad-Hoc network protocols," presented at ACM Wireless Networks journal (Winet), 2004.

-
- [129]. M.K.Sbai , C.Barakat, J.Choi, A.Al Hamra, T.Turletti, "Adapting BitTorrent to wireless ad hoc networks" In proc. of the 7th International conference on ad hoc networks and wireless 2008 (AD-HOC NOW), Sophia Antipolis, France, September 2008.
- [130]. A.Klemm, C.Lindermann, O.Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In Proc. of IEEE VTC, Oct 2003.
- [131]. G. Ding, B. Bhargava, Peer-to-Peer File-Sharing over Mobile Ad hoc Networks. In Proc. of the 2nd IEEE PERCOM-W, Orlando, FL, USA, 2004.
- [132]. S. Rajagopalan, C-C. Shen, A cross-Layer Decentralized BitTorrent for Mobile Ad hoc Networks. In Proc. of ACM/IEEE MOBIQUITOUS, San Jose, CA, USA, 2006
- [133]. S.M. Das, H.Pucha, Y.C. Hu. Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks. Technical Report TR-ECE-04-04, Purdue University, Jan 2004.
- [134]. T. Zahn and J. Schiller. "MADPastry: A DHT Substrate for Practicably Sized MANETs". In Proc. Of ASWN, June 2005.
- [135]. Mani, M., Nguyen, A., Crespi, N., "SCOPE- Service Classified Overlay for P2P Environment, a service platform for P2P services over Ad-hoc Networks (demo)", IEEE MASS 2008, Atlanta, September 2008.
- [136]. Magharei, N., Rejaie, R., Guo, Y.: Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In: Proceedings of 26th IEEE International Conference on Computer Communication (INFOCOM), May 2007, 1424–1432.
- [137]. Kumar, R., Liu, Y., Ross, K.: Stochastic fluid theory for p2p streaming systems. In: Proceedings of 26th IEEE International Conference on Computer Communication (INFOCOM), May 2007, 919–927.
- [138]. A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing Bit-Torrent for supporting streaming applications," in IEEE Global Internet, Apr. 2006.
- [139]. S. Tewari and L. Kleinrock, "Analytical model for bittorrent-based live vidéo streaming", In proc. IEEE NIME Workshop, 2007.
- [140]. C.Perkins, P.Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers in ACM SIGCOMM'94.

- [141]. W. Poon, J. Lee, D. Chiu: Comparison of Data Replication Strategies for Peer-to-Peer Video Streaming. 5th International Conference on Information, Communications, and Signal Processing, Bangkok, Thailand, Dec. 2005.

Annexe A.

A.1. Liste des publications

A.1.1. Articles de recherche

1. **M. Nafa** and N. Agoulmine, "MadP2PStream: Mobile Peer-to-peer Streaming over MANETs: Model and Optimizations", **soumis à** Springer Peer-to-Peer Networking and Applications PPNA (journal)
2. **M. Nafa** and N. Agoulmine, " MadTorStream: On P2P streaming over mobile communities", **soumis à** 1st IEEE International Workshop on Community driven Mobile Multimedia (CMM2009). (full paper)
3. V.K. Gondi, **M. Nafa** and N. Agoulmine, "Network Assisted Network Selection procedure For Heterogeneous Wireless and Cellular Networks", 4th IFIP/IEEE International Workshop on Broadband Convergence Networks (BcN 2009) (full paper)
4. **M. Nafa** and N. Agoulmine, "Analysing Joost peer to peer IPTV protocol ", 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), 2009. (poster)
5. **M. Nafa** and N. Agoulmine, "Handover Prediction-based Seamless Media Streaming for Wireless Packet Networks", The Fifth International Conference on Autonomic and Autonomous Systems, ICAS 2009 (Demo Session)
6. **M.Nafa**, V.K.Gondi and N.Agoulmine, "Multimedia Streaming in IPTV over 802.16 Experiments: Tesbed Measurements", First International Workshop on Wireless Broadband Access for Communities and Rural Developing Regions, 10 pages, Karlstad University Press, 2008. (full paper)
7. **M. Nafa**, N. Agoulmine, and Y. Ghamri-Doudane, "Topological zone/peer naming in Mobile P2P Networks", 13th Annual Workshop of HP OpenView University Association, Nice, France, Mai 21 - 24, 2006. (poster)

A.1.2. Rapports de projet

1. **M. Nafa** et al., "D1.1: State of the art analysis", Technical Deliverable Report D1.1, project ITEA WellCom, 2008.
2. **M. Nafa** et al., "D3.1: State of the art analysis for P2P Systems", Technical Deliverable Report D3.1, project ITEA ExpeShare, 2008.
3. **M. Nafa** et al., "D3.2: P2P ExpeShare Network Services Architecture (P2PNSA) and API specification ", Technical Deliverable Report D3.2, project ITEA ExpeShare, 2008.
4. **M. Nafa** et al., "D1.2: SUMO system requirements and basic functional architecture", Technical Deliverable Report D1.2, project ITEA SUMO, 2007.
5. **.M. Nafa** et al., "D2.3: Network selection, Mobility and security management", Technical Report D2.3, project ITEA SUMO, 2007.
6. **M. Nafa** et al., "D5.1 Demonstrator description & integration plan", Technical Deliverable Report D5.1, project ITEA SUMO, 2007.

A.2. Index

- AAA, 37, 38
affinité, xi, xviii, 75, 76, 78, 82, 83, 99
ALM, 22
AODV, xvi, 29, 40, 103, 113
AP, 44, 49, 50, 51, 55, 59, 60, 61
bandes, 69, 70, 71, 72, 85, 86, 87
BER, 9
BitHoc, 97, 98, 99, 100, 102
BitRate, 3, 97
BitTorrent, xi, xvi, 10, 13, 16, 17, 18, 19, 22, 25, 26, 33, 94, 95, 97, 99, 102, 107, 108, 112, 121
Bluetooth, 31
buffering, 4, 43, 45, 46, 47, 48, 49, 50, 51, 52, 56, 58, 59, 62, 63, 67
CAN, 33, 40
Chord, 33, 40, 116
churn *Voir* dynamique
client/serveur, 1, 108
Closest Deadline First, xii, 98, 99, 101, 102
Community based mobility, xvii, 32
Crossroad, 40
CSGP, 68
CUSUM, 53, 54, 55
DHT, xi, xx, 11, 12, 16, 39, 40, 41, 97, 109, 122
DSDV, 28, 40, 98, 99, 122
DVB, 8
eDonkey, xvi, 13, 16, 17, 111
EM, 72, 73, 74, 91
eMule, xi, 16, 18
ExpeShare, 4, 94, 96, 124
FaceBook, 1
FAI, 3
fast MIP, 45
FastTrack, xvi, 15, 17
FEC, 71, 120
filtre Grey *Voir* GM
flash crowd, 71
Gauss-Markov, 46
GM, xvii, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57
GPS, 31, 93
Groover, xiii, xix, 67, 84
handover, xiii, xvii, xviii, xx, 4, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 59, 60, 61, 62, 63, 105, 106, 108
Heterogeneous Random Walk, 32
hierarchical MIP, 45
HTTP, 8, 15
IARP, 30
IEEE, 8, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123
IERP, 30
IETF, 28, 112, 117, 118, 119
INRIA, 1, 28
inter-couche, 33
IP, xvi, 2, 8, 10, 12, 13, 14, 15, 18, 19, 33, 36, 37, 41, 44, 45, 48, 60, 93, 118, 119
IPSEC, 37, 38
IPTV, 2, 93
ITEA, 4, 94, 96, 124
Joost, xvi, 10, 19, 21, 93, 95, 109, 112, 123
Kademlia, 16, 111
Kazaa, xi, 10, 15, 18
MAC, 8, 33, 100
MADP2PStream, 5, 65, 66, 82, 85, 92, 103, 105, 106, 107, 108
MadTorStream, 5, 6, 93, 94, 95, 97, 98, 100, 105, 107
MANET, 3, 6, 13, 27, 28, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41, 66, 67, 68, 69, 74, 80, 82, 83, 84, 92, 93, 96, 100, 102, 105, 106, 107, 108, 109
MANETs, 4, 5, 7, 9, 24, 27, 33, 34, 36, 38, 39, 41, 82, 94, 95, 96, 99, 102, 106, *Voir* MANET
Markov cachés, 46
Markovien, 72
MDC, 71
Mesh, ix, xi, 22, 23, 24, 97, 120, 122
middleware, 33

- mobilité, xvii, xviii, xix, 1, 3, 4, 5, 6, 9, 27, 29, 30, 31, 32, 33, 34, 36, 38, 40, 43, 44, 45, 63, 66, 67, 68, 70, 74, 75, 77, 78, 79, 80, 81, 82, 83, 84, 86, 87, 89, 92, 94, 96, 98, 99, 100, 102, 105, 106, 107
- Monte-Carlo, 67
- MPEG-TS, 8
- MPR, 28, 29
- Napster, 10, 13, 14, 16, 111
- NAT, 18
- Network Selection, 44, 123
- NS-2, 5, 100, 102, 107
- OLSR, xvi, 28, 29, 40, 103, 113
- ORION, 40
- overhead, xi, xviii, 28, 29, 38, 66, 67, 79, 80, 82, 84, 90, 91, 92, 121
- P2P, 1, 2, 4, 6, 10, 11, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 33, 34, 35, 36, 37, 38, 39, 40, 41, 65, 68, 96, 100, 107, 108, 109
- Pastry, 33, 40, 41, 115, 117
- Pathchirp, 46, 58
- Pathload, 46, 118
- PPLive,, 19, 20
- proactif, xviii, 5, 6, 28, 80, 81, 83, 84, 91, 92, 94, 102, 107
- pull, 22, 24, 26, 113
- push, xvi, 24, 25, 26, 113
- QoE, 9
- QoS, 4, 8, 26, 29, 38, 43
- RAM, 45
- Random Trip, 31
- Random Walk, 31, 96
- Random Way-Point, 31, 32, 78
- RE, 73, 74
- réactif, xviii, 5, 34, 63, 80, 81, 83, 91, 92, 94, 103
- Reference Group, 32
- Reference Velocity Group, 32
- RLS, 53, 55
- RREP, 29, 30
- RREQ, 29
- RSS, xi, 44, 46, 49, 50, 51, 52, 53, 54, 55, 56, 57, 60, 61, 62
- RTCP, 8
- RTP, 8
- RTSP, 8
- sans couture* Voir seamless
- seamless*, 46, 106
- Skype, 10, 18, 19, 21, 112
- SM²Stream, xi, xvii, xix, 4, 5, 43, 61, 105, 106
- smooth handover scheme, 45
- Spruce, 46
- streaming, 2, 3, 4, 5, 6, 7, 8, 9, 13, 19, 20, 21, 22, 24, 25, 26, 37, 41, 43, 44, 45, 46, 47, 48, 49, 51, 57, 58, 60, 62, 63, 65, 66, 67, 68, 69, 70, 71, 73, 74, 82, 83, 87, 92, 93, 94, 95, 96, 97, 100, 101, 102, 105, 106, 107, 108, 109, 110, 113, 122
- Structured Group, 32
- SUMO, 4, 43, 124
- TCP, 8
- tracker, 16, 20, 99
- Tree, ix, xi, 22, 23, 24, 120
- TTL, 14, 29, 35, 36, 40
- UDP, 8, 14, 47, 48, 49, 50, 58, 112
- UMTS, 44, 119
- VLC, xviii, 19, 58, 59
- VoD, 2, 93
- VoIP, 2, 18, 95
- Wifi, 4
- WiFi, 5, 43, 44, 93, 97
- WiMax, 44
- WLAN, 4, 9, 41, 44, 47, 48, 49, 51, 52, 55, 57, 106
- YouTube, 1
- ZRP, xvii, 30

A.3. Glossaire

AAA	Authentication Autorisation Accounting
ALM	Application-Layer Multicast
AODV	Advanced On-demand Distance Vecto routing
AP	Access Point
BER	Bit Error Rate
CAN	Content Addressable Network
CUSUM	CUmulative SUM
DHT	Distributed Hash Tables
DVB	Digital Vidéo Broadcasting
FAI	Fournisseur d'Accès à Internet
MIP	Mobile Internet Protocol
GM	Grey Model
GPS	Global Positionning System
HTTP	HyperText Transfert Protocol
IARP	IntrA-zone Routing Protocol
IEEE	International Electric Electronic Engineers
IERP	IntEr-zone Routing Protocol
IETF	Internet Engineering Task Force
INRIA	Institut National de Recherche en Informatique et Automatique
IP	Internet Protocol
IPSEC	Internet Protocol SECured
IPTV	Internet Protocol TeleVision
MAC	Medium Access Control
MADP2PStream	Mobile ad hoc Peer To Peer Streaming
MadTorStream	Mobile ad hoc Torrent Streaming
MANET	Mobile Ad hoc NETwork
MPEG-TS	Moving Pictures Express Group – Transport Stream
MPR	MultiPoint Relay
NAT	Network Address Translator
NS-2	Network Simulator 2
OLSR	Optimized LinkState Routing protocol
P2P	Peer to Peer
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Acces Memory
RREP	Route REPlY
RREQ	Route REQuest
RSS	Received Signal Strength
RTCP	Real Time Control Protocol

RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SM ² Stream	Seamless Mobile Multimedia Streaming
SUMO	Service Ubiquity in Mobile and Wireless Realm
TCP	Transmission Control Protocol
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VLC	VideoLan Client
VoD	Vidéo on Demand
VoIP	Voice over Internet Protocol
WiFi	Wireless Fidelity
WiMax	Worldwide Inter-operability for Microwave Access
WLAN	Wireless Local Area Network
ZRP	Zone Routing Protocol

A.4. Media Streaming dans VLC