Content replication in mobile wireless networks

Chi-Anh La

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur

de TELECOM ParisTech

Spécialité : Informatique et réseaux

Chi-Anh LA

Réplication de contenu dans les réseaux sans fil mobiles.

Soutenu le 11 octobre 2010 devant le jury composé de

Pr. Guillaume Urvoy-Keller	Président
Pr. Carla-Fabiana Chiasserini	Rapporteurs
Dr. Luigi Liquori	
Dr. Chadi Barakat	Examinateurs
Pr. Christian Bonnet	
Pr. Pietro Michiardi	Directeur de thèse

ACKNOWLEDGEMENT

I would like to thank my thesis advisor, Prof. Pietro Michiardi for his kindness in helping me to find my research direction and giving the basic lessons on how to do research. The completion of this study would have been impossible without the his valuable helps and advices.

I want to thank to all EURECOM staff for their helps during my more than three years at this beautiful institution.

A special thank to Prof. Carla-Fabiana Chiasserini, Prof. Guillaume Urvoy-Keller, Dr. Chadi Barakat and Dr. Melek Onen, who have spent time to read and correct the preliminary version of this manuscript.

Finally, I am deeply indebted to all members of my family, who constantly support me during my study.

Abstract

The growth of mobile devices and network-based services nowadays has raised a timely question on how to efficiently distribute the data items to mobile users. Network applications need data as an input to process and provide information to users. Consequently, data traffic exerted by mobile devices fetching content is a drainage of mobile operators' network resources. Similar to the wired Internet, mobile users are now coping with the congestion at network gateways and due to the unpredictability of human mobility, mobile service providers cannot sufficiently provision infrastructures for their customers.

Content replication in this context has been proved as a good solution to enhance network performance and scalability. In this thesis, we tackle the issues of content replication in heterogeneous mobile networks. Such scheme requires us to solve two basic questions : where and how many replicas should be placed in the system. We study the solution through the lenses of facility location theory and design a distributed mechanism that reduces content access latency and avoids congestion at mobile gateways. Additionally, we consider the resource constraints of mobile devices and introduce a P2P cache-and-forward mechanism for load balancing purpose. We evaluate our mechanisms against realistic human mobility models.

Finally, to address rational users who may behave selfishly in replicating content, we derive a cost model and study content replication scheme using tools akin to game theory. We focus on the replication factor under a flash-crowd scenario with different wireless bit rates. Based on the theoretical findings, our future work is to develop the strategies to be implemented in a practical network setting.

Keywords : content replication, mobile wireless networks, facility location theory, distributed algorithms, human mobility

Résumé

La croissance des terminaux et des services de réseau mobile pose aujourd'hui une question sur la méthode de distribuer efficacement des données aux utilisateurs. Plusieurs applications de réseau ont besoin de télécharger des données afin de fournir des informations aux utilisateurs. En conséquence, l'explosion du trafic de données exercé par les clients qui cherchent des contenus en ligne provoque la saturation du réseau cellulaire des opérateurs mobiles. Similaire aux problèmes du réseau Internet, les utilisateurs mobiles ont désormais fait face à la congestion au niveau des passerelles de réseau. En raison de l'imprévisibilité de la mobilité humaine, les fournisseurs de services mobiles ne peuvent pas installer suffisamment des infrastructures pour leurs clients.

La réplication de contenu dans ce contexte a été prouvée comme une bonne solution pour améliorer la performance et l'extensibilité du réseau. Dans cette thèse, nous abordons les problèmes de la réplication du contenu dans des réseaux hétérogènes mobiles. Nous étudions deux questions fondamentales : où et combien de répliques doivent être placées dans le système. Nous modélisons le problème à l'aide de la théorie de "facility location" et nous concevons un mécanisme distribué qui est capable de réduire la latence d'accès au contenu et d'éviter la congestion au niveau des passerelles mobiles. En outre, nous examinons les contraintes de ressources des équipements mobiles et proposons des mécanismes P2P pour transférer les répliques afin de parvenir l'équilibrage de charge parmi les utilisateurs. Nous évaluons nos mécanismes en utilisant des modèles de mobilité humaine.

Enfin, pour résoudre le problème causé par les utilisateurs rationnels qui se comportent égoïstement lors de la réplication du contenu dans les réseaux hétérogènes mobiles, nous dérivons un modèle de coût et utilisons la théorie des jeux pour étudier les équilibres du système. Particulièrement, nous étudions le facteur de réplication dans un scénario "flash-crowd" avec de différents débits de réseau sans fil. A partir des résultats théoriques, nos futurs travaux sont d'élaborer des stratégies à mettre en oeuvre dans les réseaux en pratique.

Mots-clés : réplication de contenu, réseaux mobiles sans fil, théorie de la localisation des installations, algorithmes distribués, mobilité humaine

Synthèse en français

1. Introduction

La prolifération des équipements mobiles et des services basés sur le réseau a désormais soulevé une question en temps opportun sur la façon de distribuer efficacement des contenus aux utilisateurs. Pour la troisième génération (3G) de réseaux sans fil, la méthode courante de la diffusion des données est de déployer un fournisseur de contenu centralisé qui envoie directement les données aux utilisateurs via les connexions 3G. De cette façon, les demandes de certains éléments de contenu populaires peuvent malheureusement dépasser la capacité de l'infrastructure provisionnée. Cependant, aujourd'hui les technologies sans fil ont beaucoup évolué et les appareils équipés d'une interface sans fil à faible coût comme Bluetooth ou 802.11 envahissent le marché. Dans ces réseaux sans fil mobiles hétérogènes, la connexion cellulaire et la communication d'appareil à appareil (device-to-device) sont deux options réciproquement complémentaires pour les utilisateurs, ce qui est similaire au modèle "cloud data center" et au système de P2P sur Internet. L'utilisation de la communication d'appareil à appareil peut fournir plus de ressources pour le système global (la bande passante, par exemple), mais elle cause plus de latence et exige plus d'efforts pour la recherche de contenu. Au contraire, les réseaux cellulaires sont soumis à des problèmes d'extensibilité et de congestion. Si le contenu est populaire, la communication d'appareil à appareil permet de réduire la congestion cellulaire en déchargeant partiellement la distribution de contenu aux utilisateurs mobiles : certains noeuds peuvent répliquer le contenu dans leur cache locale pour servir les autres noeuds qui pourrait demander le contenu plus tard. Par exemple, pour fournir un journal électronique aux utilisateurs mobiles, dans les réseaux cellulaires chaque utilisateur doit télécharger une quantité de données. Avec l'aide de la communication d'appareil à appareil, seulement une fraction d'utilisateurs doivent télécharger les données et ensuite ils le redistribuent à d'autres utilisateurs à l'aide des techniques P2P.

Plusieurs études ont indiqué que le trafic de données exercées par les terminaux mobiles pour récupérer les contenus sur l'Internet est déjà une source de saturation des ressources du réseau des opérateurs mobiles [4, 5, 78]. Similaire à l'Internet, les utilisateurs mobiles font face maintenant à la congestion de la passerelle de réseau. Pour résoudre ce problème, la réplication du contenu a été révélée comme une bonne solution pour améliorer les performances et l'extensibilité du réseau.

Toutefois, la procédure de décision pour sélectionner quel contenu à répliquer et à quel noeud n'est pas triviale dans les réseaux mobiles. La topologie du réseau dans ce cas pourra changer rapidement à cause de la mobilité. De plus, les noeuds ne peuvent pas compter sur une infrastructure centralisée pour avoir une vue globale du réseau. Par conséquent, une solution "low-overhead" est nécessaire dans ce contexte. En outre, les terminaux sans fil ont généralement des contraintes de ressources strictes et les utilisateurs peuvent se comporter égoïstement lors de décider de répliquer le contenu.

1.1. Réplication de contenu sur l'Internet

Les solutions de réplication de contenu sur l'Internet sont souvent centralisées car il est possible de déployer un serveur que les utilisateurs peuvent envoyer des requêtes et ce serveur ensuite diffuse les requêtes vers des serveurs de caches. Un tel exemple est le service de distribution de contenu d'Akamai [1] : on utilise une infrastructure DNS pour distribuer les requêtes vers les caches du contenu. Cependant, il est très difficile de provisionner l'infrastructure nécessaire pour s'adapter aux besoins des utilisateurs mobiles en raison de la l'imprévisibilité des changements de topologie du réseau causés par la mobilité.

D'autres systèmes de diffusion de contenu sur Internet sont décentralisés et ne possèdent aucun composant central. Dans cette approche décentralisée, il existe deux options de conception principales. Dans la première option, les serveurs de réplication forment un recouvrement structuré tellement que les requêtes des utilisateurs peuvent être acheminées à un serveur grâce à une table de hachage installée à chaque noeud. Cette table contient la localisation pour chaque contenu. Cette approche toutefois n'est pas assez adaptable pour les réseaux dynamiques dont l'évolution rapide de la topologie déclenche trop de computation pour la maintenance. Dans la seconde option, le réseau est construit d'une manière non structurée et les hôtes utilisent des techniques pour trouver le contenu comme des requêtes aléatoires. Cette dernière approche est plus pratique pour les réseaux mobiles. Pourtant à cause des contraintes de l'énergie et de la bande passante, les mécanismes de localisation de contenu doivent être bien conçus pour faire face à ces limites. En outre, pour faciliter la recherche de contenu dans un système non structuré, nous avons besoin des techniques de répliquer les données très efficaces afin d'améliorer la performance.

1.2. Réplication de contenu dans les réseaux mobiles

Dans les réseaux sans fil, on fait face au problème de fiabilité, de bande passante, et d'interférence. La mobilité peut entraîner d'autres problèmes : des liens entre les noeuds deviennent instables et le réseau est partitionné d'une manière imprévisible. Dans certaines conditions, nous ne pouvons pas compter sur une infrastructure centralisée (comme dans les réseaux mobiles ad hoc - MANET et les réseaux tolérants aux retards - DTNs). Des terminaux mobiles sont généralement les petits équipements avec des ressources limitées (énergie, bande passante). Par conséquent, si le contexte d'application nécessite la coopération des utilisateurs, certains utilisateurs peuvent obtenir un comportement égoïste afin de sauver, par exemple, leur durée de vie de la batterie. Ce contexte introduit une nouvelle classe de problème pour la réplication de contenu dans les réseaux mobiles. En général, on a des problèmes suivants :

– *Mobilité :* En raison de fréquents changements de topologie, le partitionnement du réseau et les perturbations de connexion se produisent plus souvent dans les réseaux

mobiles que dans les réseaux filaires. Le partitionnement du réseau réduit considérablement la disponibilité des données lorsque le noeud qui détient des données désire de ne plus rester dans la même partition que les demandeurs de données. La réplication de données pour les partitions prévues à l'avenir peut améliorer la disponibilité des données. La redondance de contenu peut également améliorer l'opportunité pour les noeuds de trouver le contenu le plus proche tout en mouvement. Par conséquent, le mécanisme de réplication doit tenir compte tous ces dynamiques du réseau mobile afin de répliquer les éléments de données bien à l'avance. L'étude de la mobilité humaine est aujourd'hui un sujet qui attire l'attention de nombreux chercheurs. Il existe de nombreux modèles de mobilité proposés ainsi que des traces de mobilité pour étudier la performance des réseaux mobiles. Les traces ne sont cependant pas très utiles pour simuler des réseaux mobiles puisque le nombre de participants n'est pas élevé et la durée de l'expérience n'est pas suffisamment longue. C'est pourquoi on utilise ces traces seulement pour concevoir et valider des modèles de mobilité. En outre, de nombreux modèles de mobilité sont principalement construits à partir des mouvements aléatoires. Récemment, certaines tentations préliminaires ont été effectuées pour proposer des modèles "plus humains". Il est signalé que certains modèles de mobilité aident à améliorer les performances du réseau alors que ce n'est pas le cas dans la réalité [24]. Le choix judicieux d'un modèle de mobilité approprié est nécessaire pour comprendre le vrai problème dans un contexte de réseau particulier et pour évaluer la performance d'un mécanisme de distribution de contenu.

- *Contraintes énergétiques et l'équilibrage de charge :* Les noeuds mobiles fonctionnent avec des piles qui sont supposées d'avoir une capacité limitée, malgré de l'avance dans la technologie de batteries. Si un seul noeud est supposé de servir de nombreux clients, sa puissance peut d'être épuisée très rapidement. Pour améliorer la disponibilité des données, le mécanisme de réplication doit reproduire le contenu à plusieurs noeuds afin de faire partager la charge de distribution et protéger certains noeuds de l'épuisement de l'énergie. En outre, il convient également de répliquer les données de telle sorte que la consommation d'énergie de noeuds est réduite en optimisant la distance pour accéder le contenu. Dans ce cas, une approche qui embrasse le paradigme pair-à-pair (P2P) pourrait aider à résoudre le problème (c'est-à-dire il ne faut pas assumer un rôle statique comme serveur ou client à un noeud, chaque noeud doit être un client et aussi un serveur alternativement). Dans ce cas, nous devrions utiliser la version non-structurée de la conception de P2P pour faire face au contexte très dynamique des réseaux mobiles.
- *Disponibilité du contenu :* Les réseaux de téléphonie mobile peuvent impliquer de grandes populations avec des milliers de noeuds, par exemple, des utilisateurs dans un scénario très fréquentés comme dans une stade ou dans une musée. Pour retrouver le contenu dans tel dense et large réseau, une requête à partir d'un noeud client doit peut-être traverser un long chemin pour parvenir à une réplique de contenu, ce qui augmente le coût de la requête et la latence. En outre, l'existence d'un grand nombre de noeuds qui interrogent le contenu peut provoquer des interférences dans les canaux de communication, ce qui diminuent ainsi considérablement la bande passante disponible et augmentent le délai d'accès. De plus, la mobilité des noeuds peut également affecter la disponibilité de contenu. Le mécanisme de réplication doit être conçu de telle sorte que la performance de distribution de contenu ne sera pas largement affectée par ces problèmes.
- *Utilisateurs égoïstes :* Les utilisateurs mobiles sont conscients de la contrainte éner-

gétique et du coût de télécharger des données à l'aide de 3G. Compte tenu de ce fait, on peut prédire que les utilisateurs se comportent égoïstement pour minimiser leurs propres coûts, ce qui peut augmenter le coût total du système, sauf s'ils sont fournis des incitations d'aider à diffuser le contenu. Si un utilisateur choisit de répliquer le contenu, il paie le coût pour le télécharger via le réseau cellulaire et pour fournir le contenu à d'autres utilisateurs sur la demande. Par conséquent, un utilisateur peut choisir aussi de récupérer le contenu en le demandant à d'autres utilisateurs. Dans le contexte du réseau mobile actuel, la dernière solution pourrait être beaucoup moins chère et les utilisateurs peuvent avoir la tendance à répliquer le contenu moins que nécessaire. Le coût total calculé à l'équilibre de Nash dans ce cas peut dépasser le coût optimal par un grand écart. Le système devrait donc décourager les éventuels comportements égoïstes en concevant un mécanisme qui incite les utilisateurs à stocker les données si cela permet d'améliorer la performance et de réduire le coût total.

1.3. Objectifs de recherche :

Les réseaux mobiles hétérogènes posent de nouveaux défis pour la distribution de contenu en raison de la croissance rapide du nombre d'utilisateurs et de la dynamique des comportements humains. L'imprévisibilité du nombre d'utilisateurs introduit plus de difficultés pour le provisionnement de l'infrastructure. Il y a donc un problème d'extensibilité pour le service des fournisseurs. Dans ce contexte, l'utilisation de la communication d'appareil à appareil fournit une solution pour éviter la congestion au niveau des passerelles mobiles. Toutefois, comme les appareils mobiles ont des contraintes en leurs ressources (autonomie de la batterie, bande passante ...), ces problèmes devraient également être pris en compte. La coopération entre les utilisateurs afin de reproduire et de distribuer des contenus via la communication d'appareil à appareil de telle façon à réduire la latence et d'éviter les encombrements au niveau des passerelles est très appréciée. Notre objectif est de concevoir un mécanisme efficace qui travaille dans ce cadre de coopération. Une autre problème dans ce contexte d'application, c'est que comme il y a des contraintes dans les noeuds mobiles, il est raisonnable de supposer que les utilisateurs se comportent égoïstement. Par conséquent, nous devrions nous concentrer sur l'élaboration de stratégies qui peuvent être mises en oeuvre dans le cadre de réseau pratique.

– *Distribution de contenu dans les réseaux hétérogènes mobiles :* Dans les réseaux mobiles hétérogènes, le contenu peut être livré aux utilisateurs soit via la communication d'appareil à appareil, soit via une connexion 3G. Si le contenu est très populaire et tous les utilisateurs veulent aller le chercher, on peut utiliser un système de distribution de contenu en utilisant le transfert de façon d'épidémie (c'est-à-dire le transfert sur contact), car la disponibilité du contenu est élevée. Cette méthode peut réduire le délai de télécharger le contenu et l'effort de le chercher et réduire le besoin d'utilisation du service 3G. En revanche, si seulement quelques utilisateurs sont intéressés par le contenu, il n'y aurait pas de congestion pour les utilisateurs de télécharger directement depuis Internet en utilisant la connexion 3G. Ce problème devient intéressant quand il y a des contenus dont la popularité n'est pas élevée, mais n'est pas aussi faible que le problème de congestion 3G peut être négligé. La réplication dans ce contexte aide à augmenter la disponibilité du contenu. Cela réduit le retard de recherche et de chargement de contenu. De plus, elle pourra encourager les utilisateurs à changer leur choix vers la communication d'appareil à appareil. La réplication aide aussi à réduire le nombre simultané de téléchargements sur l'Internet.

Par conséquent, elle atténue la congestion à la passerelle 3G et favorise l'extensibilité du réseau. Nous avons besoin d'une conception efficace pour placer les répliques de contenu dans les régions où se situe la demande de contenu. En outre, puisque le modèle client-serveur ne s'applique pas dans ce cas, nous avons besoin d'un mécanisme de P2P pour distribuer dynamiquement le rôle de réplication aux utilisateurs mobiles.

– *Mécanismes P2P :* Pour maintenir la charge équilibrée parmi des utilisateurs il nous faut un mécanisme de partage de charge de réplication de contenu. Ce mécanisme devrait être distribué, avec des coûts faibles et ne devrait pas exiger aucune vision globale pour conformer à la nature non-structurée des réseaux mobiles hétérogènes. La mobilité humaine peut changer la topologie du réseau très souvent donc le mécanisme conçu doit être efficace dans cet environnement dynamique. Un mécanisme P2P qui est basé uniquement sur la sélection des pairs au hasard serait une bonne résolution. Notre objectif est d'étudier un mécanisme de transfert au hasard de contenu et ses performances pour voir si une telle solution peut être déployée en pratique.
– *Optimisation de la réplication du contenu :* Le mécanisme de réplication dans les réseaux mobiles devrait améliorer la disponibilité du contenu et réduire la latence de recherche de contenu. Pour le faire, il faut trouver le nombre de répliques nécessaire dans le réseau et l'emplacement pour placer ces répliques. Compte tenu de la topologie du réseau, ce problème peut être étudié à travers les lentilles de la théorie de la localisation des installations. Comme les problèmes de localisation des installations sont NP-difficiles, nous avons besoin d'un mécanisme distribué pour approximer la solution optimale dans les conditions que seules des informations locales sont disponible. Étant donné les problèmes mentionnés ci-dessus, notre travail vise à trouver une solution pour la réplication de contenu qui correspond à la nature dynamique des réseaux mobiles. Nous mettons l'accent en particulier sur un mécanisme léger et pratique qui serait efficace et fondé uniquement sur des mesures locales afin d'achever un coût faible, tout en maintenant de bonnes performances en termes d'équilibrage de charge et de délai pour la récupération de contenu.

1.4. Contributions

Dans cette thèse, nous examinons d'abord le problème de réplication du contenu dans les réseaux mobiles. Nous étudions l'état de l'art des modèles de mobilité réalistes pour avoir une image de ce qui pourrait être le problème dans un tel contexte. Nous avons ensuite modélisé notre problème comme un problème de localisation des installations. En particulier, nous découvrons qu'il s'agit d'une variante "capacitated" du problème de localisation des installations. Ce constat nous permet de concevoir un mécanisme distribué qui se rapproche des solutions optimales. Nous considérons également le problème de ressources dans réseaux mobiles et notre mécanisme vise à répartir la charge de la réplication de contenu parmi les noeuds par la méthode de "cache-and-forward" et P2P. Enfin, nous analysons les scénarios où les utilisateurs se comportent égoïstement dans la réplication de contenu. La liste récapitulative ci-dessous décrit les contributions de cette thèse :

– Nous faisons une enquête sur des modèles de mobilité et des traces qui sont appropriées pour la simulation des applications du réseau mobile, dans notre cas c'est la réplication de contenu. Nous effectuons également une étude des traces de mobilité exploitées d'un environnement de réseau virtuel (NVE). Dans cette étude, nous construisons un robot dans le NVE Second Life en utilisant le code source ouvert de

libsecondlife pour recueillir des traces de centaines d'utilisateurs au cours de plusieurs jours. Les résultats révèlent que les comportements humains posent un réel problème sur l'extensibilité du réseau mobile du fait que les gens se concentrent habituellement autour des points d'intérêt. Nos traces de mobilité Second Life sont publiquement disponible en ligne.
- Nous mettons en place des mécanismes de "cache-and-forward" pour aider les utilisateurs mobiles à partager la charge de distribution de contenu. Les résultats montrent une bonne performance en termes d'équilibrage de charge.
- Nous modélisons le problème de réplication dans les réseaux mobiles comme le problème de localisation des installations. Nous proposons ensuite la conception d'un mécanisme distribué pour approximer la solution optimale qui réduit la latence de recherche de contenu et évite la congestion au niveau des passerelles mobiles. Pour évaluer sa performance, nous développons une extension pour le simulateur ns-2 qui peut être téléchargée sur demande.
- Nous définissons et étudions un nouveau modèle pour le problème de réplication de contenu dans les réseaux sans fil hétérogènes sous un scénario "flash-crowd". En utilisant la théorie des jeux, nous modélisons ce problème comme un jeu "anti-coordination". Sur la base des conclusions théoriques, nous nous concentrons sur les paramètres de réseau pratique et le facteur de réplication dans un tel contexte. Nous démontrons ensuite la nécessité de la coopération pour améliorer l'efficacité.

1.5. Organisation de la thèse

Cette thèse est organisée comme suit. Dans le chapitre 2 nous présentons le contexte de notre problème et une liste de travaux reliés. Le chapitre 3 décrit le problème de mobilité humaine et la nécessité d'évaluer les performances des applications mobiles contre les modèles de mobilité réalistes. Dans le chapitre 4, nous examinons les mécanismes qui permettent aux utilisateurs de partager la charge de réplication de contenu. Dans le chapitre 5, nous étudions les mécanismes distribués pour répliquer le contenu dans les réseaux mobiles tout en évaluant la performance à travers les lentilles du problème de localisation des installations. Dans le chapitre 6, nous étudions le scénario de réplication lorsque les utilisateurs sont égoïstes et ont tendance à minimiser leurs propres coûts. Dans le chapitre 7, nous résumons les résultats de nos études et présentons les orientations pour les travaux futurs.

2. Etat de l'art et contexte du problème

Dans ce chapitre, nous étudions l'état de l'art de la réplication du contenu dans les réseaux mobiles. Nous rappelons que plusieurs travaux ont été faits jusqu'à présent sur les réseaux mobiles ad hoc (MANETs) et les réseaux tolérants aux retards (DTNs) qui sont difficiles à déployer dans la réalité. Nous étudions ce problème dans un environnement plus pratique : un réseau mobile hétérogène qui combine à la fois la connexion 3G et la communication d'appareil à appareil qui sont maintenant largement soutenue par la plupart des terminaux mobiles. Dans ce type de réseau, nous constatons que la réplication du contenu peut être étudiée à partir d'un point de vue de la localisation des installations.

2.1. Réplication de contenu dans les réseaux mobiles

Les futurs réseaux mobiles auront la capacité de distribuer des contenus pour les applications utilisateurs. Pour éviter la saturation du réseau 3G, la distribution de contenu pourra s'appuyer sur le fait que de nombreux appareils mobiles sont capable de partager du contenu en profitant une connexion Bluetooth ou WiFi pour partager le contenu d'une manière coopérative entre des utilisateurs. Le contenu dans ce cas peut être distribué aux utilisateurs mobiles avec plusieurs façons. La transmission "épidémie" est une méthode de diffuser un contenu à partir d'une ou de plusieurs sources (infectées) à des récepteurs (susceptibles) de contenu lorsque les noeuds mobiles sont en contact. Celui-ci marche bien quand le contenu est très populaire et tout le monde veut le télécharger. Quand le contenu n'est pas si populaire, en raison de contraintes de stockage et d'énergie il est injuste de supposer que les noeuds qui ne s'intéressent pas à ce contenu seront impliqués dans le système de distribution. La solution alternative pour ce problème consiste à répliquer le contenu sur les noeuds intéressés. Nous étudions ce problème à travers une nouvelle approche inspirée par la localisation des installations.

Dans le contexte de notre problème, les utilisateurs mobiles peuvent sélectionner deux façons de récupérer un contenu : soit à partir d'une connexion 3G ou à partir le stockage d'un voisin via la communication d'appareil à appareil. Les noeuds doivent décider leur choix avec ces objectifs :

- Les noeuds devraient limiter le nombre de fois de télécharger (ou de mettre à jour) le contenu pour éviter la congestion causée par l'accès concurrente au réseau cellulaire.
- Le contenu doit être répliqué à un endroit qui minimise la distance (ou le nombre de sauts et la latence, qui sont tous liés dans ce contexte) pour accéder le contenu.
- Les noeuds doivent partager équitablement le rôle de réplication du contenu de telle manière qu'aucun noeud ne pourrait s'épuiser d'énergie avant les autres. Dans ce cas, un noeud peut, par exemple, définir un budget (le nombre d'octets) qu'il est volontaire à servir pour un contenu et limiter le temps de stocker un contenu. La communication d'appareil à appareil peut être multi-sauts, mais nous ne supposons pas l'utilisation d'aucun protocole de routage comme nous l'avons mentionné auparavant : les noeuds qui ne sont pas intéressés par le contenu ne doivent pas être impliqués dans la distribution de contenu. Le transfert de données multi-sauts est assuré par des messages dans la couche d'application si nécessaire.

2.2. Problème de localisation des installations

Comme nous avons souligné dans le chapitre 1, la réplication du contenu avec un ensemble des noeuds mobiles peut être consultée à travers les lentilles de la théorie de la localisation des installations. Dans la recherche opérationnelle, l'optimisation du coût d'un groupe de points de la demande (ou les clients D) pour accéder à un ensemble des services (ou les installations F) est un problème important. Pour résoudre ce problème, nous devrions trouver une solution qui soit efficace en termes de coût, par exemple, le coût de construction et d'exploitation des installations. Ce problème est formulé comme un problème de placer les installations pour répondre à la demande d'un ensemble des clients. Il existe plusieurs exemples, tels que la mise en place des points d'urgence, des centres d'éducation, des stations de transport public. Prenons une entreprise alimentaire qui fournit à ses clients un accès efficace aux établissements de restauration. Il serait préférable d'assurer que tous ses clients ont un restaurant proche de chez eux. Comme l'entreprise paie des coûts importants pour la construction de chaque point de vente, l'ouverture d'un grand

nombre de points de vente peut être extrêmement coûteuses. Idéalement, la société souhaite d'ouvrir un certain nombre de points de vente tels que la distance moyenne de ses clients à leur centre de service est minimale pour un coût de déplacement raisonnable. Le problème de localisation des installations fournit des formulations mathématiques pour des aspects d'optimisation. La formulation souvent contient les coûts d'ouverture des installations et les distances entre les clients et les installations. Ces distances doivent normalement satisfaire les propriétés métriques (par exemple l'inégalité triangulaire). Le coût total dépend du nombre d'installations à ouvrir et de l'emplacement des installations. La solution à un problème de localisation des installations est définie par un ensemble des installations qui seront ouvertes et des sous-ensembles de clients attribués à chaque installation. Cette solution réduit le coût total composé de deux parties : des coûts d'ouverture et des coûts de distance (ou coûts de service).

2.3. Variantes du problème de localisation des installations

Plusieurs différentes variantes du problème de localisation des installations sont obtenues en combinant les coûts de différentes façons : le nombre d'installations ouvertes peut être un constant (problème k-median), ou le nombre de clients servis par une installation est limité (capacitated) ou illimité (uncapcacitated) ou le coût d'ouvrir une installation dépend du nombre de clients qu'elle dessert. La plupart des variantes de localisation des installations sont NP-complet [7] Les algorithmes d'approximation qui calculent donc des solutions proches de la solution optimale sont un sujet à investiguer.

2.4. Algorithmes d'approximation

Le premier algorithme d'approximation pour la localisation des installations a été proposé par Hochbaum [50] basé sur des heuristiques gourmands. Puis plusieurs recherches dans la dernière décennie ont amélioré l'état de l'art de façon spectaculaire. Différentes techniques qui sont largement utilisées pour approximer la solution de localisation des installations ont été proposées, y compris la programmation linéaire (LP-rounding), la technique "primal-dual" et la technique de recherche locale (local search). Nous nous concentrons sur la technique de recherche locale car elle est facile à implémenter en pratique.

2.5. Technique de recherche locale

Des algorithmes d'approximation pour la localisation des installations basées sur la recherche locale sont les plus faciles à comprendre et mettre en oeuvre dans la pratique. Des heuristiques de recherche locale ont été proposés par [71] et ont été largement utilisés par les praticiens. Cette technique consiste en adopter des opérations comme ajouter et supprimer des installations au hasard pour améliorer le coût. La solution obtenue par des techniques de recherche locale est appelée "minimum local" lorsque il n'y a plus d'opération locale qui peut réduire le coût. Le ratio du coût de la pire solution "minimum local" au coût d'optimum global est nommé comme "l'écart de localisation". Korupolu et al. [68] ont montré que l'analyse de la pire solution "minimum local" calculée par ce heuristique est possible et ils ont montré que cette technique peut parvenir à un facteur constant en comparaison avec la solution optimale. Pour certaines variantes de problèmes de localisation des installations, la recherche locale est la seule technique connue pour donner des approximations à un facteur constant.

2.6. Réplication du point de vue de la localisation des installations

Le problème de réplication dans les réseaux mobiles exige que le contenu doive être stocké au nombre minimal de noeuds possible tandis que la localisation du contenu doit satisfaire la demande des utilisateurs intéressés, avec une latence minimale. Avec ces propriétés, nous constatons que la réplication partage la même perspective du problème de localisation des installations. De plus, les appareils mobiles avec des contraintes de ressources (énergie, bande passante ...) ne peuvent servir qu'au plus un nombre limité de voisins. Par conséquent, la variante "capacitated" de la localisation des installations est plus appropriée dans ce cas. La dynamique et la nature décentralisée des réseaux mobiles exigent également une approche souple et distribuée pour l'optimisation des performances. Le problème de réplication de contenu avec la nature dynamique des réseaux mobiles nous introduit de nouveaux défis.

Nous considérons un réseau des utilisateurs coopératifs composé d'un ensemble $V = \{1, ..., N\}$ de noeuds mobiles. Un noeud j souhaitant accéder au contenu d'abord essaie de le récupérer à partir d'autres voisins. Si la recherche échoue, j doit télécharger une réplique de contenu à partir du serveur Internet et temporairement le stocke pour une période de stockage τ. Au cours de cette période, j sert le contenu à tous les noeuds i qui le demandent. Si l'on se réfère l'ensemble des noeuds qui contiennent une réplique de contenu comme $F \subseteq V$ et l'ensemble des noeuds qui consomment le contenu F comme $D \subseteq V$. Nous observons un chevauchement de ces deux ensembles car des noeuds "consommateurs" peuvent également agir en tant que noeuds "répliques" si nécessaire. Le problème est maintenant de déterminer l'ensemble des noeuds $\mathcal{C}$ qui minimisent le coût de répliquer le contenu aux noeuds $j \in \mathcal{C}$ et le coût de distance pour l'accès au contenu des clients $i \in V \setminus \mathcal{C}$.

Soit I l'ensemble des contenus disponibles dans le réseau $I = \{1, ..., M\}$. Chaque contenu a une popularité de contenu représenté par un certain nombre de noeuds qui sont intéressés par le contenu. Le nombre de clients attachés à un noeud de réplique peut dépasser la capacité de ce noeud, et donc peut causer plus de délai. La solution pour résoudre ce problème est de limiter le nombre maximal des clients attachés à un noeud particulier en utilisant la version "capacitated" de la localisation des installations. Étant donné que la capacité de servir du neoud j est au maximum u_j clients, nous avons :

$$C(V, f) = \sum_{\forall j \in \mathcal{C}} \sum_{\forall h \in I} f_j(h) + \sum_{\forall i \in V \setminus \mathcal{C}} \sum_{\forall h \in I} d(i, m(i, h))$$

où $m(i, h) \in \mathcal{C}$ est le noeud j qui stocke le contenu h le plus proche de i. Il y a une contrainte dans le nombre des clients i qui demandent le contenu h attaché à noeud j :

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

Le chevauchement des installations et des clients rend notre problème différent de la traditionnelle localisation des installations de la recherche opérationnelle. Ainsi le choix de l'ensemble des installations $\mathcal{C}$ comme une solution est fait à partir de l'ensemble V et non pas d'un ensemble des candidats distincts prévus F.

2.7. Solution d'approximation distribuée

Moscibroda et al. [86] ont initialement étudié la solution distribuée pour approximer la localisation des installations. En particulier, ils explorent l'enjeu entre la quantité de communication et la qualité d'approximation qui en résulte. Leur algorithme est basé sur une approche "primal-dual" distribuée. Cette technique est cependant compliquée à mettre en pratique en raison du coût de la communication. La technique de recherche locale parvient à une solution d'approximation avec un facteur constant pour la localisation des installations. Notre étude vise à établir un mécanisme qui est entièrement distribué, asynchrone et n'exige que la mesure locale au niveau des noeuds. Il est hors de la portée de cette thèse de fournir les preuves de convergence pour un tel mécanisme pour ces raisons :

- La nature dynamique du réseau mobile peut changer la localisation des noeuds tout moment. Ainsi nous ne pouvons pas avoir un ensemble statique des installations comme une solution à ce problème.
- Nous nous concentrons sur l'objectif d'équilibrage de charge, par conséquent notre mécanisme doit faire face à la mobilité très fréquente. C'est pourquoi nous devons continuer faire échanger le rôle d'installation de noeuds à noeuds. La définition de la convergence ne peut pas être aisément appliquée dans ce cas.

Notre idée est de comparer la solution optimale pour chaque image de la topologie réseau statique contre la solution capturée par notre mécanisme distribué à ce moment .Nous évaluons notre mécanisme en comparant le nombre d'installations ouvertes et leurs emplacements à l'aide du test χ^2 : on compare la répartition des installations contre la distribution proposée par la solution optimale approximée avec la technique que nous développerons dans le chapitre 5.

3. Modèles de mobilité pour les réseaux sans fil

Pour évaluer n'importe quelle application dans les réseaux mobiles, une étape essentielle est d'étudier comment les utilisateurs mobiles se déplacent dans ce contexte d'application. Effectuer des recherches réelles et à grande échelle est habituellement coûteux et peut impliquer trop de participants. En conséquence, presque tous les travaux de recherche dans ce domaine sont basés sur la simulation pour valider les résultats et évaluer les avantages par rapport à d'autres travaux. La simulation avec les réseaux mobiles exige de nombreux paramètres que les chercheurs devraient payer attention, par exemple le modèle de mobilité, la densité du réseau et le protocole de routage. Parmi eux, le modèle de mobilité joue un rôle très important.

Les modèles de mobilité sont conçus pour décrire des habitudes de déplacement des utilisateurs : leur emplacement fréquent, la vitesse et le changement de direction au cours du temps. Afin de simuler une nouvelle application dans les réseaux mobiles, il est nécessaire d'utiliser un modèle de mobilité qui représente avec précision les tendances mobiles des utilisateurs potentiels. Ce n'est dans ce contexte qu'il est possible de déterminer si cette application serait utile lors de mettre en oeuvre et de déployer dans la réalité.

Les performances des applications de réseau mobile changent énormément si l'on change le modèle de mobilité simulé [24]. Choisir un modèle de mobilité pour la simulation des réseaux mobiles est si important qu'il y ait un besoin réel d'étudier des modèles de mobilité et

leur impact sur les performances des applications. Dans la littérature, de nombreux modèles de mobilité ont été proposés pour capturer des caractéristiques différentes de la mobilité et représenter la mobilité d'une manière plus réaliste. Certains d'entre eux représentent les mouvements des noeuds mobiles qui sont indépendants les uns des autres (c'est-à-dire des modèles de mobilité "entité") ou dépendants les uns des autres (par exemple, des modèles de mobilité en groupe [52]). Une autre façon de classer ces modèles est basée sur leur caractère aléatoire [10] : pour certains modèles de mobilité, le choix de vitesse et de direction est totalement aléatoire comme "Random Waypoint", "Random Walk" ou "Random Direction". D'autre part, certains modèles supposent que les mouvements d'un noeud mobile soient affectés par le temps (le modèle Gauss-Markov [75]), par la restriction géographique (des points d'obstacles et d'intérêts [61]), par les objets dans la carte géographique [115] ou par les relations sociales des utilisateurs mobiles [87]. De nombreuses études ont été effectuées sur la base des modèles de mobilité synthétiques qui représentent principalement des mouvements aléatoires. Un des modèles de mobilité les plus fréquemment utilisés dans les simulations est le modèle "Random Waypoint". Dans ce modèle, pour chaque mouvement, un noeud choisit une destination au hasard et une vitesse également au hasard. La simplicité de "Random Waypoint" est une des raisons de son utilisation populaire dans les simulations.

3.1. Problèmes des modèles de mobilité aléatoire

Les modèles de mobilité aléatoires ont plusieurs problèmes. Les auteurs de [24, 10] ont présenté une liste exhaustive de ces problèmes qui peuvent être classés en 4 types :

- Les problèmes transitoires : La décroissance de la vitesse moyenne au cours du temps de simulation et la différence dans la distribution des noeuds entre la phase initiale et la phase stationnaire [127]. Dans le modèle "Random Waypoint", les noeuds sont répartis au hasard dans la zone de simulation alors que pendant la simulation, les noeuds ont tendance à se concentrer vers le centre de la zone de simulation. Ainsi la distribution initiale de noeuds est différente de celle lors du déplacement [16]. On a donc des confusions en interprétant des résultats de la simulation.
- Les problèmes de dépendance temporelle et spatiale : Les modèles aléatoires ne peuvent pas reproduire efficacement caractéristiques de certains scénarios réalistes, lorsque la dépendance temporelle et la dépendance spatiale sont présentes. Les modèles aléatoires supposent que la vitesse et la direction à l'époque actuelle sont indépendantes de l'époque précédente. Ainsi, certains extrêmes comportements de mobilité, telles que l'arrêt, l'accélération et le changement de direction de façon soudaine se produisent fréquemment dans la trace générée par les modèles aléatoires.
- Les problèmes géographiques : Dans les modèles aléatoires, les noeuds mobiles peuvent se déplacer librement dans le domaine de la simulation sans aucune restriction. Ce genre de mouvement omet le fait que les gens se déplacent souvent dans les rues ou les campus universitaires où la zone de déplacement est limitée par des obstacles. Dans un tel contexte, les modèles aléatoires ne peuvent pas représenter certaines caractéristiques de mobilité.
- Les problèmes de liens sociaux : Dans le modèle de mobilité aléatoire, un noeud mobile est considéré comme une entité indépendante qui se déplace librement quelque soit le mouvement d'autres noeuds. Toutefois, dans certains scénarios, y compris le champ de batail ou un environnement de travail (comme dans un bureau, une usine), le motif de déplacement d'un noeud mobile devrait être affecté par le rôle de chaque

noeud et par le groupe de noeuds auquel il appartient.

Leboudec et al. [15] ont étudié la stationnarité de la classe de mobilité aléatoire, par des moyens de calculus Palm et ont proposé le concept de la simulation parfaite utilisant la mobilité "Random Waypoint" [17]. Ils fournissent également un outil pour générer des traces de mobilité pour ns-2 qui est à librement télécharger.

3.2. Modèles de mobilité non-aléatoires

Afin de régler des problèmes dans les modèles aléatoires, plusieurs modèles de mobilité ont été proposés, mais ils sont tous très spécifiques pour des scénarios particuliers. Ces modèles habituellement nécessitent de nombreux paramètres d'entrée. Par conséquent, il n'est pas facile de faire un choix correct des paramètres. Ces modèles de mobilité sont basés sur le mouvement des groupes [52], les cartes graphiques [115], les obstacles [62] ou à base des liens sociaux [87]

3.3. Mobilité humaine

Récemment, les chercheurs concentrent leur attention sur des traces expérimentales de la mobilité humaine, afin d'établir un modèle à partir de ces traces ou de les utiliser directement dans leurs simulations. La communauté de recherche a mis les efforts visant à construire un bibliothèque de traces pour la recherche de réseau mobile (le projet Crawdad [69]). Ces traces sont pour des schémas spécifiques de mobilité qui sont enregistrés par des points d'accès sans fil, par des stations téléphoniques ou par des appareils sans fil portés par les utilisateurs mobiles dans les expériences [26], par exemple, des traces GPS, des traces des dispositifs Bluetooth ("iMotes") qui ont été distribués aux participants pendant des expériences sur le campus de l'Université de Cambridge [55].

Mais ces traces ne peuvent fournir que des informations précises quand elles impliquent un grand nombre des utilisateurs et une suffisamment longue période expérimentale. En outre il existe dans ces traces des problèmes causés par la contrainte énergétique et les dysfonctionnements des appareils mobiles. Un premier tentative par un groupe de chercheurs étaient effectué afin d'abstraire ces traces dans un modèle de mobilité humaine pour générer des mouvements par quelques paramètres d'entrée [97, 74].

En général, les traces ne peuvent pas remplacer les modèles de mobilité pour plusieurs raisons [88] :

- Les traces sont coûteuses à collecter et les grandes traces sont généralement la propriété des sociétés de télécommunication. Ces traces ne sont pas publiques, car elles peuvent être exploitées pour les buts commerciaux.
- Ces traces sont liées à des scénarios très précis donc il est actuellement difficile de les utiliser dans des cas généraux en raison de leur validité.
- De nombreuses traces disponibles ne contiennent pas toutes les données nécessaires pour analyser et caractériser les schémas de mobilité. Ces traces peuvent aussi dépendre de la technique de collecte de données (Bluetooth, Wifi...)

Pour ces raisons, les modèles de mobilité sont encore en cours d'être utilisés pour évaluer les applications des réseaux mobiles.

Cependant, les études des traces ont montré de manière surprenante des communs caractéristiques statistiques : ces traces partagent la même distribution de la durée des contacts et des intervalles “inter-contact”. On a montré que cette distribution suit la loi de puissance suivie d'une coupure exponentielle. Ces résultats ouvrent une nouvelle vague dans la recherche pour caractériser les mouvements humains à partir des traces de la mobilité. Récemment, Rhee at al.[74] ont propose SLAW, un premier modèle se compose de nombreux caractéristiques de la mobilité humaine signalées dans la littérature. SLAW est élaboré et validé par des traces de mobilité GPS avec une durée de 226 jours et 101 participants, essentiellement dans des sites en plein air. Le réseau social est également reflété dans ces traces, car les participants sont des étudiants sur le même campus ou les visiteurs au parc d'attractions. Les expériences sont assez longues pour exprimer la régularité des mouvements quotidiens de l'humain. L'outil pour générer des traces SLAW synthèse est disponible au téléchargement pour le public.

3.4. Loi de puissance dans les traces de mouvement dans le monde virtuel

Nous effectuons également une mesure de la mobilité des utilisateurs [72], mais cette fois dans un environnement virtuel. Nous présentons une nouvelle méthodologie permettant de capter la dynamique spatio-temporelle de la mobilité des utilisateurs. Notre étude exploite la croissance considérable de la popularité du réseau des environnements virtuels (NVEs), dans lequel des milliers d'utilisateurs se connectent quotidiennement pour interagir, jouer, faire des affaires et suivre des cours... Ici nous nous concentrons sur SecondLife (SL) [3], un jeu qui a récemment pris de l'ampleur dans la communauté en ligne. Nous mettons en oeuvre un robot qui se connecte à SL et extrait l'information sur les positions de tous les utilisateurs connectés simultanément à SL. Ce robot est un logiciel client SL personnalisé à partir de libsecondlife [2]. Tenté par la question si notre mesure pourrait donner des résultats similaires à ceux obtenus dans les expériences du monde réel, nous étudions la distribution statistique des contacts de l'utilisateur et nous montrons que d'un point de vue qualitative, la mobilité des utilisateurs dans SecondLife présente des traits semblables à ceux de l'humain en réalité. De plus, nous nous concentrons sur les caractères spatiaux des mouvements et nous observons que les utilisateurs de SecondLife souvent se concentrent autour de plusieurs points d'intérêt. En conclusions, les traces collectées dans cet ouvrage peuvent être très utiles pour les simulations sur la communication des utilisateurs dans les réseaux mobiles et les évaluations de performance.

En conclusion, le choix du modèle de mobilité est très important car il peut influencer de manière significative les performances des applications. Ainsi les performances des applications mobiles doivent être évaluées avec le modèle de mobilité qui correspond le mieux le scénario prévu dans le monde réel. Toutefois, comme la recherche sur les réseaux mobiles est relativement nouvelle, très peu d'études ont été menées afin de comprendre ce qui est un modèle de mobilité réaliste pour l'homme. L'étude des traces de mobilité révèle le fait que les utilisateurs se rassemblent généralement dans certains “points d'intérêt”. En conséquence, il est difficile du point de vue logistique à mettre en place des infrastructures mobiles pour répondre à la demande des utilisateurs. Il faut prendre des mesures pour réduire la congestion à la passerelle mobile surchargée. Notre étude sur la réplication dans les réseaux mobiles a pour but de faire face avec cette réalité. Lors de l'évaluation de la performance nous utilisons des modèles de mobilité différents, y compris la mobilité aléatoire et humaine.

4. Mécanismes de stockage et transfert de contenu dans les réseaux mobiles

Nous nous focalisons sur le problème de l'échange d'information entre les utilisateurs mobiles. L'information est définie comme un morceau de données qui contient un contenu intéressant, par exemple des informations touristiques locales. Dans ce contexte, la plupart des morceaux d'information sont prêts pour une utilisation générale, c'est-à-dire ils sont intéressants à un grand nombre d'utilisateurs. Ainsi une diffusion de type "stocker et transférer" serait souhaitable. Les utilisateurs forment un système coopératif pair-à-pair dont les noeuds peuvent agir simultanément à la fois comme "clients" et "serveurs" pour échanger des données. Les noeuds qui stockent une copie de l'information agissent en tant que des fournisseurs de ce contenu à des noeuds à leur proximité. Pour partager la charge de distribution de contenu (par exemple l'énergie pour transférer les données), les noeuds agissent comme fournisseurs pour un temps limité, avant de remettre cette information à d'autres noeuds.

Nos objectifs sont de proposer et d'analyser une solution répondant à ces questions :

- Le placement des copies d'information doit suivre une distribution référentielle afin de réduire la distance moyenne d'accès à l'information.
- Le rôle de fournisseur d'information peut entraîner une consommation élevée de ressources en termes de bande passante ou d'énergie. Il est donc souhaitable que ce rôle soit partagé assez fréquemment parmi les utilisateurs.

4.1. Mécanismes de diffusion de l'information

Nous étudions le problème de la diffusion de l'information dans un réseau mobile sans fil avec les mécanismes pair-à-pair. Motivé par la nécessité d'une répartition équilibrée de la charge entre les noeuds, nous étudions l'applicabilité des mécanismes de "cache-and-forward" pour diffuser des copies d'informations dans le réseau. Nous laissons des copies d'information à déplacer entre les noeuds en fonction de deux modèles de mobilité bien connus, "Random Walk" [38] et "Random Direction"[89]. Dans notre contexte, une entité mobile n'est pas un noeud de réseau, mais plutôt une copie de l'information qui "saute" entre des noeuds :

- La diffusion "Random Walk" (RWD) : Nous considérons la plus simple diffusion "Random Walk" possible, dans laquelle chaque entité mobile, c'est-à-dire chaque copie de l'information, parcourt dans le réseau en passant d'un noeud à l'autre (le nouveau noeud est sélectionné avec une probabilité égale parmi tous les voisins du noeud précédent). Chaque noeud stocke l'information pour une durée déterminée, puis il continue le processus de transfert.
- La diffusion "Random Direction" (RDD) : Elle implique que chaque entité mobile alterne des périodes de circulation et des périodes d'arrêt. Dans notre contexte, la période d'arrêt correspond au temps pendant lequel la copie d'information est stockée à un noeud fournisseur. La période de circulation commence au moment où le fournisseur actuel envoie la copie à un de ses voisins, et se termine quand le nouveau fournisseur est atteint par la copie de l'information. Le nouveau fournisseur est choisi par la sélection d'une localisation de cible et le plus proche noeud à cet

endroit devient le nouveau fournisseur. Pour le faire, au début d'une période de circulation, le fournisseur actuel sélectionne de façon indépendante la direction et la distance pour le transfert de l'information, ainsi il identifie la destination de cible dont la position est inclue dans le contenu des messages. Ce mécanisme exige que des noeuds soient capable d'estimer leur position (à l'aide de GPS).

4.2. Résultats expérimentaux

Nous utilisons ns-2 pour simuler des noeuds mobiles équipés d'une interface 802.11 avec la vitesse de transmission de données à 11Mbits/s. Les requêtes peuvent traverser un certain nombre maximum de sauts, $h_{max} = 5$. Nous améliorons la propagation des requêtes en adoptant la technique PGB [90] et en utilisant des numéros de séquence dans les messages pour éviter le phénomène de "broadcast storm" [92] et la duplication des requêtes. Dès la réception d'une requête, un fournisseur la répond avec une probabilité qui est inversement proportionnelle au nombre de sauts traversés par le message de requête. Ceci est fait pour atténuer la duplication de service car une requête pourrait atteindre plusieurs fournisseurs. Le temps de simulation est 10000 secondes. En outre, nous supposons un réseau composé de $N = 2000$ noeuds qui sont répartis sur une surface carrée $\mathcal{A}$ de $500 \times 500m^2$. Pour la simplication, nous supposons que tous les noeuds sont intéressés par le contenu. Chaque noeud a une portée de transmission radio de 20m. Pour le mécanisme RDD, les fournisseurs transfèrent le contenu en choisissant aléatoirement les angles qui sont répartis uniformément dans $[0, 2\pi]$, et une distance qui est exponentiellement distribuée sur une valeur moyenne de 100m. Nous étudions deux scénarios : statique et mobile et plusieurs méthodes de placement de noeuds : uniforme et non uniforme. Pour le modèle de mobilité des noeuds, nous utilisons "Random Waypoint" et "Random Trip" [16,17].

Nous définissons les paramètres utilisés dans notre évaluation :

- L'index χ^2 de la distribution de l'information en comparison avec les deux distribution référentielles : l'uniformité spatiale et l'uniformité des noeuds.
- Le temps cumulé de jouer le rôle de fournisseur pour chaque noeud.
- Le nombre des requêtes servies par chaque noeud en tant que fournisseur d'information.
- La distance euclidienne pour chaque noeud d'accéder aux copies d'information.

L'évaluation que nous avons effectués ont montré que, sous une variété de scénarios, y compris les réseaux statiques et mobiles, en dépit de la simplicité et la faible surcharge, les mécanismes RWD et RDD atteignent des objectifs définies dans ce travail. La diffusion de l'information donné par RWD et RDD se rapproche effectivement les distributions référentielles. En termes d'équilibrage de charge, deux stratégies de diffusion ont succédé de répartir uniformément la charge de service parmi les noeuds en particulier lorsque nous étudions le scénario mobile. La mobilité semble être un allié utile, au lieu d'un phénomène problématique dans ce cas. Intuitivement, nous voyons dans les résultats que l'index χ^2 est meilleur quand on a plus de copies d'information. Ce constat provoque une nouvelle question : il nous faut un mécanisme qui permet de déterminer le nombre de copies nécessaires stockés dans le réseau et ce nombre doit s'adapter à la dynamique de la demande. Ce problème est laissé pour le chapitre suivant.

5. Solution distribuée pour la réplication du contenu dans les réseaux mobiles

Dans le chapitre précédent, nous avons introduit le mécanisme de cache et transfert aléatoire pour maintenir la distance d'accès raisonnable du contenu et l'équilibrage de charge entre les noeuds mobiles dans de différents scénarios, en fixant un nombre prédéfini de répliques. A partir de ces résultats, on peut soulever une question : si l'on place plus de répliques dans le réseau, la performance est toujours mieux ? Combien de répliques doivent être placées dans un réseau permettant d'assurer les meilleures performances tout en réduisant au minimum l'utilisation des ressources ? Notre objectif est de résoudre ce problème en proposant un mécanisme de réplication qui permettra d'atteindre le nombre optimal de répliques. En particulier, nous étudions ce problème à travers les lentilles de la théorie de la localisation des installations. Il est intéressant de noter que les problèmes de localisation des installations et ses variantes impliquent des solutions centralisées envers un environnement statique, pendant que nous étudions un mécanisme distribué dont le but est d'approximer la solution optimale pour une topologie de réseau dynamique.

La réplication de contenu a été montrée d'être efficace pour améliorer la performance de l'accès au contenu pour les utilisateurs, en particulier quand il y a un problème de congestion ou d'extensibilité. Il existe la nécessité de décharger le trafic de données de la passerelle 3G aux réseaux sans fils qui connectent les utilisateurs. La performance de la réplication dans les réseaux mobiles est conditionnée par le nombre et les emplacements des répliques de contenu déployées dans les noeuds mobiles. Nous trouvons que ce problème peut être vu comme une version "capacitated" du problème de localisation des installations. L'objectif de ce travail est de concevoir une solution distribuée et légère pour ce problème. De plus, nous visons un mécanisme qui permet de partager la charge de stockage de contenu entre des noeuds afin de s'adapter à la demande de contenu. Nous évaluons notre mécanisme par la simulation, en explorant un large éventail de paramètres et nous étudions aussi des mécanismes réalistes pour accéder au contenu.

5.1. Enoncé du problème

Le problème de la réplication de contenu a été beaucoup étudié pour l'Intenret. Cependant, la nature dynamique des réseaux sans fil introduit de nouveaux défis pour ce problème. Nous explorons dans ce travail le concept de la réplication de contenu pour le réseau sans fil dans un environnement coopérative : nous assumons que les noeuds mobile peuvent potentiellement stocker des données et servir d'autres utilisateurs en utilisant une connexion IEEE 802.11 ou Bluetooth. Nous considérons que le contenu a une durée de validité, après laquelle une nouvelle version doit être téléchargée à partir d'un serveur sur l'Internet. En outre, tous les utilisateurs du réseau peuvent être intéressés à de différents contenus à un moment donné, ainsi une approche d'épidémie [48] qui pousse le contenu à tous les utilisateurs, ne serait pas souhaitable. La réplication de contenu doit répondre à deux questions : étant donné la demande des utilisateurs, combien de répliques de contenu devront être mises à la disposition des noeuds mobiles et à quel emplacement ? Traditionnellement, ce problème a été étudié par les lentilles de la théorie classique de la localisation des installations [85]. Le placement optimal de contenu peut être exprimé par un problème "k-median", alors que l'optimisation conjointe du placement et du nombre de répliques peut être étudiée à travers un problème de localisation des installations "capacitated". Ces

deux optimisations sont NP-difficiles.

Afin de fournir une description de base du système, nous nous concentrons d'abord sur un contenu comme un objet unique de l'information avant d'étendre nos mécanismes à plusieurs contenus. Nous supposons que l'objet est étiqueté avec une durée de validité, et à l'origine hébergé dans un serveur sur l'Internet, qui ne peut être accessible par l'accès cellulaire à large bande. Soit G = (V, E) le graphe du réseau à un moment donné, défini par un ensemble des noeuds mobiles $V = \{1, ..., N\}$ et un ensemble d'arêtes E. Un noeud j souhaitant accéder au contenu essaie d'abord de le récupérer à partir d'autres noeuds voisins. Si la recherche échoue, ce noeud ensuite télécharge une réplique du contenu à partir du serveur sur l'Internet et il le stocke temporairement pour une période de temps τ_j (le temps de stockage). Pour simplifier, nous supposons $\tau_j = \tau$, $\forall j \in V$. Au cours de la période de stockage, j sert le contenu aux noeuds qui émettent des demandes, et éventuellement télécharge à partir du serveur d'Internet une nouvelle copie du contenu lorsque son temps de validité a expiré. Nous supposons que il existe des noeud i, qui à un moment donné t demandent une copie du contenu en envoyant des requêtes à un taux constant λ_i.

Pour atteindre l'équilibrage de charge, à la fin de la durée de stockage j doit décider à transférer le contenu à un autre noeud ou à abandonner la copie ou à répliquer le contenu dans plusieurs noeuds voisins. Nous nous référons aux noeuds hébergeant une copie du contenu à un moment donné instant comme des noeuds de réplique. L'ensemble des noeuds de réplique est dénoté par $\mathcal{C}$. Ensuite, Soit I l'ensemble des contenus $I = \{1, ..., M\}$. Chaque contenu h a une popularité représentée par un pourcentage de nombre de noeuds qui y sont intéressés $p(h)$. Nous designons le coût pour un noeud j de répliquer un contenu h comme $f_j(h)$, le coût total du système peut être réécrit comme suit :

$$C(V, f) = \sum_{\forall j \in \mathcal{C}} \sum_{\forall h \in I} f_j(h) + \sum_{\forall i \in V} \sum_{\forall h \in I} d(i, m(i, h))$$

où $m(i, h) \in \mathcal{C}$ est le noeud le plus proche j avec un replique h pour i et le nombre de clients qui exigent le contenu h attaché à j est limitée par

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

Pour approximer le problème de localisation des installations avec plusieurs contenus (ou commodités) nous transformons le problème comme suit : à partir du graphe $G = (V, E)$ avec N noeuds et chaque noeud i est noté comme (x, y), supposons que nous avons M contenus, on transforme le graphe G en $G'(V', E')$ avec $M \times N$ noeuds, chaque noeud i dans G est maintenant représenté par M "instances virtuelles" dans G', notée par $i(h) = (x, y, h)$, $h = 1..M$. Le coût total du système peut être réécrit comme suit :

$$C(V', f) = \sum_{\forall j(h) \in \mathcal{C}} f_j(h) + \sum_{\forall i(h) \in V'} d(i(h), m(i, h))$$

où $m(i, h) \in \mathcal{C}$ est le noeud de réplique le plus proche $j(h)$ pour $i(h)$ et le nombre de clients $i(h)$ attachés à $j \in V$ est

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

.

5.2. Définition des coûts

– Coût d'ouverture de réplique : Nous considérons ce coût comme la charge d'un noeud de réplique pour télécharger et transférer des données à ses clients. Nous assumons que chaque noeud définit un volume de référence de données qu'il est prêt à servir son voisinage $v_j(h)$:

$$f_j = |\sum_{\forall h \in I} F(h)u_j(h) - v_j|$$

où $F(h)$ est la taille du contenu h et $F(h)u_j(h)$ est le volume de données desservi par j au cours de sa durée de stockage. Nous utilisons cette définition de coût en raison du problème de manque de ressources dans les noeuds mobiles ainsi une charge qui dépasse le volume référentiel est une source d'exhaustion pour les noeuds de réplique. Une autre importante raison est que tous les éléments de ce coût peuvent être mesurés localement au niveau des noeuds donc il est possible de concevoir un mécanisme distribué basé sur ce coût.

– Coût de service : le coût de service concerne la distance pour récupérer le réplique le plus proche pour les clients. Dans notre contexte, la distance peut représenter le délai, le nombre de sauts ou la distance euclidienne. Le délai ne peut toutefois être obtenu qu'en simulant le trafic de réseau réel. Si nous utilisons le nombre de sauts, nous avons besoin calculer ce coût de service tout en examinant tous les chemins possibles entre tous les deux noeuds. Pour simplifier, nous choisissons la distance euclidienne pour cette raison : nous supposons dans notre contexte de l'application un réseau de haute densité, donc il devrait exister un chemin entre deux noeuds avec une distance qui est proche de la plus courte distance euclidienne.

5.3. Mécanisme de placement des répliques

Nous utilisons le mécanisme "Random Walk-Diffusion" (RWD) pour placer des contenu dans le réseau. A la fin du temps de stockage, un noeud de réplique sélectionne avec une probabilité égale un de ses voisins pour transférer le contenu. Ainsi, les répliques de contenu parcourent à travers le réseau d'un noeud à l'autre, au hasard, à chaque période τ. Pour évaluer si le placement de répliques réalisé par notre technique ressemble à la distribution cible calculée par une solution approximée du problème de localisation des installations, nous effectuons le test bien connu χ^2 sur les distances entre les répliques.

5.4. Mécanisme distribué de réplication de contenu

Nous allons maintenant concevoir le mécanisme distribué pour trouver le nombre optimal de répliques à placer dans le réseau. En particulier, nous voulons répondre aux questions suivantes :

– Étant donné un ensemble des points de demande qui présentent un taux homogène des requêtes pour les contenus, quel est le nombre optimal de répliques qui doivent être déployées dans le réseau pour atteindre l'équilibrage de charge ?
– Est-il possible de concevoir un algorithme distribué léger qui approxime ce nombre optimal de répliques en présence d'une demande dynamique et variable ?

Nous abordons ces questions en proposant de simples modifications au mécanisme de placement de contenu. Ce mécanisme est inspiré de l'algorithme d'approximation de recherche locale à partir de [7], qui se compose de 3 opérations pour sélectionner heuristiquement à chaque étape afin de parvenir à la solution optimale : ajouter, supprimer ou tout simplement transférer le contenu. Au cours du temps de stockage τ, le noeud de réplique j mesure le nombre de requêtes qu'il sert $\hat{s}_j(h)$. Lorsque la durée de stockage expire, j compare $\sum_{\forall h \in I} F(h)\hat{s}_j(h)$ à v_j. Les décisions sont prises comme suit :

$$\text{if } \sum_{\forall h \in I} F(h)\hat{s}_j(h) - v_j \begin{cases} > \epsilon & \text{Repliquer le contenu} \\ < -\epsilon & \text{Supprimer le contenu} \\ \text{else} & \text{Transférer le contenu} \end{cases}$$

ϵ est une valeur de tolérance pour éviter des décisions en cas qu'il n'y a que de petits changements dans le volume de données servi et m est le nombre de différents contenues que j s'occupe actuellement.

5.5. Résultats de simulation

Nous avons implémenté le mécanisme de réplication de contenu pour le simulateur ns-2. Dans nos simulations, qui ont duré presque 3 heures (10000s), nous assumons que des noeuds sont équipés d'une interface de standard 802.11, avec la vitesse 54Mbits/s et une portée de transmission radio de 100m. Nous concentrons notre attention sur les réseaux sans fil avec une densité élevée des noeuds : nous plaçons $N = 320$ noeuds sur une surface carrée $\mathcal{A}$ de $1000 \times 1000m^2$. Nous simulons la mobilité des noeuds en utilisant le modèle stationnaire de "Random Waypoint"[16] où la vitesse moyenne des noeuds est fixée à 1 m/s et le temps moyen de repose est fixé à 100 s. Ces paramètres sont représentatifs pour des personnes qui utilisent leur téléphone mobile en marchant. Nous présentons les principaux résultats de notre travail et nous nous concentrons sur le scénario de mobile, mais nous avons aussi des résultats pour un réseau statique qui ont été présenté dans nos travaux précédents [25]. Les résultats montrent que notre mécanisme, qui n'utilise que des mesures locales, approxime de manière extrêmement précise la solution optimale du problème de localisation des installations. De plus, la solution est robuste contre la mobilité des utilisateurs et adaptable à différents taux de demande, différents tailles de contenu et différents modes d'accès au contenu. Dans le chapitre suivant, nous relâchons l'hypothèse d'un réseau de coopération et nous analysons la réplication non coopérative avec des outils de la théorie des jeux.

6. Réplication de contenu dans l'environnement non coopératif

Dans les chapitres précédents, nous avons étudié le problème de réplication dans les réseaux coopératifs. Le facteur de réplication (le nombre de répliques sur les noeuds) est supposé de dépendre du budget dédié par les noeuds en raison de leur contrainte de ressources. Toutefois, les utilisateurs peuvent également se comporter égoïstement, par exemple, ils ne veulent juste de consacrer qu'un budget minimal pour aider le système. Dans ce chapitre, nous définissons et étudions un nouveau modèle pour le problème de réplication dans le réseau sans fil hétérogène avec un scénario "flash-crowd", dans lequel les noeuds pourraient

déterminer le facteur de réplication eux-mêmes. En utilisant la théorie des jeux non coopératifs, nous projetons ce problème de réplication comme un jeu "anti-coordination". Nous commençons par définir l'optimum social dans le cas général et ensuite nous nous concentrons sur un jeu à deux joueurs pour obtenir un aperçu de la conception des stratégies de réplication efficace. Sur la base des conclusions théoriques, nos travaux actuels portent sur l'élaboration de stratégies à mettre en oeuvre dans un cadre de réseau pratique.

6.1. Modélisation du problème

Nous abordons le problème de réplication du contenu dans un réseau sans fil hétérogène : les noeuds mobiles peuvent se connecter à un réseau cellulaire (par exemple, un réseau de haut débit mobile 3G) et sont capables de former un réseau sans fils (par exemple, à l'aide d'une interface 802.11 ou Bluetooth). Nous supposons que le contenu doit être hébergé dans un serveur d'origine dans l'Internet, qui ne peut être consulté que via le réseau cellulaire. Certains noeuds sont capables de télécharger via le réseau cellulaire une nouvelle version du contenu. Ce contenu sera stocké pour servir à d'autres noeuds par la communication direct de noeud à noeud. Dans ce cas, nous supposons un scénario de "flash-crowd", dans lequel les utilisateurs découvrent un nouveau contenu et souhaitent y accéder en même temps. En conséquence, la congestion d'accès n'est pas négligeable. Par souci de simplicité, nous considérons ici un contenu d'information unique.

Supposons que les noeuds veulent accéder un contenu de taille L octets qui est disponible pour le téléchargement. Ce contenu nécessite f mises à jour par seconde à partir du serveur d'origine (afin d'obtenir une nouvelle copie) et chaque mise à jour implique un téléchargement de U octets. Nous définissons maintenant le jeu de réplication. Nous supposons un jeu de déplacement simultané : chaque joueur choisit sa stratégie dans le même temps (sans communication entre les joueurs).

- Soit V l'ensemble des joueurs (ou l'ensemble des noeuds), avec $|V| = N$.
- Soit $\mathcal{S}_i$ l'ensemble de toutes les stratégies possibles pour le joueur $i \in V$. En outre, dénotez $s_i \in \mathcal{S}_i$ comme la stratégie du joueur i, où $s_i = \{1, 0\}$. Ensuite, nous définissons $s = \{s_1, s_2, ..., s_i, ..., s_I\}$ pour un profil de stratégie.

Dans la suite, il sera utile de diviser l'ensemble des joueurs dans deux sous-ensembles. Soit $\mathcal{C} \subseteq V$ dénotant l'ensemble des joueurs dont la stratégie est d'accéder au contenu du serveur d'origine et de le stocker $s_i = 1$, $\forall i \in \mathcal{C}$, et $\mathcal{N} \subseteq V = V \setminus \mathcal{C}$ est l'ensemble des joueurs dont la stratégie est d'accéder à un contenu stocké, is $s_i = 0$, $\forall i \in \mathcal{N}$. Également, laissez $|\mathcal{C}| = x$ et $|\mathcal{N}| = N - x$.

Étant donné un profil de stratégie s, les coûts supportés par le joueur i sont définis comme suit :

$$C_i(s) = \beta_i \mathbb{I}_{s_i=1} + \gamma_i \mathbb{I}_{s_i=0}$$

où :

- β_i est le coût du temps d'accès cellulaire (c'est-à-dire la consommation des ressources 3G) si i obtient le contenu à travers le réseau 3G ;
- γ_i est le coût du temps d'accès sans fils, si i obtient une version stockée du contenu

par la communication de noeud à noeud.
– $\mathbb{1}_{s_i}$ est une fonction indicatrice.

Nous allons maintenant définir précisément les deux termes β_i et γ_i. Nous introduisons les quantités suivantes :

– R_{3G} et R_h sont les débits offerts, respectivement, par le réseau d'accès 3G et le réseau sans fils entre les noeuds.
– T_c est la durée pendant laquelle un noeud $i \in \mathcal{C}$ stocke un contenu.
– h est le nombre moyen de sauts nécessaires pour accéder au contenu le plus proche par la communication de noeud à noeud, en supposant une distribution uniforme de noeuds qui contiennent une réplique de contenu sur $\mathcal{A}$. Formellement, $h = \sqrt{\frac{\mathcal{A}}{x\pi r^2}} = \frac{R}{r}\frac{1}{\sqrt{x}}$

Avec ces définitions, nous pouvons maintenant nous concentrer sur les deux termes de coût, β_i et γ_i. Nous définissons β_i tel que :

$$\beta_i = \Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big]|\mathcal{C}|$$

où le premier terme sur le côté droit de l'équation représente le coût de télécharger le contenu pour la première fois et le deuxième terme pour le coût supplémentaire pour télécharger les mises à jour du contenu. Notez que L/R_{3G} et U/R_{3G} sont les temps d'accès consommés pour télécharger un objet entier ou ses mises à jour, tandis que $T_c f$ est le nombre de mises à jour effectuées par un noeud qui stocke actuellement l'objet. Notez également que cette équation représente la congestion causée par les noeuds qui tentent d'accéder au contenu en même temps : le débit R_{3G} est inversement proportionnel au nombre d'utilisateurs qui simultanément accédent à une seule station 3G [108].

En ce qui concerne γ_i, nous utilisons l'équation suivante :

$$\gamma_i = \Big[h\frac{L}{R_h}\Big]|\mathcal{N}|$$

où hL/R_h est le temps consommé pour accéder à la version actuelle du contenu stocké. Cette équation représente le coût de la congestion créée par l'accès simultané à plusieurs disponibles répliques de contenu par des noeuds utilisant la communication sans fils : le débit R_h est inversement proportionnelle au nombre de noeuds qui accédent à un version stockée de l'information.

Autrement dit, nous exprimons le coût C_i payé par le joueur i comme les coûts d'accès et de mise à jour de l'objet o, qui est défini par β_i si le joueur i choisit d'accéder à l'objet d'origine sur le serveur et le stocker ou par γ_i si le joueur i choisit d'accéder à la plus proche réplique de contenu, à condition qu'au moins un joueur a décidé de répliquer ce contenu.

Nous focalisons notre attention sur les coûts d'accès, en négligeant les coûts d'énergie qu'un noeud de réplique doit supporter pour servir les autres noeuds. Bien que nous reconnaissions cette simplification du problème, nous verrons dans la suite que le jeu qui en

résulte conserve son intérêt.

Le coût social d'un profil donné de stratégie est défini comme le coût total supporté par tous les joueurs :

$$\begin{aligned} C(S) &= \sum_{i\in\mathcal{C}} \Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big]|\mathcal{C}| + \sum_{i\in\mathcal{N}} h\frac{L}{R_h}|\mathcal{N}| \\ &= x^2\Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big] + (N-x)^2\frac{R}{r}\frac{1}{\sqrt{x}}\frac{L}{R_h} \end{aligned}$$

où nous avons remplacé l'expression qui tient compte le nombre moyen de sauts h. Par conséquent, le coût social peut être calculé en utilisant la fonction de fraction de joueurs x qui ont choisi d'agir comme un noeud de réplique. Notez que l'équation ci-dessus illustre un jeu qui appartient à la famille générale des jeux de congestion ou de foule ("crowd").

6.2. Optimum social

L'optimum social, dénommé $C^*(S)$, est le coût minimal social. L'optimum social servira comme une référence permettant de mesurer l'efficacité du coût de la réplication non coopérative. Nous définissons $C^*(S)$ comme suit :

$$C^*(S) = \min_s C(s)$$

L'optimum social peut aussi être réécrit comme une fonction de x

$$\begin{aligned} C^*(x) &= \min_x C(x) \\ &= \min_x \Big\{x^2\Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big] + (N-x)^2\frac{R}{r}\frac{1}{\sqrt{x}}\frac{L}{R_h}\Big\} \end{aligned}$$

6.3. Jeu à deux joueurs

Nous simplifions maintenant le problème que nous discutons ci-dessus, en assumant que seulement deux joueurs (ou noeuds) sont impliqués dans le jeu. Nous réécrivons l'équation pour le coût social comme suit :

$$\begin{aligned} C(x) &= x^2 k_1 + \frac{(N-x)^2}{\sqrt{x}}k_2 \\ k_1 &= \frac{1}{R_{3G}}\Big(L + T_c f U\Big) \\ k_2 &= \frac{1}{R_h}\Big(\frac{R}{r}L\Big) \end{aligned}$$

La version à deux joueurs du jeu de réplication implique deux joueurs, $1, 2$, dont l'ensemble des stratégies est $\{1, 0\}$: 1 implique que joueur i choisit d'aller chercher l'objet o à

partir du serveur d'origine et le stocker, tandis que 0 indique joueur i choisissant d'accéder o à partir d'une réplique. Le tableau suivant montre que lorsque les deux joueurs décident d'accéder o à partir d'une réplique, personne ne peut réellement obtenir o, donc le coût est ∞.

	$s_2 = 1$	$s_2 = 0$
$s_1 = 1$	$\Big(2k_1, 2k_1\Big)$	$\Big(k_1, k_2\Big)$
$s_1 = 0$	$\Big(k_2, k_1\Big)$	(∞, ∞)

Nous réécrivons ce tableau mais en utilisant au lieu des coûts, l'inverse des coûts qui est équivalent au profit des joueurs :

	$s_2 = 1$	$s_2 = 0$
$s_1 = 1$	$\Big(\frac{1}{2k_1}, \frac{1}{2k_1}\Big)$	$\Big(\frac{1}{k_1}, \frac{1}{k_2}\Big)$
$s_1 = 0$	$\Big(\frac{1}{k_2}, \frac{1}{k_1}\Big)$	$(0, 0)$

De toute évidence, la stratégie 0 est strictement dominée par la stratégie 1 si et seulement si $2k1 < k2$: dans ce cas, nous aurions un seul équilibre de Nash (NE), qui est $(1, 1)$. Au lieu de cela, quand $2k1 > k2$, il n'existe pas de manière stricte (ni faiblement) les stratégies dominées. Dans ce cas, nous faisons face à un jeu "anti-coordination", dans lequel il est possible de montrer que la meilleure stratégie pour un joueur sera d'alterner les périodes de réplication et de non-réplication.

Malgré qu'il n'y a pas de stratégie dominée dans notre jeu, il est facile de montrer qu'il y a deux équilibres (NE) qui correspondent à $(0, 1)$ et $(1, 0)$. Il est clair que le choix d'un NE donne plus de faveur à un joueur que l'autre. Nous notons que le jeu n'est pas un jeu à somme nulle (zéro-sum), donc nous ne pouvons pas appliquer directement le théorème minimax.

Il est à noter que ce jeu de réplication n'est pas un "jeu de poulet", ni un "Hawks-Doves" qui sont deux versions bien connues des jeux "anti-coordination".

6.4. Jeu à n-joueurs

Les résultats ci-dessus sont extensibles à un jeu de n-joueurs. Notre recherche actuelle vise à mettre en pratique nos conclusions théoriques, dans les deux directions complémentaires. D'une part, nous notons que, dans le jeu de réplication n-joueurs, un joueur peut calculer sa meilleure stratégie en répondant à celui d'autres s'il est au courant du nombre actuel de répliques x dans le réseau. Si le joueur i réplique le contenu, le coût de jouer 1 est $C(1) = xk_1$ et le coût de jouer 0 est $C(0) = \frac{(N-(x-1))k_2}{\sqrt{x-1}}$. Si le joueur i est en cours de jouer 0, le coût de jouer 1 est $C(1) = (x+1)k_1$ et le coût de jouer 0 est $C(0) = \frac{(N-x)k_2}{\sqrt{x}}$.

L'équilibre x peut être atteint lorsque aucun joueur n'a d'intérêt à changer son stratégie :

$$xk_1 = \frac{(N-(x-1))k_2}{\sqrt{x-1}}$$

Etant donné que dans la pratique la vision globale entre les joueurs ne peut pas être assurée, nous étudions quelle est l'efficacité de notre système lorsque les noeuds essaient d'estimer le nombre actuel de répliques dans le réseau $\hat{x}$. Une telle estimation peut être obtenue soit par les techniques d'échantillonnage aléatoire basées sur le commérage, ou en exploitant des mesures locales sur le nombre de requêtes reçues par chaque noeud qui stocke le contenu. Une question ouverte est comment l'équilibre est sensible à des erreurs d'estimation. D'autre part, nous observons qu'une dispositif de randomisation extérieur pourra améliorer l'efficacité, mais l'équilibre corrélé n'est pas possible lorsque les actions des joueurs ne sont pas simultanées dans un contexte asynchrone. Pour résoudre ce problème, nous pouvons permettre la communication entre les joueurs à travers d'une technique "signalling" qui implique néanmoins des connaissances sur la topologie du réseau.

En conclusion, nous avons proposé un nouveau modèle pour le problème de réplication dans un réseau hétérogène avec le scénario "flash-crowd". Nous avons fourni l'expression du coût social et avons défini un jeu à deux joueurs pour obtenir un aperçu sur la conception des stratégies de réplication efficace. Les résultats ont montré que notre problème peut être vu comme un jeu "anti-coordination" dans lequel l'utilisateur peuvent augmenter son profit en choisissant la stratégie inverse de celle d'autres utilisateurs. Nous avons effectué une analyse numérique du nombre de répliques avec différents débits de 3G et de réseau sans fils et nous avons montré la nécessité de la communication entre les noeuds pour améliorer l'efficacité.

7. Conclusion

Avec les avancements de la nouvelle technologie sans fil, les terminaux mobiles ont été largement utilisés dans la vie quotidienne comme des équipements multi-fonctionnels pour la communication et le divertissement. Des applications de réseau ont besoin de données en tant que l'entrée pour fournir des informations aux utilisateurs. Le trafic de données par des utilisateurs mobiles qui cherchent le contenu sur Internet a déjà surchargé les réseaux "backbone" des opérateurs mobiles. Les utilisateurs mobiles font face maintenant à la congestion au niveau des passerelles de réseau. La distribution de contenu aux utilisateurs mobiles d'une manière efficace avec un court délai est un problème difficile compte tenu de la nature dynamique de la mobilité et des comportements humains. Dans cette thèse, nous avons abordé le problème de la distribution de contenu dans les réseaux mobiles hétérogènes. Dans ce type de réseau, la connexion cellulaire et la communication d'appareil à appareil peuvent se compléter mutuellement. L'utilisation de la dernière technique peut fournir plus de ressources pour le système global, mais la disponibilité du contenu et le délai pour l'accès au contenu devraient être améliorés. En revanche, le réseau cellulaire est soumis aux problèmes d'extensibilité et de congestion. Si le contenu est très populaire, la communication d'appareil à appareil permet d'éliminer les goulots d'étranglement à la passerelle cellulaire en déchargeant partiellement la distribution de contenu aux utilisateurs mobiles à l'aide des techniques P2P. La réplication de contenu dans ce contexte a été

prouvée comme une bonne solution pour améliorer les performances et l'extensibilité du réseau. Toutefois, la procédure de décider le nombre des répliques à stocker et à quel endroit n'est pas négligeable dans les réseaux mobiles pour les raisons suivantes :

- La topologie du réseau dans ce cas est supposée de changer rapidement à cause de la mobilité et les noeuds ne peuvent pas s'appuyer sur une infrastructure centralisée pour avoir une vue globale du réseau.
- Les terminaux sans fil ont généralement beaucoup de contraintes de ressources sur eux donc il nous faut un mécanisme de réplication qui maintient l'équilibrage de charge. En outre, les utilisateurs peuvent se comporter égoïstement au moment de décider de répliquer le contenu. En identifiant toutes ces problèmes, nous avons d'abord décrit le problème de réplication de contenu dans les réseaux mobiles. Nous avons étudié l'état de l'art des modèles de mobilité réalistes à venir avec une bonne définition du problème et identifier les modèles que nous pouvons utiliser pour évaluer les performances du réseau mobile. Nous avons ensuite modélisé notre problème comme un problème de localisation des installations. En particulier, c'est une variante "capacitated" du problème de localisation des installations. Ce problème nous a aidés à la conception du mécanisme distribué qui approxime la solution optimale et achève l'équilibrage de charge et le raccourcissement de latence. Nous avons également examiné le problème de contraintes de ressources dans le réseau mobile et nous avons proposé un mécanisme visant à répartir la charge de réplication de contenu. Enfin nous avons analysé les scénarios lorsque les utilisateurs se comportent égoïstement dans la réplication de contenu.

En conclusion, nous avons travaillé sur un mécanisme léger et distribué pour répliquer le contenu dans les réseaux mobiles hétérogènes. Une exploration des paramètres que nous avons utilisé est nécessaire pour évaluer les performances, en particulier dans le cas lorsque les utilisateurs ont des budgets différents pour la réplication de contenu. L'analyse de la performance de notre système de réplication en utilisant de différentes techniques sans fil (Bluetooth, 802.11 ...) pourrait certainement apporter des informations plus détaillées pour une application en réalité. Le facteur de réplication dans notre mécanisme courant dépend du budget consacré par les noeuds mobiles tandis qu'il est intéressant d'étudier si les utilisateurs peuvent sélectionner un budget flexible qui s'adapte aux conditions du réseau. Pour relâcher l'hypothèse d'un réseau de coopération, nous avons analysé la réplication égoïste avec des outils de la théorie des jeux. D'après les résultats théoriques, nos futurs travaux se concentreront sur la conception de stratégies à mettre en oeuvre dans un réseau en pratique. La conception de mécanismes d'incitation pour un tel système peut également être un sujet d'étude en vue de construire un application déployable pour les terminaux mobiles. En outre, un mécanisme renforcé pour protéger le système contre les resquilleurs est un sujet important dans cette direction de recherche. Les mécanismes de sécurité pour protéger la confidentialité des utilisateurs et éviter toutes les exploitation et attaques possibles sont également critiques pour ce genre d'application. Le chiffrement de données et l'authentification devraient être introduites pour assurer efficacement la confidentialité et d'intégrité du contenu et de protéger les utilisateurs de la manipulation de l'information par des malveillantes.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The proliferation of mobile devices and network-based services nowadays has raised a timely question on how to efficiently distribute content to mobile users. In third generation (3G) wireless networks, the practical approach to data dissemination is to deploy a centralized content provider who sends data directly to users via 3G connections. In this way, the requests for some popular data items can exceed the provisioned infrastructure. Fortunately, wireless technologies have evolved a lot and devices equipped with low cost wireless connections like Bluetooth or 802.11 are invading the market. In heterogeneous mobile wireless networks, cellular networks and device-to-device communication can complement each other, similar to the model of datacenters clouds and P2P systems in the Internet. The use of device-to-device communication may provide more aggregate resources to the system (e.g. bandwidth), but it may cause more latency and require more effort in content lookup. Contrarily, cellular networks is subjected to scalability and congestion issues. If the content is popular, device-to-device communication can eliminate the cellular bottleneck by offloading partially the distribution to mobile users : e.g. some nodes can replicate contents in their local cache to serve other nodes that could query the contents later. For example, to deliver a newspaper to mobile users, with cellular networks every user would download an independent amount data. With the help of device-to-device connection, only a fraction of users need to download the data and then redistribute to other users using P2P techniques.

Network applications need data as an input to process and provide information to users. Several studies have reported that mobile data traffic exerted by mobile devices fetching content from the Internet is already a drainage of mobile operators' network resources [4, 5, 78]. Similar to the wired Internet, mobile users are now coping with the congestion at network gateway. To deal with this problem, content replication has been proved to be a good solution to enhance network performance and scalability.

However, the decision procedure to select which content to replicate at which node is not trivial in mobile networks. The network topology in this case is supposed to change rapidly due to mobility and nodes can not rely on any centralized infrastructure to have a global view of the network. Therefore a low overhead solution is required in this context. Moreover, wireless devices usually have strict energy constraint and users can behave selfishly when

deciding to replicate the content.

1.1 Content replication in the Internet

Content replication solutions in the Internet are mostly centralized since they have a server which users can send queries to and this server dispatches the queries to replicated caches. An example for this case is the content distribution service from Akamai [1]. They use a DNS infrastructure to distribute the queries to content caches. However, it is very difficult to provision the infrastructure to adapt the need of mobile users due to the unpredictability in network topology changes caused by mobility.

Other systems to distribute content in the Internet are decentralized and have no central server. In the decentralized approach, there are two main design options : In the first option, replica servers form a structured overlay such that users' queries can be routed to a server thanks to a hash table implemented at each node that contains the location of the content. This approach is reported as being not robust to high dynamic networks in which the rapid evolution of network topology triggers too much overhead. In the second option, the network is built in an unstructured manner and hosts use a flooding technique or gossiping (random probing) to find the content. The latter approach is more convenient for mobile networks. But because of the constraint in energy and bandwidth, the flooding or gossiping technique should be well designed to cope with these issues. Furthermore, to facilitate content lookup in an unstructured system, we need to replicate data in an efficient way to improve the performance.

1.2 Content replication in mobile networks

In wireless networks, one faces the problem of reliability, bandwidth, and interference. Mobility may cause another issue as node links become unstable and the network is partitioned in an unpreditable way. In certain conditions, we cannot count on any infrastructure (like in mobile ad hoc networks - MANETs and delay tolerant networks - DTNs). Mobile devices are usually small and light equipments with limited resources (battery power, limited radio range). Consequently if an application context requires the cooperation from users, some users may behave selfishly to save, for example, their battery life. This context introduces a new class of problem for replication in mobile networks.

1.2.1 Mobility

Due to frequent topology changes, network partitioning and disruption occur more often in mobile networks than in wired networks.

Network partitioning severely reduces data availability when the node that holds the desired data is not in the same partition as client nodes. Replicating data in future separate partitions before the occurrence of network partitioning can improve data availability. Content redundancy can also increase the chance for nodes to find the closest content while

moving. Therefore the replication mechanism should consider all these dynamic natures of mobile network in order to replicate data items beforehand.

The study of human mobility is nowadays a topic that attracts attention of many researchers. There are many proposed mobility models and traces to evaluate mobile networks performance. The traces however are not very helpful in simulating mobile networks since the number of participants is not high and the experiment duration is not sufficiently long. Hence they only help to design and validate mobility models. Many mobility models are mainly built from random movements. Recently, some preliminary attempts have been made to propose more human-like models. It is reported that some mobility models help improving network performance while this is not the case in real human life [24]. A careful choice of appropriate mobility model is necessary to understand the real problem in a particular network setting and to evaluate the performance of a content distribution mechanism.

1.2.2 Energy constraints and load balancing

Mobile nodes operate on batteries which are assumed to have limited capacity despite the advance in battery technology. A single node may serve many clients, which causes its power to be exhausted very quickly. To improve data availability, the replication mechnism should replicate the data items to share the content providing tasks to other nodes and prevent some nodes from energy exhaustion. Moreover, it should also replicate data in such a way that the power consumption of nodes is reduced and is balanced among the nodes that are in the network. In this case an approach that embraces the peer-to-peer (P2P) paradigm (i.e. no role is pre-assigned to a node, every node can be either a client or a server alternatively) could help solving the problem. In our case, we should use the unstructured version of P2P design to cope with the high dynamic nature of mobile networks.

1.2.3 Content availability

Many mobile networks may involve large populations with thousands of nodes, for example, in a crowded scenario like at a stadium or in a museum. To lookup content in such dense and large network, a query sent by a client node may need to traverse a long path to reach a replica, therefore increasing the query cost and latency. Moreover, the existence of a large number of querying nodes may cause more channel interference among clients, which thus decreases considerably the available bandwidth and increases channel access delay. High node mobility may also affect the availabilty of content. The replication scheme should be designed in such a way that its performance will not be greatly affected by the large number of nodes and high mobility.

1.2.4 Selfish peers

Mobile users are aware of the energy constraint and the cost to download data using 3G. Given this fact, one can predict that users will behave selfishly to minimize their own cost and do not care about the system cost unless they are provided incentives to replicate

the content. If an user select to replicate the content, she pays the cost to download it using cellular networks and to provide it to other users upon request. On other hand, an user can select retrieve the content using device-to-device communication. In the current network setting, the latter solution could be much cheaper and users may tend to replicate less than necessary. The total cost computed at the Nash equilibrium in this case can exceed the optimal cost by a large gap. The system thus should discourage potential selfish behaviors by designing a mechanism that motivates users to store the data if this allows to improve the performance and reduce total cost.

1.3 Research objectives

Heterogeneous mobile networks raise new challenges for content distribution due to the rapid growth in number of users and the dynamics of human behaviors. These networks are large in number of hosts and highly unpredictable. This introduces more difficulty for provisionning supported infrastructures, hence there is an issue of scalability for service providers. In this context, the use of device-to-device communication can be a solution to avoid the congestion at mobile gateways. However since mobile devices have constraints in their ressources (battery life, bandwith...), these issues should also be taken into consideration. Cooperation among users to replicate and distribute contents via device-to-device communication in such way to reduce latency and avoid congestion at gateways is highly appreciated. Therefore our objective is to design an efficient mechanism that works in this cooperative condition.

Another issue in this application context is that since there are constraints in mobile devices, it is rational to assume that users will behave selfishly, hence we should focus on the development of strategies that can be implemented in a practical network setting.

1.3.1 Content distribution in heterogeneous mobile networks

In heterogeneous mobile networks, content can be delivered to users either via device-to-device communication or from a 3G connection. If a content is very popular and every users want to fetch it, a content distribution scheme using epidemic forwarding, e.g nodes just look for content when they are in contact range, should be useful due to the following reasons :

- Every user is interested in the content, hence the availability of content is high. This can reduce the delay to download the content and the effort to look up for it.
- There is a congestion problem at the 3G service provider gateway if every user try to fetch the content from 3G.

In contrast, if only a few of users are interested in the content , there would be no congestion for users to download directly from the Internet by using 3G. The interesting problem comes when there are contents whose popularity is not high but is not as low as the congestion problem can be neglected. In this context, a replication scheme can be useful due to the following reasons :

– Replication helps increasing the availability of content. This reduces the delay in looking up and retrieving contents, hence encouraging users to switch their choice to use device-to-device communication.
– Replication helps reducing the concurrent number of downloads from the Internet, hence alleviating the congestion at 3G gateways and increasing network scalability.

For the replication scheme in mobile networks, we need an efficient design to place the content replica where the content demand is. Furthermore, since the client-server model is not applicable is this case, we need a P2P mechanism to dynamically distribute the replica role to users.

1.3.2 P2P mechanisms

To keep the load balanced among users' devices we need a mechanism to share the burden of content replication. This mechanism should be distributed, with low overhead and no requirement of a global view to match the unstructured nature of heterogeneous mobile networks. Human mobility may change the network topology very frequently hence the designed mechanism needs to be efficient in dealing with highly dynamic environment. A P2P mechanism that is based only on random peer selection would be a good candidate in this context. We aim to study random content hand-over mechanisms and their performance to see if such a solution can be deployed in practice.

1.3.3 Optimization in content replication

Replication mechanism in mobile networks should be done in such a way that enhances content availability and reduces content retrieval latency. To do this, the problem is to find the number of replicas needed in the network and the locations to place these replicas. Given the network topology, this problem can be studied through the lenses of facility location theory. Since facility location problems are NP-hard, we need a distributed mechanism to approximate the solution in the conditions that only local information is available.

Given the problems mentioned above, our work aims at finding a solution for content replication that matches the dynamic nature of mobile networks. We focus particularly on a lightweight and practical mechanism that is efficient and based only on local measurements in order to keep low overhead, while achieving good performance in terms of load balancing and content retrieval delay.

1.4 Contributions

In this thesis, we first discuss the problem of content replication in mobile networks. We study the state of the art of realistic mobility models to have an image of what could be the problem in such context. We then cast our problem as a facility location problem. In particular, we find out that this is a capacitated variant of facility location problem. This finding helps us to design a distributed mechanism that approximates well optimal solutions

according to our objective metrics. We also consider the problem of resource constraints in mobile networks and our mechanism aims at distributing the burden of content replication while balancing the load among nodes by P2P cache-and-forward schemes. Finally we analyze the subsequent scenario where users behave selfishly in content replication.

The following is a summary list of the contributions of this thesis :

- We make a survey on mobility models and traces that are appropriate to use in simulation mobile network application, particularly in our content replication context. We also conduct a mobility trace measurement and analysis in a Network Virtual Environment (NVE). To do that, we build a crawler for *Second Life* NVE using the available open source from *libsecondlife* to collect traces of hundred users during several days. The results reveal that human behaviors pose a real problem on mobile network scalability as people usually concentrate around points of interest. Our traces and crawler is publicly available online.
- We introduce cache-and forward mechanisms that help mobile users to share the burden of content distribution. The results show good performance in terms of load balancing.
- We cast the problem of replication in mobile networks as a capacitated facility location problem. We then design a distributed and low overhead mechanisms to approximate the optimal solution that reduces content retrieval latency and avoids congestion at mobile gateways. To evaluate its performance, we develop an extension for *ns-2* simulator which can be downloaded upon request.
- We define and study a new model for the caching problem in heterogeneous wireless networks under a flash-crowd scenario. Using non-cooperative game theory, we cast this caching problem as an anti-coordination game. Based on the theoretical findings, we focus on the practical network settings and the replication factor in such condition. We point out that there is the need of cooperation to improve efficiency.

1.5 Thesis organization

The remaining of this thesis is organized as follows. In the next chapter we introduce our problem background and present a list of related works. Chapter 3 describes the problem caused by human mobility and the need to evaluate mobile application performance against realistic mobility model. In Chapter 4 we examine the mechanisms that allow users to share the burden of storing content. In Chapter 5 we study the distributed mechanisms to replicate content in mobile networks while evaluating the performance through the lens of facility location problem. In Chapter 6 we study the replication scenario when users are selfish and tend to minimize their own cost. In Chapter 7, we summarize the results of our study and outline directions for future work.

Chapter 2

Background and problem statement

In this chapter we study the state of the art of content replication in mobile networks. We point out that several works have been done so far while considering mobile ad hoc networks (MANETs) and delay tolerant networks (DTNs) which are hard to be deployed in reality. We, on the other hand, study the problem in a more practical environment : a heterogenous mobile network that combines both 3G connection and device-to-device communication which are now widely supported by most of mobile devices. In such a kind of network, we find that content replication can be viewed from a facility location perspective. We present the facility location problem variants together with the approximation algorithms and show that it is applicable to build a distributed mechanism inspired by such algorithms for the replication problem in mobile networks.

The remainder of this chapter is organized as follows : Sec.2.1 provides an overview on state of the art in mobile replication and the context of our study. Sec. 2.2, 2.3 and 2.4 gives a brief background on the facility location problem and its approximation solutions. Sec.2.5 describes the problem of replication from a facility location perspective.

2.1 Content replication in mobile networks

With the proliferation of mobile devices, more and more data should be delivered to mobile users. Future mobile networks will have the capability to support content distribution to meet the need of data input for users applications. In this context, there is a need to design a content distribution model that matches the current mobile infrastructures. Such scheme should rely on the fact that many mobile devices could share content based on device-to-device communication using Bluetooth or WiFi connections. The resulting mobile content distribution model may reduce the time to obtain new content and also reduce the workload hence congestion at mobile gateways. The usual solution for this content distribution scheme is to replicate the content in a cooperative manner at users' devices.

Content can be distributed to mobile users in several ways. Epidemic forwarding is a content distribution scheme to spread content from one or more source (infected) nodes to

a group of content receivers (susceptible) when nodes of each type are in communication range. Because of storage or energy constraints, it is unfair to assume that nodes which are not interested in that content will be involved in the distribution scheme. The alternative solution to this issue is to replicate content at interested nodes. One way to study this problem is through a new approach inspired by the facility location perspective.

2.1.1 Problem statement

Recent studies in this field mainly focus their efforts on replication in mobile ad hoc networks (MANETs) and delay tolerant networks (DTNs). In MANETs, there is no centralized infrastructure and routing protocols can allow multi-hop communication. However it is not guaranteed that there is a path between two arbitrary nodes since network may experience disruption. Nodes thus replicate data in order to enhance the content availability and to deal with disruption and routing issues. In DTNs, there is no centralized infrastructure and only single-hop communication is allowed. However, nodes can store and forward messages to the destination with a tolerated delay. In this case nodes should replicate the content to minimize the delay and storage time.

We study our problem in a more practical environment. We consider a heterogeneous mobile network where nodes have both cellular access and device-to-device communication capability. Hence nodes can select 2 ways to retrieve a content : either from the cellular networks or from a neighbor storing that content (device-to-device networks). In such context, nodes should decide their strategy with these objectives :

- Nodes should limit the number of times to download (or to update) the content to avoid congestion in accessing cellular networks.
- Content should be replicated at a location that minimizes the distance (or hop count and delay, which are all related in this context) to retrieve the content with device-to-device communication.
- Nodes should equally share the role of content replication in such manner that no node could run out of energy before others. In this case a node can, for example, define a budget as number of bytes it is willing to serve for a content and limit the time to hold a content.

The device-to-device communication can be multi-hop but we do not assume to use any MANET-like routing protocol since as we mentioned before : nodes which are not interested in the content should not be involved in the content distribution. The multi-hop connection is guaranteed by messages at application layer if needed.

2.1.2 Related works

Epidemic technique

The epidemic content dissemination has been explored by several studies. In [66], authors showed that information dissemination by a simple epidemic algorithm, while assuming a random waypoint movement model, can be represented by a deterministic epidemic model characterized by a parameter : the infection rate. Authors established an analytical

expression for the infection rate based node density, which is an important factor for the performance of information dissemination strategies.

In [105] authors proposed an epidemic algorithm for collecting information in a hybrid network consisting of mobile nodes and fixed infostations. Their architecture, called as shared wireless infostation model (SWIM), actively transfers information among wireless nodes upon each contact, until information is unloaded to one of the infostations. They assumed unlimited buffers and only consider the spreading of a single data item.

In [77], the authors studied the epidemic forwarding with a finite buffer size and multiple items. Their approach represents the spread of multiple data items, finite buffer capacity at mobile devices and a least recently used (LRU) buffer replacement scheme. Using the introduced modeling approach, they analyzed the seven degrees of separation (7DS), one of well-known approaches for implementing P2P data sharing in a MANET using epidemic forwarding and provide many insights for optimizing the design.

Despite of numerous studies of epidemic dissemination in mobile networks, *none of them has been done yet while considering the content popularity.* In [66], the infection rate has been considered but is assumed to be homogeneous for all nodes.

Replication in MANETs

To improve data accessibility in ad hoc networks, in [45, 46], authors proposed methods of replicating data items in MANETs by considering the data access frequencies from mobile hosts to each data item and the stability of radio links among mobile hosts. They proposed three techniques to improve data accessibility in a MANET environment : "static access frequency", "dynamic access frequency" and "neighborhood and dynamic connectivity based grouping". These techniques are proposed under several assumptions :

- Each data item and each mobile host has a unique identifier in the network.
- Every mobile host has finite memory space to store replicas.
- The system requires no update transactions.
- The access frequency of each data item by mobile host is known and does not change. The decision of item replication is based on the data items access frequencies during a constant relocation period.

In [107] the authors proposed a system that allocates and maintains replicas based on the location of mobile nodes. Mobile nodes can communicate to obtain information in a peer-to-peer fashion. The technique assumes the availability of position information of all mobile hosts which requires GPS support at each node. The probability of data access by a given node is defined by the distance between the node and the location where the data was generated (i.e. mobile hosts want to access data relevant to their location). In such context, authors suggested that to improve data accessibility, replicas should be distributed across the network and be far away from each other. However, in order to maintain good performance in MANETs, they also have to ensure that the replicas are not separated by more than R hops. This technique however, does not aim to save the power of mobile hosts and deal with disconnection or partitioning. The overhead of exchanged information in this case is high.

Luo et al. [79] proposed a collection of protocols (PAN probabilistic quorum system for

ad hoc networks) that use a gossip-based multicast protocol to probabilistically disseminate data in a quorum system to achieve high reliability even when there are large concurrent update and query transactions. According to them, the unpredictability of mobile networks makes probabilistic protocols very appealing for such environments. The overhead of exchanged information in a probabilistic quorum system is still very high.

Wang and Li [121] used a sequential clustering algorithm to identify network partitions and disconnection of mobile nodes, while assuming a system based on the reference velocity group mobility model. Each partition is then provided with a parent or a child server to offer services. However, this technique did not address the issue of power limitation of mobile hosts.

Thanedar et al. [114] proposed a replication scheme, called Expanding Ring Replication (ERR) that combines the push-based and the pull-based data delivery approaches. In the pull-based approach, when a node wants to access some data items, it broadcasts to its neighbors an "interest advertisement" containing a description of the data items required. If the request is not satisfied within a given time period, the node initiates an information request to the server. In the push-based approach, the data server measures the frequency of requests f_h for each data item h. If f_h exceeds a threshold value t_h set by the server, the server decides to replicate the data on one or more capable nodes in the network. To replicate the data in the network, the server probes nodes j at k hops away in the network, soliciting their capabilities to replicate data items. The pull-based technique requires more overhead, while the push-based technique requires an appropriate setting of threshold value.

In [131], a set of mobile nodes are grouped into clusters which are defined as a set of stable links. There is at least one stable path between any pair of nodes in a cluster (a path is composed by one or many links between source and destination nodes). A path is said to be stable if the product of connectivity probability of the links that compose the path is higher than a threshold. Every cluster head maintains states of all other cluster heads in the networks. When a node requests to access a data item, the node broadcasts the access request in the whole of cluster K that it belongs to. If there are some replicas of the data item in the cluster, the closest replica node serves the access request. If there is no replica for the requested data object in K, the request is propagated from the cluster head of K to all other cluster heads. If there is replica in some cluster $K^{'}$, the cluster head of $K^{'}$ sends the data to the cluster head of K, and the cluster head of K forwards the data to the requesting nodes. The node which has requested the data item is chosen to be a replica for this data item. Again this approach requires many messages exchanged within clusters.

In [41], authors proposed a cooperative information cache mechanism based on passive local measurement to enhance the diversity of information and retrieving latency. This mechanism is low overhead and fully distributed. This work however did not consider the fairness among caching nodes.

In [93] authors casted the problem of energy-aware cache placement in MANETs as a facility location problem. They designed caching strategies that optimally tradeoff between energy consumption and access latency. Authors devised a polynomial time algorithm which provides a sub-optimal solution and can be implemented in a distributed and asynchronous manner. In the case of a tree topology, the algorithm gives the optimal solution. Given an

arbitrary topology, it finds a feasible solution with an objective function value within a factor of 6 of the optimal solution.

In [42], authors introduced the optimal configuration of wireless sensor network as facility location problem and tried to solve it using a distributed algorithm to approximate the optimal solution. This approach however requires a lot of messages exchanged among sensor nodes and does not deal well with mobility.

Replication in DTNs

Recently, Chaintreau et al. [96] studied the problem of optimal content replication for delay tolerant networks. This work assumed that the storage at mobile node is limited and they focused more on the cache eviction than the location and availability of content. In [57], authors proposed a distributed mechanism to update and replace cache contents for mobile users to reduce content access delay via device-to-device communication. This work focused on the frequency of users to meet and not the location of users.

2.2 Facility location problem

As we pointed out in Chapter 1, content replication with a set of mobile nodes can be explored through the lenses of facility location theory. In operations research, optimizing the cost of a group of demand points (or clients D) to access a set of services (or facilities F) is an important problem. To solve this problem, we should find a solution that is efficient in terms of effective cost, e.g. cost to build and to operate the facilities. This problem is formulated as a "facility location problem" in which many facilities are set up, each of which is assigned with the demands of a subset of clients. Examples of facility location problem are several, e.g. essential services such as emergency points, education centers, public transport stations and retail services. Consider a retail company aiming to provide its customer with efficient access to food outlets. It would be preferable to ensure that, all its customers have a nearby outlet. Since the company incurs significant cost in building up each outlet, opening a large number of outlets may be prohibitively costly. Ideally, the company would like to open a number of outlets such that the average distance of its customers to their nearest service facility is minimum for a reasonable cost.

Facility location problems provide mathematical formulations of optimization aspects of these issues. The formulation consists of the cost to open facilities and the distances between the clients and facilities. These distances normally satisfy metric properties,e.g the triangular inequality. The total cost depends on the number of facilities to open and the location of facilities. A solution to a facility location problem is specified by a set of facilities to be opened and an assignment of the clients to the open facilities. The sum of costs to open the facilities is called as facility cost (or opening cost), and the sum of distances of each client to the facility it is assigned to is called as service cost (or distance cost) of the solution. The set of facilities F and the set of clients D are the inputs for a typical facility location problem. The output solution is to open a subset of the facilities $\mathcal{C} \subseteq F$ and assign clients to their closest open facility. This solution minimizes the total cost consisting of two parts *opening cost* and *distance cost* :

Opening cost : The opening cost for a facility j denoted by f_j depends on the problem we are targeting. For a given solution, the sum of cost for all open facilities is its opening (or facility) cost.

Distance cost : The distance from a client $j \in D$ to a facility $i \in F$ is denoted by $d(i, j)$. This distance is assumed to be symmetric and satisfies triangle inequalities. If a client j is assigned to its closest facility i, then the distance $d(i, j)$ is the distance cost of client j. The sum of distance costs of all clients is the distance (or service) cost of the solution.

2.3 Facility location variants

Different variants of the facility location problem are obtained by combining these costs in different ways : the number of facilities to open can be a constant (k-median problem), the number of clients served by a facility is limited (capacitated facility location problem) or unlimited (uncapacitated facility location problem) or the cost to open a facility depends on the number of clients it serves. Most variants of facility location are NP-Complete [7] hence approximation algorithms which compute solutions close to the optimal solution are under investigation.

2.3.1 k-median problem

This problem is motivated by scenarios in which a limited budget is available for opening the facilities and the cost of all the facilities are roughly the same. The solution consists of the choice of k facilities to minimize the distance cost : $\forall j \in F$ select up to k facilities so as to minimize the cost $C(F, D, k)$:

$$C(F, D, k) = \sum_{\forall i \in D} d(i, m(i)) \tag{2.1}$$

where $m(i) \in \mathcal{C}$ is the facility j *closest* to i.

2.3.2 Uncapacitated facility location

In case we have a cost to open facility and the number of facilities to open depends on a joint optimization for opening cost and distance cost, we have the uncapacitated facility location problem. The solution is to open a set of facilities $\mathcal{C}$ to minimize the joint cost $C(F, D, f)$ of opening the facilities and serving the demand while ensuring that each facility j can serve an unlimited number of clients :

$$C(F, D, f) = \sum_{\forall j \in \mathcal{C}} f_j + \sum_{\forall i \in D} d(i, m(i)) \tag{2.2}$$

where $m(i) \in \mathcal{C}$ is the facility j *closest* to i.

2.3.3 Capacitated facility location

The uncapacitated facility location ignores the fact that the cost of a facility could depend on the number of clients it serves. In its capacitated variant, we assume that a facility can have a constraint in resources dedicated to its clients, so it is necessary to limit the number of clients assigned to a facility.

The solution is to open a set of facilities $\mathcal{C}$ to minimize the joint cost $C(F, D, f)$ of opening the facilities and serving the demand while ensuring that each facility j can only serve at most u_j clients :

$$C(F, D, f) = \sum_{\forall j \in \mathcal{C}} f_j + \sum_{\forall i \in D} d(i, m(i)) \tag{2.3}$$

where $m(i) \in \mathcal{C}$ is the facility j *closest* to i and c_j is the number of clients i attached to facility j :

$$c_j \leq u_j$$

.

For capacitated facility location problem, there are 2 variants :
- *splittable demand* : the demand from a client can be split across more than one facility.
- *unsplittable demand* : the demand from a client can only be served totally by one facility. Hence it is more likely to have the capacity constraint violated in this case.

2.3.4 Multiple commodity facility location

Facility location problem can be extended to address the case there are multi commodities served at a facility. Let I denote the set of commodities I=$\{1, ..., M\}$. Each commodity $h \in I$ has a subset of clients. To extend the cost function, we consider an optimization for all commodities and assume the same opening cost f for every commodity h, the joint cost can be expressed as :

$$C(F, D, I, f) = \sum_{\forall j \in \mathcal{C}} \sum_{\forall h \in I} f_j(h) + \sum_{\forall i \in D} \sum_{\forall h \in I} d(i, m(i, h)) \tag{2.4}$$

where $m(i, h) \in \mathcal{C}$ is the facility j holding h *closest* to i. If we consider the capacitated version of facility location problem, we have the number of clients i demanding any commodity h attached to facility j :

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

2.4 Approximation solutions for facility location problem

The first approximation algorithm for facility location was proposed by Hochbaum [50] based on greedy heuristics. Research in the last decade has improved the state of the art

dramatically. Different techniques which are widely used to approximate facility location problems have been proposed, including linear programing (LP) rounding, primal-dual and local search technique

2.4.1 Greedy heuristic

Greedy heuristics for facility location problems were the first proposed by Hochbaum [50]. Hochbaum reduced the facility location problems to set cover problem variants and applied algorithms of the greedy heuristics for the set cover. Recently, Jain et al. [58] used the method of dual fitting and the idea of factor-revealing LP to design and analyze two greedy algorithms for the metric uncapacitated facility location problem and provided constant factor approximation. An improvement of [58] has been proposed in [59], in which the algorithm computes a solution together with an infeasible dual-solution having the same value. The approximation factor in this case can be the factor to shrink the dual solution to a feasible one.

2.4.2 LP rounding technique

Approximation algorithms based on rounding the fractional optimal solution to the LP relaxation of the original integer programs were proposed by Shmoys et al. [104]. They used the filtering idea proposed by Lin and Vitter [76] to round the fractional solution to the LP and obtained constant factor approximations for many facility location problems. This idea was also combined with randomization by Chudak et al. [31].

There are four steps in LP rounding technique :

- Formulate the problem as an integer programming (IP) problem.
- Solve the corresponding LP-relaxation. LP-relaxed solution is a lower bound on IP.
- Round the relaxed optimal solution.
- Prove that the rounding does not increase much the cost.

The drawback of this technique is that LP rounding usually involves large linear program which causes long running time [59].

2.4.3 Primal-dual technique

The primal-dual technique for approximation consists of a primal integer programming formulation of the problem and the dual of a linear programming relaxation of the integer program. It solves the problem with a two-phase primal-dual scheme. The technique's main idea is to relax the primal conditions while satisfying all the complimentary slackness conditions of dual variables. This method has a solution to the primal integer problem within a bounded factor of optimal solution. Approximation algorithms for uncapacitated facility location based on primal-dual techniques were proposed by Jain and Vazirani [60]. Their primal-dual technique consists of first constructing a feasible dual solution and then using the dual solution to construct an integer feasible solution of LP.

2.4.4 Local search technique

Approximation algorithms for facility location based on local search are the easiest to understand and implement in practice. Local search heuristics were proposed by [71] and have been widely used by practitioners. Local search technique applied for facility location problems consists of adopting operations like random facility add or drop to improve the cost. The solution obtained by Local search techniques is called "local minimum" when there is no more local operations to reduce the cost. The ratio of the cost of the worst case local minimum to the cost of the global optimum is termed as "locality gap". Korupolu et al. [68] showed that a worst case analysis of the local minimum computed by this heuristic is possible and they showed constant factor approximations to many facility location problems which were comparable to those obtained by other techniques. The significance of these results lies in the fact that local search is widely implemented by many practitioners of operations research. For certain variants of facility location problems, local search is the only technique known to give constant factor approximations.

Algorithm 2.1 localSearch (F, D)

$\mathcal{C} \leftarrow$ an arbitrary feasible solution $\mathcal{C}$
while $\exists \mathcal{C}^{'}$ such that $C(\mathcal{C}^{'}) < C(S)$ **do**
  $\mathcal{C} \leftarrow \mathcal{C}^{'}$
end while
return $\mathcal{C}$

Alg.2.1 describes the generic procedure of local search technique to find the approximated solution $\mathcal{C}$. For example, to find any solution $\mathcal{C}^{'}$ having a cost C less than the current cost, in k-median problem we can find a candidate facility s that exists already in the current solution $\mathcal{C}$ to *swap* with another facility $s^{'}$ that did not belong to $\mathcal{C}$. This *swap* operation constitutes the new solution $\mathcal{C}^{'}$ as shown in Alg.2.2. This algorithm has been shown to have a locality gap of 5 in [7].

Algorithm 2.2 localSearch (F, D) (for k-median problem)

select arbitrary $\mathcal{C}$ such that $\mathcal{C} \in F$ and $|S| = k$
while $\exists s \in \mathcal{C}, s^{'} \in F$ such that $C(\mathcal{C} - s + s^{'}) < C(\mathcal{C})$ **do**
  $\mathcal{C} \leftarrow \mathcal{C} - s + s^{'}$
end while
return $\mathcal{C}$

For an uncapacitated facility location problem (UFL), we can open any number of facilities that minimizes the sum of facility cost and the total service cost. As a solution, we need to identify a subset $\mathcal{C} \subseteq F$ and assign clients in C to facilities in $\mathcal{C}$. Besides *swap* we consider two more operations : *add* and *drop* as shown in Alg. 2.3. This algorithm has been proved in [7] to have a locality gap of 3.

In a capacitated facility location problem (CFL), we are given integer capacities $u_j > 0$ for each $j \in F$. We can put multiple copies at a facility j. Each copy incurs a cost f_j and is capable of serving at most u_j clients. The capacities u_j may be different for different facilities j. In [7], authors demonstrated a locality gap of at most 4 on a local search procedure for the capacitated facility location problem. At each step of the local search,

Algorithm 2.3 localSearch (F, D) (for UFL)

select arbitrary $\mathcal{C}$ such that $\mathcal{C} \in F$
while $\exists s \in \mathcal{C}, s^{'} \in F$ such that $min(C(\mathcal{C} - s + s^{'}), C(\mathcal{C} - s), C(\mathcal{C} + s^{'})) < C(\mathcal{C})$ **do**
  $S \leftarrow \{\mathcal{C} - s + s^{'} | \mathcal{C} - s | \mathcal{C} + s^{'}\}$ such that $C(S)$ is minimum
  $\mathcal{C} \leftarrow S$
end while
return $\mathcal{C}$

there are two operations : *add* a facility $s^{'} \in F$ or *add and drop* : add l copies of a facility $s^{'} \in F$ and drop a subset of the open facilities $T \in \mathcal{C}$ (See Alg. 2.4). Due to capacity constraints, a copy of $s^{'}$ can serve at most $u_{s'}$ clients, hence to find the subset T, we must define a Knapsack problem : let the client set assigned to T as $weight(T)$ and the profit of dropping T and adding $s^{'}$ as :

$$profit(T) = \sum_{s \in T} f_s + \sum_{s \in T} \sum_{i \in C(T)} (d(s, i) - d(s^{'}, i))$$

where $C(T)$ is the set of clients attached to T. Given $s^{'}$, the Knapsack problem should find a subset $T \subseteq \mathcal{C}$ having $weight(T) < lu_{s'}$ that maximizes the $profit(T)$.

Algorithm 2.4 localSearch (F, D) (for CFL)

select arbitrary $\mathcal{C}$ such that $\mathcal{C} \in F$
while $\exists T \subseteq \mathcal{C}, s^{'} \in F$ such that $min(C(\mathcal{C} + s^{'}), C(\mathcal{C} + s^{'} - T)) < C(\mathcal{C})$ **do**
  $S \leftarrow \{\mathcal{C} + s^{'} | \mathcal{C} + s^{'} - T\}$ such that $C(S)$ is minimum
  $\mathcal{C} \leftarrow S$
end while
return $\mathcal{C}$

2.5 Replication from facility location perspective

The replication problem in mobile networks requires the content to be stored at the smallest number of nodes as possible while the content location satisfies the need to retrieve the content from interested users with minimum latency. With these properties we find that the replication problem share many perpsectives with facility location problems. Therefore, it can be casted as a facility location problem (which aims to minimize the average distance to access facility) [1]. In mobile networks, mobile devices with constrained resources (energy, bandwidth...) can only serve at most a limited number of neighbors, hence the capacitated variant of facility location problem is more appropriate. The dynamic and decentralized nature of mobile networks also requires a flexible and distributed mechanism for the performance optimization purpose.

[1] Replication problem can also be casted as a "center selection problem" which aims to minimize the maximum distance to access content. However we found that it is more appropriate from a system's global view to optimize the average distance instead of the maximum distance.

2.5.1 Problem formulation

The problem of content replication together with dynamic nature of mobile networks introduce new challenges with respect to the wired networks. In this context, we study a scenario involving users equipped with devices offering Internet broadband connectivity as well as device-to-device communication capabilities (e.g. IEEE 802.11).

We then consider a *cooperative user network* composed of a set $V = \{1, ..., N\}$ of mobile nodes. A node j wishing to access the content first tries to retrieve it from other devices; if its search fails, the node downloads a fresh content replica from the Internet server and temporarily stores it for a period of time τ, termed *storage time*. During the storage period, j serves the content to all nodes i issuing requests for it. If we refer the set of nodes that hold a replica of content as $F \subseteq V$ and the set of nodes that consume content from F as $D \subseteq V$, clearly we see an overlap of these two sets as consumer nodes can also act as replica nodes if needed. The problem now is to determine the set of nodes $\mathcal{C}$ that minimize the joint cost of storing content replicas $j \in \mathcal{C}$ and the distance cost to access from clients $i \in V \setminus \mathcal{C}$

Let I denote the set of contents available in network I=$\{1, ..., M\}$. Each content h has a content popularity represented by a number of nodes that are interested in the content. The number of clients assigned to a replica node may overload that node and hence may cause more delay. Possible way to overcome this issue is to bound the maximum number of clients assigned to a particular node by using the capacitated version of facility location problem. Given that node capacity to serve is at most u_j clients, we have :

$$C(V, f) = \sum_{\forall j \in \mathcal{C}} \sum_{\forall h \in I} f_j(h) + \sum_{\forall i \in V \setminus \mathcal{C}} \sum_{\forall h \in I} d(i, m(i, h))$$

where $m(i, h) \in \mathcal{C}$ is the facility j holding h *closest* to i. There is a constraint in the number of clients i demanding any content h attached to facility j :

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

The overlap of facility set and client set made our problem different from the traditional facility location in operation research. Hence the choice of facility set $\mathcal{C}$ as a solution is made from the whole set V and not from a distinct candidate set F.

2.5.2 Distributed solution

Moscibroda et al. in [86] initially studied the distributed solution to approximate the facility location problem. In particular, they explore the trade-off between the amount of communication and the resulting approximation ratio. Their algorithm is based on a distributed primal-dual approach for approximating a linear program and the approximation factor depends on communication rounds and message size. The distributed primal-dual approach however is complicated to implement in practice due to the communication cost.

We focus our interest on the local search algorithms which are more applicable for a distributed system where a global view can not be assumed.

Local search algorithms have been known among practitioners as easy to understand and implement. Hence for many optimization problems, local search heuristics are a good choice for implementation. The main idea of local search in approximating facility location problem is to perform randomly a set of operations e.g. *add*, *drop*, *swap* of facilities if the subsequent cost is lower than the current cost. We aim to develop a distributed approximation mechanism inspired by the local search technique presented in [7]. These techniques have been proved to reach a solution with constant factor for capacitated facility location. Our study aims to build a mechanism that is fully distributed, asynchronous and requires only local measurement at nodes.

It is out of the scope of this thesis to provide any proof of convergence for such distributed mechanism due to these reasons :

- the dynamic nature of mobile network can change nodes' location any time. Hence we can not have a static set of facility as a solution for this problem.
- we focus on the load balancing objective and our scheme should deal with constant mobility. This is why we should keep swapping the facility role from nodes to nodes. The definition of convergence cannot be readily applied in this case.

Our idea is to compare the optimal solution of every snapshot of static network topology against the captured solution given by our distributed mechanism at that time. We evaluate our mechanism by verifying the number of facilities opened and their locations using χ^2 test : we compare the distribution of facilities against the distribution given by optimal solution with the technique that we will develop further in Chapter 5

Chapter 3

Mobility models for wireless networks

To evaluate any application in mobile networks, a vital step is to study how mobile users move in that application context. Carrying out research on real and large scale mobile networks is usually expensive and may involve too many participants. Therefore almost all research works in this area are based on simulation to validate results and evaluate the performance against other works.

Simulation with mobile networks has many parameters that researchers should pay attention to, e.g. mobility model, network density and traffic patterns. Among them, the mobility model plays a very important role. Mobility models are designed to describe the movement patterns of mobile users, i.e. how their location, speed and direction change over time. In order to simulate a new application in mobile networks, it is necessary to use a mobility model that accurately represents the mobile patterns of people that are its potential users. Only in such context, it is possible to determine whether or not the proposed application would be useful to implement and deploy in reality.

Performance results of mobile network applications drastically change as a result of changing the mobility model simulated [24]. Choosing a mobility model in the simulation for mobile network is so important that there is a real urge to understanding mobility models and their impact on application performance.

In the literature, many mobility models have been proposed to capture different characteristics of mobility and to represent mobility in a more realistic fashion. Some of them represent mobile nodes' movements which can be either independent of each other (i.e., entity mobility models) or dependent on each other (i.e., group mobility models [52]).

In [10], another way to classify these models is based on their randomness : for some mobility models, the choice of velocity and direction is totally random like random waypoint, random walk and random direction models. On the other hand, some models assume that movements of a mobile node should be affected by time (Gauss-Markov model [75]), geographic restriction (with obstacle and interest point [61], graph-based and map-based [115]) or social relations of mobile users [87].

Many studies were carried out based on synthetic mobility models which are mainly

random movements. One of the most frequently used mobility model in simulations is the Random Waypoint model. In this model, nodes move independently to a randomly chosen destination with a randomly selected velocity. The simplicity of Random Waypoint model may have been one reason for its widespread use in simulations.

Recently, researchers focus their attention on experimental mobility traces in order to derive the model from these traces or to use them directly in their simulations. One intuitive method to create realistic mobility patterns would be to construct trace-based mobility models, in which accurate information about the mobility traces of users could be provided. In some recent papers, authors validate the performance of mobile network application against realistic mobility traces. The research community has put efforts to build a trace repository for mobile network research purpose (the CRAWDAD project [69]). Traces are those mobility patterns that are recorded by wireless access points, by cellular base stations or wireless devices carried by mobile users in real life experiments [26]. But these traces can provide only accurate information when they involve a large number of users and a sufficient long experimental period. The drawback of these traces is that the number of participants is not high and can introduce inaccurate data when experimental devices have problems (energy constraint, users' misbehaviors). As a very first attempt, a group of researchers are trying to abstract these traces into a human mobility model and thus can generate the mobility pattern based on some input parameters [97, 74].

Since different mobile applications need to be validated through their own context with specific mobility patterns, the first thing we need to do is to find the mobility models with mobility characteristics that are applicable in our application. In this chapter we present several mobility models that may be used in the simulations of mobile networks proposed by recent research literature together with the model's drawbacks and alternative model if any. The remainder of this chapter is organized as follows. In Section 3.1, we describe the commonly used random models with their properties and variants, point out the issues from these models e.g. the speed decay problem, and introduce the alternative solutions. Section 3.2 presents some other mobility models that deal with more specific mobile scenarios. In Section 3.3 we discuss the human traces study and the new human mobility model derived from these traces. Section 3.4 concludes the proper choice of model for our application scheme.

3.1 Random mobility models

In random-based mobility models, mobile nodes are supposed to move randomly and freely without any restriction. The destination, speed, moving time and pause time are all chosen randomly and independently of other nodes. However, nodes can choose the next move according their movement history.

In this section, we present some random mobility models and their properties that have been proposed and currently used for the performance evaluation of many mobile network applications. The first two models presented, the Random Walk Mobility Model and the Random Waypoint Mobility Model, are the two most common mobility models used by researchers. We point out some limitations of the random-based models and their potential impact on the accuracy of simulation.

3.1.1 Random Walk

The Random Walk Mobility Model was first proposed by Einstein, originally to emulate the unpredictable movement of particles in physics, which is sometimes referred to as Brownian Motion [38]. Since many entities in nature move in extremely unpredictable ways, the Random Walk Mobility Model was developed to mimic this kind of movement.

In this mobility model, a node moves from its current location to a new location by randomly choosing a direction and speed to travel. The new speed v_i and direction angle θ_i are both chosen (uniformly or following a Gaussian distribution) from predefined ranges, $[v_{min}, v_{max}]$ and $[0, 2\pi]$ respectively.

Each movement in the Random Walk Mobility Model occurs in either a constant time interval T or a constant distance traveled d (In the Random Walk Mobility Model, there are 2 variants : node may change direction after traveling a specified distance instead of a specified time, at the end of which a new movement with new speed and direction is recalculated).

If the node which moves according to this model reaches a simulation boundary, it "bounces" from the simulation boundary within an angle determined by the incoming direction (θ_i or $\pi - \theta_i$). Node then continues with this new direction.

In [123], authors proved that a random walk on a one or two-dimensional surface returns to the origin with complete certainty, i.e., a probability of 1.0. This characteristic ensures that the random walk represents a mobility model in which the mobile entities are ensured to move around their starting points.

The Random Walk Mobility Model is sometimes simplified, for example, by assigning the same speed to every node in the simulation. The Random Walk model has similarities with the Random Waypoint model because the node movement has strong randomness in both models. The main difference is that there is no pause time in Random Walk model.

Issues in Random Walk model

The Random Walk model is a memoryless mobility process where the information about the previous status is not used for the future decision, because it retains no knowledge concerning its past locations and speed values [75]. The current speed and direction of a node is independent of its past speed and direction. However, this is not the case of mobile nodes in many real life applications. This characteristic can generate unrealistic movements such as sudden stops and sharp direction change. Other models, such as the Gauss-Markov Mobility Model, which we discuss in Sec. 3.1.4, can fix this issue.

If the constant time interval T (or constant distance d) is set to a small value, node movements is random but bounded to a small portion of the simulation area. If the goal of the performance investigation is to evaluate a low mobility network, then the parameter to change a node's direction should be set with a small value. Otherwise, a larger value should be used.

3.1.2 Random Waypoint

The Random Waypoint Model was proposed in [22]. It has been widely used to evaluate the MANETs since it is easy to use and it is supported by many available tools. (e.g. the setdest tool from CMU Monarch group included in network simulator ns-2 [21]).

In the network simulator ns-2, the implementation of this mobility model is as follows : at the beginning of simulation, each mobile node is randomly placed in the simulation area and randomly selects one location as the next destination. It then moves towards this destination with constant velocity chosen uniformly and randomly from $[0, v_{max}]$. The velocity and direction of a node are chosen independently of other nodes. When arriving to the destination, the node stops for a duration defined by the "pause time" parameter T_p. After this duration, it again chooses another random destination in the simulation field and moves towards it. The whole process is repeated again and again until the simulation ends. The movement pattern of a node using the Random Waypoint Mobility Model is similar to the Random Walk Mobility Model if $T_p = 0$. In the Random Waypoint model, v_{max} and T_p are the two key parameters that determine the mobility behavior of nodes. If the v_{max} is small and the pause time T_p is long, the topology of network becomes relatively stable. On the other hand, if the node moves fast (i.e., v_{max} is large) and the pause time T_p is small, the topology is expected to be highly dynamic. If the Random Waypoint Mobility Model is used in a performance evaluation, appropriate parameters need to be evaluated. Slow speeds, and large pause times, the network topology hardly changes and the results are likely applicable for a static network.

There is also a complex relationship between node speed and pause time in the Random Waypoint Mobility Model. For example, a scenario with fast velocity and long pause times actually produces a more stable network than a scenario with slower velocity and shorter pause times. The link breakage rate is more sensitive to the pause time, i.e long pause times produce a stable network (i.e., few links change) even at high speeds [14].

The introduction of pause time makes it more difficult to predict node average speed. Considering that v_{max} is uniformly and randomly chosen from $[0, v_{max}]$, we can find that the average nodal speed is $v_{max}/2$, only if we assume that the pause time $T_p = 0$ (which is the case of Random Walk).

Issues in Random Waypoint model

In Random Waypoint model, nodes tend to move toward the center of simulation area, causing a non-homogeneity in node density [16].

Since original Random Waypoint model fails to provide a steady state, it is commonly known that the average speed consistently decays with simulation time [127]. What happens is that in the end of simulation we have a nearly static scenario. This problem makes the performance evaluation unreliable.

3.1.3 Random Direction

In the Random Waypoint Mobility Model [98], the probability of a node choosing a new destination that is located in the center of the simulation area, or a destination which requires travel through the center of the simulation area, is high. Thus, the nodes appear to converge, disperse, and converge again.

Random Direction Model alleviates this effect and maintains a constant node degree during the simulation. In Random Direction Mobility, a node chooses a random direction to travel similar to the Random Walk Mobility Model. The node then travels to the border of the simulation area in that direction. When the simulation boundary is reached, each node pauses for a timeout, then chooses another direction from $[0, \pi]$ and continues the movement.

Since nodes tend to spread out to the border of the simulation area, the average hop count for data transferring using the Random Direction Mobility Model is higher than those of other mobility models. In addition, network partitions occur more frequently with the Random Direction Mobility Model.

There are some modified variants of Random Direction Mobility Model. In [98], nodes continue to choose random directions but they are not forced to travel to the boundary before stopping to change direction. Instead, a node chooses a random direction and selects a destination at any point in that direction. The node then pauses at this destination before choosing a new random direction. This modification however is similar to Random Waypoint Model.

Issues in Random Direction model

The scenario that Random Direction model represents is somehow unrealistic : it is difficult to find a context where nodes concentrate at the edge of the simulation area.

3.1.4 Gauss-Markov model

The Gauss-Markov Mobility Model was originally proposed for the simulation of a wireless personal communication service network (PCS) [75]. It was designed to adapt to different levels of randomness by tuning a parameter denoted as γ. The Gauss-Markov Mobility Model was also widely utilized (e.g. [54, 24]). In this model, the velocity of mobile node is assumed to be correlated over time and modeled as a Gauss-Markov stochastic process.

Gauss-Markov Mobility Model was implemented in [116] as following : Initially each node is assigned a current speed and direction. At fixed intervals of time n movement occurs by updating the speed and direction of each node. Specifically, the value of speed and direction at the n instance is calculated based upon the value of speed and direction at the $n-1$ instance and a random variable using the following equations :

$$v_n = \gamma v_{n-1} + (1-\gamma)\bar{v} + \sqrt{1-\gamma^2}v(x_{n-1})$$

$$\theta_n = \gamma\theta_{n-1} + (1-\gamma)\bar{\theta} + \sqrt{1-\gamma^2}\theta(x_{n-1})$$

where v_n and θ_n are the new speed and direction of node at time interval n ; $\gamma \in [0, 1]$, is the tuning parameter used to vary the randomness, $\bar{v}$ and $\bar{\theta}$ are constant values expressing the mean of speed and direction and $v(x_{n-1})$ and $\theta(x_{n-1})$ are random variables from a Gaussian distribution.

Gauss-Markov model is a temporally dependent mobility model where the degree of dependency is determined by the memory level parameter γ. If the Gauss-Markov Model is memoryless, i.e. $\gamma = 0$, obviously, the model becomes Random Walk or Brownian Motion model. Intermediate levels of randomness are obtained by varying the value of γ between 0 and 1. If the Gauss-Markov Model has strong memory, i.e. $\gamma = 1$ the velocity of mobile node at time slot t is exactly same as its previous velocity.

At each time interval the next location is calculated based on the current location, speed, and direction of movement. To ensure that a node does not remain near an edge of the grid for a long period of time, nodes are forced away from an edge when they move closely to a given distance from the edge. When the node is going to travel beyond the boundaries of the simulation field, the direction of movement θ is forced to bounce with π.

Other implementations of the model exist, in which Markov process can be applied to the coordinates (x,y) directly instead of through speed and direction variables. Some implementations used a velocity vector instead of a direction equation.

Issues in Gauss-Markov model

The Gauss-Markov model is designed to eliminate most of issues in Random Walk and Random Direction. For example Gauss-Markov Mobility can eliminate the sudden stops and sharp turns encountered in the Random Walk Mobility Model by allowing previous speed and direction to influence the choice of next speed and directions. The only problem from using Gauss-Markov model is that it is hard to make a proper choice of model parameters.

3.1.5 Issues in random mobility models and the Random trip model

Random mobility models have several issues. In [24, 10] authors presented exhaustive lists of these problems. These issues can be classified into 4 types :

Transient problems

The simulation of mobility models such as the random waypoint often cause transient problems :

– The decay of average speed in simulation time.

– The change in node distribution from the initial phase to a steady state phase [127].

In most of the performance investigations that use the Random Waypoint Mobility Model, nodes are initially distributed randomly around the simulation area while during the simulation, nodes tend to gathering in the center of simulation area. This initial distribution of nodes is different of the distribution of nodes when moving [16]. This causes confusion in interpreting the simulation results.

There is a study that shows how to simply obtain the stationary distribution of nodes and speeds and shows how to perform a perfect (i.e. transient free) simulation [127, 17].

Memory-less problems :

The random models are designed to represent the movement of mobile nodes in a simple way. Because of its simplicity of implementation and analysis, they are widely used. However, they may not effectively mimic certain mobility characteristics of some realistic scenarios, when temporal dependency, spatial dependency are present. The random models assume that the velocity of mobile node is a memory-less random process, i.e., the velocity and direction at current epoch is independent of the previous epoch. Thus, some extreme mobility behavior, such as sudden stop, sudden acceleration and sharp direction change, may frequently occur in the trace generated by the random models. In many real life scenarios, the speed of mobile nodes should not change suddenly and the direction change needs long time to complete.

Geographical problems :

In random models, mobile nodes can move freely in the simulation area without any restriction. This kind of movement omits the reality that people can move in streets, highways or university campus where the moving area is bounded by obstacles, boundaries and public traffic. In such a specific context, random models cannot represent some mobility characteristics. Depending on the simulation requirements we can choose another models e.g : map-based mobility.

Social-based problems :

In random mobility model, a mobile node is considered as an independent entity that moves freely regardless of other nodes' movement. This kind of movement is called as entity mobility model in [24]. However, in some scenarios including battlefield communication and working environment (like in an office, a factory), the movement pattern of a mobile node should be affected by the role of each node and by the group of nodes it belongs to. In these cases, a group or social-aware mobility is more applicable.

In [128], authors proposed three simple solutions to avoid the transient problems :

- Save the locations of the nodes after a simulation has executed long enough to eliminate this initial transient effect, and use this as the initial starting point of nodes in next simulations.
- Initially distribute the nodes in a manner that maps to a distribution more identical to the model.
- Discard the first 900 seconds of simulation time produced by the Random Waypoint mobility model in each simulation trial. [128] demonstrates that this simple solution avoids the transient effect even in slow mobility scenario.

In [91], authors proposed to derive "steady state" of the speed, location, and pause time of a node moving in a rectangular area under the random waypoint mobility model, then they modified the implementation of random waypoint to begin a simulation with a "stationary" distribution. In another work [15], Leboudec et al. also studied the stationarity of random mobility class but by means of Palm calculus and proposed the perfect simulation

using the Random Trip mobility [17]. They also provide a tool to generate mobility files in ns-2 which is freely to download.

In [16, 17], the authors present a generalization of the Random Walk and Random Waypoint mobility models that they call Random Trip model. The authors introduce a technique to sample the initial simulation state from the stationary regime (a methodology that is usually called perfect simulation) based on Palm Calculus [15] in order to solve the problem of reaching time-stationarity.

3.2 Non-random mobility models

In order to deal with issues in random models, several mobility models have been proposed, but all of them are very specific for particular scenarios. These models usually require many input parameters hence it is not easy to make a choice with proper simulation settings.

Group-based mobility

In [52], a novel group mobility model has been presented, namely Reference Point Group Mobility (RPGM). In this model, they introduced the relationships of mobile users by dividing nodes into groups and each individual movement toward new destination is affected by the checkpoint of their group. The relationships can be tuned by model parameters. Simulations with RPGM showed a significant difference to random mobility models.

Graph-based and road-based mobility

[115] proposed a novel graph-based mobility model, in which nodes do not move randomly, but along the edges of a graph restricted by the real infrastructure.. Simulation results show that the graph-based constraints have a big impact on the performance of MANETs.

Several mobility models have been designed to simulate vehicular networks. In [30], authors proposed STRAW, a simple vehicular mobility model with real map data. Similarly in [82], GrooveSim has been introduced as a simulator to models vehicular communication using a street map-based topography.

[100] proposed a realistic model of node movements based on the motion of vehicles on real street maps. Authors then compared this model with the Random Waypoint mobility mode. *Results showed that the Random Waypoint mobility model is a good approximation for simulating vehicles motions*, but there are situations in which the new model is more appropriate, for example when the street network becomes sparse, the street mobility model will restrict the motion of vehicles even more, thus makes movements less random than those in the Random Waypoint model.

In [40], they proposed a framework for vehicular mobility simulation, named VanetMobiSim to generate realistic vehicular movement traces for network simulators. VanetMobiSim is validated by realistic vehicular traffic.

Obstacle-based mobility

In [62], authors proposed to design mobility model that allows the placement of obstacles that restrict movement and signal propagation. The idea is to use Voronoi diagram of obstacle vertices to construct movement paths. Authors also observed that the performance of ad hoc network protocols is affected differently with this new mobility model.

Social-based mobility

In [87], authors argued that human movement is strongly affected by the needs of of socializing and cooperating. They proposed a new mobility model with connectivity matrix based on social network theory. This model allows nodes to be grouped together based on social relationships among people in a community and these relationships can also change in time. They validated the model with real traces in [55] and showed that the synthetic mobility traces approximated closely human movements.

In [39], a new movement model to be used in DTN simulations, called Working Day Movement Model has been presented. The model represents the daily activities of working people who come to work in the morning and go back home at evening. The model is compared and validated with the statistical data of real-world traces.

3.3 Human mobility

Recently, there are some public data repository of traces capturing movement of humans e.g. GPS traces and Bluetooth connectivity traces which contains the Bluetooth identifiers of the devices that have been in radio range of a device.

Bluetooth devices or "iMotes" were distributed to experiment participants in the campus of Cambridge University, in order to collect data about human movements and study the characteristics of the contact between people [55]. There are similar projects like the project at UCSD [83] and the wireless traffic measurements at Dartmouth College [49]. The purpose of these projects is to provide a repository of traces for the mobile wireless research community.

In general, traces cannot replace synthetic models due to several reasons [88] :

- Traces are expensive to collect and large data traces usually owned by telecommunication companies. They do not make them public since these traces can be exploited for commercial use.
- These traces are related to very specific scenarios and it is currently difficult to generalize their validity.
- Many available traces do not contains all necessary data to analyze and characterize mobility patterns. These traces can also depend on the data collection technique (Bluetooth, Wifi...)

For these reasons, many mobility models are still in use to evaluate mobile network applications. Random Walk and Random Waypoint mobility model are used in most of simulation studies thanks to their simplicity. Recently, several improvements for random

mobility models for ad hoc network research have been presented in [16, 61, 80].

However, trace studies have surprisingly shown common statistical characteristics : the same distribution of the duration of the contacts and inter-contact intervals. They showed that this distribution can be fitted to a power-law followed by an exponential cut-off. These findings open a new wave in research focusing in characterize human movement patterns in mobility traces.

3.3.1 Mobility trace studies

Recently, many researchers have tried to study existing models in order to make them more realistic by exploiting the available mobility traces [69].

The main idea of these models is the exploitation of available measurements such as connectivity logs to generate synthetic traces that are characterized by the same statistical properties of the real ones. Several measurement studies have been done in several wireless settings : MANETs, DTNs, VANETs (Vehicular ad hoc networks) and WLANs (Wireless Local Area Networks)... to gain insight of users' mobility. We can group these traces in three distinct types [64] :

- Infrastructure-based traces that reflect connectivity between Access Points (APs) or base stations (BSs) and wireless devices [83, 37]. In these traces the geographical information can be derived from the position of infrastructure gateways.
- Device-to-device traces : recorded contacts directly between mobile devices collected by distributing devices to a few people (students, conference volunteers [26, 55, 124]). These traces contains contact times for each pair of devices. There is no geographical information. And the number of participants is not high.
- GPS-based traces : contacts through a trace collected by tracking the movements of individual people through GPS devices. In [97] the traces contain the latitude and longitude coordinates of each mobile device every 10 seconds.

First traces studies were measurements of WLANs which have been done in [110, 11, 12]. In [110], authors presented the results of mobility measurement (pause time, travel length) of radio networks. Traffic measurement and online behavior of mobile users of a high speed wireless access network are presented in [56] which paid more attention on network measurement. From the traces of WLAN deployed at Dartmouth College campus, authors [49] made an detailed measurement and analysis on network usage behaviors. Another interesting work [67] tried to derive a model from user mobility characteristics of these traces : they divided the campus area into popular regions and measured the movements to and from these areas by a Markovian model. The results showed that pause time and velocity of users in these traces follow a log-normal distribution. In [118], authors constructed a mobility model based on traces from wireless network at ETH in Zurich. Similar to [67], the simulation area divided into squares and authors computed the probability of transitions between adjacent squares. In this study, the power-law distribution of session durations has been reported .

Authors in [106] built a contact traces based on the class schedule of 22341 students in a campus. They argued that since performance studies on wireless networks requires only

contact information instead of full information on mobility, these traces can help to derive the same results. They used these contact traces to to study contact opportunity, number of contacts per student... and evaluate the spread of mobile computer viruses by epidemic forwarding in such context.

In [80], authors presented another method to collect mobility traces that recorded pedestrians' movement in downtown without using wireless devices. Their data were obtained by simple observation by deploying points of observation using digital cameras. Then they proposed a new method to generate a mobility scenario called Urban Pedestrian Flows (UPF).

In [130], authors studied traces collected by Wifi devices attached to buses. In these traces, buses encounter each other buses on their daily routes and can forming a DTN. Authors thus analyzed the bus-to-bus contact opportunities to derive the subsequent performance of DTN routing. They found that the inter-contact times aggregated at a route level exhibit periodic behavior which can support the predictability assumption for some kinds of DTNs.

3.3.2 The heavy-tail in inter-contact time distribution

Recent studies [111, 53, 11, 12] focussed on analyzing the characteristics of mobility traces in order to gain more insights on human mobility patterns.

Chaintreau et al. [26] conducted the first study on contact and inter-contact patterns of mobile users. They show that contact time and inter-contact time between individuals can be fitted into power-law distributions and that these patterns may be exploited to develop more efficient opportunistic protocols. Fig. 3.1 shows the curve of power law distribution of inter-contact time in Cambridge iMotes data. These data are collected from 40 iMotes deployed to undergraduate students for 11 days. iMotes detect proximity using Bluetooth.

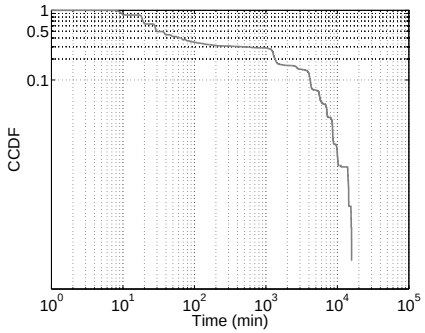


Fig. 3.1 – Power-law distribution of inter-contact time in Cambridge imotes data.

The work confirms the results of other studies conducted at Dartmouth [49], UCSD [83] : these patterns are different to the exponential decay of inter-contacts time intervals that the authors of [102] found in random mobility models.

However, Le Boudec et al. in [64] presented another perspective to the problem of fitting

these distributions. The authors consider 6 sets of traces and derive several analytical results that can be summarized as follows. The authors verify the power-law decay of inter-contacts time CCDF between mobile devices. They found that beyond a specific time τ (which is varying for each trace) the CCDF exhibits exponential decay. *As an important finding they demonstrated that mobility models such as the Random Waypoint model should not be abandoned since they are able to represent power-law decay of inter-contact time with an exponential tail after this τ time.*

3.3.3 The power-law of human mobility in virtual world traces

In [72], we also conduct a measurement study of user mobility but in a virtual environment. We present a novel methodology to capture spatio-temporal dynamics of user mobility that overcomes most of the limitations of previous attempts : it is cheap, it requires no logistic organization, it is not bound to a specific wireless technology and can potentially scale up to a very large number of participants. Our measurement approach exploits the tremendous raise in popularity of Networked Virtual Environments (NVEs), wherein thousands of users connect daily to interact, play, do business and follow university courses just to name a few potential applications. Here we focus on the SecondLife (SL) "metaverse" [3] which has recently gained momentum in the on-line community. Tempted by the question whether our methodology could provide similar results to those obtained in real-world experiments, we study the statistical distribution of user contacts and show that from a qualitative point of view user mobility in Second Life presents similar traits to those of real humans. We further push our analysis to the network topology that emerge from user interaction and show that they are highly clustered. We focus on the spatial properties of user movements and observe that users in Second Life revolve around several points of interest traveling in general short distances. Besides our findings, the traces collected in this work can be very useful for trace-driven simulations of communication schemes in mobile networks and their performance evaluation.

Our work differs from [64, 28, 27] which conduct several experiments mainly in confined areas and study analytical models of human mobility with the goal of assessing the performance of message forwarding in Delay Tolerant Networks (DTNs). Each user taking part to such experiments is equipped with a wireless device (for example a sensor device, a mobile phone, ...) running a custom software that records *temporal information* about their contacts. Individual measurements are collected, combined and parsed, originating elegant but complex algorithms [28] because the only available information is the temporal distribution of contact times, which are bound to the specific wireless technology used in the experiments. In general, position information of mobile users is not available, thus a spatial analysis is difficult to achieve [28]. Some experiments with GPS-enabled devices have been done in the past [70, 97], but these experiments are limited to outdoor environments.

Our primary goal is to perform a temporal, spatial and topological analysis of user interaction in SL. We implement a crawler to connect to SL and extracts position information of all users concurrently connected to a sub-space of the metaverse. This crawler is a custom SL client software (termed a *crawler*) using `libsecondlife` [2]. The crawler is able to monitor the position of **every** user located on the target land and measurement data is stored in a database. The crawler connects to the SL metaverse as a normal user, thus it is

not confined by limitations imposed by private lands : any accessible land can be monitored in its totality ; the maximum number of users that can be tracked is bounded only by the SL architecture (as of today, roughly 100 concurrent users per land) ; communication between the crawler and the database is not limited by SL.

During our experiments, we noted that introducing measurement probes in a NVE can cause unexpected effects that perturb the normal behavior of users and hence the measured user mobility patterns. Since our crawler is nothing but a stripped-down version of the legacy SL client and requires a valid login/password to connect to the metaverse, it is perceived in the SL space as an avatar, and as such may attract the attention of other users that try to interact with it : our initial experiments showed a steady convergence of user movements towards our crawler. To mitigate this perturbing effect we designed a crawler that mimics the behavior of a normal user : our crawler randomly moves over the target land and broadcasts chat messages randomly chosen from a small set of pre-defined phrases.

One striking evidence of our results is that they qualitatively fit to real life data, raising the legitimate question wether measurements taken in a virtual environment present similar traits to those taken in a realistic setting. Our methodology allows performing large experiments at a very low cost and generate data that can be used for trace-driven simulations of a large variety of applications : the study of epidemics and information diffusion in wireless networks are just some prominent examples.

Using the physical coordinates of users connected to a target land, we create snapshots of *radio communication networks* : given an arbitrary communication range r, a communication link exists between two users v_i, v_j if their distance is less than r. In the following we use a temporal sequence of networks extracted from the traces we collected using our *crawler* and analyze contact opportunities between users, their spatial distribution and graph-theoretic properties of their communication network.

A precondition for being able to gather useful data is to select an appropriate target land and measurement parameters. Choosing an appropriate target land in the SL metaverse is not an easy task : *i)* a large number of lands host very few users ; *ii)* lands with a large population are usually built to distribute virtual money : all a user has to do is to sit and wait for a long enough time to earn money (for free) ; iii) an automatic synchronization of the crawler to special events supposed to attract many users is very difficult to achieve. While we are currently working on a solution to the latter problem, we manually selected and analyzed the following popular areas : *Apfel Land*, a german-speaking arena for newbies ; *Dance Island*, a virtual discotheque ; *Island of View*, an open-space land in which an event (St. Valentines day) was organized.

In this study, we present results for 24 hours traces : while the analysis of longer traces yields analogous results to those presented here, long experiments are sometimes affected by instabilities of `libsecondlife` under a Linux environment and we decided to focus on a set of shorter but stable measurements. A summary of the traces we analyzed can be defined based on the total number of unique users and the average number of concurrently logged in users : Isle of View had 2656 unique visitors with an average of 65 concurrent users, Dance Island had 3347 unique users and 34 concurrent users in average and Apfel Land had 1568 users and 13 concurrent users in average.

We launched the crawler on the selected target lands and set the time granularity (intervals at which we take a snapshot of the users' positions) to $\tau = 10$ sec. We selected a communication range r to simulate users equipped with a bluetooth and a WiFi (802.11a at 54 Mbps) device, respectively $r_b = 10$ meters and $r_w = 80$ meters. In this work we assume an *ideal wireless channel* : radio networks extracted from our traces neglect the presence of obstacles such as buildings and trees.

User location in SL is expressed by coordinates $\{x, y, z\}$ which are relative to the target land whose size is by default 256×256 meters.

Temporal analysis

The metrics we use to analyze mobility patterns are inspired by the work of Chaintreau *et. al.* [27] and allow the analysis of the statistical distribution of contact opportunities between users :

- *Contact time* (CT) *:* is defined as the time interval in which two users (v_i, v_j) are in direct communication range, given r ;
- *Inter-contact time (ICT) :* is defined as the time interval which elapses between two contact periods of a pair of users. Let

$$[t^1_{(v_i,v_j)s}, t^1_{(v_i,v_j)e}], [t^2_{(v_i,v_jj)s}, t^2_{(v_i,v_j)e}], ...[t^n_{(v_i,v_j)s}, t^n_{(v_i,v_j)e}]$$

  be the successive time intervals at which a contact between user v_i and v_j occurs ; then, the inter-contact time between the $k-th$ and the $(k+1)-th$ contact intervals is :

$$IC^k_{(v_i,v_j)} = t^{k+1}_{(v_i,v_j)s} - t^k_{(v_i,v_j)e}$$

- *First contact time (FT) :* is defined as the waiting time for a user v_i to contact her first neighbor (ever).

We now discuss the results of our measurements for the three selected target lands and study the influence of the communication range (r_b or r_w). Fig. 3.2 illustrates the distribution of the temporal metrics we used in this work for $r_b = 10$ meters and $r_w = 80$ meters.

A glance at the complementary CDF (CCDF) of the contact time CT, showed in Fig. 3.2a-3.2d, indicates that the *median* contact time is roughly 30, 60 and 100 seconds respectively for Apfel Land, Isle of View and Dance Island when $r = r_b$, and about 70, 200 and 300 seconds for the same set of islands when $r = r_w$. Fig. 3.2a-3.2d also indicate that transfer opportunities are proportional to r : larger transmission ranges imply larger transfer opportunities.

The CCDF of the inter contact time ICT is shown in Fig. 3.2b-3.2e : the median ICT is around 400 seconds for the two open-space lands and between 700 and 800 seconds for the Dance Island. Analyzing the same trace of user movement yields surprisingly similar results with different communication ranges. We believe this result is due to the fact that users are concentrated around points of interest (as discussed below), but it would be interesting to compare such findings with real-world experiments.

Although the distribution of contact opportunities appears to be similar for the two open-space lands, the CCDF of the first contact time FT, depicted in Fig. 3.2c-3.2f, illus-

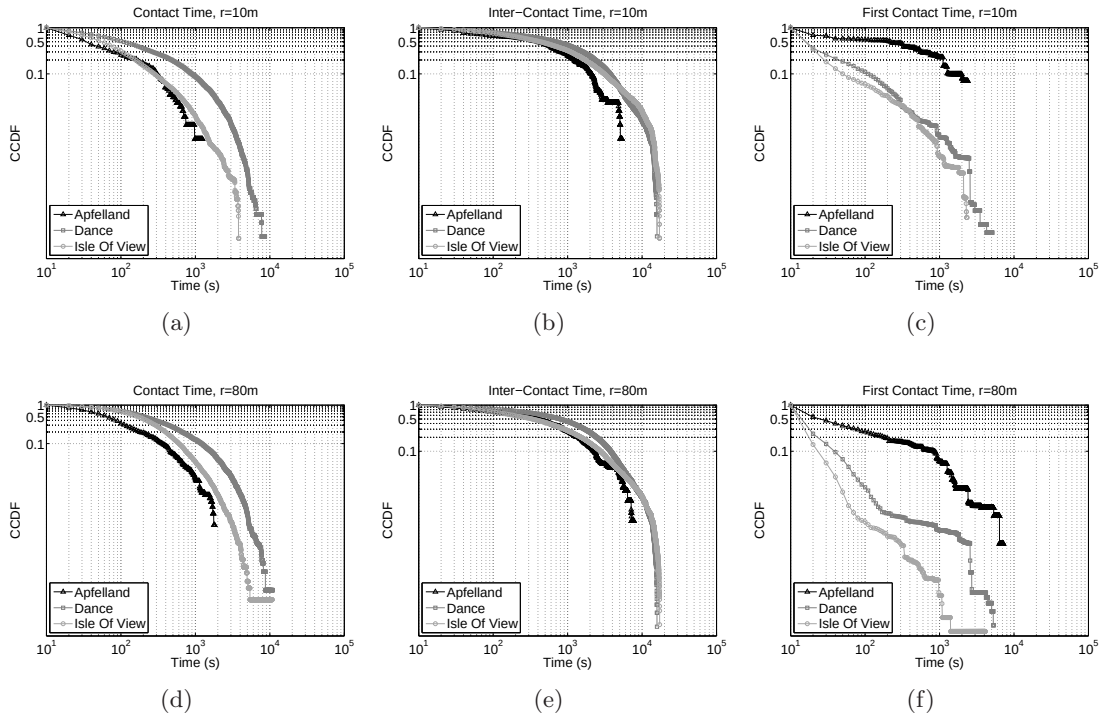


Fig. 3.2 – Temporal Analysis : CCDF of contact opportunity metrics for three SL lands.

trates some differences between these lands : in Apfel Land users have to wait for a long time before meeting their first neighbor. The median FT is around 300 seconds for Apfel Land, while it is less than 20 seconds for the other two lands when $r = r_b$. The FT improves a lot when increasing r : the median is around 30 seconds for Apfel Land and less than 5 seconds for the other lands.

In this work, we used Maximum likelihood estimation (MLE) [33] for fitting our traces to some well-known mathematical models of contact-time distributions. The three baseline models we used are summarized in Table 3.1. For each distribution we give the basic functional form $f(x)$ and the appropriate normalization constant C.

name	$f(x)$	C
power-law	$x^{-\alpha}$	$(\alpha-1)x_{min}^{\alpha-1}$
power-law with cutoff	$x^{-\alpha}e^{-\lambda x}$	$\frac{\lambda^{\alpha-1}}{\Gamma(1-\alpha,\lambda x_{min})}$
exponential	$e^{-\lambda x}$	$\lambda e^{\lambda x_{min}}$

Tab. 3.1 – Definition of the power-law distribution and other reference statistical distributions used for the Maximum likelihood estimation.

We applied MLE to analyze the distribution of contact times. The CCDF of the contact time CT depicted in Fig. 3.2a-d can be best fit to an *exponential distribution* with coeffients λ as shown in Tab. 3.2 when $r = r_b$ and $r = r_w$ for all three islands ApfelLand, Dance Island and Isle Of View.

land name	λ with $r = r_b$	λ with $r = r_w$
ApfelLand	0.010	0.007
Dance Island	0.003	0.002
Isle Of View	0.008	0.004

TAB. 3.2 – Coeffients of exponential distribution in users' contact time CT.

MLE applied to our empirical data on inter contact times indicates that the best fit is the *power-law with cutoff distribution*. We observe in Fig. 3.2b-e that the CCDF of the inter contact time ICT has two phases : a first power-law phase and an exponential cut-off phase. The values of the coefficients of these distributions are shown in Tab. 3.3 for ApfelLand, Dance Island and Isle Of View, when $r = r_b$ and $r = r_w$. Note that in order to improve the clarity of the Figures, in Fig. 3.2a-b-d-e we do not show the slope corresponding to fitting distributions with the coefficients we computed using MLE.

land name	α with $r = r_b$	λ with $r = r_b$	α with $r = r_w$	λ with $r = r_w$
ApfelLand	0.34	0.00049	0.46	0.00045
Dance Island	0.47	0.00041	0.44	0.00037
Isle Of View	0.42	0.00046	0.59	0.00041

TAB. 3.3 – Coeffients of power-law with cutoff distribution in users' inter-contact time ICT.

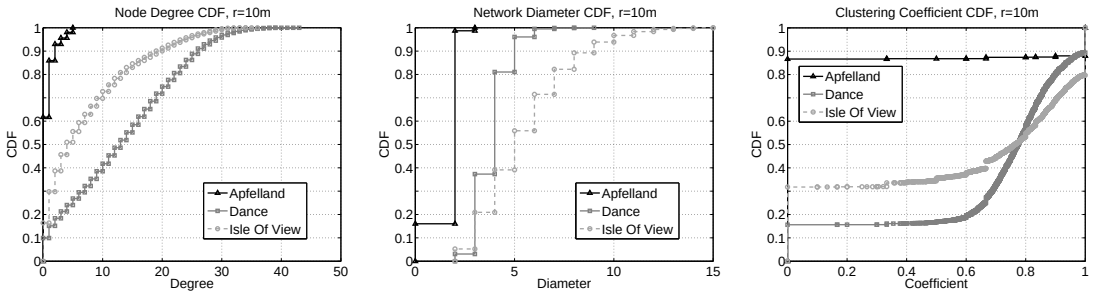
These results are quite surprising : we obtained a statistical distribution of contact opportunities that mimics what has been obtained for experiments in the *real world* [70, 28, 97]. It should be noted, however, that human activity roughly spans the 12 hours interval, while even the most assiduous user which we were able to track in our traces spent less than 4 consecutive hours on SL.

Spatial analysis

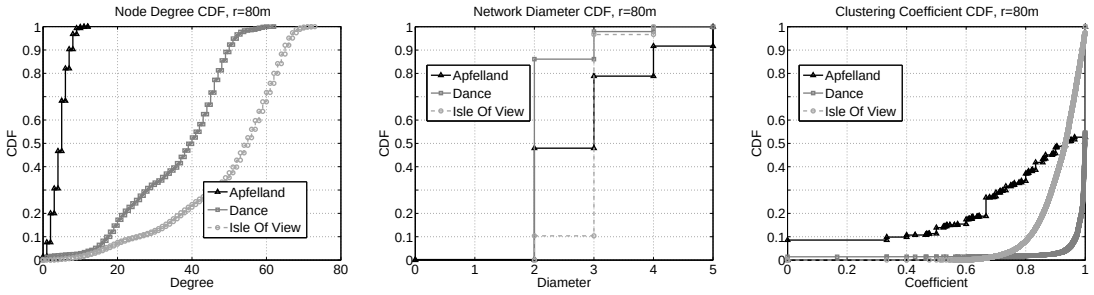
We present here the metrics we used to perform the spatial analysis of our traces :

- *Node degree :* is defined as the number of neighbors of a user when the communication range is fixed to r ;
- *Network diameter :* is computed as the longest shortest path of the largest connected component of the communication network formed by the users. We used the largest component since, for a given r, the network might be disconnected ;
- *Clustering coefficient :* is defined as in [122] : we compute it for every user and take the mean value to be representative of the whole communication network ;
- *Travel length :* for every user v_i we compute the distance covered from its login to its logout coordinates in SL ;
- *Travel time :* for every user v_i we compute the total time spent while moving ; hence, this metric does not include *pause* times ;
- *Zone occupation :* we divided lands in several square sub-cells of size $L \times L$ and computed the number of users in every sub-cell, when $L = 20$ meters.

Network topology : We now delve into a detailed analysis of the communication



(a) Node degree (b) Network diameter (c) Clustering coefficient



(d) Node degree (e) Network diameter (f) Clustering coefficient

Fig. 3.3 – Graph theoretic properties for three selected SL lands.

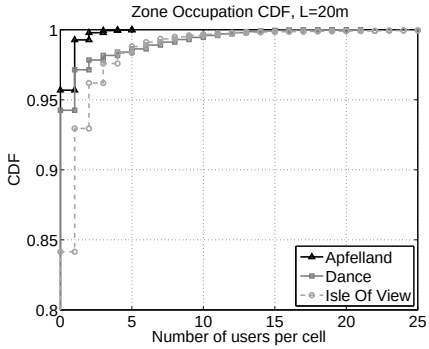


Fig. 3.4 – Spatial distribution of SL users.

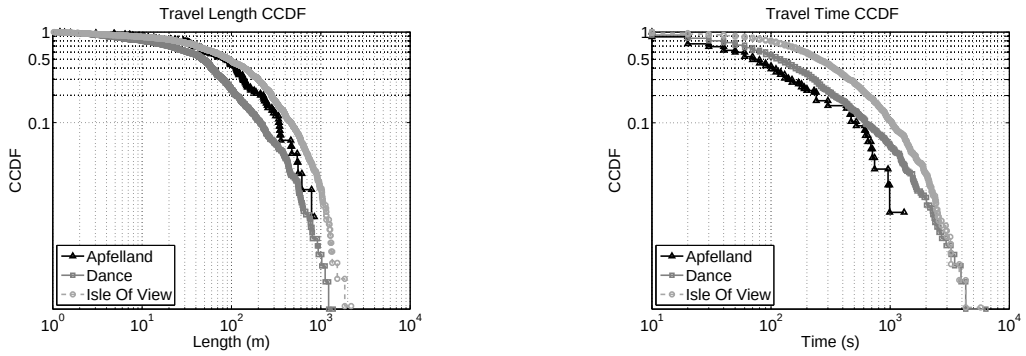
networks that emerge from user interaction when we assume them to be equipped with a wireless communication device covering a range $r \in \{r_b, r_w\}$. Fig. 3.3 illustrates the aggregated (over the whole measurement period) CDF of the node degree, the aggregated CDF of the network diameter and clustering coefficient.

The node degree CDF illustrates a diverse user behavior in each target land : for Apfel Land we observe that 60% of users have no neighbors, for the Dance Island only 10% of users have no neighbors while in the Isle of View, all users have at least one neighbor when $r = r_b$. When the communication range is set to $r = r_w$ all users have at least one neighbor in all lands. The maximum degree and the whole distribution varies a lot between target lands : the main reason lies in the physical distribution of users on a land. In Apfel Land users are relatively sparse while in the Dance Island, for example, most of the users spend

most of the time in a tiny portion of the land : this observation is corroborated[1] by our study on the spatial distribution of users as shown in Fig. 3.4. Although the general trend for all target lands we inspected is that a large fraction of the land has no users, some lands (e.g. Dance Island) are characterized by hot-spots with several tens of users.

The CDF of the network diameter illustrates the impact of different transmission ranges : it is clear that the diameter shrinks for $r = r_w$. We note, however, that for Apfel Land there is an apparent contradiction : for $r = r_b$ the maximum diameter is smaller than for $r = r_w$. This phenomenon is due to the fact we compute the diameter of the largest connected component of the temporal graph formed by users : when the radio range is small (and users are scattered through the target land) we observe the emergence of relatively small connected components, whereas for larger ranges the connected component is large (eventually it includes all users), hence a larger diameter.

In Fig. 3.3 we also plot the CDF of the clustering coefficient for the whole measurement period. Our results clearly point to high *median* values of the clustering coefficient which indicate that the networks we observe are not Erdos-Renyi random graphs[2] : these networks are highly clustered but, due to the small number of concurrent users that can log in to a land and the results on the network diameter, we cannot claim at this time that the graphs that emerge from user interaction have small world characteristics.



(a) (b)

FIG. 3.5 – Trip analysis for three selected SL lands

Trip analysis : using physical coordinates, we were able to study the statistical distribution of the distance travelled by users on the three target lands we analyze in this section. Fig. 3.5 illustrates the aggregate CCDF of the travel length and the travel time for all users.

Fig. 3.5-a provides further hints towards a better understanding of user mobility in the selected target lands. For a confined area such as Dance Island, the vast majority of users travel less than 200 meters ($90th$ percentile). This observation however applies also for open spaces : for Apfel Land, the $90th$ percentile is around 400 meters while it grows up to 500 meters for Isle of View. There is a small fraction of users who travel a very long distance : for the Isle of View, around 2% of users travel more than 2000 meters. Fig. 3.5-b

[1]There is an intuitive reason for this phenomenon : in a discotheque users spend most of their time on the dance floor or by the bar, while in an open space users are generally located more sparsely.

[2]Which are usually characterized by a very small clustering coefficient [122].

is useful to infer the distribution of the times a user takes to travel from her initial point (the first time our crawler tracked the user) to her final point (the last time the user has been seen on the target land).

We applied the MLE method to these metrics and found that for the Travel Length and Travel Time CCDF, showed in Fig. 3.5-a-b, the best fit is again the power-low with cutoff distribution (see Table 3.1).

Our analysis indicated that mobility patterns in a virtual environment share common traits, from a qualitative point of view, with those in the real world. Users are generally concentrated around points of interest and travel small distances in the vast majority of cases. We characterized the graph theoretic properties of network topology emerging from user interaction and found results indicating they are highly clustered.

3.3.4 Levy flights similarity

Rhee et al. proposed a model of human mobility by means of Levy flights [97]. This model can generate similar power-law inter-contact time distributions observed in previous human mobility studies. Based on 1000 hours of GPS traces from 44 volunteers in outdoor mobility settings, they observed that human walks are similar to a truncated Levy walks reported by studies carried out on animals [119, 120, 8]. Levy flight is a type of random mobility in which the trip lengths are distributed according to a heavy-tailed probability distribution i.e. the movement consists of many short flights followed by seldom very long flights. The truncation in heavy-tailed distribution which differs human mobility from a pure Levy flight is explained by the fact that people move within geographical constraints like public traffic and obstacles. However with the coarse granularity in measurement and the few number of participants, the traces used in [97] are useful only when we consider a very sparse network.

In [43] by analyzing the movements of thousand mobile phone users with their registration logs, authors show that mobile users usually visit only their frequented locations. These findings reaffirm that human rarely makes a long trip as indicated by Levy flight model. Recently in [63] authors analyzed about 72000 people's movements recorded by 50 taxicabs during six months. Results show that in such large data, mobility patterns are mainly attributed to the traffic network. Authors found that in this case a random walk model can reproduce with very high correlation coefficient the similar human mobility pattern observed in the obtained traces.

3.3.5 The SLAW mobility

In [74], authors present a new mobility model, namely Self-similar Least-Action Walk (SLAW). SLAW produces synthetic human mobility traces containing the following features which are found recently in trace studies :

– *Truncated power-law distribution in flight length, inter-contact times (ICTs) and pause-time* : In [23, 43, 97], they shown that the flights, which are defined as the straight line distance between two consecutive waypoints, follow a truncated power-

parameter	description	range
α	preference of short distance	1-6
N_{wp}	Number of waypoints	
v_{Hurst}	Hurst parameter to set self-similar waypoints	0.5-1
B_{range}	Clustering range (meter)	
β	Levy exponent for pause time	0-2
T_p^{min}, T_p^{max}	Minimum and maximum pause time (second)	

TAB. 3.4 – Input parameters for SLAW model.

law distribution. This is because upon coming to an area, people plan to reduce the distance of travel by visiting the nearby points before visiting farther places (the "least-action" principle proposed by Maupertuis [35]) The distribution of inter-contact times can also be modeled by a truncated power law distribution [26]. Recently, other studies [97, 67] show that the pause time distribution of human mobility also follow a truncated power-law distribution.

– *Bounded mobility areas* : Gonzalez et al. [43] report that people mostly move only within their frequently visited areas and people may have different mobility area sizes.
– *Fractal waypoints* : The waypoints of humans can be modeled by fractal points [97]. A set of fractal points can be divided into subset such that each subset is a smaller copy of the whole set [81]. This feature is called as "self-similarity" which is attributed to the fact that people are always moving around some popular places.

SLAW is the first model consists of many human mobility patterns reported in the literature. SLAW is developed and validated against 226 days and 101 volunteers GPS traces of human mobility mainly in outdoor sites. The social network is also captured in these traces since participants are students in the same campus or visitors at a theme park. The experiments are long enough to express the regularity of daily movements of humans.

The tool to generate SLAW synthetic traces is available to download for public. The input parameters for SLAW models are describes in Tab. 3.4

3.4 Conclusion

The performance of mobile network applications changes drastically with different mobility models. It can vary as well when the same mobility model is used with different parameters. The choice of a mobility model may significantly influence application performance. Hence mobile application performance should be evaluated with the mobility model that most closely matches the expected real-world scenario. Researchers should define the expected real-world scenario, then make a proper choice of the mobility model to use. In fact, the anticipation of real-world scenario at the beginning of application development can also help to consider whether such application is useful in reality. However, since research in mobile networks is relatively new, very few studies have been carried out to understand what is a realistic model for human. Mobility models may have various properties and exhibit different mobility characteristics. To thoroughly evaluate mobile network applica-

tion performance, it is interesting to use a set of different mobility models. By properly choosing mobility models with different characteristics, we can produce a set of various mobility scenarios spanning the mobility space.

Random mobility models, despite of their simplicity are still useful in mobile network study by carefully setting parameters [64, 63]. Each model has its own advantages and drawbacks :

- The Random Walk Mobility Model produces Brownian motions with small flight lengths, therefore it can be used to evaluate a network where mobility is not high and mobile nodes do not move far away from the initial position. To simulate high mobility the flight length should be large.
- The Random Waypoint Mobility Model is largely used in many simulation studies of mobile networks. This model is simple and supported by many tools. It has been reported to generate realistic mobility patterns. The concerns with this model are : nodes are more likely to cluster in the center of the simulation area and the speed decays over time. But these issues can be eliminated by discarding the transient phase or establishing the steady state as initial settings.
- The Random Direction Mobility Model is unrealistic because it is unlikely that people would spread themselves evenly throughout an area (e.g. a building). Furthermore, it is unlikely that people will only pause at the edge of a given area. A modified version of Random Direction Mobility Model allows nodes to pause and change directions before reaching the simulation boundary but in this case the model is similar to the Random Walk Mobility Model with pause times.
- The Gauss-Markov Mobility Model also provides movement patterns that one might expect in the real-world, if appropriate parameters are chosen. In addition, the method used to force nodes away from the edges of the simulation area. The choice of parameters for Gauss-Markov model requires more experiences from practitioners.

In brief, we can use either the Random Waypoint Mobility Model, the Random Walk Mobility Model or the Gauss-Markov Mobility Model, with proper parameters choice and make sure that the transient phase is eliminated.

Further study on mobility models for mobile network performance is very important. More effort should be invested in examining the movements of human in the real world to produce more realistic mobility models, by studying traces from wireless gateways or cellular networks. Studies to assessing current models against their scope of application are also important.

The study of mobility traces reveals a fact that users usually gather together in some “interest points”. As a consequence, it is difficult from the logistics perspective to set up mobile gateway infrastructure to meet users’ demand. The device-to-device communication in this case can be used to reduce the congestion at overloaded gateway. Our study on replication in mobile networks aims to deal with this reality. In our performance evaluation we use various mobility models, including random and human mobility.

We are not aware of any drawback of the new mobility model SLAW, which does not mean that there is no issue in using this model. However we can try it since it has been validated with the human mobile patterns from real life traces. In our work we do not want

to study the opportunity to meet between individuals, hence we do not need to establish the social connection that requires for example a group mobility model. And considering the density of network we are targeting, a random model with fixed issues like random trip is suitable for our simulation choice. The results in our next chapters confirm this fact by showing very little difference when we compare the results for random mobility (random trip model) and human mobility (SLAW model). These results also confirm the findings of [100], in which authors claimed that there is no difference in simulating random models and map-based models if the road system is sufficiently dense.

3.5 Relevant publication

La, Chi Anh ;Michiardi, Pietro, *Characterizing user mobility in Second Life*, In Proc. of ACM SIGCOMM Workshop on Online Social Network 2008

Chapter 4

Content cache and forward mechanisms in mobile networks

In this chapter, we focus on the problem of sharing information content among mobile users. Information is defined as a piece of data that contains a commonly interesting content e.g. the latest news or local sightseeing information. In this context, most pieces of information are likely to be of general use, i.e. to be interesting to a large number of users that form a "content popularity", and therefore a sensible dissemination and caching policy would be desirable. In such an environment, few and far between access points or gateway nodes, we consider a highly populated network area where user devices are equipped with a data cache and communicate through the device-to-device networking paradigm. Users create a cooperative environment where information is exchanged among nodes in a peer-to-peer fashion. In particular, they form a pure peer-to-peer system, whose nodes may simultaneously act as both "clients" and "servers" to the other nodes in the network. We assume that users create a cooperative environment where information is exchanged among nodes in a peer-to-peer fashion. The nodes storing an information copy are supposed to act as *providers* for this content to nearby nodes. To share the content distribution burden (energy consumption), nodes act as providers for a limited time, before handing over the information to other nodes. We then try to answer the following questions :

- Regardless of how the information is distributed at the outset, can simple cache-and-forward mechanisms achieve a target information distribution ? Is the system able to identify, in a distributed manner, where the information should be stored in the network in order to reduce information access distance ?
- As mentioned above, a node storing the information acts as provider for that information ; of course, this role may result in a high toll from nodal resources in terms of bandwidth or power consumption ; it is therefore advisable that the role of content provider be handed over to neighboring nodes quite frequently, without altering the information distribution. Given such cache-and-forward mechanisms, do nodes evenly share the role of provider ? And, are they equally burdened when they take on the role of provider ?

Traditional approaches to information caching in communication networks [9, 94, 93, 109] are based on the solution of linear programming problems, which often require global knowledge of the network condition, or lead to quite complex solutions that involve significant communication overhead. Distributed algorithms for allocation of information replicas are instead proposed in [109, 18, 19]. These solutions typically involve significant communication overhead, especially when applied to mobile environments, and focus on minimizing the information access cost or the query delay. Unlike previous approaches, we propose and analyze a solution addressing the above issues that is a fully distributed, uncoordinated cache-and-forwarding approach and is information-oblivious, i.e., not requiring knowledge of the content stored by users.

4.1 System objectives

We investigate the problem of spreading information contents in a mobile wireless network with mechanisms embracing the peer-to-peer paradigm. In our vision, an information dissemination mechanism should be :

- Fully distributed : There should be no required centralized component.
- Content-transparent : The system should not require knowledge of the contents stored by the neighboring users to reduce the overhead.
- The mechanism should result in a desirable distribution of information replicas in the network.
- The information should be evenly and fairly carried by all nodes in their turn to deal with the load balancing in distributed systems.

We show that these goals can be achieved by simple cache-and-forward mechanisms, provided that a sufficient number of information replicas are injected into the network, by letting the information move across nodes according to two well-known mobility models, namely random walk and random direction. The proposed approach works under different network scenarios, is fully distributed and comes at a very low cost in terms of protocol overhead.

In particular, motivated by the need of a balanced load distribution among the provider nodes and of an equal quality of service provisioning to the users, we target a uniform distribution of contents, either over the network spatial area or over the network nodes. With this aim in mind, we investigate the applicability of the two cache-and-forward mechanisms to disseminate information across the network. Both strategies, using the simulation setup are proven to yield a distribution of the information copies that is close to the target distribution, regardless of the considered network scenario. Also, the obtained results show that the level of fairness in distributing the burden among provider nodes depends on the number of information copies stored in the network.

4.2 Related work

Our study is related to the problem of optimal cache placement in wireless networks. Several works have addressed this issue by exploiting its similarity to the k-median problems. These problems are NP-hard and a number of constant-factor approximation algorithms have been proposed for each of them [60, 9, 93] ; these algorithms however are not amenable to an efficient distributed implementation.

Distributed algorithms for allocation of information replicas are proposed, among others, in [45, 109, 125, 51]. These solutions typically involve significant communication overhead, especially when applied to mobile environments, and focus on minimizing the information access cost or the query delay. In our work, instead, we consider a cooperative environment and aim at a uniform distribution of the information copies, while evenly distributing the load among the nodes acting as providers.

In the context of sensor networks, approaches based on active queries following a trajectory through the network, or agents propagating information on local events have been proposed, respectively, in [99] and [20]. Note that both these works focus on the forwarding of these messages through the network, while our scope is to make the desired information available by letting it move through nodes caches.

From the content distribution perspective, Sbai et al. [101] propose the adapted Bittorrent version for spontaneous multi-hop wireless networks in order to provide a new environment for sharing content among communities of end users. This approach also focuses on the time to download and sharing ratio among users, but requires the use of routing protocols and overlay structures which introduces a lot of overhead.

4.3 P2P cache and forward mechanisms

We start by addressing the problem of where the information copies should be cached in the network so as to obtain the desired content distribution.

We consider a tagged[1] information and we target the two desired distribution : the first uniform over the spatial area covered by the network (*spatial uniformity*) the second following the layout of the network topology (*nodal uniformity*). Spatial uniformity is motivated by the need to guarantee equal access to the information over the whole service area (e.g., probability of finding the content and information delivery latency) to all network users, while nodal uniformity allows the information density to match the node density and, therefore, to cluster the information where the demand is higher :

– *Spatial uniformity :* since we consider a square area where nodes are deployed and we seek a uniform dissemination of content over the network area, the target distribution is the solution to the bidimensional case of the hypercube line picking problem [117],

[1] I.e., we assume information to be uniquely identifiable.

which is known to be :

$$q(x) = \begin{cases} 2x\left(x^2 - 4x + \pi\right) & \\ & \text{if } 0 \leq x < 1, \\ 2x\left[4\gamma - \left(x^2 + 2 - \pi\right) - 4\tan^{-1}\gamma\right] & \\ & \text{if } 1 \leq x < \sqrt{2}, \end{cases}$$

with $\gamma = \sqrt{x^2 - 1}$.

– *Nodal uniformity :* in order to test the uniformity of providers over the network nodes, we take as a reference distribution the empirical distribution of node inter-distances measured in simulation.

To achieve the target distribution, we let the information move across nodes according to two well-known mobility models, namely the random walk [38] and the random direction [89] models, which are often used to represent the movement of user nodes in wireless networks. In our context, a mobile entity is not a network node but, rather, a copy of the tagged information which "hops" from a user node that just stopped being a provider for that information onto another node which will become the new content provider. We apply the two mobility models and develop the dissemination strategies detailed below.

– **The random walk dissemination (RWD) strategy.** We consider the simplest random walk possible, in which each mobile entity, i.e., each copy of the information content, roams the network by moving from a node to a one-hop neighbor selected with equal probability. Each node caches the information for a fixed amount of time, and then hands it over to the next selected hop in the information copy visit pattern. This approach requires trivial node operations and introduces low overhead, thus representing a lower-bound benchmark for more advanced information mobility models.
– **The random direction dissemination (RDD) strategy.** It implies that each mobile entity alternates periods of movement (move phase) to periods during which it pauses (pause phase). In our context, the pause phase corresponds to the time period during which the information copy is stored at a provider node. The move phase starts at the time instant when the current information provider hands over the content to one of its one-hop neighbors, and it ends when the new provider is reached by the information copy. The new provider is identified by first selecting a target location : the closest node to that location becomes the new provider. To this end, at the beginning of a move phase, the current provider independently selects the direction and the distance[2] for the movement of the information content, thus identifying a target location whose position is included in the content messages. We introduce a simple broadcast-based application-level routing scheme that allows information to be moved towards the target location, with each forwarder selecting as a next hop the neighbor that best fits the ideal trajectory designed by the original provider. The neighbor selection process is performed in a reactive manner, as it involves an exchange of advertisement (by the forwarder) and reply (by candidate next hop neighbors) messages at each movement hop. When a node has no neighbors closer than itself to the target position, it elects itself as the new provider, and the pause phase starts again. We make some remarks as follows. First, this scheme

[2]Note that randomly selecting a travel distance is equivalent to randomly selecting speed and travel time.

requires nodes to be capable to estimate their position (i.e., through GPS), a fair assumption in most practical scenarios. Second, the information moves across user nodes, thus it may be transmitted along a direction that just approximates the planned trajectory, or it may be stored at a node that is nearby (but not exactly at) the selected geographical destination. Third, geographical areas devoid of nodes that can support the information movement may be encountered during move phases : in that case, the current forwarder assumes a *boundary* has been hit, and applies a reflection to the movement angle.

Algorithm 4.1 infoHandOver

```
if randomWalk then
  Select random i ∈ neighborSet
  handOverTo(i)
else if randomDirection then
  Select random (x, y) ∈ A
  Select i ∈ neighborSet | distance(i,x,y)=min
  handOverTo(i)
end if
```

Algorithm 4.2 uponReceiveInfo(j)

```
if randomWalk then
  if hasInfo then
    bounceInfo()
  else
    storeInfo()
  end if
else if randomDirection then
  if distance(j, x, y) = min then
    if hasInfo then
      bounceInfo()
    else
      storeInfo()
    end if
  else
    Select i ∈ neighborSet | distance(i,x,y)=min
    handOverTo(i)
  end if
end if
```

If a node receives the information while it stored that information already, it makes a "bounce" of information to another node immediately using the same RWD/RDD procedure. The algorithm for hand-over information with RWD and RDD are described in Alg. 4.1 and Alg. 4.2. As already mentioned, using the RWD and RDD strategies translates into a fully-distributed, low-overhead solution. The characterization of the spatial distribution of randomized algorithms applied to node mobility has been investigated in the literature from an analytic perspective, in an ideal setting. Indeed, if the network topology can be represented as an undirected, connected, non-bipartite graph, then the distribution of nodes moving according to the random walk model converges to a unique

stationary distribution regardless of the initial distribution, and this stationary distribution is uniform in the case of regular graphs [3] [13]. As for the random direction model, in [89] it has been shown that, if at time $t = 0$ the position and the orientation of mobile nodes are independent and uniform over a finite square area, they remain uniformly distributed over the area for all time instants $t > 0$, provided that the entities move independently of each other.

In the context of this work, we cannot trivially use similar techniques to [89, 13] and show that the same randomized algorithms applied to information instead of mobile nodes achieve a uniform distribution. The dissemination mechanisms we apply to information operate on realistic network deployments that do not have a regular structure, hence the results for the random walk model do not directly apply. Furthermore, the combined effects of node mobility and information mobility hinder the analytical task, especially for the RDD strategy where the information only approximately reaches its geographical destination.Lastly, in this work we are interested in both spatial and nodal uniformity, and for the latter we are not aware of any previous studies that prove convergence to a uniform distribution in a general scenario. Therefore, in the following, we carry out a thorough simulation campaign to investigate the actual distribution of the information that is obtained through our approach and it's distance from the target uniform distribution.

4.4 Experimental set-up and methodology

We use the *ns*-2 network simulator, where all nodes are equipped with standard 802.11b interfaces, with 11 Mbps fixed data transmission rate. To evaluate the behavior of the cache-and-forward strategies discussed in Section 4.3, we implemented a simple *application* that allows nodes to query providers through limited-scope flooding. Queries can traverse a maximum number of hops, $h_{max} = 5$, before being discarded [4]. We improve the query propagation process by adopting the PGB technique [90] to select forwarding nodes that relay queries to their destinations. Sequence numbers are used to detect and discard duplicate queries and avoid the *broadcast storm* phenomenon [92]. Upon reception of a query, a provider replies with a probability that is inversely proportional to the number of hops traversed by the query message. This is done to further mitigate the overhead of any duplicate query that would reach multiple providers.

In the following, we define the simulation settings we analyze in this work. Note that all results presented in the remainder of this work are averaged out over 10 simulation runs, each with a randomized selection of initial information providers. Simulation time is set to 10,000 seconds, unless specified otherwise. Moreover, we assume a network composed of $N = 2000$ nodes that are spatially distributed on a square area $\mathcal{A}$ of 500 m side. For the sake of simplicity in this first step, we assume that all nodes are interested in the content. Each node has a transmission range of 20 m resulting into 9–10 neighbors for each node on average. When employing the RDD scheme, providers characterize the information move phase by randomly choosing angles that are uniformly distributed in $[0, 2\pi]$, and exponentially distributed distances, with the mean value 100 m. We study both *static* and

[3] A graph is regular if each of its vertices has the same number of neighbors.

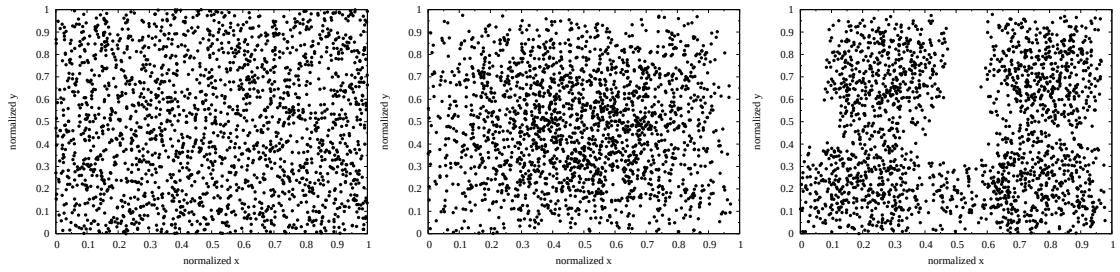
[4] The choice of $h_{max} = 5$ is arbitrary : queries can roughly propagate over half of the network diameter, given our settings.

mobile cases, as will be detailed below.

4.4.1 Nodes placement

We define the following static node deployments, samples of which are depicted in Fig. 4.1 :

- *Uniform distribution* : nodes are uniformly placed on $\mathcal{A}$;
- *Stationary distribution* : as we will compare results for both static and mobile cases, [5], we consider a deployment, where, as discussed in [16], nodes are more often located towards the center of the network area ;
- *Clustered distribution* : we assume nodes to be deployed in four equally sized clusters. Each cluster corresponds to a "point of interest" around which nodes are located. Nodes are also placed in-between clusters so as to ensure network connectivity. In practice, we implement the random trip model as defined in [17] and take a snapshot of the network topology as our initial node distribution.



(a) Uniform (b) Stationary (c) Clustered

Fig. 4.1 – Snapshots of node placement used in our experiments.

4.4.2 Node mobility

The impact of node mobility on the dissemination mechanisms we designed is analyzed for the following mobility models :

- *Stationary Random Waypoint* : nodes are initially deployed according to the stationary distribution of the node mobility model to reduce the transient effect[16] (resulting in nodes being more often located towards the center of the network area) ; then, each node selects a random destination in $\mathcal{A}$ and moves towards it at a constant speed selected uniformly at random from the interval [0,6] m/s with a mean of 3 m/s (pedestrian speed). The pause time is set to 10 s.
- *Random trip* : following the definition in [17], nodes revolve around four "points of interest". The initial node deployment conforms to the clustered distribution defined for the static case in Fig.4.1(c). The stationary random waypoint model defined above guides node movements inside a cluster. Inter-cluster mobility is allowed with probability 0.3.

[5] Note that the stationary distribution is in connection to the random waypoint mobility model.

4.4.3 Parameter space

We now define the parameters used in our evaluation, accounting for the initial distribution and number of information providers in the network, as well as the query behavior of mobile nodes.

- *Number of information providers* : at the beginning of each simulation run, a predefined number of providers is randomly chosen among all nodes in the network. $\mathcal{C}(t)$ is the number of providers available at time t ; we choose $\mathcal{C}(0) \in \{20, 50, 100, 200, 400\}$.
- *Information caching time* : when taking up the role of information provider, a node i keeps a local copy for a time τ_i. In this work, we assume $\tau_i = \tau \ \forall i$ with $i = 1, \ldots, N$. In the following we present results for $\tau \in \{10, 100\}$ seconds.
- *Information demand* : we assume nodes to issue queries to information providers using the simple application defined above. Without loss of generality, we focus on one information content (of size equal to 1KB) that is made available in the network. Users' demand for the available information is modeled through a query rate which we assume to be common to all users, $\lambda_i = \lambda = 0.0025$ req/s $\forall i$ with $i = 1, \ldots, N$. The aggregate query rate Λ over all nodes depends on the number of information providers currently active in the network[6], i.e., $\Lambda(t) = (N - \mathcal{C}(t))\lambda$.

4.4.4 Evaluation metrics

To understand to which extent the information distribution achieved by our dissemination techniques resembles the desired content diffusion, we employ the well-known χ^2 goodness-of-fit test on the inter-distance between information copies. As a matter of fact, we can compare the measured inter-distance distribution against the theoretical distribution of the distance between two points, whose position is a random variable following the target distribution. Using inter-distances instead of actual coordinates allows us to handle a much larger number of samples (e.g., $\mathcal{C}(t)(\mathcal{C}(t) - 1)$ instead of just $\mathcal{C}(t)$ samples) thus making the computation of the χ^2 index more accurate. As discussed before, we consider the two reference distributions in Sec. 4.3 :

To compare information and reference distributions, we will resort to a visual comparison of the PDFs, as well as to the χ^2 goodness-of-fit test in time-serie plots. Then, we provide a basic performance evaluation of the information query process achieved by our application, and focus on the following metrics.

- *Cumulative provider time :* we evaluate the load balancing properties of the different information dissemination strategies by computing the cumulative time $\hat{\tau}_i$ each node i spends as an information provider. Given that the cache time τ is deterministic, we can compute $\hat{\tau}_i = \tau \times \mathcal{I}_i$, where $\mathcal{I}_i$ accounts for the number of times node i takes up the role of information provider during the simulation time.
- *Served queries at each information provider* : we measure the cumulative number of served queries for each information provider j. Note that this metric is also useful to understand the impact of the hop-based reply policy implemented by provider nodes (i.e., the likelihood of replies decreases with the increase of hops traversed by the

[6]Indeed, providers do not issue requests to access the content

query).
– *Euclidean distance to access information* : we measure the cumulative Euclidean distance from a node to its *closest* information provider, every τ seconds. The distance to access information is the result of the spatial distribution of information in the network and can be used to measure how "fair" our maechanisms are toward to each querying node.

4.5 Simulation results

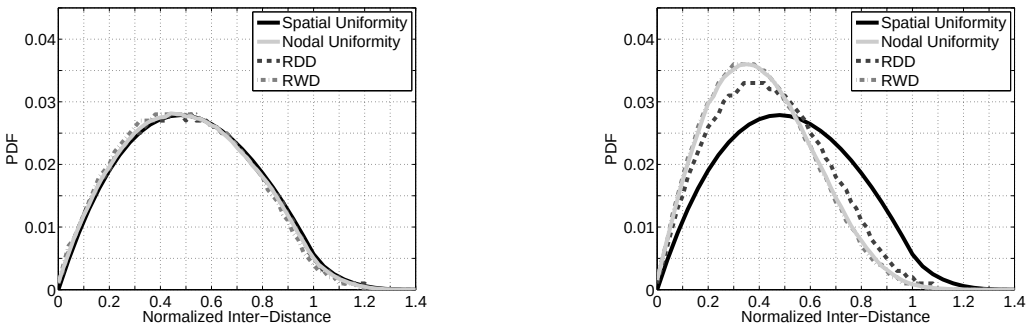
In this section, we look at how the RWD and RDD strategies can achieve the first two objectives outlined in the introduction of this chapter : a desirable distribution of the information and a fair distribution of information burden across the provider nodes. In the set of results we present, no information drop is allowed ; indeed, for both the RWD and RDD strategies, a provider that hands the information over to another node considers the transfer as successful only if it receives an acknowledgment message, otherwise it repeats the procedure by selecting a different neighbor. The duplication probability we obtained by implementing such an application was negligible (order of 10^{-5}). Thus, we can consider that the overall number of providers does not change during the simulation time (i.e., $\mathcal{C}(t) = \mathcal{C}(0)$) ; additionally, the query rate λ is set to a constant value equal to 0.0025 req/s, resulting, as discussed in Sec. 4.4.3, in $\Lambda = 4.5$ req/s.

4.5.1 Spatial distribution of content

We now focus on the properties of the information distribution of our cache-and-forward strategies RWD and RDD, that is, we study where information replicas are cached in the network in a variety of scenarios. The following results are obtained for different static node deployments (uniform and clustered) and node mobility models (stationary random waypoint and random trip models) when $\mathcal{C}(0) = 200$ and the caching time $\tau = 10$ s. Note that the probability density functions (PDF) we show hereafter are computed from samples collected over all the simulation time.

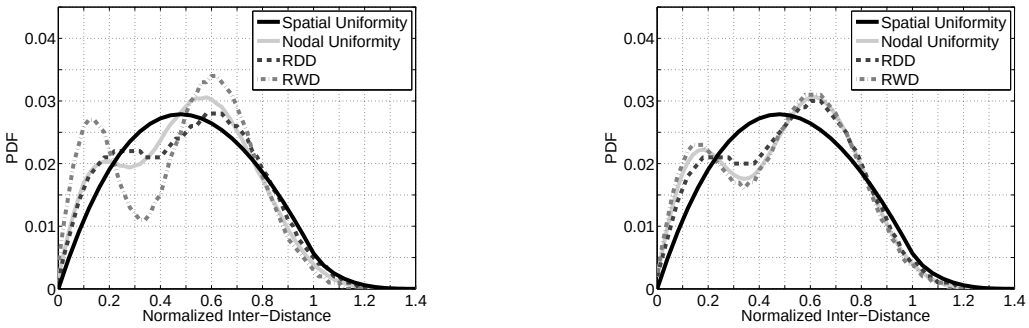
Both Fig. 4.2 and Fig. 4.3 indicate a target PDF corresponding to the two information distributions we take as reference, *spatial uniformity* and *nodal uniformity*, as defined in Sec. 4.4.4. Explicitly, Fig. 4.2 shows the experimental PDF of the information copies for both dissemination policies when nodes are deployed according to the uniform distribution and move according to the stationary random waypoint model. Similarly, Fig. 4.3 shows the PDF of the inter-distance between information copies in the static and mobile case when nodes are deployed in clusters and move according to the random trip model.

As shown in Fig. 4.2(a), both RWD and RDD strategies yield information distributions that closely overlap both the nodal and spatial uniformity targets. Indeed, if nodes are static and uniformly distributed on the network area, then information mobility can be thought of as equivalent to node mobility, with the constraint that information can only move in well-defined positions that are given by the original network deployment. Our simulation



(a) Static scenario (b) Mobile scenario

FIG. 4.2 – PDF of the inter-distance between information copies normalized to $\mathcal{A}$, for the RWD and RDD dissemination policies in static (*uniform* and *stationary*) and mobile scenarios (*random waypoint*) when $\mathcal{C}(0) = 200$ and $\tau = 10$ s (The target distribution is the spatial uniformity).



(a) Static scenario (b) Mobile scenario

FIG. 4.3 – PDF of the inter-distance between information copies normalized to $\mathcal{A}$, for the RWD and RDD dissemination policies in static (*clustered*) and mobile (*random trip*) scenarios when $\mathcal{C}(0) = 200$ and $\tau = 10$ s (The target distribution is nodal uniformity).

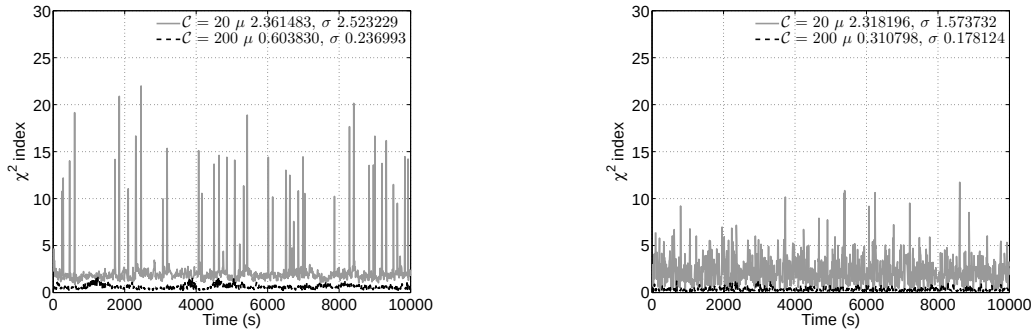
results, backed up by prior analytical studies [89, 13] on node mobility, indicate that information replicas achieve a uniform distribution, both in space and over the nodes.

Fig. 4.2(b) illustrates the implications of the combined effects of node and information mobility. In this case, the RWD cache-and-forward strategy approximates very well the nodal uniformity target, whereas spatial uniformity, represented now by a different distribution, is not achieved. The reason lies in the fact that, by moving the information of a single random hop at a time, the RWD scheme is strictly bound to the nodal distribution. The RWD distribution and the reference distribution for nodal uniformity exhibit a lower mean than the spatially uniform distribution : indeed, node mobility reduces the inter-distance between nodes, which are more likely to be moving around the center of the network area. When considering the RDD strategy, we observe that the information distribution it achieves falls in between the two reference distributions. As a matter of fact, the RDD strategy tends to a uniform distribution over space ; however, the movement of carriers biases such distribution towards that shaped on the nodes layout. In brief, the RDD policy outperforms the RWD policy in better approximating the reference distribution in the static case, while, in the mobile case, the RWD and the RDD perform similarly.

The results for the clustered scenario are shown in Fig. 4.3. When nodes are static and distributed in clusters, there are some parts of the network area that are not home to any provider. Hence, it is reasonable that spatial uniformity cannot be achieved. In this case, having a cache storing information where there are no nodes to access it would serve little purpose. For this reason, in a clustered scenario, nodal uniformity appears to be a more sensible target and our results confirm the effectiveness of our cache-andforward strategies to approximate a desirable distribution of information. While Fig. 4.3(a) shows that the RDD policy is more accurate than the RWD strategy in approximating nodal uniformity, in Fig. 4.3(b) the two cache-and-forward schemes achieve similar results. Indeed, when nodes revolve around several points of interest and are free to move from one cluster to another, provider nodes can also be found in parts of the network area with a low node density. In order to assess the impact of the number of information replicas, i.e., the number of providers, in the network, we focus on a single scenario, the static uniform one, where we already observed that the two reference distributions match (see Fig. 4.2(a)). Further insights can be gained by observing more closely the behavior of the two information dissemination techniques in a simple case : we therefore focus on static stationary scenarios and emphasize, using the time series of the χ^2 index, the differences between the target spatially uniform and the experimental distributions. The χ^2 index is computed considering the measured and the objective probability density function : the smaller the index, the better the fit. The evolution of the χ^2 index is plotted over time when the RWD and the RDD are applied in Fig. 4.4 in which we considered the number of information copies concurrently moving through the network to sum to $\mathcal{C}(0) = 20,\,200$ and the caching time to be equal to $\tau = 10$ s ; In this figure we also report the average μ and standard deviation σ of the χ^2 index. We note that an increase of one order of magnitude (from 20 to 200) in the number of providers, which implies more nodes bearing the cost of serving information, differently affects our mechanisms : Fig. 4.4(a) indicates that for the RWD policy the mean χ^2 index improves by almost four times while Fig. 4.4(b) shows a tenfold improvement for the RDD mechanism, although both schemes exhibit similar average values when the number of providers is low. It should be noted that an increased number of providers greatly helps in stabilizing the information distribution, as testified by the standard deviation of

the χ^2 index in both schemes. Fig. 4.4(b) also pinpoints an important property of the RDD strategy : a small standard deviation of the χ^2 index implies that at all times, the achieved distribution does not diverge too much from the target, which ensures a fair access to information by client nodes. The results we presented in this section support the idea we advocate in our work : exploiting well-known mobility models to derive cache-and-forward mechanisms is indeed an efficient, light-weight alternative to complex (and centralized) techniques akin to facility location and k-median problems previously appeared in the literature. Provided that simple distributed schemes can achieve a desirable information distribution, we now move forward and examine the implications of our mechanisms from the perspective of provider and client nodes.

When nodes are static and distributed in clusters there are some parts of the network area that do not physically host any provider. Hence, it is reasonable that spatial uniformity cannot be achieved. In this case, having a cache storing information where there are no peers to access it would serve little purpose. Hence, in a clustered scenario, nodal uniformity seems a more reasonable target and our results indicate that our policies approximate well enough the target disribution. As opposed to our previous observations, it is not straightforward to conclude that our information dissemination mechanisms are able to mimic nodal uniformity, and we are not aware of any theoretic studies in the literature that support our simulation results. We emphasize here that nodal uniformity in the cluster scenario implies both an improved "quality of service" to peers and a better load balancing among providers.



(a) RWD strategy

(b) RDD strategy

FIG. 4.4 – χ^2 index for the static stationary scenario : mean (μ) and standard deviation (σ) with 10 s observation intervals, for $\mathcal{C}(0) = \{20, 200\}$ and $\tau = 10$ s.

4.5.2 Load balancing

We now turn our attention to the important question of load balancing across providers. For brevity, below we present just a subset of the results we derived. In particular, since the RDD manages to provide a better approximation to the target information distribution than RWD, we only show the performance of the RDD policy. Also, we present results only for the static uniform scenario and the mobile network with random waypoint mobility, since a similar performance is achieved under clustered network topologies.

In Fig. 4.5 we plot the complementary distribution function (CCDF) of the cumulative

time a node is serving as an information provider (i.e., the provider time) over the whole duration of our experiments, that is, we normalize the provider time to the simulation time. The results are presented for the RDD policy in a static uniform scenario, for different values of the caching time τ and when the initial number of information providers sums to $\mathcal{C}(0) = 20$ and to $\mathcal{C}(0) = 200$ (which correspond, respectively, to 1% and 10% of the total number of nodes). Looking at the figure, we observe that when we increase $\mathcal{C}(0)$ from 20 to 200, the load is spread more uniformly across the nodes since there is an increased opportunity for being (randomly) selected as information provider. The effect of an increased caching time τ from 10 s to 100 s, is, instead, a translation of the CCDF to higher values, without affecting the load distribution.

In Fig. 4.6, we plot the same result for a mobile scenario with random waypoint mobility. We observe that the CCDF is less skew than in the static case, which means the load is shared more uniformly thanks to the mobility.

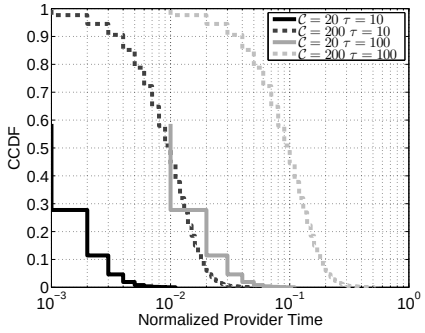


Fig. 4.5 – CCDF of the time a node spends in provider mode, normalized to the total simulation time, for the RDD policy in a static uniform scenario.

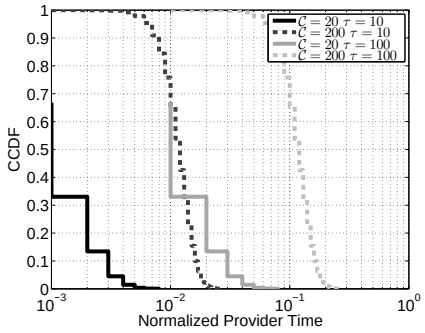
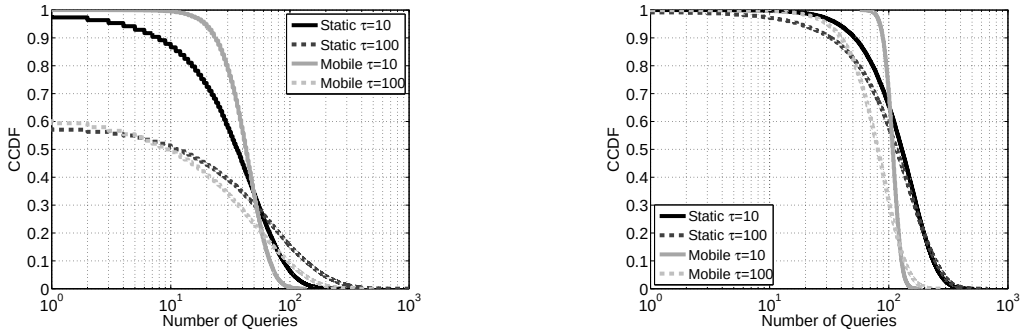


Fig. 4.6 – CCDF of the time a node spends in provider mode, normalized to the total simulation time, for the RDD policy in a mobile uniform scenario with random waypoint mobility.

We now look deeper at the impact of different scenarios and simulation parameters on the effective load that an information provider supports in terms of number of served queries. Note that the number of served queries is not equal to the number of queries a provider receives because of the reply behavior described in Section 4.4.

Fig. 4.7(a) and Fig. 4.7(b) present the CCDF of the number of queries served by the provider nodes, respectively, when $\mathcal{C}(0) = 20$ and $\mathcal{C}(0) = 200$. Both the static uniform

scenario and the mobile scenario with random waypoint mobility are considered. Looking at the plots, we note that an increased number of initial providers is effective in spreading the query load more evenly, especially in the static case. In the case of the static topology, when $\mathcal{C}(0) = 20$, roughly 50% of providers never get a chance to satisfy a user request, whereas with $\mathcal{C}(0) = 200$, about 60% of providers are serving a number of queries comprised in the interval $[70, 150]$. The combined effect of node mobility and an increased number of initial providers is striking : Fig. 4.7(b) indicates that approximately 95% of providers serve roughly the same amount of queries. Thus, node mobility, that at a first sight could be considered harmful to information distribution mechanisms, turns out to be a good ally in terms of load balancing.



(a) $\mathcal{C}(0) = 20$ (b) $\mathcal{C}(0) = 200$

FIG. 4.7 – CCDF of the total number of queries served by information providers for the RDD policy in static and mobile scenario.

4.5.3 Information access distance

Lastly, we take the perspective of users issuing queries to access information held by providers. In Fig. 4.9 we plot the cumulative distribution function (CDF) of the Euclidean distance from a querying node to the closest provider, for the mobile scenario with random waypoint mobility and $\tau = 10$ s. More specifically, we study the impact of an increasing number of initial providers $\mathcal{C}(0)$, when we let this simulation parameter grow from 20 to 400 providers (i.e., from 1% to 20% of the total number of nodes). Both the mean distance to access information, ranging from 50 m to 15 m, and the variance of the CDF, shrink considerably when increasing the number of initial providers. Given that the node radio range is set to 20 m, the implications of this result are the following : when a sufficient number of initial providers is injected into the network (i.e., 200–400), nodes may access the information within one hop, whereas an insufficient number of initial information copies (i.e., 20–100) may constrain a node to propagate its query over multiple hops to retrieve the information. Fig. 4.8 shows the same result for the static case. We see the distance is slightly longer but the difference is not considerable.

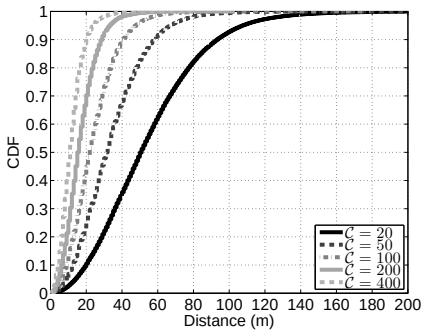


Fig. 4.8 – CDF of the Euclidean distance to closest information replica, for the RDD policy in a static scenario.

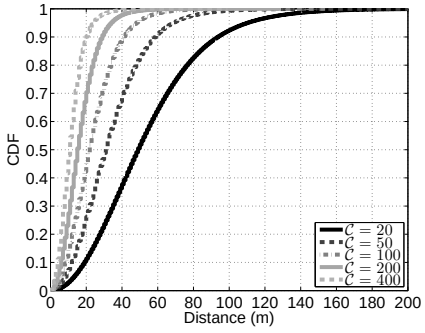


Fig. 4.9 – CDF of the Euclidean distance to closest information replica, for the RDD policy in a mobile scenario with random waypoint mobility.

4.6 Contribution

We considered a peer-to-peer wireless network, where nodes may act as both clients and providers to other network nodes. In such a cooperative environment, we addressed the problem of achieving a desired distribution of information and a fair load distribution among the provider nodes. We designed low-overhead, content-transparent, distributed algorithms that regulate the information storage at the network nodes and allow a fair selection of the nodes acting as providers.

The evaluation we carried out showed that, under a variety of scenarios including static, mobile, and clustered network topologies, despite their simplicity and low overhead, the proposed cache-and-forward schemes (RWD and RDD) achieve the first two objectives defined in this work. Indeed, as long as enough providers are injected into the network (in our experiments, 10% of the total number of nodes), we have that :

- Under static scenarios, the information distribution yielded by RWD and RDD effectively approximates the spatially uniform distribution ; instead, both schemes well approximate a uniform distribution on nodes when they are grouped around point of interests as simulated in the clustered scenario.
- Both cache-and-forward mechanisms achieve a good approximation of nodal uniformty when nodes revolve around landmarks in the clustered scenario ; instead, when nodes move uniformly at random on the network area the quality of approximation of a desired spatially uniform distribution deteriorates for both policies.
- In terms of load balancing, both dissemination strategies evenly distribute the service load across the provider nodes especially when we study the mobile case. Mobility appeared to be a useful ally, instead of a problematic phenomenon, since it helped to achieve an even distribution of the load on providers.

In conclusion, the random cache-and-handover mechanisms provide a solution to achieve a content placement that helps to reduce the average distance to access information. Intuitively as we can see in Fig. 4.4, the more replicas we have, the better χ^2 we can achieve. This observation calls for a further step : we need a mechanism that can determine the number of replicas needed to be stored in the network. In the realistic case where the user content demand varies over time, we need a content replication/drop strategy to adapt the number of information replicas to the changes in the information query rate. We therefore devise a distributed, lightweight scheme that performs efficiently in a variety of scenarios. Additional issues, such as the dynamic adaptation of number of information replicas to time-varying and space-varying content demand or information survival are left for the next chapter.

4.7 Relevant publication

Casetti, Claudio ; Chiasserini, Carla-Fabiana ; Fiore, Marco ; La, Chi Anh ; Michiardi, Pietro ; *P2P cache-and-forward mechanisms for mobile ad hoc networks*, In Proceedings of IEEE Symposium on Computers and Communications, ISCC 2009.

Chapter 5

Distributed solution for content replication in mobile networks

In previous chapter, we introduced random cache and forward mechanisms to maintain a reasonable content access distance and load balancing among nodes in various mobile scenarios, by fixing a predefined number of replicas. From those results, one may raise a question : if we place more replicas in network, is the performance always better ? How many replicas should be placed in a given network to achieve the best performance while minimizing resource usage ? We aim to solve this problem by studying replication mechanisms that can achieve the optimal number of replicas, i.e. to minimize content access distance while taking into account the resource constraints at mobile nodes. Particularly, we study this problem through the lenses of facility location theory. It is worth noting that facility location problems and its variants imply centralized solutions and static environment, while we are studying a distributed mechanism whose the aim is to approximate the optimal solution in a dynamic network topology.

Content replication has been shown to be effective in enhancing performance and reliability of content access for end-users, especially when there is a problem of network congestion and scalability [1]. Recently together with the growth of mobile users, content replication becomes an appealing topic in mobile network research since there is the need to offload the data fetching to device-to-device communication as the solution to deal with the congestion at mobile gateway.

Performance and reliability of content access in mobile networks is conditioned by the number and location of content replicas deployed at the network nodes. Facility location theory has been the traditional, *centralized* approach to study content replication : computing the number and placement of replicas in a network while considering the limited resources at at each mobile device. We find out that this problem can be casted as an capacitated facility location problem.

The endeavour of this work is to design a *distributed, lightweight* solution to the above

[1] See [36] for a detailed survey on the topic

joint optimization problem, while taking into account the network dynamics with human mobility factor. In particular, we devise a mechanism that lets nodes share the burden of storing and providing content, so as to achieve load balancing, and decide whether to replicate or drop the information so as to adapt to a dynamic content demand and time-varying topology. We evaluate our mechanism through simulation, by exploring a wide range of settings and studying realistic content access mechanisms that go beyond the traditional assumption matching demand points to their closest content replica.

5.1 Problem formulation

The problem of content replication and caching has received a lot of attention in the past due to its importance in enhancing performance, availability and reliability of content access for Web-based applications. Here, we inherit the problem of replication typical of the wired-Internet and we discuss why the dynamic nature of wireless networks introduces new challenges with respect to the wireline counterpart.

We explore in this work the concept of content replication in a cooperative wireless environment, where content demand and topology are dynamically changing. Nodes can potentially store data and serve other users through device-to-device communications (e.g., using IEEE 802.11 or Bluetooth). We consider that content has a validity time, after which it has to be discarded and a new version has to be downloaded from a server in the Internet. Furthermore, not all users in the network may be interested in a given content at a given time ; hence, disseminating the information to the nodes according to an epidemic approach [48], or pushing the content to all users, might not be desirable.

Such a scenario introduces several problems to content replication. *Optimal replica placement* is one of those : selecting the location that is better suited to store content is difficult, especially when the network is dynamic. Another prominent issue is *how many content replicas* should be made available to mobile nodes. Clearly, decisions on the placement and number of replicas to be deployed in a network are tightly related problems : intuitively, the latter introduces a feedback loop to the former as every content replication triggers a new instance of the placement problem.

Traditionally, the above content replication problems have been studied through the lenses of classic Facility Location Theory [85]. Optimal placement can be cast as the *k-median* problem, whereas the joint optimization of placement and number of replicas can be studied as an *capacitated facility location* problem ; both these problems are NP-hard for general network topologies.

In this work we focus on *replication* and *replica placement* problems, i.e., we view content replication as a process of its own, rather than a by-product of a query/caching mechanism [36].

Let us now define the context of our work. We investigate a scenario involving users equipped with devices offering Internet broadband connectivity as well as device-to-device communication capabilities (e.g., through IEEE 802.11). Although we do not concern ourselves with the provision of Internet access in ad hoc wireless networks, we remark that

broadband connectivity is where new content is fetched from (and updated).

In order to provide a basic description of the system, we focus first on content being represented by a *single* information object and then extend our mechanisms to multiple objects. We assume the object to be tagged with a validity time, and originally hosted on a server in the Internet, which can only be accessed through the broadband access we hinted at. We then consider a network composed of a set $V = \{1, ..., N\}$ of mobile nodes. A node j wishing to access the content first tries to retrieve it from other devices; if its search fails, the node downloads a fresh content replica from the Internet server and temporarily stores it for a period of time τ_j, termed *storage time*. For simplicity of presentation, in the following we assume $\tau_j = \tau$, $\forall j \in V$. During the storage period, j serves the content to nodes issuing requests for it and, possibly, downloads from the Internet server a fresh copy of the content if its validity time has expired. We assume that a node i, which at a given time t does not store any copy of the content and which will later be referred to as "content consumer", issues queries at a constant rate λ_i.

To achieve load balancing, at the end of the storage time j has to decide whether (1) to hand the content over to another node, (2) to drop the copy, or (3) to replicate the content and hand over both copies. We refer to the nodes hosting a content copy at a given time instant as *replica nodes*, and we denote their set by $\mathcal{C}$. Only replica nodes are responsible for updating the content and for injecting a new version in the wireless network.

Next, to highlight our contribution with respect to related work, we relate our study to the formulation of the replication and replica placement problems typically used in the literature. Let us fix the time instant and drop the time dependency for ease of notation. Then, let $G = (V, E)$ represent the network graph at the given time, defined by a node set V and an edge set E. Let $\mathcal{C}$ denote the set of facility nodes, i.e., nodes holding a content replica. The specification of the placement of a given number of replicas, k, amounts to solving the uncapacitated k-median problem, which is defined as follows. We consider both cases when we have single commodity (or content) and multi commodity. For readers' convenience, we recall the formulations for facility location variants already mentioned in Chapter 2 and develop them to adapt to the content replication problem.

5.1.1 Single commodity

Definition 1 *k-median. Given the node set V with pair-wise distance function d, $\forall j \in V$ select up to k nodes to act as facilities so as to minimize the joint cost $C(V, k)$:*

$$C(V, k) = \sum_{\forall i \in V} d(i, m(i)) \tag{5.1}$$

where $m(i) \in \mathcal{C}$ is the facility j closest to i.

The replica node set $\mathcal{C}$, instead, can be obtained by solving the following capacitated facility location problem at a given time instant.

Definition 2 *Capacitated facility location. Given the node set V with pair-wise distance function d and cost for opening a facility at j $f(j)$, $\forall i \in V$, select a set of nodes to act as*

facilities so as to minimize the joint cost $C(V, f)$ of acquiring the facilities and servicing the demand while ensure that each facility j can only serve at most u_j clients :

$$C(V, f) = \sum_{\forall j \in \mathcal{C}} f_j + \sum_{\forall i \in V} d(i, m(i)) \tag{5.2}$$

where $m(i) \in \mathcal{C}$ is the facility j closest to i and the number of clients i attached to facility j :

$$c_j \leq u_j$$

For general graphs, the above problems are NP-hard [65] and a variety of approximation algorithms have been developed, which however require global (or extended) knowledge of the network state [7]. However our problem is even more sophisticated : there is no more distinction of client set and facility set since any node can be chosen to replicate the content, i.e. to be a facility node. Furthermore, node mobility makes network topology change all the time. For these reasons we should define a new class of facility problem which is hard to prove that any approximation algorithm can converge in a polynomial time. We therefore use the simulation to validate our scheme and define a way to evaluate the convergence of our mechanism.

5.1.2 Multi commodity

We extend the baseline capacitated facility location problem definition in Def. 2 to address the case of multi commodity [1]. Let I denote the set of items I=$\{1, ..., M\}$. Each item h has a content popularity represented by a number of nodes that are interested in the content [2].

To extend the cost function to include multiple contents, we consider two possibilities as in [95] :

- We consider M separate optimizations for each content and combine the results as an approximation solution.
- We consider an optimization for all contents.

We assume the same opening cost f for every content h, the cost in Eq.5.2 can be rewritten as :

$$C(V, f) = \sum_{\forall j \in \mathcal{C}} \sum_{\forall h \in I} f_j(h) + \sum_{\forall i \in V} \sum_{\forall h \in I} d(i, m(i, h)) \tag{5.3}$$

where $m(i, h) \in \mathcal{C}$ is the facility j holding h *closest* to i and the number of clients i demanding any content h attached to facility j is limited by :

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

[1] In the following we use interchangeably *content*, *commodity*, *information item* or simply *item as* synonymes.

[2] We use i, j, h to indicates consumer node, facility node and information item respectively.

To approximate the multi-commodity facility location problem is not trivial. Therefore we transform the problem from multi-commodity to single-commodity by doing the following : from the graph $G = (V, E)$ with N nodes and each node i is denoted as (x, y), suppose that we have M contents, we transform the graph into $G'(V', E')$ with $M \times N$ nodes, each node i in G now has M 'virtual instances' in G', denoted as $i(h) = (x, y, h)$, $h = 1..M$.

Eq.5.2 now is :

$$C(V', f) = \sum_{\forall j(h) \in \mathcal{C}} f_j(h) + \sum_{\forall i(h) \in V'} d(i(h), m(i, h)) \tag{5.4}$$

where $m(i, h) \in \mathcal{C}$ is the facility $j(h)$ *closest* to $i(h)$ and the number of clients $i(h)$ attached to facility $j \in V$ is

$$\sum_{\forall h \in I} c_j(h) \leq u_j$$

.

5.1.3 Content popularity

In our study we assume that nodes are not interested in all contents that exist in the system. The set of clients for a content constitutes its content popularity defined as following : We have N nodes and M contents. Each content h has a percentage of nodes that are interested in that content as $p(h)$. $p(h)$ varies from 0 to 1 and can follow a predefined distribution. In brief content h can have $p(h)N$ nodes that are interested in content and the maximum popularity is N in case $p(h)$=1.

5.1.4 Discussion

Our main contribution is to design a mechanism for content placement and replication that achieves load balancing and adapt to the network topology and demand change, while taking into account the implications of query propagation towards replica nodes. Several new problems are introduced in the context of our work :

- Node mobility introduces the problem of a dynamic graph G, requiring that the facility location problem be solved upon every network topology or demand rate change.
- Even under static topology and constant demand, solving the facility location problem does not yield load balancing among nodes.
- The input to the facility location problem is the content demand workload generated by users : both replicas location and the number of replicas to deploy in a network depend on content consumption patterns. While the approach traditionally adopted is to assume content demand to be directed to the closest facility, as stated in Defs. 2, the wireless nature of our system allows content requests to propagate in the network, potentially reaching multiple facilities (replica nodes).

In the literature, facility location problem are solved by approximation algorithms like lagrangian relaxation, primal dual or local search technique. We focus our interest on local search algorithm which is more applicable for a distributed system where a global view can not be assumed. However as we mentioned before, the mobility and the overlap of facility set and client set made our problem more difficult to prove that it has a converged solution. We design a distributed mechanism inspired of local search algorithm for our system. Furthermore, we pointed out that the infinite topology change of mobile network would trigger operations for the optimization all the time, and thus the algorithm would never end. We accept the fact that the optimization procedure runs in a infinite loop, but we hence introduce a new concept of convergence in which we set a tolerance factor to evaluate the convergence of optimization result. Our simulation shows that the number of replicas approximates within a small tolerance factor the converged result computed by a centralized local search algorithms.

Note that several caching policies have been proposed mainly in the context of mobile ad-hoc networks [126, 109, 41], but they focused more on cache replacement. Simple, widely used techniques for replication are gossiping and epidemic dissemination [48, 47], where the information is forwarded to a randomly selected subset of neighbors. Although our RWD scheme may resemble this approach in that a replica node hands over the content to a randomly chosen neighbor, the mechanism we propose and the goals it achieves (i.e., approximation of optimal number of replicas) are significantly different.

Another viable approach to replication is represented by quorum-based [79] and cluster-based protocols [129]. Both methods, although different, are based on the maintenance of quorum systems or clusters, which in mobile network are likely to cause an exceedingly high overhead. Node grouping is also exploited in [45, 46], where groups of nodes with stable links are used to cooperatively store contents and share information. The schemes in [45, 46], however, require an a-priori knowledge of the query rate, which is assumed to be constant in time. Note that, on the contrary, our lightweight solution can cope with a dynamic demand, whose estimate by the replica nodes is used to trigger replication. We point out that achieving content diversity is the goal of [125] too, where, however, cooperation is exploited among one-hop neighboring nodes only.

Threshold-based mechanisms for content replication are proposed in [114, 103]. In particular, in [114] it is the original server that decides whether to replicate content or not, and where. In [103], nodes have limited storage capabilities : if a node does not have enough free memory, it will replace a previously received content with a new one, only if it is going to access that piece of information more frequently than its neighbors up to H-hops. Our scheme significantly differs from these works, since it is a totally distributed and extremely lightweight mechanism, which accounts for the content demand by other nodes and ensures a replica density that autonomously adapts to the changes in the query rate over time and space.

Relevant to our study are the numerous schemes proposed for handling query/reply messages ; examples are [29], which resembles the perfect-discovery mechanism, and [20, 112] where queries are propagated along trajectories so as to meet the requested information. Also, we point out that the RWD scheme was proposed in Chapter 4, which showed preliminary results indicating that a uniformly distributed replica placement can be well approximated using distributed store-and-forward mechanisms, in which nodes store content

only temporarily. The results shown in Chapter 4, however, besides being a preliminary study, focused on mechanisms for content handover only : no replication or content access were addressed.

5.2 Cost definition

In FL problems, the costs can be arbitrarily defined in order to adapt to the system's objectives In this section we study several ways to define the costs and select the definitions that match our replication goals. We aim to design a system cost that considers resource constraints and the latency to access contents for mobile nodes.

5.2.1 Opening cost

We call $f_j(h)$ as the cost to replicate content h at j. We have identified many options for this opening cost :

- **Constant cost** : We consider in this cost a constant value Ω to install a replica of h at every j.
$$f_j(h) = \Omega \tag{5.5}$$
Usually in FL problem, we consider a fixed cost to open every facility. However it is not the case in our replication scheme in mobile networks, where every node is supposed to have energy constraint, and the consumption of energy for replica role depends heavily on the number of connected clients.
- **Node degree-based cost** : In [73] authors make an observation that since replicating at a node with higher degree will have high probability to serve more clients hence it costs more to replicate at a high degree node for the reason of congestion and power consumption. Therefore we can take into account the node degree in the cost to open the facility :
$$f_j(h) = deg(j) \tag{5.6}$$
However, we still find out a problem in applying this cost to our replication schemes : because this cost assumes that every node is interested in the content, it considers only the number of edges connected to a node (i.e physical topology) but not the content popularity, hence we may count a high cost at a high node degree replica while there may not be any client that is interested by that replica content.
- **Client size-based cost** : We define the cost to install a replica that considers the client set $u_j(h)$ which counts every node i interested in the content and the distance to replica j is minimum.
$$f_j(h) = u_j(h) \tag{5.7}$$
For this cost, it is intuitive that the cost decreases when the client set size is smaller. Hence whatever the topology is, the solution is to fully replicate the information to minimize the client set. Since the objective of replication is to find a reasonable number of replicas in the system in order to reduce workload caused by the concurrent content downloads from cellular networks, we come to the idea that replica node should define an expected number of clients it is willing to serve that meets its own

energy constraint and capacity. Given the that expected number, the installment cost in this case is defined as following :

$$f_j(h) = |u_j(h) - u_j| \tag{5.8}$$

With this cost definition, the cost increases when replica nodes serve more than its expected number (which causes an excess in energy constraint) or less than its expected number (which increases the number of nodes need to access cellular network to download the content, hence may increase the congestion at providers gateway)

– **Workload-based cost** : In reality, we can not consider only the number of client because contents can have different size. We should consider also the workload in term of data size that a replica node transfers to its clients hence it is more practical that each node defines a reference volume of data it is willing to serve its neighborhood $v_j(h)$:

$$f_j(h) = |F(h)u_j(h) - v_j(h)| \tag{5.9}$$

where $F(h)$ is the size of content h and $F(h)u_j(h)$ is the workload served by replica node j during its storage time.
We consider also the case if a replica node is holding more than one content and define a common reference volume of data (or budget) for all the contents it holds as v_j. The opening cost is this case is :

$$f_j = |\sum_{\forall h \in I} F(h)u_j(h) - v_j| \tag{5.10}$$

In our replication mechanism, we use the cost definitions in Eq. 5.9 and Eq. 5.10 due to the fact that they consider the problem of resource constraints. Another important reason is that all elements in these equations can be measured locally at nodes hence it is feasible to design a distributed mechanism based on these costs.

5.2.2 Service cost

In a FL problem, service cost concerns the distance to reach the closest facility from all clients. If the FL problem is metric, these distances should conform to the triangular rule. In our replication context, the distance can be latency, hop count or euclidean distance. The latency however can only be derived by simulating the real traffic. If we use the hop count, we need to compute this service cost while consider all possible paths between every two nodes. To simplify, we choose euclidean distance for this reason : we assume in our application context a high density network, thus there should be alway a shortest path in terms of hop count connecting two nodes with a distance that is very close to the shortest euclidean distance. This assumption allows us to avoid considering every detail of network topology when compute the distance cost.

5.3 Distributed mechanism for replication and placement problems

We now outline our content distribution and replication procedures. Firstly, several techniques for query distribution and content access are detailed; next, we examine the challenging problem of replica placement, i.e., of which nodes are to be selected as carriers of content replicas to achieve load balancing; finally, we discuss the behavior of replica nodes as a function of the system workload, in search of a cooperative, distributed content replication strategy in presence of changing demand.

5.3.1 Replica placement

Next, we overview the distributed lightweight algorithm that we use to solve the replica placement problem. Recall that any mobile device can be selected to host a content replica for a limited amount of time, that we term *storage time*, τ.

Also, as the first step to our study, we focus on the case where every node is interested in all contents.

As discussed in Sec. 5.1, at a fixed time instant, replica placement can be cast as the k-median problem. Given a set of potential locations to place a replica, the problem is to position an a-priori known number k of replicas according to Def. 1, i.e., so as to minimize the distance between replica node and requesting node. For a generic distribution of nodes over the network area, the solution of the k-median problem for different instances of the network graph yields replica placements that are instances of a random variable uniformly distributed over the graph. This is quite an intuitive result, confirmation of which we found by applying the approximation algorithm in [7] to the solution of the k-median problem in presence of various network deployments.

We evaluate our lightweight distributed mechanism whether it well approximates the target distribution of the replicas over the network nodes, which is obtained by the solution in [7].

According to our mechanism, named *Random-Walk Diffusion (RWD)* [1], at the end of its storage time, a replica node selects with equal probability one of its neighbors to store the content for the following storage period. Thus, content replicas roam the network by moving from one node to another, randomly, at each time step τ.

To understand the extent to which replica placement achieved by our simple technique resembles the target distributions, in Sec. 5.5 we employ the well-known χ^2 goodness-of-fit test on the inter-distance between content replicas. Whenever the computation complexity allows us, we compare the temporal evolution of the inter-distance distribution of replicas obtained by our scheme against the optimal replica placement computed by solving the k-median problem. Otherwise, we consider as term of comparison the empirical distribution of the distance between two nodes measured in simulation. Note that using inter-distances instead of actual coordinates allows us to handle a much larger number of samples (e.g.,

[1] For a detailed description of RWD, please refer to Chapter 4

$|V| \cdot (|V| - 1)$ instead of just $|V|$ samples) thus making the computation of the χ^2 index more accurate.

It is clear that the quality of approximation of the target replica distributions achieved by our store-and-forward mechanism depends on the node density : the higher the density, the better our approximation.

5.3.2 Content replication

We now focus on the more general problem of the capacitated facility location, defined in Sec. 5.1, where the optimal number of replicas (facilities) to be placed in the network is to be determined along with their location. In particular, we want to answer the following questions.

1. Given a set of demand points that exhibit a homogeneous querying rate, what is the optimal number of content replicas that should be deployed in the network to achieve load balancing ?
2. Is it possible to design a lightweight distributed algorithm that approximates this optimal number of replicas in presence of a dynamic demand and time-varying topology ?

We address these questions by suggesting simple modifications to the RWD mechanism described in Sec. 5.3.1.

Again, we fix the time instant and, for simplicity, we drop the time dependency from our notation. Let the network be described by the graph $G = (V, E)$, with $|V| = N$ nodes deployed on an area $\mathcal{A}$. Also, recall that $\mathcal{C}(h)$ and $V \setminus \mathcal{C}(h)$ represent the sets of content replicas and of nodes issuing requests for h, respectively.

Given G and the number of items M, the capacitated facility location problem amounts to the joint optimization of the number of replicas and their locations in the network. The RWD mechanism achieves a good approximation of the optimal placement in mobile networks, but ignores the cost to deploy a content replica. Now, with reference to Def. 2, we define the cost function to deploy content replicas in the network f_j, $\forall j \in \mathcal{C}$ as in Eq. 5.10. Given the load balance we should achieve at each facility node and node capacity constraint, the total number of demanding clients for all contents $\sum_{\forall h \in I} F(h) u_j(h)$ should not exceed or fall behind the facility capacity υ_j.

In practice, replica nodes should estimate $u_j(h)$ by measuring the number of queries it served to its neighborhood $s_j(h)$ within its storage time. We assume both the cases where all replica nodes are willing to serve the same volume of data and whether υ_j is a stochastic variable which follows a predefined distribution.

Note that in case we do independant opimization for each content h, we set υ_j for each content as $\upsilon_j(h)$ and we take only $s_j(h)$ for a content instead of the sum of workload for all contents : $f_j(h) = |F(h)s_j(h) - \upsilon_j(h)|$. Eq. (5.10) indicates that the cost for replica node j grows with the gap between its workload and the reference volume of data υ_j. By using the cost function in (5.10) in the facility location problem in Def. 2, we can determine

the location and number of replicas so that load balancing is achieved under the idealistic assumption that each query reaches one replica only.

Our replication mechanism only involves replica nodes, which are responsible to decide whether to replicate, hand over or drop content based on local measurements of their workload. This procedure is inspired from the local search FL approximation algorithm from [7] which consists of 3 operations to select for each heuristic round toward the solution : **add**, **drop** or simply **swap** the content. During storage time τ, the generic replica node j *measures* the number of queries that it serves, i.e., $\hat{s}_j(h)$. When the storage time expires, the replica node compares $\sum_{\forall h \in I} F(h)\hat{s}_j(h)$ to υ_j. Decisions are taken as follows :

$$\text{if } \sum_{\forall h \in I} F(h)\hat{s}_j(h) - \upsilon_j \begin{cases} > \epsilon & \text{replicate} \\ < -\epsilon & \text{drop} \\ \text{else} & \text{hand over} \end{cases}$$

where ϵ is a tolerance value to avoid replication/drop decisions in case of small changes in the node workload and m id the number of items that node j is currently holding. We show the algorithm executed at replica node in Alg. 5.1.

The rationale of our mechanism is the following. If $\sum_{\forall h \in I} F(h)\hat{s}_j(h) > \upsilon_j$, replica node j presumes the current number of content replicas in the area to be insufficient to guarantee the expected volume of data υ_j, hence the node replicates the content and hands the copies over two of its neighbors (one each), following the RWD placement mechanism (Sec. 5.3.1). The two selected neighbors will act as replica nodes for the subsequent storage time. Instead, if $\sum_{\forall h \in I} F(h)\hat{s}_j(h) < \upsilon_j$, replica node j thinks that the current number of replicas in the area is exceeding the total demand, and just drops the content copy. Finally, if the experienced workload is (about) the same as the reference value, j selects one of its neighbors to hand over the current copy.

Algorithm 5.1 replicate (j, h)

```
w ← 0
if jointOpt then
  for i = 1 to M do
    w ← w + F(i)ŝ_j(i)
  end for
else
  w ← F(h)ŝ_j(h)
end if
if w ≥ υ_j + ε then
  add(h)
  handOver(h)
else if w ≤ υ_j + ε then
  drop(h)
else
  handOver(h)
end if
```

We stress that replication and placement are tightly related. For example, if content demand varies in time or in space (e.g., only a fraction of all nodes located in a sub-zone of the network area issue queries), both the number of replicas and their location

must change. Thanks to the fact that replica nodes take decisions based on the measured workload, our solution can dynamically adapt to a time- or space-varying query rate, as will be shown by our simulation results. On the contrary, when the content demand is constant and homogeneous, our handover mechanism ensures load balancing among the network nodes.

In the following, we set up a simulation environment to evaluate the behavior of our mechanism when the wireless network is both static and dynamic. We also characterize the time the system takes to reach an optimal number of content replicas and we investigate the impact of the content access scheme on the performance of our solution.

5.4 Simulation set-up

We implemented our replica placement and content replication mechanism in the *ns*-2 simulator. For each experiment described in the following, we execute 10 simulation runs and report averaged results. Our statistics are collected after an initial warm-up period of 500 s.

In our simulations, which lasted for almost 3 hours of simulated time (10000 s), we assume nodes to be equipped with a standard 802.11 interface, with an 54 Mbps fixed data transmission rate and a radio transmission range of 100 m. We consider a single content, whose size is of the order of 1 MB. In our evaluation we do not simulate cellular access. We point out that all standard MAC-layer operations are simulated, which implies that both queries and replies may be lost due to typical problems encountered in 802.11-based ad hoc networks (e.g., collisions or hidden terminals). This explains why, in the following, even nominally "ideal" access techniques may not yield the expected good performance.

Notation	Description	Default
N	Number of nodes	320
$\mathcal{C}$	Replica nodes	
$\mathcal{A}$	Simulation area	1km^2
τ	Storage time	100s
v_j	Replica budget (volume of data)	15MB
$\hat{s}_j$	Workload measured by replica node	
F	File size	1MB
ϵ	Workload tolerance	2MB
m	Number of items currently hold by a node	
M	Number of contents	4
H	Hop limit for query to travel	5

TAB. 5.1 – Notations used in simulation and default values.

We focus our attention on wireless networks with high node density : we place $N = 320$

nodes uniformly at random on a square area $\mathcal{A}$ of 1000×1000 m^2, with a resulting average node degree of 9–10 neighbors. We simulate node mobility using the *stationary* random waypoint model [16] where the average node speed is set to 1 m/s and the pause time is set to 100 s. These settings are representative, for example, of people using their mobile devices as they walk.

Unless otherwise stated, the default values of our simlation are presented in Tab. 5.1. For the content access mechanisms, we set the scope of flooding and scanning to $H = 5$ hops : e.g., a node can cover half of the network diameter with scoped-flooding. In the case of scoped-flooding or perfect-discovery, if a query fails (i.e., no answer is received after 2 s), a new request is issued, up to a total of 5 times. If the scanning mechanism is used, a complete scan of 2π is divided into $S = 5$ angular sectors, each of which is visited for a maximum of 0.5 s, at most 5 times[1].

Finally, the tolerance value ϵ used in the replication/drop algorithm is equal to 2, unless otherwise stated ; for all nodes, the storage time τ is set to 100 s and the reference workload for a replica node is equal to $\upsilon_j = 15$MB.

We present the main results of our work organized in a series of questions. We focus on the mobile scenario, but we have also results for a static network which were presented in our previous work [25].

5.5 Single content

We present the main results of our work organized in a series of questions. We focus on the mobile scenario, but present results for a static network when the comparison is relevant.

5.5.1 Replication with single content

How well does our replica placement approximate the optimal distribution ?

Here we assume a known number of content replicas to be deployed ($|\mathcal{C}|$=30), i.e., we consider the k-median problem discussed in Sec. 5.1. We measure the accuracy of our distributed replica placement mechanism using the χ^2 goodness-of-fit test on the inter-distance between replicas, as explained in Sec. 5.3.1. Considering a mobile network, we compute the distribution of replica nodes as follows : every τ seconds we take a snapshot of the network in its current state, we compute the optimal replica placement, by solving the k-median problem through the centralized local-search algorithm in [7], and we use the χ^2 test against the distribution achieved by our mechanism.

Fig. 5.1 shows that our scheme does an excellent good job of approximating the optimal replica placement. In particular, the temporal evolution of the χ^2 index suggests that our replica placement mechanism is able to approximate very well the optimal solution[2], despite

[1] We use the parameters that give the best results in terms of content access performance.
[2] A $\chi^2 \approxeq 3$ is assumed to indicate a good match between two distributions [6].

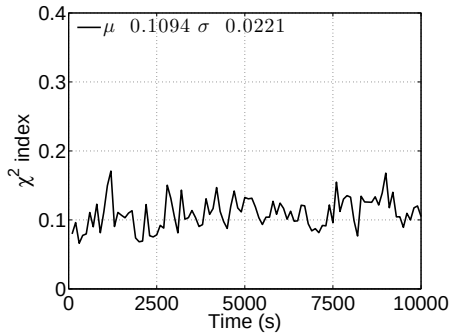


FIG. 5.1 – Temporal evolution of the χ^2 index in a mobile scenario ($|\mathcal{C}|$=30 and τ=100 s).

network dynamics.

Is the replication mechanism effective in reaching a target number of replicas ?

We now turn our attention to the *capacitated facility location* described in Sec. 5.1 and study how well the replication mechanism defined in Sec. 5.3.2 approximates the joint problem of replication and placement.

Here we consider a scenario in which only one copy of the content is initially present in the network and we focus on the evolution in time of the number of replicas in the system. We omit the temporal evolution of the χ^2 index, since our results are consistent with what we have observed for the placement scheme without replication.

Fig. 5.2 shows the temporal evolution of the total number of replicas $|\mathcal{C}|$ for the mobile scenario, against a reference line representing the optimal number of content replicas. Finding the optimal number of content replicas amounts to solving the uncapacitated facility location problem for a given network graph. To this end, we have implemented the *centralized* algorithm in [7] and computed an approximation to the optimal solution over several snapshots of the network graph. With reference to Def. 2, we set a non-uniform cost to open a facility as defined in Eq. 5.10. Intuitively, the cost to select a node to hold a content replica is proportional to its degree : a highly connected node will most likely attract more demand from content consumers.

For the parameters used in our simulations, the solution of the centralized algorithm indicates that the target number of replicas the system should reach is $|\mathcal{C}^*| = 30$.

Fig. 5.2 indicates that the number of content replicas we achieve with our scheme strikingly matches the target value : in steady state, the average relative error is less than 2%.

5.5.2 Load balancing

How is the total workload shared among replica nodes ?

As before, we study the joint placement and replication problem and we use the extreme

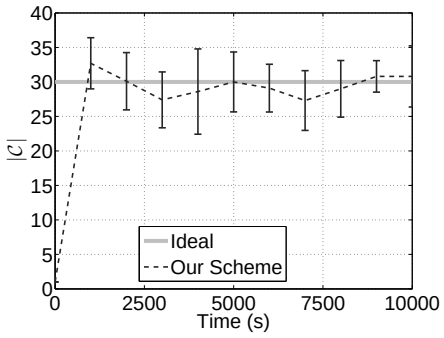


Fig. 5.2 – Temporal evolution of the number of replicas, for a network bootstrapping with $|\mathcal{C}| = 1$ in a mobile scenario ($\lambda = 0.01$, $\upsilon_j = 10MB$, $\tau = 100$ s, $|\mathcal{C}^*| = 30$).

scenario in which the network is initialized with only one content replica. Tab. 5.2 shows the 25%, 50% and 75% quantiles of the workload for each replica node, aggregated over the simulation time. As expected, the average workload roughly matches the budget $\upsilon_j = 10MB$, both in the static and mobile scenario.

Percentile	25th	50th	75th	Mean
Static	4	8	14	9.73
Mobile	5	8	13	9.77

Tab. 5.2 – Aggregate workload distribution for replicas for a network bootstrapping with $|\mathcal{C}| = 1$ ($\lambda = 0.01$, $\upsilon_j = 10MB$, $\tau = 100$ s).

5.5.3 Convergence time

What is the convergence time of the replication mechanism ?

Convergence time should be carefully defined in our context : clearly, our mechanism cannot settle to a static, unique content replica placement, nor can it stabilize on a unique number thereof. For placement, it is not our intent to statically assign the role of content replica to a node and deplete nodal resources : we seek to balance the workload across all network nodes. We assume the network to have converged to a steady state when the difference between the reference value computed using the centralized local search algorithm and the experimental number of replicas is within 2%.

Again, we consider a scenario in which only one copy of the content is initially present in the network. Tab. 5.3 illustrates how convergence time (labelled t_s) varies with the storage time τ and the tolerance value ϵ. We also performed experiments to study the impact of the network size : we have observed a linear growth of the convergence time with N. Since the storage time τ is used to trigger replication/drop decisions, we expect to see a positive correlation between τ and convergence time : Tab. 5.3 confirms this intuition. We note that there is a trade-off between the convergence time and the message overhead : a small

storage time shortens the convergence time at the cost of an increased number of content movements from a node to another. As for the impact of the tolerance parameter ϵ, our experiments indicate that a very reactive scheme would yield smaller convergence times, at the risk of causing frequent oscillations around a target value.

τ (s)	t_S (s)
20	800
100	1700
200	2300

ϵ	t_S (s)
0	700
2	1700
5	1900

TAB. 5.3 – Average convergence time t_S as a function of the storage time τ ($\epsilon = 2$) and the tolerance factor ϵ ($\tau = 100$ s).

5.5.4 Adaptation to demand change

What is the impact of variations in time and in space of the content demand ?

We now focus on the behavior of content replication in presence of a dynamic workload. We first examine workload variations in time. In a first phase, from time 0 to time 5000 s, we set the content popularity at 100% . In a second phase, from 5000 s to the end of the simulation, the popularity is 50%, i.e. we randomly select 50% of nodes to continue querying and the remainder stop querying. Thus the demand reduces to a half.

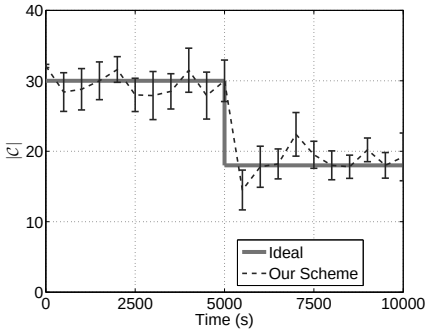


FIG. 5.3 – Temporal evolution of the number of replica nodes in case of variations in time of the content demand, for a mobile network. $|\mathcal{C}^*|$ is equal to 30 and 18 in the first and second phase, respectively.

Fig. 5.3 shows the temporal evolution of the number of replicas in a mobile network. The figure is enriched with two reference values : in the first phase $|\mathcal{C}^*| = 30$, in the second phase $|\mathcal{C}^*| = 18$. Our mechanism achieves a very good approximation of the target number of replicas : despite node mobility, not only is our scheme able to correctly determine the number of replicas but also their target location. As a consequence, the load distribution is minimally affected by a variation in time of content demand. This result is reported in Tab. 5.4, where we indicate the 25%, 50% and 75% quantiles of the workload, and the average load per replica node.

Percentile	25th	50th	75th	Mean
1st Phase	4	8	13	9.98
2nd Phase	3	7	13	9.91

Tab. 5.4 – Workload distribution of replica nodes for variations in time of the content demand, in a mobile network.

We now turn our attention to variations in space of content demand : we describe the behavior of the content replication mechanism with the following example. For the initial 5000 s of the simulation time, content queries are issued by all nodes deployed on the network area $\mathcal{A}$ of size 1 km^2. Subsequently, we select a smaller square area α of size 500 m^2 in the bottom left corner of $\mathcal{A}$ and instruct only nodes within that zone to issue content queries, while all other nodes exhibit a lack of interest.

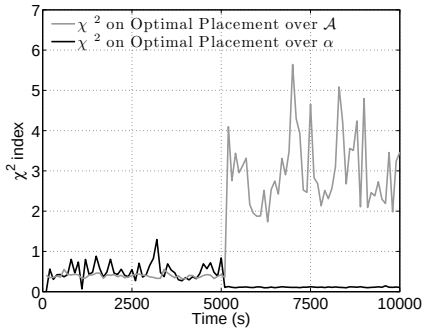


Fig. 5.4 – Temporal evolution of the χ^2 index for variation in space of the content demand, in a mobile network.

Fig. 5.4 compares the empirical and the approximate optimal distributions using the temporal evolution of the χ^2 index. We observe a very good match (i.e., low values χ^2) over the network area $\mathcal{A}$ and on the sub-area α when content demand comes, respectively, from $\mathcal{A}$ for $t \leq 5000$ s and α for $t > 5000$ s. This suggests that when content demand varies in space, our scheme allows content replicas to migrate to the location where the demand is higher and meet a variation in the workload.

5.6 Multiple contents

In this section we try to obtain the results to show whether our replication scheme works with the multiple contents under the difference in content size and content popularity. We also aim to study the scalability of replication system under several factors : network size, network density and human mobility.

5.6.1 Replication with multiple contents

We explore the replication scheme for multiple contents. In particular, we try to reply the question if we can use independent optimization for each content instead of a joint optimization for all contents. We show also the results with multiple contents with the difference in popularity and content size to see how well our scheme copes with the content dynamic.

How different is the replication scheme based on joint optimization and independent optimizations ?

In order to understand how joint optimization differs from independent optimization, we apply CFL centralized algorithms for different υ_j on a snapshot of a mobile network topology. We present the numerical result obtained from the CFL centralized algorithm with 4 items of the same content size 1MB. Fig. 5.5 shows the number of replicas we obtain with υ_j in the range of 10–40MB. For the independant optimization this mean that each content is assigned with a budget $\upsilon_j(h)$=$\upsilon_j/4$. The result shows a higher number of replicas when we apply independant optimization for each item.

From these reference values we validate the above numerical results by running the simulations with 4 items of 1 MB using υ_j=40MB for the joint optimization and $\upsilon_j(h)$=10MB (i.e. υ_j=40MB in total) for the independant optimizations. Note that for the sake of clarity we plot only the results for one item among the 4 items since the results for the remaining items are very similar. Fig. 5.6 shows us the difference in simulation results for the two algorithms. Fig. 5.6(a) shows that $|\mathcal{C}|$ obtained are similar for both algorithms even $\upsilon(j)$ for independent optimization is triple than the value used in joint optimization : in average we have 17 replicas for joint and 43 for independent optimization. Fig. 5.6(b) shows that replica placement approximates well the optimal placement χ^2 error of 0.16-0.18. To have a per-node detail on how well nodes share the replica role, we compute the time that a replica node holds 1, 2, 3 or 4 items. To do this we take the snapshot of the network every 10s and count the number of items at every node. In Fig. 5.6(c), the distribution of stored items per node are similar for both cases and are good in terms of fairness : the replicas are well spread among nodes as only a few of times nodes are holding all replicas and in 80–90% of cases, nodes hold only 1 replica. Finally Fig. 5.6(d) gives us an image on how fair the workload is shared among nodes : 50% of nodes have the workload up to 0.3% the total workload for joint optimization which is around $1/N$. We make an observation that the independent optimization requires a budget υ_j of 40MB but in average the measured workload can not reach this value. In other words, independent optimization does not work well in the multi item case : nodes tends to replicate too many while the total budget is not reached yet. However as we can see in Fig. 5.7, the query delay for joint optimization is higher. This is a tradeoff of distance cost (in terms of delay) and opening cost (in terms of number of replicas).

Does our replication scheme work with various content popularity ?

We study the scenario when not all nodes are interested in a content. In this case we assume that a node will take the role of replica k uniquely when it is interested in content k. We vary the percentage of interested node $p(h)$ from 100% to as low as 25% (Table. 5.5) by setting the percentage of nodes that are interested in the content and υ_j is set to 15MB

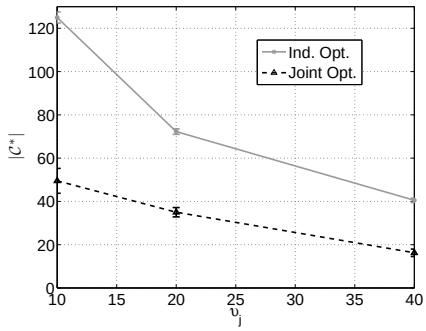
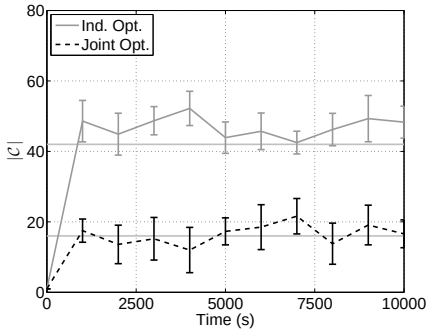
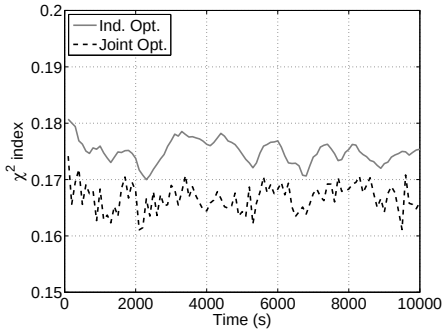


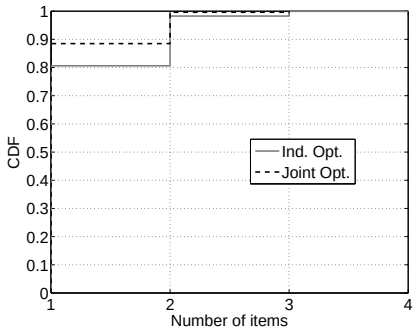
Fig. 5.5 – $\mathcal{C}$ computed by joint optimization and independent optimizations with the centralized CFL algorithm.



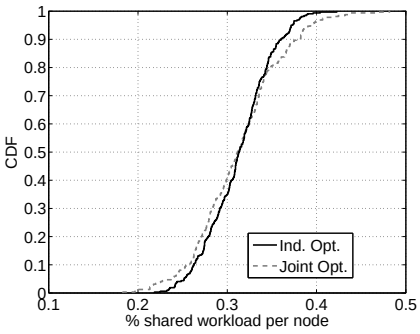
(a) Avg. number of replicas per content



(b) χ^2 index



(c) Number of stored items



(d) Workload per storage time

Fig. 5.6 – Comparison of the replication scheme between joint optimization and independent optimization.

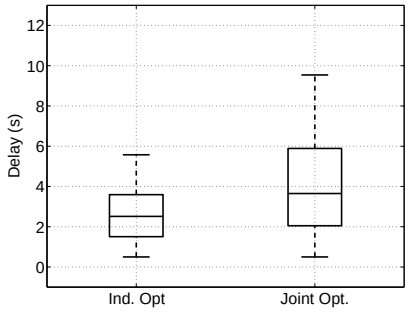


Fig. 5.7 – Query delay with joint opt. and ind. opt. υ_j=40MB.

for joint optimization and 40MB for independent optimization. We use these setting to have the same number of replicas for both joint and independent optimization in order to compare their performance.

Item h	Percentage	Ind. Opt. v_j=40MB	Joint Opt. v_j=15MB
1	100%	39	42
2	75%	30	29
3	50%	19	18
4	25%	14	15

TAB. 5.5 – $|\mathcal{C}^*|$ computed by the centralized CFL algorithm while varying the percentage of interested nodes.

Fig. 5.8(a) shows that $\mathcal{C}$ is oscillating around the target value even with a low content popularity as 25%. The load is also well shared among replica nodes. Fig. 5.8(c) shows that a node at mimimum served 0.21% the total workload and 50% of nodes served up to 0.3% (This value is exactly the expected mean workload since we have N=320 nodes in the network, every node should share in average $\frac{1}{N}$ the total workload which is roughly 0.3%). In the worst case node served at maximum 0.41%. Fig. 5.8(e) shows the stored items per node which shows that very rarely a node (only 0.5% of the cases) is selected to hold all items and 10% of the cases a node is holding more than 1 item. This means that our scheme achieves a good result in spreading the role of replica among nodes. The fact that replicas for each content are spreading to all nodes instead of grouping at some good candidates can be explained for these reasons :

- Our distributed scheme keeps swapping the replica role from node to node to maintain the load balance.
- We do not want a node to keep content replicas for a long time to address the dynamic of mobile network.

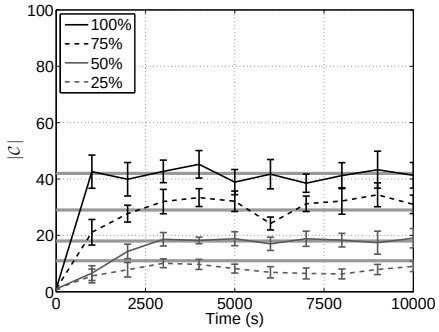
Fig. 5.8(b) shows $\mathcal{C}$ in the case we separate the budget for each content. Results show that the number of replicas at steady state oscillates with less variation. This result can be attributed to the strict budget for each content instead of a flexible budget for all content used in joint optimization. Fig. 5.8(f) shows there are 18% of the cases a node is holding more than 1 item which is slightly higher than joint optimization. In terms of load balancing, there are less nodes that experience high workload as shown in Fig. 5.8(d).

How does our replication scheme behave when the content size is different ?

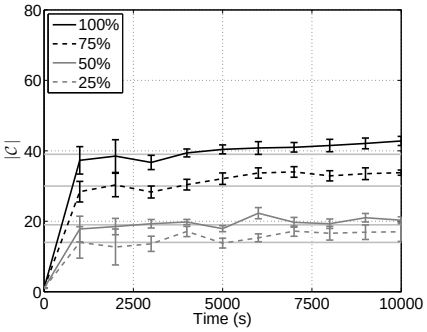
We study the scenario where the 4 items have different sizes. We set up 4 content sizes as in Table 5.6. The target number of replica $|\mathcal{C}^*|$ computed by centralized CFL algorithm while assuming an equal demand among nodes for item h.

Item h	F(h)	Ind. Opt. v_j=40MB	Joint Opt. v_j=15MB
1	1MB	39	42
2	2MB	62	67
3	3MB	87	91
4	4MB	115	117

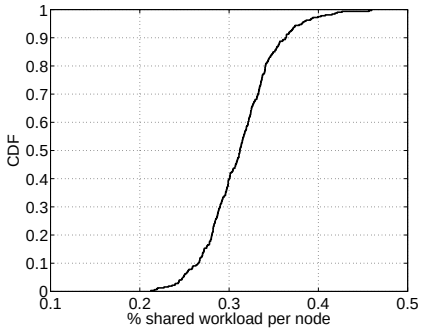
TAB. 5.6 – $|\mathcal{C}^*|$ computed by the centralized CFL algorithm with different content sizes.



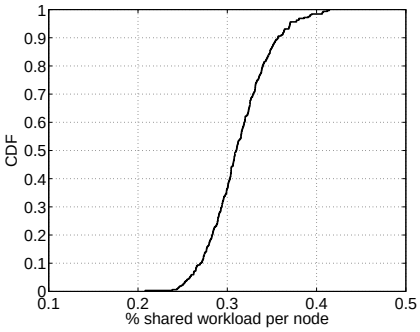
(a) Number of replicas (Joint Opt.)



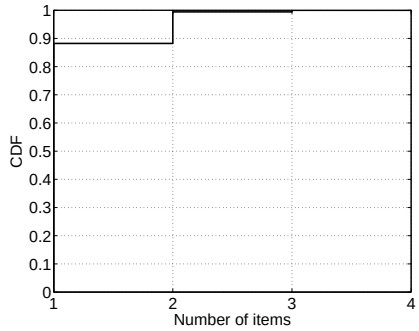
(b) Number of replicas (Ind. Opt.)



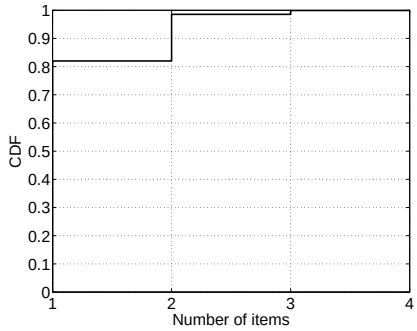
(c) Total workload (Joint Opt.)



(d) Total workload (Ind. Opt.)



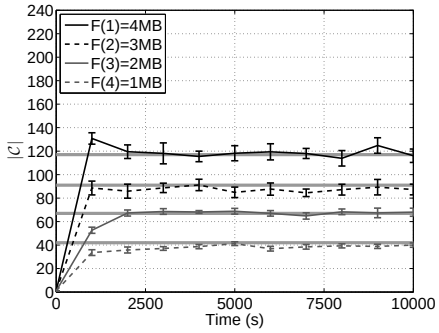
(e) Number of stored items (Joint Opt.)



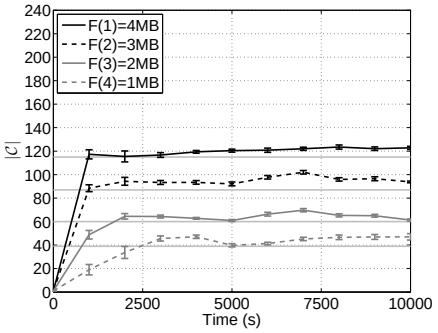
(f) Number of stored items (Ind. Opt.)

Fig. 5.8 – Replication based on joint optimization with different content popularity.

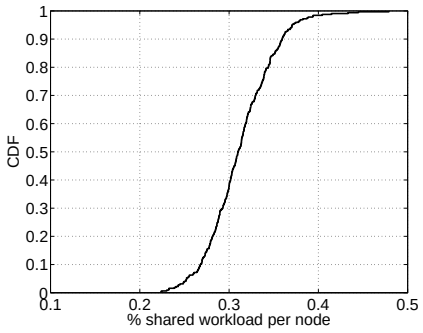
Fig. 5.9(a) shows that $\mathcal{C}$ is oscillating around the target value. However the bigger content size is, the more replicas are present in the network and the value $|\mathcal{C}|$ fluctuates more since the workload measurement error is scaled with the content size. Fig. 5.9(c) shows that a node at mimimum served 0.21% the total workload and 50% of nodes served up to 0.3%. In the worst case node served at maximum 0.41% : this number is much more than the case where we have multi-rate with the same size and can be explained also by the reason that big size items introduce more error for the local workload measurement. Fig. 5.9(e) shows the stored items per node which shows that very rarely a node (only 0.5% of the cases) is selected to hold all items and 15% of the cases a node takes up the replica role of more than 1 item. This means that our scheme achieves a good result in spreading the role of replica among nodes. Fig. 5.9(b) shows again when we separate the budget for each content, the obtained $\mathcal{C}$ oscillates with less variation. Fig. 5.9(f) shows there are more nodes (about 18%) holding more than 1 item. There are less nodes that experience high workload as shown in Fig. 5.9(d).



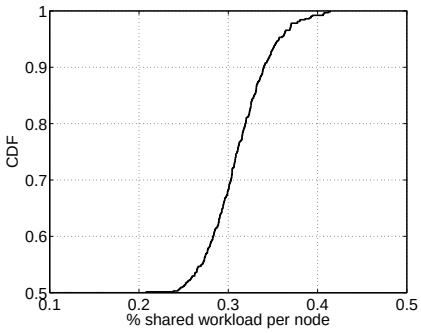
(a) Number of replicas (Joint Opt.)



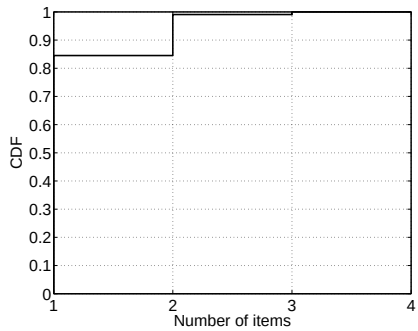
(b) Number of replicas (Ind. Opt.)



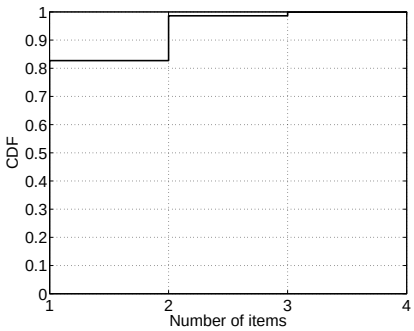
(c) Total workload (Joint Opt.)



(d) Total workload (Ind. Opt.)



(e) Number of stored items (Joint Opt.)



(f) Number of stored items (Ind. Opt.)

Fig. 5.9 – Replication with different content sizes.

5.6.2 Impact of mobility

How does our mechanisms work with realistic human mobility model ?

Recently the human mobility characteristics are widely studied by the reasearch and some mobility models that mimick the human pattern are introduced. We adopt the SLAW model in [74] and we aim to study whether our scheme works with realistic human mobility.

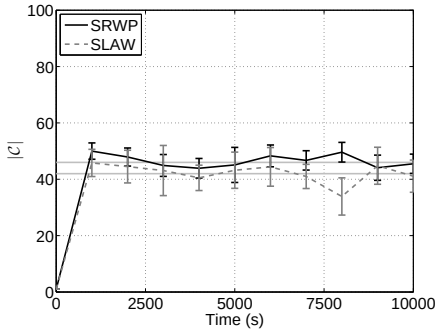
We generate a synthetic trace of 3-hours duration. The scenario consists of a 1 km^2 area, with 320 nodes and 600 waypoints that are Pareto-distributed with Hurst parameter equal to 0.75. Nodes move at a speed of 1 m/s ; their pause time obeys a Levy distribution with coefficient equal to 1 and has minimum and maximum values equal to, respectively, 100 s and 1000 s. The distance weight which determines how much node gives priority to nearby locations before go to farther locations is set to 3.

We run the simulation for 4 items and comapre the results with target number of replicas computed by the CFL algorithm over the network snapshots for every 100 s.

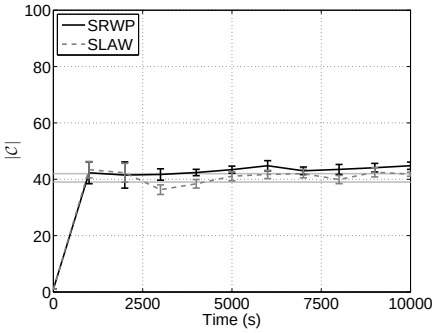
Fig. 5.10(a) shows us the replica evolution during simulation time for SLAW and stationary RWP. We observe that for SLAW although the number of replicas fluctuates a lot, this number matches approximately the target value. The error rate is higher than the result for stationary random mobility but this is reasonable since human mobility creates clusters in network topology and introduce more error in local workload measurement. Fig. 5.10(c) shows that a node at mimimum served 0.12% the total workload and 50% of nodes served up to 0.3% but in the worst case a node served at maximum 0.45% : the non-uniform network topology can be the factor that cause this imbalance in comparison with stationary RWP model. However we may not be too pessimistic because only 10% of replica nodes served more than 0.45% hence the result is still good for up to 90% of nodes. Fig. 5.10(e) shows the stored items per node which shows that very rarely a node (only 2% of the cases) is selected to hold all items and 20% of the cases nodes hold more than 1 items. This means that our scheme still achieves a good result in replica role sharing with realistic human mobility. Fig. 5.10(b) there is less deviation of $\mathcal{C}$ when using separate the budget for each content. Fig. 5.10(f) shows there are more nodes holding more than 1 item. Fig. 5.10(d) and Fig. 5.10(c) show the interesting impact of the SLAW mobility on the joint and independent optimization : we know that the joint optimization tends to put less replica items on nodes and in this case when thee network topology is highly clustered, replica nodes can not use efficiently the dedicated budget hence in Fig. 5.10(c) we see a high percentage of nodes having the workload less than the average. Contrarily as independent optimization tends to put more replica items on a node, Fig. 5.10(d) shows that there is more nodes experiencing high workload than the average.

How does our mechanisms work with different node velocity ?

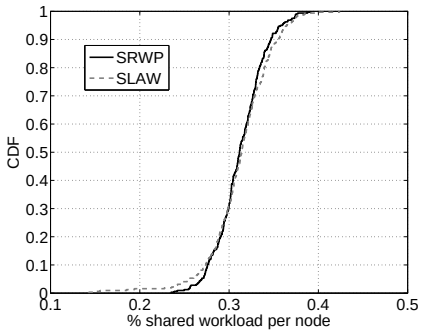
We also study our scheme with different node velocity. Fig. 5.11 shows an interesting result in which we found that the workload and delay just slightly increase when the mobility is high. In Fig.5.11(a) the number of replicas just increases from 43 to 48 when the speed is set from 1m/s to 10m/s. Fig.5.11(c) shows no difference in term of average workload at facility nodes. Fig.5.11(e) shows the distribution of delay for successful content demands in which we observe just a slight increasing delay when the speed is high.



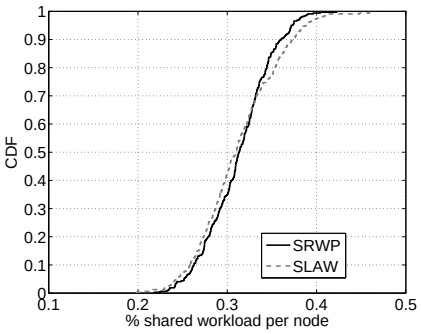
(a) Avg. number of replicas per content (Joint Opt.)



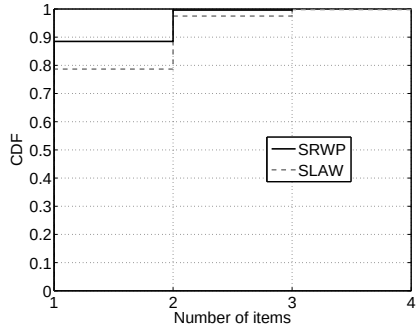
(b) Avg. number of replicas per content (Ind. Opt.)



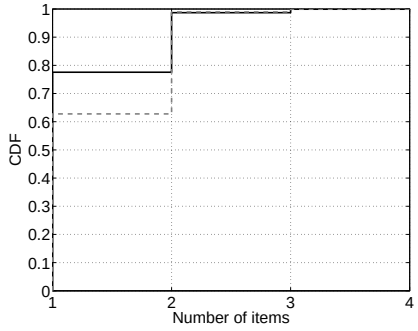
(c) Total workload (Joint Opt.)



(d) Total workload (Ind. Opt.)

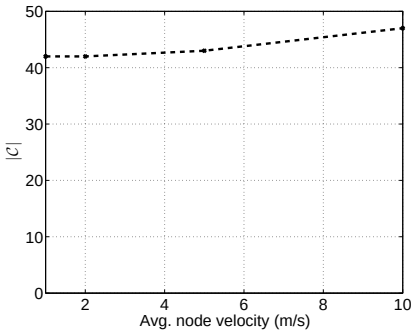


(e) Number of stored items (Joint Opt.)

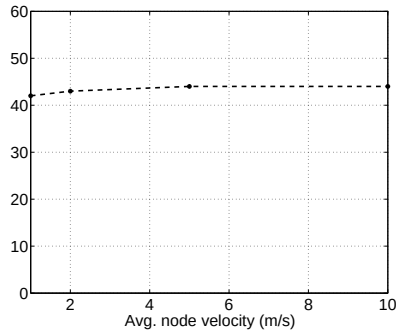


(f) Number of stored items (Ind. Opt.)

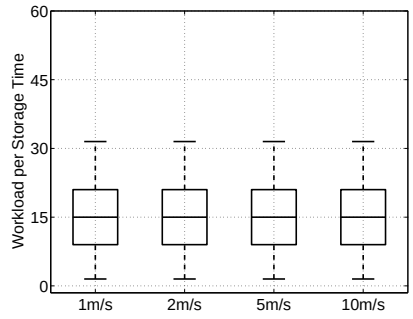
FIG. 5.10 – Replication with 2 mobility model stationary random waypoint and SLAW



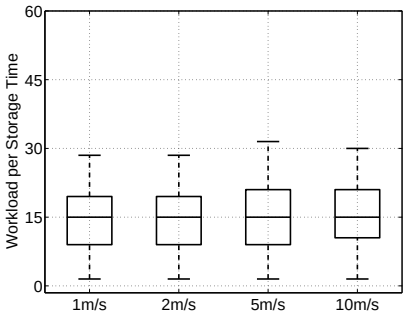
(a) Avg. number of replicas per content (Joint Opt.)



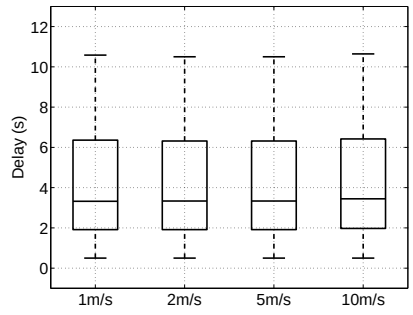
(b) Avg. number of replicas per content (Ind. Opt.)



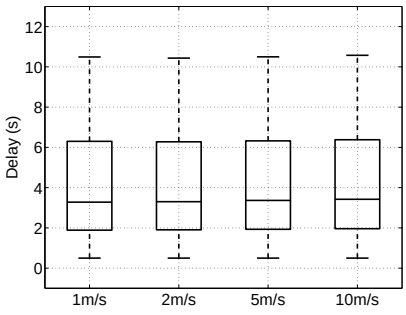
(c) Workload per storage time (Joint Opt.)



(d) Workload per storage time (Ind. Opt.)



(e) Delay (Joint Opt.)



(f) Delay (Ind. Opt.)

Fig. 5.11 – Delay and workload with different node velocity : 1, 2, 5, 10 m/s

5.6.3 Scalability

We study the scalability of our system by simulations with different node degree (i.e node density), number of contents and number of nodes (i.e. network size).

How does our mechanisms work with different node degree ?

Fig. 5.12 shows the simulation results when the average node degree is set to 5, 10 and 20 (i.e. we set 160, 320 and 640 nodes in the square area of 1km^2). In Fig.5.12(a) the number of replicas increases accordingly to the optimal number of facilities computed by CFL local search algorithm. Fig.5.12(c) shows again no difference in term of average workload at facility nodes. Fig.5.12(e) shows the delay for successful content demands and we can see that for a sparse network topology with average degree of 5, the delay is longer while for node degree of 20 the delay is slightly higher than 10 due to nodes' interference. Figs.5.12(b), 5.12(d), 5.12(f) show the same results for independent optimization scheme in which we see just a slight difference.

How does our system scale with the number of contents ?

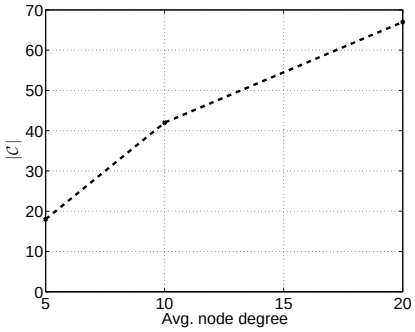
Fig. 5.13 shows the results when the number of content simulated is set to 1, 2, 4 and 8. In Fig.5.13(a) the number of replicas increases with number of contents which is reasonable since nodes have a capacity constraint. Fig.5.13(c) shows the difference in term of average workload at facility nodes. Fig.5.13(e) shows that the delay is longer while for more contents due to nodes' interference. Figs.5.13(b), 5.13(d), 5.13(f) show very similar results for independent optimization scheme.

How does our system scale with network size ?

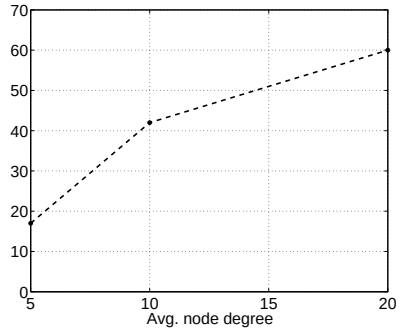
Fig. 5.14 shows the results when we vary the network size from 100 to 1000 nodes with $v_(j)$=15MB (We do this while keeping average node degree at 10 by extending the simulation area from 0.3, 1.5 and 3 km^2 for 100, 500 and 1000 nodes respectively). Fig.5.14(c) shows the difference in term of average workload at facility nodes in which we just see a little difference when the number of nodes is set to as low as 10. When the network size increases from 100 to 1000 we see the same distribution of workload which mean our system scale well with number of nodes. Fig.5.14(e) shows that the delay is not higher when the network size increase which is also a good indicator for system scalability. We can observe similar results for independent optimization scheme in Figs.5.14(b), 5.14(d), 5.14(f).

5.6.4 Replica allocation

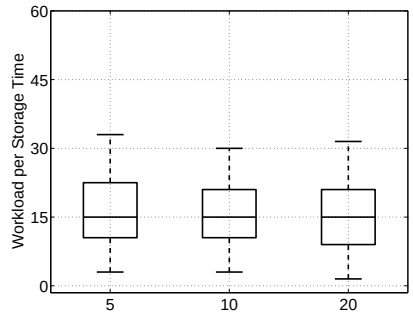
We now turn out attention to the allocation of replica for each content. To the best of our knowledge there is no work on the allocation of replicas for content in wireless network. In the Internet, authors in [34] worked on replication stategy in unstructured peer-to-peer networks. They assumed that users search for content in random nodes and proved that the strategy to allocate replica to content is optimal in terms of successful queries lies between the uniform and the proportional distribution based on the query rate, namely the square root distribution. The allocation percentage $\frac{\mathcal{C}(h)}{\sum_h \mathcal{C}(h)}$ for a content h is proportinal to the square root of total demand per second $\Lambda(h)$ for that content :



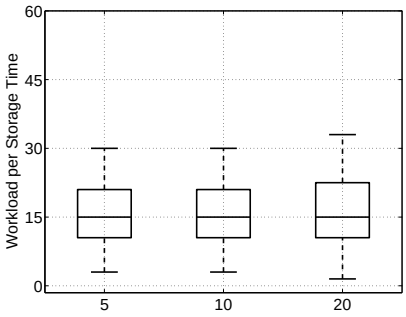
(a) Avg. number of replicas per content (Joint Opt.)



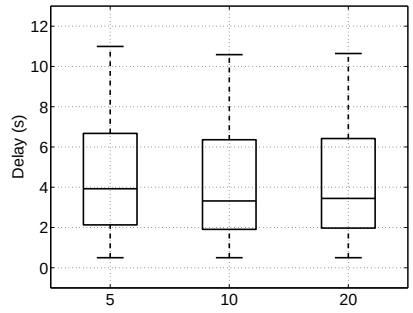
(b) Avg. number of replicas per content (Ind. Opt.)



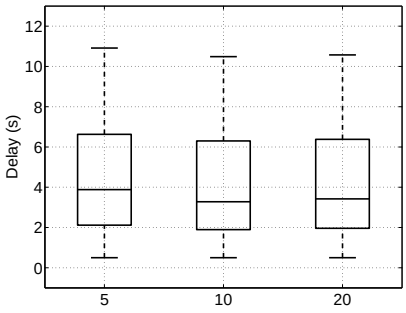
(c) Workload per storage time (Joint Opt.)



(d) Workload per storage time (Ind. Opt.)

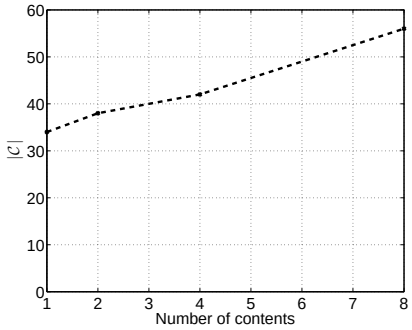


(e) Delay (Joint Opt.)

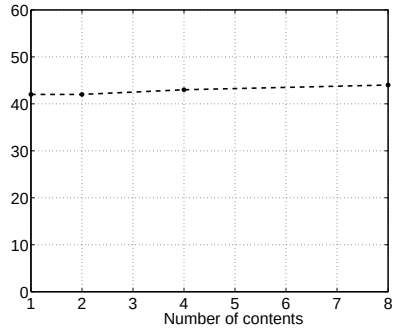


(f) Delay (Ind. Opt.)

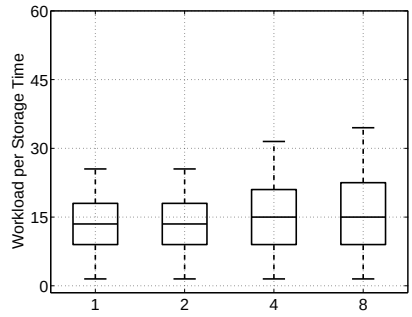
Fig. 5.12 – Delay and workload with different node degree : 5, 10 and 20



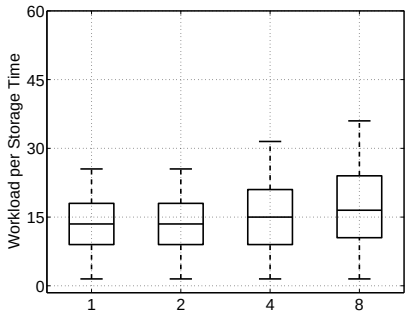
(a) Avg. number of replicas per content (Joint Opt.)



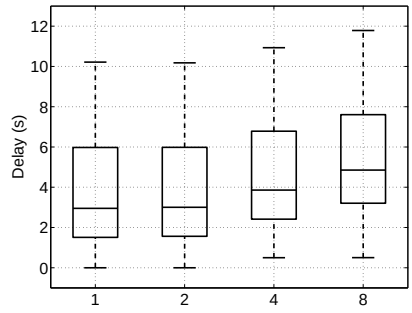
(b) Avg. number of replicas per content (Ind. Opt.)



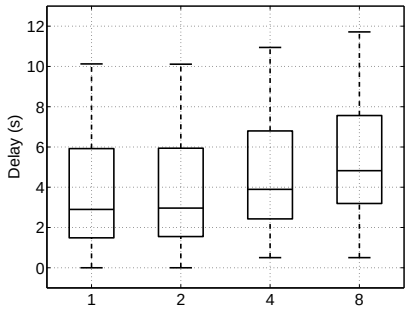
(c) Workload per storage time (Joint Opt.)



(d) Workload per storage time (Ind. Opt.)

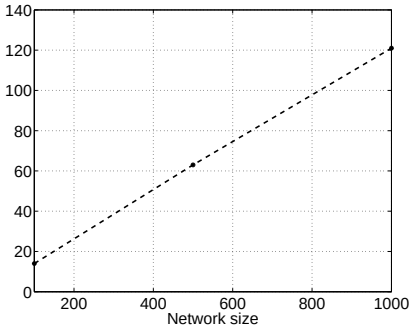


(e) Delay (Joint Opt.)

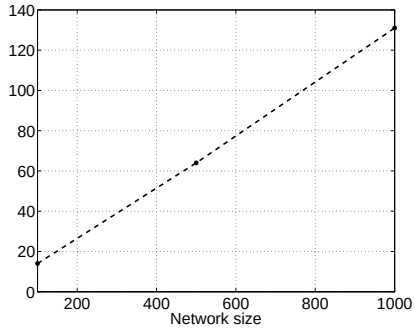


(f) Delay (Ind. Opt.)

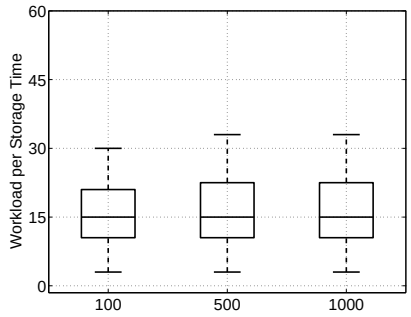
FIG. 5.13 – Delay and workload with 1, 2, 4, 8 contents



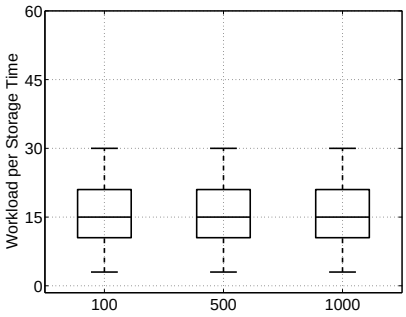
(a) Avg. number of replicas per content (Joint Opt.)



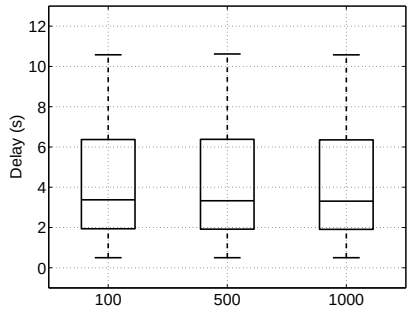
(b) Avg. number of replicas per content (Ind. Opt.)



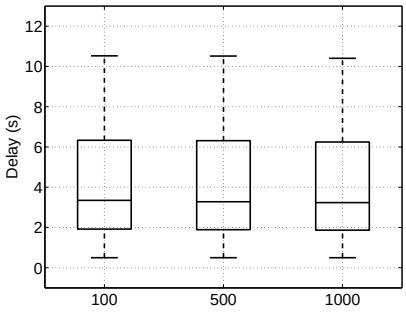
(c) Workload per storage time (Joint Opt.)



(d) Workload per storage time (Ind. Opt.)



(e) Delay (Joint Opt.)



(f) Delay (Ind. Opt.)

Fig. 5.14 – Delay and workload with various network size : 10, 100, 1000 nodes

$$\frac{\mathcal{C}(h)}{\sum_{h=1}^{M}\mathcal{C}(h)} = \frac{\sqrt{\Lambda(h)}}{\sum_{h=1}^{M}\sqrt{\Lambda(h)}}$$

In [113] authors assumed that nodes use an expanding ring search for content and in such context they showed that a allocation of replicas proportional to content demand probability is optimal.

We plot our simulation results in replica allocation percentage for 4 contents with different querying popularity and υ=5–40MB Fig. 5.15 shows the proportion of replica for each item and the corresponding proportion of query rate. We observe that our scheme achieves an allocation between the square root and the proportional distribution which means our results approximate roughly the optimal replication strategy. The allocation when we have υ_j=5–15MB is closer to the square root distribution than when υ_j=40MB, which allows us to say that when replica nodes reserve more generously the resources to serve requests, the allocation tends to follow the proportional distribution. When the budget is stricter, the allocation follows better the square root rule.

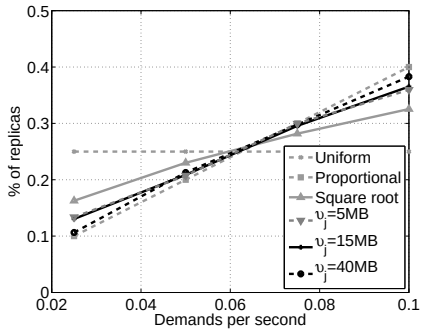


FIG. 5.15 – Distribution of $\mathcal{C}$ in comparison with uniform, proportional and square root allocation.

5.7 Content access mechanisms

The perfect-discovery mechanism is difficult to implement in pratice. In this section, we study the impact of alternative content access mechanisms like flooding and scanning. Tab. 5.7 summarizes the notations used in our figures to refer to content access mechanisms.

Content access mechanism	*Notation*
Perfect-discovery	PM
Scanning	SM
Scoped-flooding	FM
Scoped-flooding with selective reply	FM*

TAB. 5.7 – Notation for different content access mechanisms.

The workload experienced by a replica node is determined by the mechanism used by nodes to access the content through device-to-device communications. We identify two

phases : a content query transmission, and a query reply transmission (by the replica node carrying the desired content). We investigate several mechanisms for content access focusing on the content query transmission phase, and we assume that the identity of the nodes that have relayed the query is added to the query message itself. After a replica node with the desired content is found, it will reply to the node issuing the query through a multihop transmission process that backtracks the path from the replica node to the querying node, exploiting the identity of relay nodes included in the query message. This backtracking, although possibly occurring through multiple hops, makes no use of ad hoc routing protocols, as it is completely application-driven.

As far as the query transmission phase is concerned, the following three mechanisms are envisioned.

Scoped-flooding : content requests are simply flooded with a limited scope using application-layer broadcast. The "scope" can be defined as the maximum number of hops through which a query propagates, i.e., neighboring nodes propagate a query until it has traversed a maximum number of hops H, after which it is dropped. Clearly, if the request is received by a replica node, the content is served and the query is not propagated any further.

The main drawback of flooding is that multiple content replicas within reach of a node will be "hit" by a request. Beside causing congestion when a large number of replica nodes reply to the querying node, this also creates an artificially inflated workload, which conflicts with the underlying assumptions in Defs. 1 and 2. In our experiments, we explore the benefits of a *selective reply* mechanism that replica nodes can use to mitigate excessive workloads due to flooding. When selective reply is enabled, a replica node replies to a query with a probability that is inversely proportional to the hop-count of query messages.

Scanning : instead of flooding in all directions, the node issuing the query specifies an angular section within which the query is to be propagated by other nodes. In order to do so, it includes its own position (e.g., obtained through GPS), and the angle boundaries. All nodes receiving the query rebroadcast it only if their position satisfies the angular requirements, until a replica node is found or the query has traversed a maximum number of hops H. Nodes that are not within the angular section specified in the query will discard the message. If no reply is received after a timeout, a new sector is scanned, and the scanning of all sectors is repeated till either a reply is received or a maximum number of retries has been achieved. The number of sectors S, each of width $2\pi/S$, is a parameter of the system.

The complexity of this mechanism is comparable to that of scoped-flooding, however we will show that it reduces the overhead experienced with flooding. On the downside, scanning requires nodes to be able to estimate their position and reduces the probability of solving a query with respect to flooding-based solutions. Indeed, when a replica is within the sector currently scanned by the requesting node but it is farther than one hop away, one or more relay nodes would be needed to reach the replica. However, if at least one of the available relays are located outside the sector, the replica is not reached and the content query remains unsolved. Thus, the narrower the sector, the more likely that the query is unsuccessful.

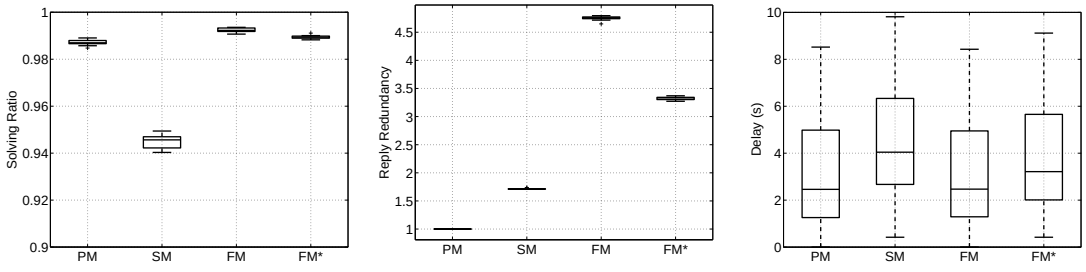
Perfect-discovery : in this case, which is added for comparison purposes, nodes are assumed to be able to access a centralized content-location service that returns the identity of the closest content replica in terms of euclidean distance. We do not address the problem of how the centralized service is updated, save by noting that it is certainly responsible for additional overhead and complexity, and that it can be managed through a separate protocol using unicast or multicast transmissions. A query is propagated using application-driven broadcast, but only the intended replica node (specified in the query) will serve the content. Any other replica node will discard the request.

On the one hand, this content access mechanism is the most demanding because it requires the presence of an auxiliary service to discover the closest replica. On the other, only one replica node carries the workload generated by the closest users, which is the hypothesis to the optimization problems stated in Sec. 5.1.

Finally, we improve the query/reply propagation process by adopting the PGB technique [90] for selecting forwarding nodes and sequence numbers to detect and discard duplicate queries.

We evaluate the performance of the four content access mechanisms listed in Tab. 5.7, in terms of the following metrics :

- *solving ratio*, i.e., the ratio of satisfied requests to the total number of queries generated in the network. The target value is 1, corresponding to 100% of solved queries ;
- *reply redundancy*, i.e., the number of replies to the same request, received from different replica nodes. The target value is 1, corresponding to one reply to each query ;
- *latency*, i.e., the delay experienced by nodes to access information.



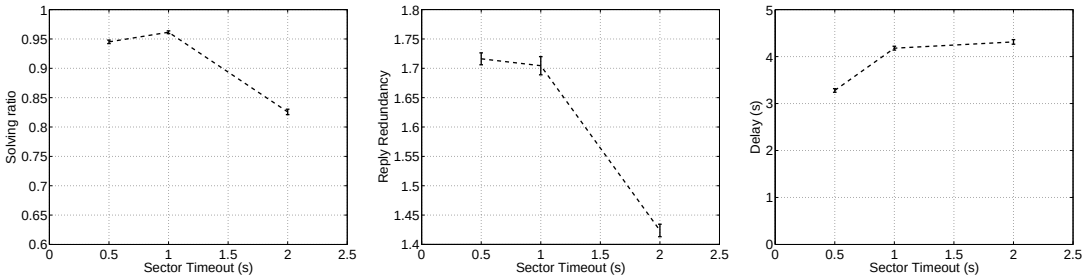
(a) Solving ratio (b) Reply redundancy (c) Latency

FIG. 5.16 – Performance of content access mechanisms, in a mobile scenario ($|\mathcal{C}|$=30 and τ=100 s).

Fig. 5.16 shows the following quantiles of the access performance metrics for $|\mathcal{C}|$=30 : the 25% (resp. 75%) as the lower (resp. higher) boundary in the plot box, the 50% as the line within the plot box. The brackets above and below the plot box delimit the support of the CDF for that metric. For all access mechanisms, the median solving ratio (Fig. 5.16.a) is higher than 0.9, which indicates that only a small fraction of queries cannot reach a content replica. The scheme that exhibits a slightly worse performance appears to be the scanning scheme, which is seldom unable to reach a replica through relay nodes (see Sec. 5.3).

Fig. 5.16.b depicts the extent to which flooding-based mechanism can artificially inflate the workload of replica nodes : in our experiments, a single query can hit almost 6 replicas in the worst case[3]. High redundancy has a direct consequence on the behavior of the replication mechanism, as we discuss in detail later. We observe that the selective reply mechanism can halve the level of redundancy typical of flooding, and that node mobility helps in reducing redundancy in all schemes. It is important to notice that the scanning mechanism achieves a low reply redundancy, without requiring the presence of an auxiliary mechanism to help consumer nodes target the closest replica.

Latency for each content access scheme is shown in Fig 5.16.c : scanning is clearly the outlier in this figure.



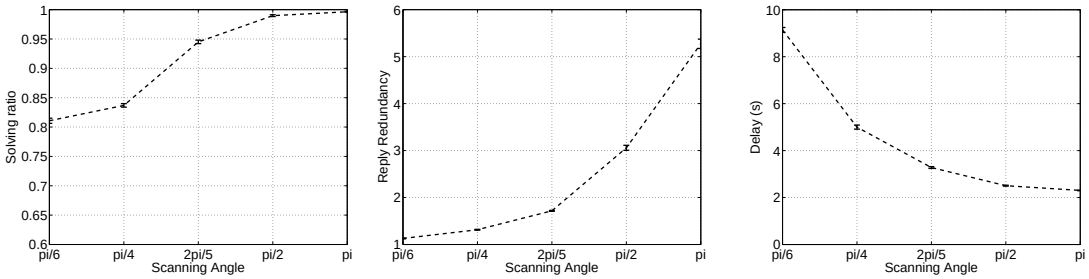
(a) Solving ratio (b) Reply redundancy (c) Latency

Fig. 5.17 – Performance of the scanning mechanism as a function of the sector timeout (scanning angle $2\pi/5$, $|\mathcal{C}|$=30 and τ=100 s).

We now provide more details on the performance of the scanning mechanism. Fig. 5.17 shows the impact of the time spent waiting for a reply on each sector composing the scanning horizon ; we term this time *sector timeout*. The solving ratio is marginally affected by this parameter. Indeed, delaying the search in the next sector by a longer time has the mobile node skip larger portions of the area : two consecutively scanned sectors turn out to be non-adjacent due to the change in the position of the node issuing the query. The redundancy decreases with longer sector timeouts : indeed, the longer the timeout the higher the probability that a node scans another sector, i.e., it issues another query, only when no replica is available in the current sector. Instead, the latency deteriorates with a longer sector timeout because it will take more time to hit the sector where the replica is located. Mobility seems to have a positive effect on the delay, even with longer sector timeouts, since most solved queries are due to close-by replica nodes (farther nodes may reply after the querying node has moved away).

Fig. 5.18 shows the impact of the number of angular sectors in which the space around a node is partitioned, as determined by the scanning angle parameter. As explained above, a small scanning angle might reduce the probability for a query to reach a replica, hence the lower solving ratio with small angles. We observe a similar effect on redundancy : smaller angles limit the number of replicas "hit" by a query. Instead, the latency decreases with larger scanning angles because the probability to find a replica within a sector increases.

[3]We also run experiments with lower values of the flooding scope H : redundancy, which is proportional to H, remains higher in flooding compared to other schemes.



(a) Solving ratio (b) Reply redundancy (c) Latency

FIG. 5.18 – Performance of the scanning mechanism as a function of the scanning angle (sector timeout = 0.5s, $|\mathcal{C}|$=30 and τ=100 s).

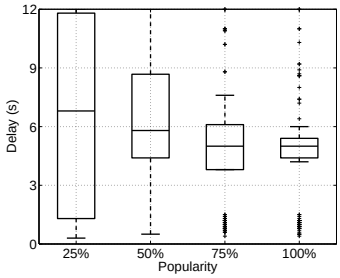
In summary, we analyzed the performance of several content access mechanisms, ranging from simple flooding-based to complex schemes requiring perfect-discovery. With the setting used in our tests, we showed that a content query hits at least one replica with very high probability (Fig. 5.16) and that access delay can be slightly larger than 1 s with the scanning mechanism. Despite having larger delays, our results (Figs. 5.17 and 5.18) showed that the scanning mechanism achieves very low redundancy (comparable to perfect-discovery) and bears little costs in terms of complexity (which is comparable to flooding).

5.8 Performance vs. the epidemic content distribution

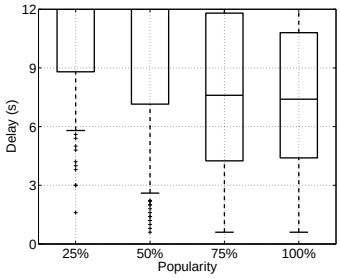
We now study the advantage of our replication scheme against a simple epidemic content distribution scheme in which nodes are willing to store and transfer the replica whenever they are in contact range. In such scenario, we consider a pull-based mechanism for issuing content query. First, a node sends query to ask for the content via device-to-device communication. If the query hits a replica node, the content is sent to the querying node. Otherwise, the query will be ignored. In case that node cannot receive the reply after a determined number of retries, it can download the content from an external server. After successfully downloading the content, regardless from server or via device-to-device communication, node becomes a content replica and stores it for infinite time to serve neighbors' demand following the epidemic scheme. In contrast, for the replication scheme nodes become replica only if it download the content from server or the content replica role is hand-over to them and nodes store the content for only a storage time of τ seconds. In such context, we define two metrics to evaluate the performance of the two schemes : the delay and the number of external downloads defined as following.

- Delay : the number of seconds from when a node starts sending the first query until the query is fullfilled by other nodes or by downloading from server. Hence the delay is bound by the request timeout.
- Ratio of external downloads : the ratio between the number of queries that are fullfilled by an external server and the number of queries that are served by replica nodes.

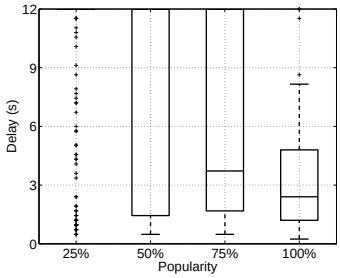
For the epidemic scheme, we consider two querying mechanisms : at a given time node can send a query to only a random neighbor (unicast) or to all neighbors (flooding) in its communication range. We limit the TTL to 1 and the query will not be rebroadcasted by neighbors. For the replication scheme we use the perfect-discovery and scanning mechanisms with scanning angle $\pi/5$. The maximum number of retries is set to 6 and retry interval is 2 seconds. In the replication scheme, τ is set to 100 seconds. In both replication and epidemic scheme, we run simulation with 1 content of 1MB and we bootstrap with 1 replica node. We use again the scenario of 320 nodes with Stationary Random Waypoint mobility model. We focus our study in different content popularity settings (e.g. nodes that do not belong to content popularity will not participate in storing and forwarding the content).



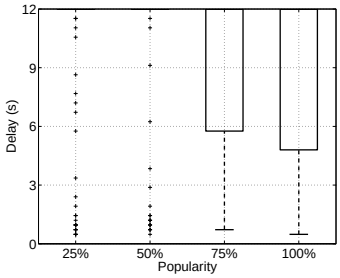
(a) Delay with replication and perfect discovery query mechanism



(b) Delay with replication and scanning query mechanism



(c) Delay with epidemic diffusion and flooding query scheme



(d) Delay with epidemic diffusion and unicast query scheme

Fig. 5.19 – Performance vs the epidemic content distribution scheme.

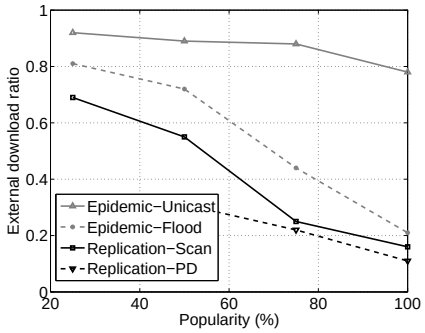


Fig. 5.20 – Percentage of external downloads.

Fig. 5.19(a) and Fig. 5.19(d) show the delay for the replication and epidemic (unicast) scheme for different content popularity. In average for the replication scheme 50% of nodes

reach the content in no more than 6s by device-to-device communication and at most 20% of nodes should download the content from external network. For epidemic scheme we see that in epidemic scheme the percentage of nodes that have a query fulfillment delay up to the timeout (12s) is high and increases when the popularity decreases and even at 100% popularity only 28% of nodes reach the content within 6s and at least 67% of nodes can not take advantage of the device-to-device connection. Fig. 5.19(c) shows the delay for the epidemic scheme but with query flooding mechanism for different content popularity : the result is improved comparing with the unicast querying scheme but for the case of less than 100% popularity its performance is still worse than the replication scheme. Fig. 5.20 gives us the ratio of external downloads in which we see that even for the flooding mechanism, the number of external downloads is consistently higher than the replication scheme, which means that there will be more congestion at server. These results can be explained as following : *The replication scheme helps to place the content replica at querying nodes that are surrounded by content demanders.* Contrarily, epidemic scheme just places the content at any node issuing a query without considering the content popularity in that node's vicinity. In brief, if the content popularity is not 100%, a content distribution based on replication scheme is performing better than an epidemic forwarding.

5.9 Conclusion

In this work, we focused on content replication problem in mobile networks where users can access content through device-to-device communications, and we addressed the joint optimization problem of :

- establishing the number of content replicas to deploy in the network.
- finding their most suitable location.
- letting users efficiently access content through device-to-device communications.

To achieve these goals, we proposed a distributed and lightweight mechanism that lets content replicas move in the network according to random patterns : network nodes temporarily store content, which is handed over to randomly selected neighbors. Hence the burden of storing and providing content is evenly shared among nodes and load balancing is achieved. In our mechanism, replica nodes are also responsible for creating content copies or drop them, with the goal of obtaining an ideal number of content replicas in the network. The workload experienced by a replica node is the only measured signal we use to trigger replication and drop decisions.

We studied the above problems through the lenses of facility location theory and showed that our lightweight scheme can approximate with high accuracy the solution obtained through centralized algorithms. Clearly, network dynamics cause a high toll in terms of complexity to reach an optimal replication and placement of content, and we showed that our distributed mechanism can readily cope with such a scenario. Moreover, we removed the typical assumption of assigning content demand points to their closest replica and investigated several content access schemes, their performance, and their impact on content replication.

Lastly, we studied the flexibility of our scheme when content demand varies in time and in space : our experiments underlined the ability of our approach to adapt to such variations while maintaining accuracy in approximating an optimal solution obtained through centralized algorithms.

Instead of designing distributed approximation algorithms of the optimal solution to facility location problems, which either require global (or extended) knowledge of the network [7, 73] or are unpractical [86], we extend our store-and-forward mechanism with a distributed replication algorithm that bases its decisions on local measurements only and aims at evenly distributing among nodes the demanding task of being a replica provider. Results show that our mechanism, which uses *local measurements only*, is extremely precise in approximating an optimal solution to content placement and replication, robust against network mobility, adaptable to different initial distributions of replicas and flexible in accommodating various content access patterns, including variation in time and space of the content demand.

In the next chapter , we will study the behavior of our scheme considering user selfishness. We will relax the assumption of a cooperative setting and analyze selfish replication with tools akin to game theory. In [84] we show that the system we study can be modeled as an anti-coordination game, and our goal is to understand how to modify or extend the ideas presented in this work to achieve strategy-proofness.

5.10 Relevant publication

La, Chi Anh ;Michiardi, Pietro ;Casetti, Claudio ;Chiasserini, Carla-Fabiana ;Fiore, Marco *A lightweight distributed solution to content replication in mobile networks*, In proceedings of IEEE Wireless Communications & Networking Conference, WCNC 2010

La, Chi Anh ;Michiardi, Pietro ;Casetti, Claudio ;Chiasserini, Carla-Fabiana ;Fiore, Marco *Content Replication in Mobile Wireless Networks*, Journal version, In preparation

Chapter 6

Content replication in selfish environment

In previous chapters, we studied the problem of replication in a cooperative networks The replication factor (e.g. number of replicas) is assumed to depend on the dedicated budget of nodes due to their resource constraint. However, users can also behave selfishly, e.g. they just want to dedicate only a minimum budget to help the system working for them. In this chapter, we define and study a new model for the replication problem in a heterogeneous wireless network under a flash-crowd scenario, in which nodes could determine the replication factor themselves. Using non-cooperative game theory, we cast the replication problem as an anti-coordination game. We start by defining the social optimum in the general case and then focus on a two-player game to obtain insights into the design of efficient replication strategies. Based on the theoretical findings, our current work focuses on the development of strategies to be implemented in a practical network setting.

6.1 Problem modeling

We address the problem of content replication in a heterogeneous wireless network : mobile nodes can connect to a cellular network (e.g., a mobile broadband network such as 3G) and are able to form a temporary multi-hop network (e.g., a 802.11-based device-to-device network). We assume content to be hosted at an origin server in the Internet, which can only be accessed through the cellular network. Some nodes are able to download through cellular network a fresh version of the content. This content which will be stored and served to other nodes via device-to-device network. Content popularity drives access behavior : in this chapter we assume a "flash crowd" scenario, in which users discover a new content and wish to access it concurrently. As a consequence, access congestion determines to a large extent the download performance, for both the cellular and the device-to-device network. For simplicity, here we consider nodes to be interested in a single information object.

The problem of replication has received a lot of attention in the past due to its importance in enhancing performance, availability and reliability of content access in wireless systems. However, this problem has been addressed often under the assumption that nodes

would cooperate by following a strategy that aims at optimizing the system performance, regardless of the costs incurred by each individual node. Our goal, instead, is similar to the one in [32], in that we build a model where nodes are selfish, i.e., they choose whether to replicate or not the content so as to minimize their own cost. Our work however differs from [32] in how content demand is modeled.

Let V be a set of nodes uniformly deployed on area $\mathcal{A} = \pi R^2$. We assume the presence of a single base station for mobile broadband access and we let the radio range associated to device-to-device communications to be r for every node. For sake of simplicity, we consider only one information object of size L bytes to be available for download ; the object requires f updates per second from the origin server (in order to obtain a fresh copy) and each update implies the download of U bytes.

We now define the replication game. We assume a simulatenous move game : every player selects their strategy at the same time (with no communication among players possible).

- Let V be the set of players (or the set of nodes, we call "node" as "player" later on in this Chapter), with $|V| = N$;
- Let $\mathcal{S}_i$ be the set of all possible strategies for player $i \in V$. Additionally, let $s_i \in \mathcal{S}_i$ be the strategy of player i, where $s_i = \{1, 0\}$. Also, define $s = \{s_1, s_2, ..., s_i, ..., s_I\}$ to be a strategy profile ;

In the following, it will be useful to split the set of players in two subsets. Let $\mathcal{C} \subseteq V$ be the set of players whose strategy is to access object from the origin server and store it, that is $s_i = 1$, $\forall i \in \mathcal{C}$, and $\mathcal{N} \subseteq V = V \setminus \mathcal{C}$ the set of players whose strategy is to access a stored object, that is $s_i = 0$, $\forall i \in \mathcal{N}$. Also, let $|\mathcal{C}| = x$ and $|\mathcal{N}| = N - x$.

Given a strategy profile s, the cost incurred by player i is defined as :

$$C_i(s) = \beta_i \mathbb{1}_{s_i=1} + \gamma_i \mathbb{1}_{s_i=0} \tag{6.1}$$

where :

- β_i is the air time cost (*i.e.,* the radio resources consumption) if i obtains the content through the 3G network ;
- γ_i is the air time cost if i obtains a stored version of the content through device-to-device communication ;
- $\mathbb{1}_{s_i}$ is the indicator function

We now define precisely the two terms β_i and γ_i. To this end, let us introduce the following quantities :

- R_{3G} and R_h are the bit rate offered, respectively, by the 3G access network and the (per hop) device-to-device communication ;
- T_c is the time for which node $i \in \mathcal{C}$ stores an object ;
- h is the *average* number of hops required to access the closest object through device-to-device communication, assuming a uniform distribution of nodes on $\mathcal{A}$ and a uniform distribution of nodes replicating the information object. Formally, $h = \sqrt{\frac{\mathcal{A}}{x\pi r^2}} = \frac{R}{r}\frac{1}{\sqrt{x}}$.

With these definitions at hand, we can now focus on the two cost terms, β_i and γ_i. We define β_i such as :

$$\beta_i = \Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big]|\mathcal{C}| \tag{6.2}$$

where the first term on the right-hand side of the equation accounts for the cost to download the information object for the first time and the second term accounts for the additional cost to download the information updates. Note that L/R_{3G} and U/R_{3G} are the air time consumed to download an entire object or its updates, while $T_c f$ is the number of updates performed by a node currently storing the object. Note also that Eq. 6.2 models the congestion incurred by nodes trying to access the information object at the same time : the bit rate R_{3G} is inversely proportional to the number of concurrent users accessing a *single* 3G base station [108].

As for γ_i, we let :

$$\gamma_i = \Big[h\frac{L}{R_h}\Big]|\mathcal{N}| \tag{6.3}$$

where hL/R_h is the air time consumed to access the current version of the stored object. Eq. 6.3 models the congestion cost created by multiple simultaneous access to available objects by device-to-device nodes : the bit rate R_h is inversely proportional to the number of nodes accessing a stored version of the information [1].

In words, we express the cost C_i incurred by player i as the access and update costs to object o, which is given by β_i if player i choses to access the object from the origin server and store it and by γ_i if player i choses to access the object from the nearest replica, *provided that at least one player decided to replicate the object.*

We focus on access costs, neglecting the energy costs a replica node has to bear to serve other nodes. Although we rekon this to be a simplification of the problem, we will see in the following that the resulting game conserves its interest.

The social cost of a given strategy profile is defined as the total cost incurred by all players, namely :

$$\begin{aligned} C(S) &= \sum_{i\in\mathcal{C}} \Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big]|\mathcal{C}| + \sum_{i\in\mathcal{N}} h\frac{L}{R_h}|\mathcal{N}| \\ &= x^2\Big[\frac{L}{R_{3G}} + (T_c f)\frac{U}{R_{3G}}\Big] + (N-x)^2\frac{R}{r}\frac{1}{\sqrt{x}}\frac{L}{R_h} \end{aligned} \tag{6.4}$$

where we replaced the expression accounting for the average number of hops h. Hence, the social cost can be computed as a function of the fraction of players that chose to act as replica nodes, namely x. Note that Eq. 6.4 illustrates a game that belongs to the general family of *congestion* or *crowd* games.

[1] Our congestion model is more conservative than the capacity scaling law defined in [44].

6.2 Socially optimal cost

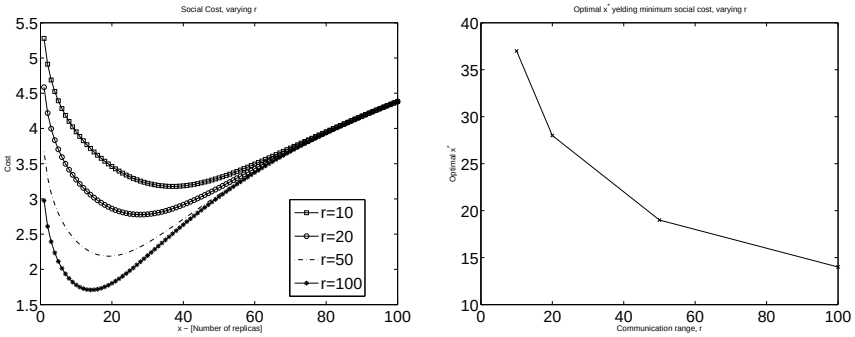
The social optimum cost, referred to as $C^*(S)$ for the remainder of this chapter, is the minimum social cost. The social optimum cost will serve as an important base case against which to measure the cost of selfish replication. We define $C^*(S)$ as :

$$C^*(S) = \min_s C(s) \tag{6.5}$$

The social optimum cost can be also rewritten as a function of x, that is :

$$\begin{aligned} C^*(x) &= \min_x C(x) \\ &= \min_x \Big\{ x^2 \Big[\frac{L}{R_{3G}} + (T_c f) \frac{U}{R_{3G}} \Big] + (N-x)^2 \frac{R}{r} \frac{1}{\sqrt{x}} \frac{L}{R_h} \Big\} \end{aligned} \tag{6.6}$$

We now plot the social cost as a function of the number x of players choosing strategy $s_i = 1$, and analyze the impact of the following system parameters : the communication range of a mobile node $r \in \{10, 20, 50, 100\}$ meters, the node density which is obtained by fixing N and varying $R \in \{100, 500, 1000, 5000\}$ meters and finally the device-to-device bit rate $R_h \in \{2, 11, 24, 54\}$. Parameters that are not varied are : communication range $r = 20$ meters, the 3G bit rate $R_{3G} = 2$ Mbps, and $N = 100$. Furthermore, we let : $L = 1000$ Bytes, $U = 100$ Bytes, $T_c = 100$ sec. and $f = 0.01$ req/sec.

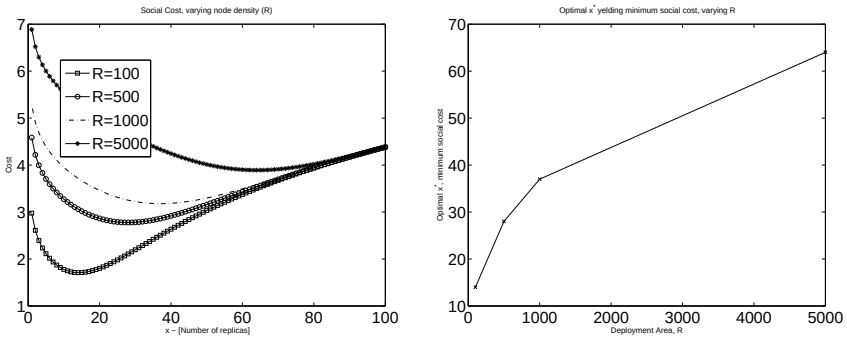


(a) (b)

FIG. 6.1 – $C(x)$ (a) and $C^*(x)$ (b) with varying communication range.

Fig. 6.1(a) represents the social cost $C(x)$ as a function of the number of replicas, when varying the communication radio range of nodes. We are interested in the minimum social cost, which can be easily spotted on the figure due to the convexity of $C(x)$. We observe that increasing the communication range of nodes implies that the minimum social cost can be achieved with fewer players selecting to replicate object o. Indeed, increased communication capabilities imply a decreased average hop count, hence nodes are better off accessing o through a replica. Fig. 6.1(b) illustrates the optimal number of providers x^* that minimizes the social cost as a function of the communication range of nodes, that is :

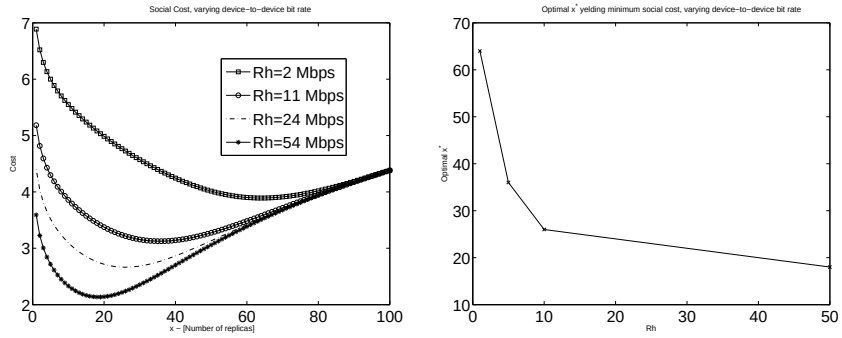
$$x^* = \arg\min_x C(s)$$



(a) (b)

Fig. 6.2 – $C(x)$ (a) and $C^*(x)$ (b) with varying node density.

Fig. 6.2(a) depicts the social cost $C(x)$ when the node density is varied. Similarly to our previous observation, higher node density (*i.e.*, lower R) imply fewer hops to reach the closest replica, and the minimum social cost is achieved with fewer providers. Fig. 6.2(b) depicts x^* as a function of the node density.



(a) (b)

Fig. 6.3 – $C(x)$ (a) and $C^*(x)$ (b) with varying throughput ratio.

Fig. 6.3 shows $C(x)$ as a function of the bit rate accessing o through a multi-hop route when the bit rate for a direct access from the origin server is fixed. Fig. 6.3 illustrates x^* using as parameter the device-to-device bit rate.

6.3 A two player game

Let's now further simplify the problem we discuss in this chapter, and assume only two players/nodes are involved in playing the game. We revert to the normal form game with

the payoff matrix described in Tab. 6.1, where :

$$C(x) = x^2 k_1 + \frac{(N-x)^2}{\sqrt{x}} k_2$$
$$k_1 = \frac{1}{R_{3G}}\Big(L + T_c f U\Big)$$
$$k_2 = \frac{1}{R_h}\Big(\frac{R}{r} L\Big)$$

The two-player version of the replication game involves two players, $1, 2$ whose strategy set is $\{1, 0\}$: 1 implies that player i choses to fetch the object o from the origin server and store it, while 0 indicates a player chosing to access o through a replica. Tab. 6.1 indicates that when both players decide to access o through a replica, none can actually get o, hence the cost is ∞.

	$s_2 = 1$	$s_2 = 0$
$s_1 = 1$	$\Big(2k_1, 2k_1\Big)$	$\Big(k_1, k_2\Big)$
$s_1 = 0$	$\Big(k_2, k_1\Big)$	(∞, ∞)

TAB. 6.1 – Matrix form of the two-player game : entries indicate the cost to each player.

Let's rewrite Tab. 6.1 using payoffs instead of costs : Tab. 6.2 contains the inverse of the costs.

	$s_2 = 1$	$s_2 = 0$
$s_1 = 1$	$\Big(\frac{1}{2k_1}, \frac{1}{2k_1}\Big)$	$\Big(\frac{1}{k_1}, \frac{1}{k_2}\Big)$
$s_1 = 0$	$\Big(\frac{1}{k_2}, \frac{1}{k_1}\Big)$	$(0, 0)$

TAB. 6.2 – Matrix form of the two-player game : entries indicate the payoff to each player.

Let's examine the payoff matrix illustrated in Tab. 6.2. Clearly, strategy 0 is strictly dominated by strategy 1 if and only if $2k_1 < k_2$: in this case, we would have only one Nash Equilibrium (NE), which is $(1, 1)$. When the benefits from fetching o from the origin server are higher than by accessing it through a replica the best strategy is to chose 1.

Instead, when $2k_1 > {k_2}^2$, there are no strictly (neither weakly) dominated strategies. In this case we face a so called *anti-coordination game*, in which it is possible to show that the best strategy for a player would be to alternate replicating and non-replicating periods. For the sake of clarity, we show an example were we use the same parameters we analyzed in the previous section on the socially optimal cost.

[2]This is the case that happens in practice, e.g., with the values of the system parameters used to compute the minimum cost.

$$k_1 = \frac{1}{2 * 10^6}(8000 + 100 * 0.1 * 800) \approx 0.008$$
$$k_2 = \frac{1}{21 * 10^6}(\frac{500}{20} * 4000) \approx 0.0095$$

Hence, $2k_1 > k_2$ which implies that, in a realistic setting, our players face an anti-coordination game, in which players randomize their strategies. Indeed, there are two conflicting (in terms of payoffs) NE points, i.e., the $(0, 1)$ and $(1, 0)$ strategy profiles. It is well known that mixed-strategies profiles and expected payoffs π_i can be derived as follows. Suppose player 2 chooses 1 with probability $p_2(1)$ then the expected payoff for player 1 to play 1 corresponds to $(1, 1)$ with the probability of $p_2(1)$ and $(1, 0)$ with the probability of $1 - p_2(1)$:

$$\mathbb{E}[\pi_1(1, p_2(1))] = p_2(1)\frac{1}{2k_1} + (1 - p_2(1))\frac{1}{2k_1} = \frac{2 - p_2(1)}{2k_1}$$

Similarly, the expected payoff for player 1 to play 0 corresponds to $(0, 1)$ with the probability of $p_2(1)$ and $(0, 0)$ with the probability of $1 - p_2(1)$:

$$\mathbb{E}[\pi_1(0, p_2(1))] = p_2(1)\frac{1}{k_2} + (1 - p_2(1))0 = \frac{p_2(1)}{k_2}$$

Hence, $p_2(1) = \frac{2k_2}{2k_1+k_2}$. Due to the symmetry of the game, player 1 chooses 1 with probability $p_i(1) = p_1(1) = p_2(1) = \frac{2k_2}{2k_1+k_2} \forall i \in 1, 2$. Considering the joint mixing probabilities, the expected payoff for both players is :

$$\mathbb{E}[\pi^*{}_i] = p_i(1)\frac{2 - p_i(1)}{2k_1} + (1 - p_i(1))\frac{p_i(1)}{k_2} = \frac{2}{2k_1 + k_2} \forall i \in 1, 2$$

It is worth noting that in this anti-coordination game the mixed strategy NE is inefficient. Indeed, when players can correlate their strategies based on the result of an observable randomizing device (i.e., a correlated equilibrium can be achieved), the expected payoff which corresponds only to $(1, 0)$ and $(0, 1)$ is :

$$\mathbb{E}[\hat{\pi}_i] = p_i(1)\frac{1}{k_2} + (1 - p_i(1))\frac{1}{2k_1} = \frac{k_1 + k_2}{2k_1k_2} \forall i \in 1, 2$$

We observe that $\mathbb{E}[\hat{\pi}_i]$, is strictly larger than $\mathbb{E}[\pi_i]$. This clearly suggests that some correlation among the nodes' actions should be introduced in order to improve system performance.

Despite there are not dominated strategies in our game (neither strictly nor weakly), it is straight-forward to show that there are two Nash Equilibirum (NE) that corresponds to the $(0, 1)$ and to the $(1, 0)$ strategy profile. It is clear that one NE is more favorable to one player than the other. We can also derive the mixed-strategy NE point by analyzing

the expected payoff for a player, given the probability for the other player to chose one strategy. We can compute the expected payoff for a player from the above example, given the probability for the other player to chose a strategy (Fig.6.4) and we can display the best response mappings (Fig.6.5)

We note that the game is not a zero-sum game, hence we cannot apply directly the minimax theorem. The mixed strategy NE exisists, and in this particular example, it turns out to be achieved when players randomize their strategies leaning towards the caching strategy. This comes from the fact that players cannot incur the risk of having the lowest payoff in case no player selects the caching strategy.

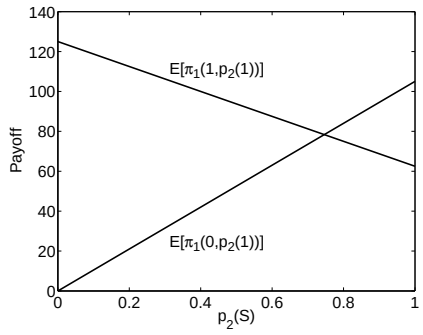


FIG. 6.4 – Expected payoff of player 1 in the two player game with k_1= 0.008 and k_2=0.0095.

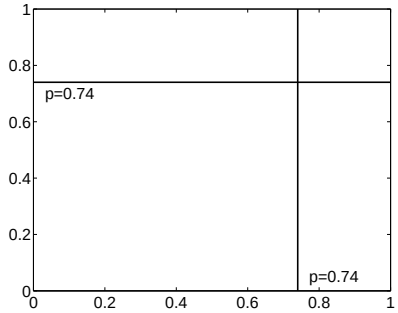


FIG. 6.5 – Best response mappings for the two player game.

	$s_2 = 1$	$s_2 = 0$
$s_1 = 1$	(A,a)	(C,b)
$s_1 = 0$	(B,c)	(D,d)

TAB. 6.3 – Matrix form of the two-player game : entries indicate the payoff to each player.

It is worth noting that this replication game is neither a "Game of Chicken" nor a "Hawks and Doves Game", two well-known versions of anti-coordination games. Considering the generic payoff in Tab. 6.3, the conditions for an anti-coordination game expressed by the payoff matrix are the following :

$$B > A$$
$$C > D$$
$$b > a$$
$$c > d$$

In our case, we clearly have $C > D$ and $c > d$ since $\frac{1}{k_1} > 0$. $B > A$ and $b > a$ if we have $\frac{1}{k_2} > \frac{1}{2k_1}$. We have showed in the above example that this inequality holds in realistic settings.

The Game of Chicken is usually illustrated by the scenario where two drivers are moving towards each other on a narrow road. The first driver who decides to swerve will lose his face. But if nobody swerves, an collision occurs. In a Game of Chicken, this condition must hold $A > C$, which cannot be satisfied in our case. So our game is not a Game of Chicken.

The Hawks and Doves Game is similar to Game of Chicken except that it does not requires $A > C$. But the following inequality must hold :

$$C > D > B > A$$

$$c > d > b > a$$

Clearly, we cannot have $d > b$ or $D > B$ which implies that $\frac{1}{k_2} < 0$ in our setting, hence our game is not a Hawks and Doves Game.

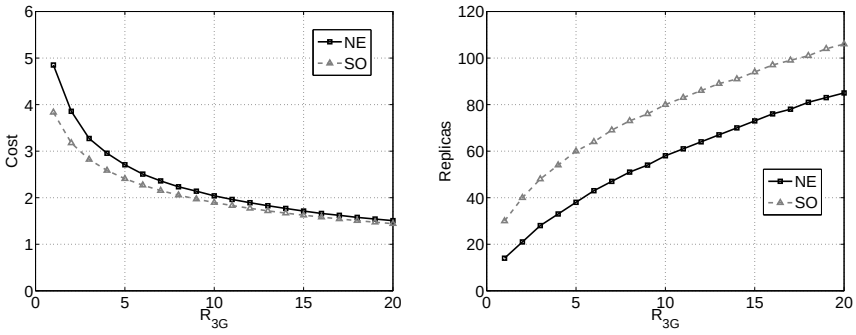
6.4 The n-player game

The results above can be extended to an n-player setting, which will be treated in detail in an extended version of this work. Our current research aims at putting into practice our theoretic findings, following two complementary directions. On the one hand, we note that, in the n-player replication game, a player can compute its best response to other players' strategies if it is aware of the current number x of replicas in the network. If player i is a replica node, the cost to play 1 (i.e. maintaining replica role) is $C(1) = xk_1$ and the cost to play 0 is $C(0) = \frac{(N-(x-1))k_2}{\sqrt{x-1}}$. If player i is currently playing 0, the cost to play 1 is $C(1) = (x+1)k_1$ and the cost to play 0 is $C(0) = \frac{(N-x)k_2}{\sqrt{x}}$. The steady x can be reached when no player has incentive to change its strategy :

$$xk_1 = \frac{(N-(x-1))k_2}{\sqrt{x-1}}$$
$$(x+1)k_1 = \frac{(N-x)k_2}{\sqrt{x}}$$

Hence, if every node i can not benefit from changing strategy according to best response to current x, we can find an equilibrium. We conduct a numerical analysis with several setting of 3G and device-to-device bit rate. For the equilibrium, we study the basic game : each node i chooses to download or to access content via device-to-device network to minimize its own cost based on the strategy of other nodes. We start with a random set of replica nodes and let nodes change their replication strategy as stated in Alg.6.1. It is also interesting to study the basic game in an asynchronous condition, e.g. in each iteration each node does not know whether other nodes select to switch their strategies, but we let this game for future work. The obtained number of replicas x however is not optimal while considering the social optimum $C(x) = x^2 k_1 + \frac{(N-x)^2}{\sqrt{x}} k_2$.

Fig. 6.6 shows the difference between the Nash equilibrium and the social optimum given by a static network of 320 nodes. Results show that when 3G bit rate is low, the cost for social optimum is much lower than the equilibrium which gives space to cooperation mechanisms to reduce the cost. This is because users prefer to use the device-to-device access and tend to replicate less than the optimal number. In our numerical results, we observe that until a very large 3G bit rate ($\approx$ 70Tbps[3]), the number of replicas for both cases converges with a full replication scenario. This bit rate is unrealistic. Therefore the full replication would not happen in real life, our anti-coordination game hold in current network configuration.



(a) (b)

FIG. 6.6 – Nash Equilibrium and Social Optimum with varying 3G bit rate R_{3G}=1-20Mbps.

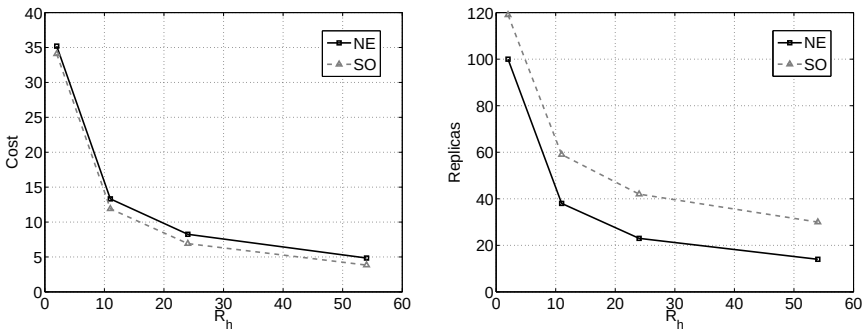
Fig. 6.7 shows the difference between the Nash equilibrium and the social optimum given by a network of 320 nodes when the device-to-device bit rate is varying. It is intuitive that x decreases when R_h increases, but the gap between the cost of the equilibrium and the social optimum always exists, hence there is a need of improvement in this case. In Fig. 6.8, we see that with a unrealistic high wireless bit rate R_h=220Tbps[3], the number of replicas for NE and SO converges to 1.It is out of the scope of this work to study the price of anarchy, as we tend to focus only on the replication factor in realistic settings.

Since in practice global knowledge cannot be assumed, we are investigating how far from efficiency our system settles when nodes compute an estimate $\hat{x}$ of the current number of replicas in the network. Such an estimate can be obtained either through random sampling techniques based on gossiping, or by exploiting *local measurements* of the number of queries received by each node storing the content. An open question is how sensitive

[3]This rate is not realistic and is used for illustrative purpose

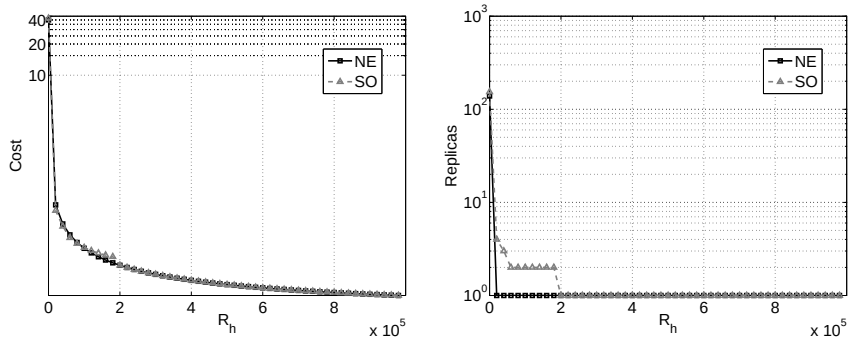
Algorithm 6.1 IteratedBestResponse

$K \leftarrow$ number of iterations
for $k = 1$ to K **do**
  for $i = 1$ to N **do**
    $x \leftarrow$ number of replicas
    if i is a replica **then**
      $C(1) \leftarrow xk_1$
      if $x > 1$ **then**
        $C(0) \leftarrow \frac{(N-x-1)k_2}{\sqrt{x-1}}$
      else
        $C(0) \leftarrow \infty$
      end if
    else
      $C(1) \leftarrow (x+1)k_1$
      $C(0) \leftarrow \frac{(N-x)k_2}{\sqrt{x}}$
    end if
    if $C(1) \geq C(0)$ **then**
      if i is a replica **then**
        i changes its strategy
        $x \leftarrow x - 1$
      end if
    else
      if i is not a replica **then**
        i changes its strategy
        $x \leftarrow x + 1$
      end if
    end if
  end for
end for



(a) (b)

Fig. 6.7 – Nash Equilibrium and Social Optimum with device-to-device bit rate R_h=2-54Mbps .



(a) (b)

FIG. 6.8 – Nash Equilibrium and Social Optimum with device-to-device bit rate R_h up to 1000Tbps.

to estimation errors the achieved equilibrium is. On the other hand, we observe that an external randomization device can help in improving efficiency, but correlated equilibrium is impractical when players' actions are not simultaneous, i.e., in a *asynchronous setting*. To address this issue, we allow communication between players through *signalling*. Simply stated, *signalling* replaces the external randomization device cited above and is used by a player to notify its strategy to others. The use of signalling however implies to take into account neighboring relations among players, as dictated by the underlying communication graph defined by the network topology.

6.5 Contribution

We proposed a novel model for the replication problem in a heterogeneous network under a "flash-crowd" scenario. We provided the expression for the social cost and defined a two-player game to obtain insights into the design of efficient replication strategies. The results showed that our problem can be casted as an "anti coordination" game in which user can increase their payoff while choosing the opposite strategy of other user. We conducted a numerical analysis of number of replicas with different 3G and device-to-device bit rate and showed the need of communication to improve efficiency. Based on the theoretical findings, our future work will focus on the design of strategies to be implemented in a practical network setting.

6.6 Relevant publication

Michiardi, Pietro ; Chiasserini, Carla-Fabiana ; Casetti, Claudio ; La, Chi Anh ; Fiore, Marco, *On a selfish caching game*, In proceedings of ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, PODC 2009

Conclusion and perspectives

Conclusion

With the advance in new wireless technology, mobile devices have been widely used in human daily life as a multi-functional equipment for entertainment and information purpose. Network applications need data as an input to process and provide information to users. It was reported that data traffic by mobile devices fetching content from the Internet is already overloaded mobile operators' backbone networks. Similar to Internet, mobile users are now coping with the congestion at network gateways. Distributing content to mobile users in a efficient way with low latency and without congestion at gateway is a challenging problem considering the dynamic nature of human mobility and behaviors. In this thesis, we addressed the problem of content distribution in heterogeneous mobile networks. In such network conditions, cellular networks and device-to-device communication can complement each other. The use of device-to-device communication may provide more aggregate resources to the system, but content availability is low and thus more latency in content access. In contrast, cellular networks is subjected to scalability and congestion issues. If the content is highly popular, device-to-device communication can eliminate the bottleneck at cellular gateway by offloading partially the distribution to mobile users using P2P techniques. Content replication in this context has been proved as a good solution to enhance network performance and scalability. However, the decision procedure to decide how many replica to be stored and at which location is not trivial in mobile networks due to the following reasons :

- The network topology in this case is supposed to change rapidly due to mobility and nodes can not rely on any centralized infrastructure to have a global view of the network.
- Wireless devices usually have many resource constraints on them hence we need a replication mechanism that helps load balancing. Moreover, users can behave selfishly when deciding to replicate the content.

By identifying all these issues, we first described the problem of content replication in mobile networks. We studied the state of the art of realistic mobility models to come up with a good problem definition and to identify the models we can use to evaluate mobile network performance. We then casted our problem as a facility location problem, in

particular this is a capacitated variant of facility location problem. This problem helped us to design distributed mechanism that approximates well optimal solutions to our objective metrics : the latency and load balancing. We also considered the problem of ressource constraints in mobile network and our mechanism aimed to distribute the burden of content replication while maintaining the load balancing among nodes by P2P cache-and-forward schemes. Finally we analyzed the subsequent scenario when users behave selfishly in content replication.

The contributions of this thesis are the following :

- We made a survey on mobility models and traces that are appropriate to use in simulation mobile network applications, particularly in our content replication context. We also conducted a mobility trace measurement and analysis in a network virtual environment. The results reveal that human behaviors pose a real problem on mobile network scalability as people usually concentrate around points of interest.
- We introduced cache-and forward mechanisms that help mobile users to share the burden in content distribution. The results shows good performance in terms of load balancing.
- We casted the problem of replication in mobile networks as a capacitated facility location problem. We hence designed a distributed and low overhead mechanisms to approximate the optimal solution that reduces content retrieving latency and avoids congestion at mobile gateways.
- We defined and studied a new model for the caching problem in heterogeneous wireless networks under a flash-crowd scenario considering the cost to access content using different wireless technologies. Using non-cooperative game theory, we casted the caching problem as an anti-coordination game. Based the theoretical findings, we focus on the replication factor in practical network settings and pointed out the need of cooperation to enhance content distribution performance.

Perspectives

Our work presented a lightweight and distributed mechanism to replicate content in heterogeneous mobile networks. An exploration of the parameters we used is required to evaluate the performance, especially the case when users have different budgets for content replication. A performance analysis of our replication scheme using different wireless technique (Bluetooth, 802.11...) would certainly bring more detailed insights for realistic application deployment. The replication factor in our current mechanism depends on the dedicated budget chosen by mobile nodes while it is interesting to study whether users can select a flexible budget that adapts to network conditions.

To relax the assumption of a cooperative setting, we have analyzed selfish replication with tools akin to game theory. From the theoretical findings, our future work will focus on the design of strategies to be implemented in a practical network settings. The design of incentive mechanisms for such system can be also a topic to study in order to build a real application deployable at mobile devices. Moreover, enhanced mechanism to protect the system from free riders is an important topic in this research direction.

The security mechanisms to protect user privacy and avoid any possible exploitation and attack are also critical for these kinds of application. Encrypted data and authentication should be introduced to efficiently ensure the confidentiality and intergrity of contents and protect users from information manipulation for malicious purposes.

Glossary

-0-9-

3G : Third Generation, a set of standards for cellular networks which allow simultaneous use of speech and data services.
7DS : Seven Degrees of Separation, a technique that allows wireless users to exchange data in a local disconnected network.
802.11 : A set of standards carrying out wireless local area network computer communications.

-B-

Bluetooth : Open wireless technology standard for exchanging data using short wavelength radio transmissions.

-C-

CCDF : Complementary Cumulative Distribution Function.
CDF : Cumulative Distribution Function.
CT : Contact Time : time elapsed when two mobile nodes are in contact range.
CFL : Capacitated Facility Location problems.

-D-

DHT : Distributed Hash Table.
DTN : Delay Tolerant Networks or Disruption Tolerant Networks.
DNS : Domain Name Server.

-E-

ERR : Expanding Ring Replication.

-F-

FL : Facility Location problems.

-G-

GPS : Global Positioning System.

-I-

ICT : Inter-contact Time : time elapsed from when two mobile nodes are not longer in contact range until the first moment they are in contact again.

-L-

LP : Linear Programming technique.
LRU : Least Recently Used.

-M-

MANET : Mobile Ad Hoc Networks.
MLE : Maximum Likelihood Estimation.

-N-

NE : Nash Equilibrium
NVE : Network Virtual Environment.
ns-2 : Network Simulator version 2.

-S-

SL : Second Life Virtual Environment.
SLAW : Self Similar Least Action Walk : a mobility model consists of most of human mobility patterns reported in the literature.

SO : Social Optimum.
SWIM : Shared Wireless Infostation Model.

-P-

P2P : Peer-to-Peer.
PAN : Probabilistic quorum system for ad hoc networks.
PDF : Probability Distribution Function.

-R-

RDD : Random Direction Dissemination : a mechanism using to hand-over content replica to the node that is closest to a random location.
RWD : Random Walk Dissemination : a mechanism using a list of neighbors to randomly hand-over content replica.
RWP : Random Waypoint Mobility Model.

-U-

UFL : Uncapacitated Facility Location problems.

-T-

TTL : Time-to-Live.

Bibliography

[1] Akamai : http ://www.akamai.com.

[2] Libsecondlife : www.libsecondlife.org.

[3] Second life : http ://www.secondlife.com.

[4] Customers angered as iphones overload at&t, 2009. `http://www.nytimes.com/2009/09/03/technology/companies/03att.html?_r=1`.

[5] iphone users eating up at&t's network, 2009. `http://venturebeat.com/2009/05/11/iphone-users-eating-up-atts-network/`.

[6] *NIST/SEMATECH e-Handbook of Statistical Methods*. 2009. `www.itl.nist.gov/div898/handbook/`.

[7] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. *Local Search Heuristic for k-median and Facility Location Problems*. ACM STOC, 2001.

[8] D. Austin, W.D. Bowen, and J.I. McMillan. *Intraspecific variation in movement patterns : modeling individual behaviour in a large marine predator*. Oikos, 105(1) :15-30, 2004.

[9] I. Baev and R. Rajaraman. *Approximation Algorithms for Data Placement in Arbitrary Networks*. ACM-SIAM SODA, 2001.

[10] F. Bai and A. Helmy. *A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks*. Book Chapter in the book on Wireless Ad Hoc and Sensor Networks, Kluwer Academic Publishers, 2004.

[11] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. *Characterizing user behavior and network performance in a public wireless LAN*. ACM SIGMETRICS, 2002.

[12] M. Balazinska and P. Castro. *Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network*. ACM MOBISYS, 2003.

[13] S. Bandyopadhyay, E.J. Coyle, and T. Falck. *Stochastic Properties of Mobility Models in Mobile Ad Hoc Networks*. IEEE Transactions on Mobile Computing, vol.6, no.11, pp.1218–1229, Nov., 2007.

[14] J. Boleng. *Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks*. Local and Metropolitan Area Networks Workshop (LANMAN), 2001.

[15] J.-Y. Le Boudec. *Understanding the simulation of mobility models with Palm calculus.* Perform. Eval. 64(2) : 126-147, 2007.

[16] J.-Y. Le Boudec and M. Vojnovic. *Perfect Simulation and Stationarity of a Class of Mobility Models.* IEEE Infocom, 2005.

[17] J.-Y. Le Boudec and M. Vojnovic. *The Random Trip Model : Stability, Stationary Regime, and Perfect Simulation.* ACM /IEEE Trans. on Networking, vol.14, no.6, pp.1153–1166, Dec., 2006.

[18] A. Boukerche. *Handbook of Algorithms for Wireless Networking and Mobile Computing.* CRC/Chapman Hall, 2005.

[19] A. Boukerche. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks.* Wiley & Sons, 2008.

[20] D. Braginsky and D. Estrin. *Rumor Routing Algorithm For Sensor Networks.* First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, Sept, 2002.

[21] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. *Advances in Network Simulation (ns).* IEEE Computer, vol. 33, No. 5, pp. 59-67, May, 2000.

[22] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva. *A performance comparison of multi-hop wireless ad hoc network routing protocols.* ACM MOBICOM, 1998.

[23] D. Brockmann, L. Hufnagel, and T. Geisel. *The scaling laws of human travel.* Nature 439 :462-465, January, 2006.

[24] T. Camp, J. Boleng, and V. Davies. *A Survey of Mobility Models for Ad Hoc Network Research.* Wireless Communications & Mobile Computing (WCMC) : Special issue on Mobile Ad Hoc Networking Research, Trends and Applications, Vol. 2, No. 5., pp. 483-502., 2002.

[25] C. Casetti, C.-F. Chiasserini, M. Fiore, C.-A. La, and P. Michiardi. *P2P Cache-and-Forward Mechanisms for Mobile Ad Hoc Networks.* IEEE ISCC, Sousse, Tunisia, 2009.

[26] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. *Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms.* INFOCOM, 2006.

[27] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, J. Scott, and R. Gass. *Impact of human mobility on opportunistic forwarding algorithms.* IEEE Transactions on Mobile Computing, 2007.

[28] A. Chaintreau, A. Mtibaa, L. Massoulie, and Christophe Diot. *The Diameter of Opportunistic Mobile Networks.* Proc. of CoNEXT, 2007.

[29] K. Chen and K. Nahrstedt. *An integrated Data Lookup and Replication Scheme in Mobile Ad Hoc Networks.* SPIE ITCom, Aug. , 2001.

[30] D.R. Choffnes and F.E. Bustamante. *An integrated mobility and traffic model for vehicular wireless networks.* ACM VANET, 2005.

[31] F. Chudak and D. Shmoys. *Improved approximation algorithms for capacitated facility location problem.* ACM SODA, 1999.

[32] B.G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C.H. Papadimitriou, and J. Kubiatowicz. *Selfish caching in distributed systems : a game-theoretic analysis.* ACM PODC, 2004.

[33] A. Clauset, C. R. Shalizi, and M. E. J. Newman. *Power-law distributions in empirical data*. Submitted to SIAM Reviews, 2007.

[34] E. Cohen and S. Shenker. *Replication strategies in unstructured peer-to-peer networks*. SIGCOMM, 2002.

[35] P.L.M. de Maupertuis. *Accord des differentes lois de la nature qui avaient jusqu'ici paru incompatibles*. Memoires de l'Academie des Sciences, page 417, 1744.

[36] A. Derhab and N. Badache. *Data Replication Protocols for Mobile Ad-hoc Networks : A Survey and Taxonomy*. IEEE Communications Surveys & Tutorials, vol. 11, no. 2, pp. 33–51, 2009.

[37] N. Eagle and A. Pentland. *Reality mining : Sensing complex social systems*. Journal of Personal and Ubiquitous Computing, 2005.

[38] A. Einstein. *Investigations on the Theory of the Brownian Motion*. D. Cowper (1926, reprinted Einstein, Collected Papers, vol.2, pp., pp.206-22, 1926.

[39] F. Ekman, A. Keraenen, J. Karvo, and J. Ott. *Working Day Movement Model*. ACM SIGMOBILE workshop on Mobility models, 2008.

[40] M. Fiore, J. Haerri, F. Filali, and C. Bonnet. *Vehicular Mobility Simulation for VANETs*. IEEE ANSS, 2007.

[41] M. Fiore, F. Mininni, C. Casetti, and C.-F. Chiasserini. *To Cache or Not To Cache ?* IEEE Infocom, Rio de Janeiro, Brazil, Apr. , 2009.

[42] Christian Frank and Kay Romer. *Distributed Facility Location Algorithms for Flexible Configuration of Wireless Sensor Networks*. DCOSS, p124-141, 2007.

[43] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi. *Understanding individual human mobility patterns*. Nature, 453(7196) :779-782, 2008.

[44] P. Gupta and P.R. Kumar. *The Capacity of Wireless Networks*. IEEE Transactions on Information Theory, 2000.

[45] T. Hara. *Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility*. IEEE INFOCOM, 2001.

[46] T. Hara, Y.-H. Loh, and S. Nishio. *Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks*. DEXA, 2003.

[47] M. Hauspie, A. Panier, and D. Simplot-Ryl. *Localized Probabilistic and Dominating Set Based Algorithm for Efficient Information Dissemination in Ad Hoc Networks*. IEEE MASS, Oct. , 2004.

[48] H. Hayashi, T. Hara, and S. Nishio. *On Updated Data Dissemination Exploiting an Epidemic Model in Ad Hoc Networks*. BioADIT, 2006.

[49] T. Henderson, D. Kotz, and I. Abyzov. *The changing usage of a mature campus wide wireless network*. ACM MOBICOM, 2004.

[50] D.S. Hochbaum. *Heuristics for the fixed cost median problem*. Mathematica Programming, 1982.

[51] J. Holliday, R. Steinke, D. Agrawal, and A.E. Abbadi. *Epidemic Algorithms for Replicated Databases*. IEEE Transactions on Knowledge and Data Engineering, vol.15, no.3, pp.1–21, 2003.

[52] X. Hong, M. Gerla, G. Pei, and C. Chiang. *A group mobility model for ad hoc wireless networks*. ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), 1999.

[53] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. *Modeling time-variant user mobility in wireless mobile networks.* IEEE INFOCOM, 2007.

[54] Y.-C. Hu and D.B. Johnson. *Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks.* ACM MOBICOM, 2000.

[55] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. *Pockets Switched Networks and Human Mobility in Conference Environments.* ACM SIGCOMM CHANTS, 2005.

[56] R. Hutchins and E.W. Zegura. *Measurements from a campus wireless network.* IEEE International Conference on Communications (ICC), 2002.

[57] S. Ioannidis, L. Massoulie, and A. Chaintreau. *Distributed caching over heterogeneous mobile networks.* SIGMETRICS, 2010.

[58] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. *Greedy facility location algorithms analyzed using dual fitting factor revealing LP.* Journal of ACM, 2003.

[59] K. Jain, M. Mahdian, and A. Saberi. *A new greedy approach for facility location problems.* ACM STOC, 2002.

[60] K. Jain and V.V. Vazirani. *Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation.* J. ACM, vol.48, no.2, 2001.

[61] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri. *Towards Realistic Mobility Models for Mobile Ad hoc Networks.* Ninth Annual International Conference on Mobile Computing and Networking (MobiCom), 2003.

[62] A.P. Jardosh, E.M. Belding-Royer, K.C. Almeroth, and S. Suri. *Real-world Environment Models For Mobile Network Evaluation.* IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad hoc Networks, March, 2005.

[63] B. Jiang, J.J. Yin, and S.J. Zhao. *Characterizing the human mobility pattern in a large street network.* Phys. Rev. E 80, 021136, 2009.

[64] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic. *Power law and exponential decay of inter contact times between mobile devices.* ACM MOBICOM, 2007.

[65] O. Kariv and S. Hakimi. *An Algorithmic Approach to Network Location Problems, Part II : p-medians.* SIAM Journal on Applied Mathematics, vol. 37, pp. 539–560, 1979.

[66] A. Khelil, C. Becker, J. Tian, and K. Rothermel. *An Epidemic Model for Information Diffusion in MANETs.* ACM MSWiM, 2002.

[67] M. Kim, D. Kotz, and S. Kim. *Extracting a mobility model from real user traces.* IEEE INFOCOM, 2006.

[68] M. Korupolu, C. Plaxton, and R. Rajaraman. *Analysis of a local search heuristic for facility location problems.* ACM SODA, 1998.

[69] D. Kotz and T. Henderson. *CRAWDAD : A Community Resource for Archiving Wireless Data at Dartmouth.* IEEE Pervasive Computing, 4(4) :12-14, October-December, 2005.

[70] J. Krumm and E. Horvitz. *The Microsoft Multiperson Location Survey.* Microsoft Research, MSR-TR-2005-13, 2005.

[71] A. Kuehn and M.J. Hamburger. *A heuristic program for locating warehouses.* Management Sciences, 1963.

[72] C.-A. La and P. Michiardi. *Characterizing User Mobility in Second Life.* ACM SIGCOMM WOSN, 2008.

[73] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros. *Distributed Placement of Service Facilities in Large-Scale Networks.* IEEE Infocom, 2007.

[74] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong. *SLAW : A Mobility Model for Human Walks.* IEEE INFOCOM, 2009.

[75] B. Liang and Z. Haas. *Predictive distance-based mobility management for PCS networks.* IEEE INFOCOM, 1999.

[76] J. Lin and J. Vitter. ϵ *approximation with minimum packing constraint violation.* ACM STOC, 1992.

[77] C. Lindemann and O.P. Waldhorst. *Modeling Epidemic Information Dissemination on Mobile Devices with Finite Buffers.* SIGMETRICS, 2005.

[78] A. Lindgren and P. Hui. *The Quest for a Killer App for Opportunistic and Delay Tolerant Networks.* ACM MobiCom Workshop on Challenged Networks (Chants, 2009.

[79] J. Luo, J.-P. Hubaux, and P. T. Eugster. *PAN : Providing Reliable Storage in Mobile Ad Hoc Networks with Probabilistic Quorum Systems.* ACM MobiHoc, 2003.

[80] K. Maeda, K. Sato, K. Konishi, A. Yamasaki, A. Uchiyama, H. Yamaguchi, K. Yasumotoy, and T. Higashino. *Getting urban pedestrian flow from simple observation : Realistic mobility generation in wireless network simulation.* ACM MSWiM, 2005.

[81] B.B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman, 1982.

[82] R. Mangharam, D.S. Weller, D.D. Stancil, R. Rajkumar, and J.S. Parikh. *GrooveSim : a topography-accurate simulator for geographic routing in vehicular networks.* ACM VANET, 2005.

[83] M. McNett and G. M. Voelker. *Access and Mobility of Wireless PDA User.* Mobile Computing Communications Review, 9(2) :40-55, April, 2005.

[84] P. Michiardi, C.-F. Chiasserini, C. Casetti, C.-A. La, and M. Fiore. *On a Selfish Caching Game.* ACM PODC, Calgary, Canada, Aug. , 2009.

[85] P. Mirchandani and R. Francis. *Discrete Location Theory.* Wiley, John and Sons, 1990.

[86] T. Moscibroda and R. Wattenhofer. *Facility Location : Distributed Approximation.* ACM PODC, 2005.

[87] M. Musolesi and C. Mascolo. *A Community Based Mobility Model for Ad Hoc Network Research.* ACM REALMAN, 2006.

[88] M. Musolesi and C. Mascolo. *Mobility Models for Systems Evaluation.* Middleware for Network Eccentric and Mobile Applications, 2009.

[89] P. Nain, D. Towsley, B. Liu, and Z. Liu. *Properties of Random Direction Model.* IEEE INFOCOM, 2005.

[90] V. Naumov, R. Baumann, and T. Gross. *An Evaluation of Inter-vehicle Ad Hoc networks Based on Realistic Vehicular Traces.* ACM MobiHoc, 2006.

[91] W. Navidi and T. Camp. *Stationary distributions for the random waypoint mobility model.* IEEE Transactions on Mobile Computing, 3(1) :99-108, 2004.

[92] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu. *The Broadcast Storm Problem in a Mobile Ad Hoc Network.* ACM Mobicom, 1999.

[93] P. Nuggehalli, V. Srinivasan, C.-F. Chiasserini, and R. R. Rao. *Efficient Cache Placement in Multi-hop Wireless Networks.* IEEE/ACM Transactions on Networking, vol.14, no.5, pp.1045–1055, Oct., 2006.

[94] S. Prabh and T. Abdelzaher. *Energy-Conserving Data Cache Placement in Sensor Networks.* ACM Transactions on Sensor Networks, vol.2, no.1, pp.178–203, 2005.

[95] R. Ravi and A. Sinha. *Multicommodity facility location.* ACM-SIAM symposium on Discrete algorithms, 2004.

[96] J. Reich and A. Chaintreau. *The age of impatience : Optimal Replication Schemes for Opportunistic Networks.* ACM CoNEXT, 2009.

[97] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. *On the Levy-walk Nature of Human Mobility.* IEEE INFOCOM, 2008.

[98] E. Royer, P.M. Melliar-Smith, and L. Moser. *An analysis of the optimum node density for ad hoc mobile networks.* IEEE International Conference on Communications (ICC), 2001.

[99] N. Sadagopan, B. Krishnamachari, and A. Helmy. *Active Query Forwarding in Sensor Networks.* Ad Hoc Networks Journal, Elsevier, Vol.3, no.1, pp., Janvol.3, no.1, pp., Jan, 2005.

[100] A.K. Saha and D.B. Johnson. *Modeling mobility for vehicular ad-hoc networks.* ACM VANET, 2004.

[101] M.K. Sbai, E. Salhi, and C Barakat. *P2P Content sharing in spontaneous multi-hop wireless networks.* COMSNETS, 2010.

[102] G. Sharma and R. R. Mazumdar. *Scaling laws for capacity and delay in wireless ad hoc networks with random mobility.* IEEE International Conference on Communications (ICC), 2004.

[103] M. Shinohara, H. Hayashi, T. Hara, and S. Nishio. *Replica Allocation Considering Power Consumption in Mobile Ad Hoc Networks.* IEEE PERCOMW, 2006.

[104] D. Shmoys, E. Tardos, and K. Aardal. *Approximation algorithms for facility location problems.* ACM STOC, 1997.

[105] T. Small and Z. Haas. *The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm.* ACM MobiHoc, 2003.

[106] V. Srinivasan, M. Motani, and W.T. Ooi. *Analysis and Implications of Student Contact Patterns Derived from Campus Schedules.* ACM MOBICOM, 2006.

[107] M. Tamori, S. Ishihara, T. Watanabe, and T. Mizuno. *A replica distribution method with consideration of the positions of mobile hosts on wireless ad hoc networks.* ICDCS Workshop, 2002.

[108] W.L. Tan, F. Lam, and W.C. Lau. *An Empirical Study on the Capacity and Performance of 3G Networks.* IEEE Transactions on Mobile Computing, 2007.

[109] B. Tang, H. Gupta, and S. Das. *Benefit-based Data Caching in Ad Hoc Networks.* Trans. on Mobile Computing, vol.7, no.3, pp.208–217, Mar., 2008.

[110] D. Tang and M. Baker. *Analysis of a metropolitan-area wireless network.* ACM MOBICOM, 1999.

[111] D. Tang and M. Baker. *Analysis of a local-area wireless network.* ACM MOBICOM, 2000.

[112] J. B. Tchakarov and N. H. Vaidya. *Efficient Content Location in Wireless Ad Hoc Networks.* Mobile Data Management, 2004.

[113] S. Tewari and L. Kleinrock. *Proportional replication in peer-to-peer networks.* INFOCOM, 2006.

[114] V. Thanedar, K. C. Almeroth, and E. M. Belding-Royer. *A Lightweight Content Replication Scheme for Mobile Ad Hoc Environments.* Network, 2004.

[115] J. Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel. *Graph-based Mobility Model for Mobile Ad Hoc Network Simulation.* 35th Annual Simulation Symposium, in cooperation with the IEEE Computer Society and ACM, 2002.

[116] V. Tolety. *Load reduction in ad hoc networks using mobile servers.* MSc thesis, Colorado School of Mine, 1999.

[117] M. Trott. *The Mathematica Guidebooks Additional Material : Average Distance Distribution.* 2004.

[118] C. Tuduce and T. Gross. *A Mobility Model Based on WLAN Traces and its Validation.* IEEE INFOCOM, 2005.

[119] G.M. Viswanathan, V. Afanasyev, S.V. Buldyrev, E. J. Murphy, P.A. Prince, and H.E. Stanley. *Levy flight search patterns of wandering albatrosses.* Nature, (381) :413-415, 1996.

[120] G.M. Viswanathan, S.V. Buldyrev, S. Havlin, M.G.E. da Luz, E.P. Raposo, and H.E. Stanley. *Optimizing the success of random searches.* Nature, (401) :911-914, 1999.

[121] K. Wang and B. Li. *Efficient and guaranteed service coverage in partitionablemobile ad hoc networks.* IEEE INFOCOM, 2002.

[122] D. J. Watts and Steven Strogatz. *Collective dynamics of 'small-world' networks.* Nature, 1998.

[123] E.W. Weisstein. *The CRC Concise Encyclopedia of Mathematics.* CRC Press, 1998.

[124] H. Wu, M. Palekar, R. Fujimoto, R. Guensler, M. Hunter, J.S. Lee, and J.H. Ko. *An empirical study of short range communications for vehicles.* VANET '05 : Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, New York, NY, USA, 2005.

[125] L. Yin and G. Cao. *Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks.* IEEE SRDS, 2004.

[126] L. Yin and G. Cao. *Supporting Cooperative Caching in Ad Hoc Networks.* IEEE Trans. on Mobile Computing, vol. 5, no. 1, pp. 77–89, Jan. , 2006.

[127] J. Yoon, M. Liu, and B. Noble. *Random Waypoint Considered Harmful.* IEEE INFOCOM, 2003.

[128] J. Yoon, M. Liu, and B. Noble. *Sound mobility models.* ACM MOBICOM, 2003.

[129] H. Yu, H. Hassanein, and P. Martin. *Cluster-based Replication for Large-scale Mobile Ad-hoc Networks.* International Conf. on Wireless Networks, Communications and Mobile Computing, 2005.

[130] X. Zhang, J. Kurose, B.N. Levine, D. Towsley, and H. Zhang. *Study of a bus-based disruption-tolerant network : mobility modeling and impact on routing.* ACM MOBICOM, 2007.

[131] J. Zheng, J.S. Su, and X.C. Lu. *A clustering-based data replication algorithm in mobile ad hoc networks for improving data availability.* IPSA, 2004.