

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
جامعة عمار ثلجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT



FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE
Mémoire de MASTER

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Réseaux, Systèmes et Applications Réparties

Par :
GUERGAB Fatima Zohra Djahida

Thème :

Mise en œuvre et évaluation
de la mise en cache coopérative "Mobile Edge Computing"

Les membres du jury :

| | | |
|-----------|-------|------------------|
| Encadreur | : Mr | Noureddine CHAIB |
| Président | : Mme | Benarous LEILA |
| Examineur | : Mr | Bendouma TAHAR |

Année universitaire 2021/2022

DÉDICACES

Je dédie ce travail spécialement à mes chers parents, mes frères et mes sœurs, sans que j'oublie mes amis de mon enfance en particulier, à toute ma famille petits et grands ainsi que mes amis intimes qui m'accompagnent long des parcours universitaire.

All Love to you 'Djahida'

REMERCIEMENT

Tout d'abord, je remercie Allah, de m'aider et de m'avoir permis d'accomplir ce modeste travail.

Et après je remercie mon encadreur qui m'a beaucoup aidé par ses conseils et m'a accompagné à chaque étape de ce travail, Merci beaucoup, et je remercie les membres du jury, Merci beaucoup aussi.

Je remercie mes parents et mes amis, en particulier, mon amie Amina, pour leur soutien moral et leur soutien dans les moments les plus difficiles de ma vie.

Merci à tous mes estimés professeurs, du primaire au professeur d'université, qu'Allah vous récompense bien en notre nom.

Finalement, je remercie ma famille et tous ceux qui contribuaient de près ou de loin à l'aboutissement de ce travail.

| | | |
|---|--|-----------|
| Dédicaces | | i |
| Remerciement | | i |
| Résumé | | 1 |
| Abstract | | 1 |
| 1 Introduction générale | | 2 |
| 1.1 Problématique | | 4 |
| 1.2 Objectifs de mémoire | | 4 |
| 1.3 Organisation de mémoire | | 5 |
| 2 Mobile-Edge Computing | | 6 |
| 2.1 Introduction | | 6 |
| 2.2 Mobile-Cloud Computing | | 7 |
| 2.3 Cloudlets | | 8 |
| 2.4 Mobile-Edge Computing | | 10 |
| 2.5 Mise en cache coopérative mobile | | 12 |
| 2.6 Conclusion | | 14 |
| 3 Système de mise en cache coopérative Mobile-Edge Computing | | 15 |
| 3.1 Introduction | | 15 |
| 3.2 Aperçu | | 15 |
| 3.3 Cas d'utilisation d'une demande de contenu mobile | | 17 |
| 3.4 Conception du système de Co-Caching MEC | | 19 |

| | | |
|----------|--|-----------|
| 3.4.1 | Faible latence et expérience utilisateur riche | 20 |
| 3.4.2 | Cohérence | 20 |
| 3.4.3 | Évolutivité | 21 |
| 3.4.4 | Utilisation minimale de la bande passante | 21 |
| 3.5 | Les lignes de base | 23 |
| 3.5.1 | Nœud de balise | 23 |
| 3.5.2 | Probabilité | 24 |
| 3.5.3 | Sans relais | 24 |
| 3.6 | Conclusion | 24 |
| 4 | L'implémentation et Évaluation des performances | 25 |
| 4.1 | Introduction | 25 |
| 4.2 | Modélisation du système et formulation des problèmes | 25 |
| 4.3 | La technique proposée de mise en cache collaboratif distribué Edge | 29 |
| 4.3.1 | La phase de sélection de nœud avec énergie | 29 |
| 4.3.2 | Stratégie de mise en cache collaborative Edge | 30 |
| 4.4 | Configuration expérimentale | 30 |
| 4.5 | Résultats expérimentale | 31 |
| 4.6 | Conclusion | 34 |
| 5 | Conclusion générale | 35 |
| | Bibliographie | 36 |

TABLE DES FIGURES

| | | |
|-----|--|----|
| 2.1 | L'architecture de Cloudlet | 9 |
| 2.2 | Synthèse dynamique de VM cloudlet | 10 |
| 2.3 | Architecture de MEC | 12 |
| 2.4 | Réseau ad hoc de mise en cache coopérative mobile | 13 |
| 3.1 | Diagramme du processus de demande de contenu | 18 |
| 3.2 | Architecture du système de mise en cache coopérative Mobile-Edge Computing | 19 |
| 3.3 | Parties du système de cache coopératif MEC sur chaque serveur MEC | 20 |
| 3.4 | Évolutivité de la mise en cache coopérative MEC | 22 |
| 3.5 | Consommation de bande passante intra-réseau | 23 |
| 3.6 | Cas d'utilisation d'économie d'utilisation de la bande passante | 23 |
| 4.1 | Un exemple de système de mise en cache collaboratif en Edge | 26 |
| 4.2 | Le taux de succès | 31 |
| 4.3 | Le délai d'accès moyen | 32 |
| 4.4 | L'utilité de l'utilisateur | 33 |
| 4.5 | Disponibilité des nœuds | 33 |

LISTE DES TABLEAUX

4.1 Plateforme expérimentale 31

LISTE DES ABRÉVIATIONS

- API : Application Interface
- CP : Content Providers
- CSP : Communication Service Providers
- ETSI : European Telecommunications Standards Institute
- ISG : Industry Specification Group
- LTE : Long Term Evolution
- MCC : Mobile Cloud Computing
- MEC : Mobile-Edge Computing
- MEC Co-Caching : CO-Caching Mobile-Edge Computing Cooperative Caching
- MNO : Mobile Network Operators
- QoS : Quality of Service
- RAN : Radio Access Network
- RNC : Radio Network Controller
- VM : Virtual Machine

Le paradigme de *Mobile Edge Computing* (MEC) repose sur le placement des ressources de calcul et de stockage aux « extrémités » du réseau à proximité des utilisateurs finaux. Le terme « edge » désigne n'importe quel type de station de base du réseau. Les motivations pour l'adoption de ce nouveau concept sont principalement la réduction de la charge au cœur du réseau et la diminution de la latence grâce à la proximité des ressources afin d'améliorer l'expérience utilisateur. La plupart des études existantes se sont focalisées sur l'allocation de cache, la popularité du contenu ou encore la manière de concevoir le cache. Dans ce mémoire, nous présentons une nouvelle stratégie de mise en cache basée sur la coopérative, dans lequel, nous avons proposé un scénario de demande dans le système de la mise en cache collaborative d'Edge . Ce système a été construit en deux phases : la première est de sélectionner un noeud avec énergie, et la deuxième phase est la stratégie de mise en cache collaborative Edge qui donnait les indicateurs de performance essentiels pour analyser et évaluer la performance de notre proposition par rapport aux autres techniques existence, nous avons considéré les techniques suivantes : noeud de balise, probabilité et sans relais, ces techniques sont utilisées pour tester et analyser des indicateurs de performance importants tels que le taux d'accès, le délai d'accès moyen, etc. Les résultats expérimentaux montrent que notre proposition réduit la latence moyenne d'accès aux demandes des utilisateurs, améliorant ainsi l'utilité des utilisateurs, et surtout, il assure la disponibilité du service.

The *Mobile Edge Computing* (MEC) paradigm is based on placing compute and storage resources at the edge of the network near the end users. The term "edge" refers to any type of base station on the network. The motivations for adopting this new concept are mainly to reduce the load at the core of the network and to reduce latency by placing the resources close to the end users in order to improve the user experience. Most of the existing studies have focused on cache allocation, content popularity or how to design the cache. In this dissertation, we present a new cooperative-based caching strategy, in which, we proposed an application scenario in the Edge collaborative caching system. This system was built in two phases : the first is to select a node with energy, and the second phase is the Edge collaborative caching strategy that gave the essential performance indicators to analyze and evaluate the performance of our proposal compared to other techniques existence, we considered the following techniques : beacon node, probability and without relay, these techniques are used to test and analyze important performance indicators such as access rate, average access delay, etc. The experimental results show that our proposal reduces the average access latency to user requests, thus improving user utility, and most importantly, it ensures service availability.

يعتمد نموذج الحوسبة المتنقلة على وضع موارد الحوسبة والتخزين على حافة الشبكة بالقرب من المستخدمين النهائيين. يشير المصطلح "حافة" إلى أي نوع من أنواع المحطات الأساسية على الشبكة. تتمثل الدوافع وراء تبني هذا المفهوم الجديد بشكل أساسي في تقليل الحمل في قلب الشبكة وتقليل زمن الوصول عن طريق وضع الموارد بالقرب من المستخدمين النهائيين من أجل تحسين تجربة المستخدم. ركزت معظم الدراسات الحالية على تخصيص ذاكرة التخزين المؤقت أو شعبية المحتوى أو كيفية تصميم ذاكرة التخزين المؤقت. في هذه المذكرة ، نقدم استراتيجية تخزين مؤقت جديدة قائمة على التعاون ، حيث اقترحنا سيناريو تطبيق في نظام التخزين المؤقت التعاوني على الحافة. تم بناء هذا النظام على مرحلتين: الأولى هي تحديد عقدة مع الطاقة ، والمرحلة الثانية هي استراتيجية التخزين المؤقت التعاوني على الحافة التي أعطت مؤشرات الأداء الأساسية لتحليل وتقييم أداء اقتراحنا مقارنةً بالتقنيات الأخرى الموجودة ، نعتبر التقنيات التالية: عقدة المنارة ، الاحتمالية وبدون مرحل ، تُستخدم هذه التقنيات لاختبار وتحليل مؤشرات الأداء المهمة مثل معدل الوصول ومتوسط تأخير الوصول ، وما إلى ذلك. توضح النتائج التجريبية أن اقتراحنا يقلل متوسط زمن الوصول لطلبات المستخدمين ، وبالتالي تحسين أداة المستخدم ، والأهم من ذلك أنها تضمن توفر الخدمة.

CHAPITRE 1

INTRODUCTION GÉNÉRALE

La demande importante des utilisateurs d'avoir différents types de services sur leurs appareils mobiles et les problèmes liés à ces appareils comme la consommation d'énergie, les faibles ressources et la faible connectivité [1] à conduire à l'apparition du *Mobile Cloud Computing* (MCC) pour tenter de décharger les tâches d'application et également le stockage nécessaire sur un serveur cloud. MCC atténue certains défis des appareils mobiles d'une manière souhaitable. Par exemple, l'efficacité énergétique et la fiabilité sont atteintes par plusieurs solutions intelligentes, principalement par des méthodes de déchargement.

Malgré ces réalisations, des problèmes tels que la latence élevée, la disponibilité du service, *Quality of Service* (QoS) et le coût du service [2] doivent encore être résolus. Ces préoccupations découlent principalement de la croissance rapide du nombre d'utilisateurs mobiles et de leurs attentes à des services MCC. Les utilisateurs ont besoin de plus de disponibilité malgré le manque de connectivité des appareils mobiles et ils exigent une QoS plus élevée avec un coût de service moindre. De plus, la latence du réseau est un gros fardeau pour l'amélioration de la qualité de service et de l'expérience utilisateur lors de l'utilisation d'un cloud distant. Ces problèmes sont plus tangibles dans les applications qui offrent des services de cognition ou de réalité virtuelle qui exigent une faible latence [3].

Considérant ces problèmes, les chercheurs ont réalisé que l'utilisation de ressources et de services avec plus de localité est plus rentable avec une meilleure disponibilité, une connectivité plus rapide et latence minimale. Cela a conduit au concept de "Cloudlet" [3]. Un cloudlet est un ordinateur ou un groupe d'ordinateurs connectés à l'extrémité du réseau pour fournir un accès à faible latence aux ressources informatiques pour les appareils mobiles. La mission des cloudlets est d'alléger les contraintes de ressources des appareils mobiles et également

d'atteindre une meilleure latence du réseau. La reconnaissance vocale, le traitement du langage naturel, l'apprentissage automatique, la réalité augmentée et d'autres applications à forte intensité de calcul bénéficieraient le plus de l'approche cloudlet [3][4].

Réalisant les avantages de rapprocher les services et les ressources cloud, les opérateurs de *Radio Access Network* (RAN) ont commencé à étudier l'idée de cloudlets sur les stations de base RAN comme le tout premier saut à l'extrémité du réseau pour servir les utilisateurs mobiles. De sérieux efforts ont commencé à rendre cette idée pratique pour l'environnement des affaires. Cela a conduit à l'émergence du *Mobile-Edge Computing* (MEC).

Dans le briefing exécutif de *European Telecommunications Standards Institute* (ETSI) [5], a défini MEC comme suit : " *Mobile-Edge Computing* offre aux développeurs d'applications et aux *Content Providers* (CP) des capacités de cloud computing et un environnement de services IT à l'extrémité du réseau mobile. Cet environnement se caractérise par une latence faible ainsi qu'un accès en temps réel aux informations du réseau radio qui peuvent être exploitées par les applications."

Comme mentionné, la satisfaction des exigences de QoS des services émergents en temps réel, interactifs et multimédias a conduit aux paradigmes MEC, où l'accent est mis sur la fourniture d'une QoS élevée, à de faibles coûts opérationnels. En outre, la croissance rapide du nombre d'utilisateurs mobiles utilisant ces services entraîne une croissance importante du trafic réseau qui doit être prise en compte en fonction de la capacité limitée de la bande passante de liaison et du temps de réponse souhaitable des applications mobiles.

L'utilisation de la mise en cache coopérative à l'extrémité du réseau améliore la localité des données et aide à équilibrer la charge de travail du réseau. Avec l'émergence de *Mobile-Edge Computing*, il est désormais possible de déployer des approches de mise en cache coopérative à l'extrémité du réseau à proximité des utilisateurs mobiles en utilisant les stations de base MEC. De cette manière, en dissociant le temps pendant lequel un contenu est téléchargé depuis un cloud distant du moment où le contenu est livré à l'utilisateur mobile, un réseau de mise en cache coopérative *Mobile-Edge Computing* peut améliorer l'expérience de service et réduire latence et économisant les ressources du réseau, en d'autres termes, *Mobile-Edge Computing Cooperative Caching* (MEC Co-Caching) fait référence à l'utilisation de la mise en cache coopérative dans des stations de base intelligentes équipées de capacités de serveur MEC, pour stocker le contenu au tout premier saut des utilisateurs finaux, par exemple, si un utilisateur mobile demande

un contenu populaire qui est mis en cache, il recevra le contenu directement de la station de base pertinente plutôt que du serveur cloud distant d'origine jusqu'à ce que la cohérence des enregistrements de session ou du cache expire.

1.1 Problématique

Chaque *Mobile Network Operators* (MNO) utilise peu de passerelles pour acheminer le trafic des utilisateurs mobiles hors de leur inter-réseau vers la dorsale Internet. Cette architecture d'infrastructure, combinée à un nombre croissant d'utilisateurs mobiles, entraîne des problèmes de bande passante et de ressources du réseau [6][7]. De plus, une grande quantité de consommation de trafic sur Internet est causée par la demande de contenus populaires par les utilisateurs. A titre d'exemple sur Youtube, 80% des vues ne sont dédiées qu'à 10% des contenus populaires [8]. De cette façon, les réseaux de MNO souffrent de fournir le même contenu plusieurs fois alors que ce n'est pas nécessaire [9]. De même, cette congestion entraîne une qualité de service et une expérience utilisateur médiocres. La mise en cache coopérative est l'un des moyens de gérer ce trafic supplémentaire dans l'extrémité du réseau. D'autre part, les réseaux de diffusion de contenu peuvent souffrir considérablement de la latence du réseau causée par la distance par rapport au serveur cloud d'origine ou au serveur proxy intermédiaire. MEC *Co-Caching* tente de résoudre ces problèmes dans un système de mise en cache coopérative implémenté sur MEC. Bien qu'il existe des études sur la mise en cache dans les réseaux cellulaires, les efforts ont porté principalement sur l'analyse théorique par modélisation. De plus, des décisions doivent être prises sur la manière de déployer un système de mise en cache coopérative MEC. Il devrait également y avoir des approches pour la cohérence des enregistrements de cache.

1.2 Objectifs de mémoire

Ce mémoire démontre la conception, la mise en œuvre et l'évaluation d'un système de mise en cache coopérative *Mobile-Edge Computing* qui fournit du contenu aux utilisateurs mobiles. L'accent est mis sur l'étude de la faisabilité et des performances de l'utilisation de l'approche MEC pour la mise en cache coopérative. Les performances de livraison du contenu, d'expérience utilisateur sont mesurées et analysées.

1.3 Organisation de mémoire

Le reste du mémoire est structuré comme suit : chapitre 2 présentent l'étude de fond et les concepts nécessaires à la mise en œuvre du système de mise en cache coopérative *Mobile-Edge Computing*, et résume les concepts du *Mobile Cloud Computing* (MCC), du *Mobile-Edge Computing* (MEC), et présente aussi différentes perspectives sur la mise en cache coopérative dans les réseaux liés aux mobiles. Différentes approches ont été étudiées pour la mise en cache coopérative, telles que la mise en cache sur les nœuds mobiles, la mise en cache sur des nœuds intermédiaires ou proxy ou la mise en cache à l'extrémité du réseau. chapitre 3 est dédié à fournir des détails plus approfondis sur la mise en cache coopérative MEC en général . Dans le chapitre suivant , nous présentons la mise en œuvre et les résultats de l'évaluation. Enfin, le mémoire se termine par une conclusion generate.

2.1 Introduction

Ce chapitre présente le contexte et les concepts nécessaires à la mise en œuvre du système de mise en cache coopérative *Mobile-Edge Computing*. Cette section résume les concepts de MCC, MEC et La mise en cache coopérative . MEC est une nouvelle technologie qui est actuellement en cours de standardisation dans ETSI [5]. MEC fournit un environnement de services informatiques et des capacités de Cloud Computing à l'extrémité du réseau mobile. L'objectif est de réduire la latence, d'assurer un fonctionnement du réseau et une prestation des services hautement efficaces, et d'offrir une expérience utilisateur améliorée. MEC augmente le temps de réponse depuis d'Edge et permet d'accélérer les contenus, les services et les applications afin d'améliorer l'expérience de l'utilisateur mobile sur un réseau et des services efficaces. Contrairement aux réseaux mobiles centralisés traditionnels, MEC fournit des capacités de cloud computing pour amener les contenus populaires à l'extrémité du réseau (par exemple, les passerelles, les stations de base et les appareils des utilisateurs finaux) afin que le contenu soit plus proche de l'utilisateur final. Les applications et les analyses sur les serveurs MEC ne sont pas affectées par la congestion et la latence. En fait, en effectuant des analyses ou en mettant en cache le contenu sur les serveurs MEC, le volume de données transmises au cœur pour traitement est réduit et l'impact du trafic de données via le réseau est minimisé. Le concept de mise en cache coopérative est basé sur l'idée de demander les données nécessaires à un nœud voisin du réseau au lieu de la ressource d'origine. Différentes approches ont été proposées pour la mise en cache coopérative, telles que la mise en cache sur les nœuds mobiles, la mise en cache sur les nœuds intermédiaires ou proxy ou la mise en cache à l'extrémité du réseau. Dans ce chapitre, nous étudions aussi les différentes perspectives sur la mise en cache coopérative dans les réseaux mobiles.

2.2 Mobile-Cloud Computing

Ces dernières années, nous assistons à une demande importante d'utilisateurs souhaitant disposer de différents types de services cloud sur leurs appareils mobiles, par exemple, les services de divertissement, de réseaux sociaux, d'affaires, d'actualités, de jeux ou de santé [10]. Cependant, cette demande est confrontée aux appareils mobiles avec des problèmes tels que la faible consommation d'énergie, les faibles ressources et la faible connectivité [1]. Pour résoudre ce problème, le terme MCC est apparu et les chercheurs tentent de définir les limites et de donner des définitions appropriées.

Il existe plusieurs définitions existant pour MCC. En général, il s'agit d'un service en cours d'exécution sur un serveur Cloud riche en ressources qui sont utilisées par un client mobile léger [10]. Il peut également être référé lorsque les nœuds mobiles jouent le rôle de fournisseur de ressources dans un réseau pair-à-pair [11]. De même, comme il est mentionné dans Fernando et al dans [10] on peut considérer le MCC comme un réseau avec certaines caractéristiques. Nous pouvons prendre le besoin d'adaptabilité, d'évolutivité, de disponibilité dans le concept de Cloud Computing [12] et l'étendre au MCC.

Alternativement, le MCC pourrait être défini de manière plus complète comme il est cité dans MCC Forum [13] dans Dinh et al [14] comme suit : « Le Cloud Computing mobile dans sa forme la plus simple fait référence à une infrastructure où le stockage et le traitement des données se produisent à lieu hors de l'appareil mobile. Les applications cloud mobiles déplacent la puissance de calcul et le stockage des données des appareils mobiles vers le Cloud, apportant des applications et MCC non seulement aux utilisateurs de smartphones, mais à un éventail beaucoup plus large d'abonnés mobiles ».

MCC s'attaque à certains défis des appareils mobiles d'une manière souhaitable. L'efficacité énergétique est atteinte par plusieurs solutions comme l'accès intelligent au disque ou à l'écran [15] et principalement par les techniques de déchargement. Diverses recherches démontrent que le déchargement des tâches des applications mobiles vers le Cloud distant peut notamment économiser de l'énergie [16] et réduire la consommation d'énergie de l'appareil mobile. De plus, la capacité de stockage des données et la puissance de traitement sont améliorées grâce au stockage et à l'accès aux mégas données sur le Cloud [14]. Nous pouvons également avoir plus de fiabilité en stockant nos données sur le Cloud sur différents serveurs Cloud.

Cependant, malgré toutes les améliorations apportées par le MCC, il reste encore des problèmes à résoudre. Des problèmes tels que la faible bande passante, la latence élevée, la disponibilité du service, QoS et le coût du service [14]. La bande passante est limitée dans les réseaux sans fil par rapport aux réseaux filairesC normaux. Les utilisateurs ont besoin d'une haute disponibilité malgré l'absence de connectivité aux appareils mobiles et ils exigent une meilleure qualité de service et des coûts de service réduits. De plus, la latence du réseau est toujours un gros fardeau pour améliorer l'expérience utilisateur en obtenant la voie des services Cloud. Ces problèmes sont plus tangibles dans les applications qui fournissent des services de réalité virtuelle nécessitant une faible latence et une bande passante élevée [3].

Par conséquent, considérant des faiblesses discutées précédemment, l'utilisation des ressources à proximité de l'utilisateur et l'amélioration de la disponibilité, la connectivité et la latence du réseau. Satyanarayanan dans [3] propose le concept de "Cloudlet" pour faire face à ces objectifs qui mène plus tard à *Mobile-Edge Computing*.

2.3 Cloudlets

Comme le montre la figure 2.1, le cloudlet est considéré comme le niveau intermédiaire d'une hiérarchie à 3 niveaux : appareil mobile, cloudlet et Cloud. Un cloudlet peut également être considéré comme un centre riche en ressources à proximité des utilisateurs. Cloudlet est connecté à un serveur Cloud plus important et son objectif est de rapprocher les services Cloud de l'utilisateur final [3].

Dans le concept de cloudlet, l'appareil mobile décharge sa charge de travail sur un cloudlet local riche en ressources. Les cloudlets seraient situés dans des espaces communs tels que les cafés, les bibliothèques ou les résidences universitaires, afin que les appareils mobiles puissent se connecter et fonctionner comme un client léger au cloudlet [10]. Un cloudlet peut être n'importe quel élément de premier saut à l'extrémité du réseau. Il n'a qu'un état logiciel, il doit être riche en ressources et bien connecté, avec une faible latence de bout en bout et il suit également une certaine norme de déchargement (par exemple, la migration de la machine virtuelle) [17]. En d'autres termes, la défaillance d'un cloudlet n'est pas critique, il a une forte connectivité interne et un réseau local sans fil à large bande passante et il doit être à proximité logique et physique de l'utilisateur pour réduire la latence du réseau.

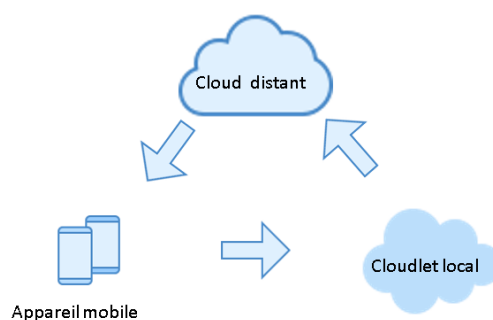


FIGURE 2.1 – L’architecture de Cloudlet

Il existe deux approches principales pour mettre en œuvre une infrastructure cloudlet à l’aide de la technique de *Virtual Machine* (VM). Dans ces deux architectures, il est important que le cloudlet puisse revenir à son état initial après avoir été utilisé (par exemple, par un nettoyage après utilisation). Une approche basée sur les machines virtuelles est largement utilisée car elle peut encapsuler et séparer proprement l’environnement logiciel invité transitoire du logiciel hôte permanent de l’infrastructure cloudlet et elle est moins fragile que d’autres approches comme la migration de processus ou la virtualisation logicielle [3].

En ce qui concerne la mise en œuvre, il devrait être possible de transférer un état de VM de l’application mobile des utilisateurs vers l’infrastructure du cloudlet. La première approche est la migration de VM dans laquelle une VM déjà en cours d’exécution est suspendue et son état de processeur, de disque et de mémoire sera transféré vers la destination et l’exécution reprendra à partir de l’état exact de suspension dans l’environnement hôte cloudlet. La deuxième approche est la synthèse dynamique de VM, l’appareil mobile fournissant une petite superposition de VM au lieu des états mentionnés dans la première approche à l’infrastructure cloudlet qui possède la base de VM. La superposition est calculée par l’appareil mobile sur la base de l’image personnalisée encapsulant les exigences de l’application, voire figure 2.2 [4]. Ensuite, la superposition est exécutée dans l’état exact où elle a été suspendue et le résultat sera renvoyé par le cloudlet [3].

Les cloudlets utilisent des machines virtuelles déployées rapidement que le client peut personnaliser librement selon ses besoins pour créer l’image de machine virtuelle ou la superposition de machine virtuelle qui contient l’application et toutes les exigences nécessaires pour fonctionner correctement [4]. Dans les deux types d’implémentations, l’image ou la superposition de VM est créée au moment de l’exécution par l’utilisateur, ce qui est assez flexible pour décharger la charge de travail sur le cloudlet. Néanmoins, malgré cette flexibilité, la procédure de création

d'une image ou d'une superposition de VM ainsi que l'encapsulation de l'état de l'application peuvent prendre beaucoup de temps. En fin de compte, le choix d'utiliser ou non les cloudlets comme sources riches en ressources dépend totalement de la conception, des besoins et de l'environnement de l'application.

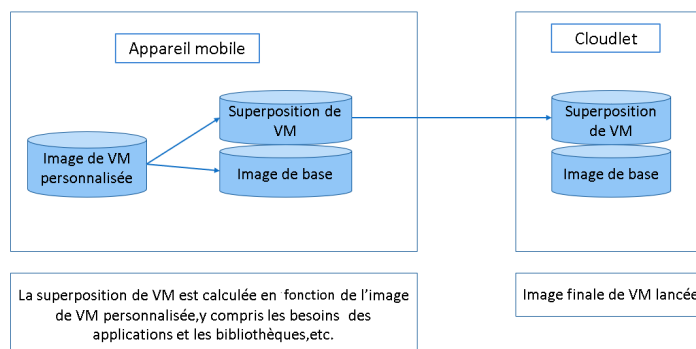


FIGURE 2.2 – Synthèse dynamique de VM cloudlet

2.4 Mobile-Edge Computing

Dans Satyanarayanan et al. [3] il est indiqué qu'un cloudlet pourrait être n'importe quel nœud de premier saut à l'extrémité du réseau qui ressemble à un cluster d'ordinateurs multicœurs en interne avec une connexion Internet à large bande passant vers un cloud distant. En d'autres termes, qui va jouer le rôle à l'extrémité du réseau? Va-t-il être uniquement dédié aux clouds privés qui ont la possibilité de déployer des cloudlets dans leur réseau local? Y aura-t-il des normes pour augmenter la simplicité et aussi pour encourager les développeurs à créer des applications utiles?

Réalisant les avantages de rapprocher les services et les ressources Cloud, le concept de cloudlet a attiré l'attention des opérateurs de RAN, car c'est une idée raisonnablement valable d'utiliser les stations de base RAN en ce tout premier saut à l'extrémité du réseau pour servir les utilisateurs mobiles. Par conséquent, de sérieux efforts ont commencé à rendre cette idée pratique pour l'environnement des affaires. À titre d'exemple, Nokia Networks a récemment présenté sa nouvelle génération de stations de base intelligentes qui sont considérées comme des stations de base compatibles avec une plate-forme Edge computing.

En outre, ETSI avec la coopération d'opérateurs tels que Huawei, IBM, Intel, Nokia Networks, NTT DOCOMO et Vodafone ont formé *Industry Specification Group* (ISG) [5] pour créer une plate-forme d'environnement standardisée et ouverte pour apporter services Cloud plus proches des utilisateurs finaux et de formuler une intégration logique des applications mobiles sur cette plate-forme entre les vendeurs, les fournisseurs de services et les développeurs tiers, en d'autres termes, l'objectif est de créer une initiative pour MEC.

L'objectif de MEC est de transformer les stations de base en centres de services intelligents personnalisables hautes performances à l'extrémité des réseaux mobiles tout en générant des revenus et une valeur unique pour les opérateurs en offrant différentes propositions de valeur aux utilisateurs mobiles, telles que la proximité des ressources, le contexte et connaissance de l'emplacement, agilité et rapidité [18].

Une station de base compatible MEC permet aux développeurs d'exécuter une application à l'extrémité du réseau à l'aide d'une plate-forme standard prédéfinie. Cette plate-forme peut également offrir des services supplémentaires tels que le stockage en cloud, la mise en cache, l'informatique, etc. Concrètement, cela transforme la station de base en un serveur MEC. Un serveur MEC peut être déployé sur différents types de stations de base *Long-Term Evolution* (LTE) tels que ENodeB ou 3G *Radio Network Controller* (RNC) [18]. La figure 2.3 illustre de l'architecture générale de MEC.

Pour conclure, MEC est un nouvel écosystème qui permet aux MNO de fournir à des tiers autorisés une plate-forme pour accéder à Edge de RAN et déployer des applications uniques basées sur les fonctionnalités MEC. Enfin, ces avantages améliorent la qualité d'expérience (QoE) pour les abonnés mobiles et apportent de la valeur aux opérateurs, leur permettant de jouer des rôles complémentaires et rentables au sein de leurs modèles commerciaux respectifs [18].

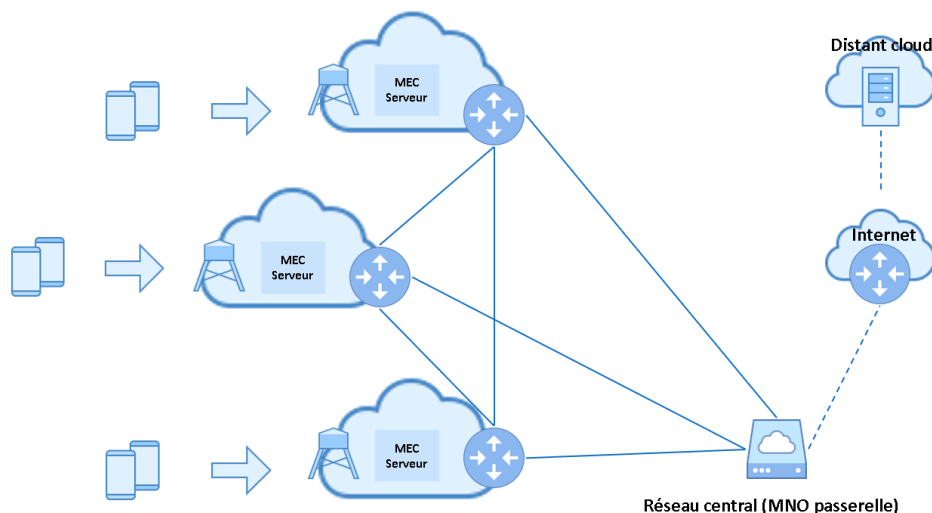


FIGURE 2.3 – Architecture de MEC

2.5 Mise en cache coopérative mobile

Grâce aux améliorations technologiques des appareils mobiles, les clients mobiles sont capables de partager des données entre eux en tant que pairs. De cette façon, ils peuvent rester indépendants du serveur d'origine d'où proviennent les données. La mise en cache coopérative mobile est une agrégation de cette alternative avec le concept de mise en cache pour les appareils mobiles [19]. Dans la mise en cache coopérative mobile, les appareils mobiles tentent de former un réseau ad hoc avec d'autres nœuds mobiles à proximité pour partager les données pertinentes, voir Figure 2.4. Pour développer ce type de réseau, il convient d'envisager des politiques appropriées et de sélectionner des algorithmes efficaces concernant l'invalidation d'enregistrement de cache, le niveau de cohérence, le placement et la recherche d'enregistrement de cache. Certaines études connexes sont examinées ci-dessous.

Il existe plusieurs résultats souhaitables résultant de la mise en cache coopérative dans les réseaux mobiles ad hoc, tels que : l'amélioration de la latence d'accès, l'équilibrage de la charge de travail du serveur principal et l'atténuation de la congestion des canaux point à point. Cependant, il pourrait y avoir quelques inconvénients à augmenter le surcoût de communication entre les appareils mobiles [20]. De même, CP et *Communication Service Providers* (CSP) bénéficient financièrement de la réduction de la consommation de bande passante dans le réseau qui est obtenue en réduisant la demande de téléchargement vers la source de données par les utilisateurs finaux mobiles [21] [22]. Certaines préoccupations existent concernant la mise en œuvre et l'efficacité de la mise en cache coopérative dans les environnements mobiles. Le système peut souffrir considérablement de l'égoïsme de l'utilisateur, d'une solution non

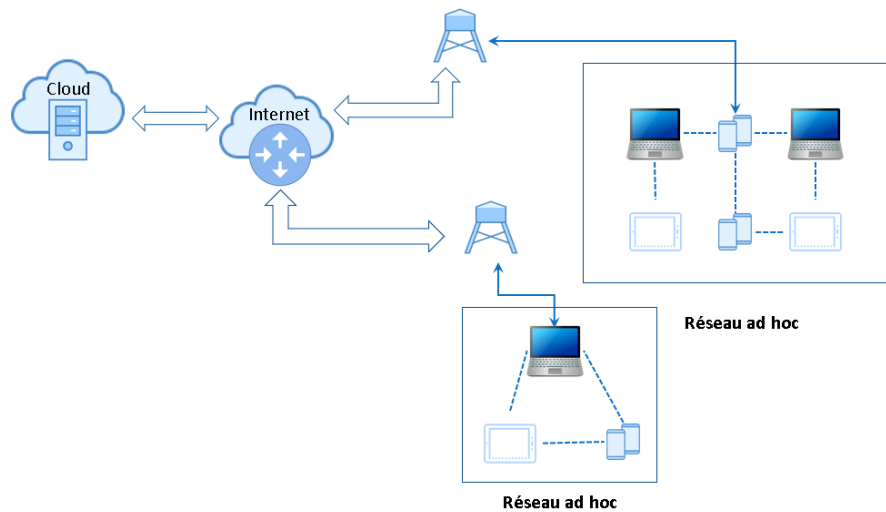


FIGURE 2.4 – Réseau ad hoc de mise en cache coopérative mobile

optimale pour placer des enregistrements en cache ou d'un mauvais modèle de recherche [21]. Un modèle de recherche est une approche pour trouver la version cohérente des données dans le réseau. De plus, pour maintenir la cohérence des données, une approche concrète doit être utilisée pour l'invalidation du cache [23]. L'invalidation du cache est un processus de mise à jour des valeurs des entrées mises en cache sur tous les nœuds du réseau. Pourtant, ce processus peut être différent dans chaque cas en fonction du niveau de cohérence requis par le système [24][25][26].

Il existe deux approches principales pour l'invalidation du cache : la méthode push et la méthode pull. Lors de méthode pull, l'appareil mobile demande au propriétaire du contenu mis en cache de vérifier la validation de l'enregistrement mis en cache existant. Cette approche entraîne une surcharge sur les nœuds. De plus, le niveau de cohérence dépend des intervalles d'extraction des autres nœuds. Néanmoins, il convient aux réseaux mobiles ad hoc car ce ne sont pas des réseaux stables et n'importe quel nœud peut partir à tout moment. À l'inverse, le point de départ de la méthode push est le source original du contenu. En cas de modification ou d'invalidation, la source d'origine doit en informer les propriétaires de la copie en cache. Cette méthode est une méthode basée sur les événements. Dans cette méthode, le niveau de cohérence est plus élevé car le message d'événement peut se propager rapidement à travers le réseau et les nœuds ne dépendent pas les uns des autres des intervalles d'extraction. D'autre part, l'inconvénient de cette méthode est la surcharge sur la source d'origine pour l'indexation des abonnés. La source doit connaître tous les abonnés à un contenu, afin qu'elle sache où envoyer les messages d'invalidation. La méthode push convient aux réseaux stables où le changement de

nœud ne se produit pas souvent.

De plus, les méthodes de placement et de recherche dans les enregistrements en cache et leurs performances sont un autre domaine d'étude. Taghizadeh et al dans [21] étudient une politique coopérative de placement dans les enregistrements en cache pour la minimisation des coûts dans un environnement de réseau mobile avec des demandes de contenu homogènes. Ils recherchent une répartition optimale entre la duplication des objets mis en cache et l'unicité dans chaque nœud. Une approche gourmande pour chaque nœud consiste à mettre en cache autant d'enregistrements distincts que possible, ce qui entraîne une non-coopération et une duplication de contenu importante. Dans l'autre cas extrême, un nœud essaie de maximiser le nombre total d'enregistrements uniques mis en cache dans le réseau en évitant les duplications. L'étude explique que trouver une approche entre ces deux, se traduirait par un meilleur coût de réseau pour les fournisseurs. De plus, l'impact de l'égoïsme des utilisateurs sur le coût du réseau est simulé et analysé.

2.6 Conclusion

La conception de l'Internet actuel crée des problèmes que l'utilisateur ne peut pas supporter. Par ailleurs, les différents défis comme la mobilité, la sécurité, la mise en cache et la QoS ont fait naître de nouvelles exigences. De plus, les utilisateurs commencent à se concentrer sur les informations souhaitées, ce qui augmente le besoin de recentrer la communication sur l'accès au contenu plutôt que sur l'interaction hôte à hôte. Les chercheurs pensent que MEC pourraient jouer un rôle clé dans la croissance universelle d'internet en traitant le contenu des données comme une entité de première classe dans l'architecture du réseau et en étant capable de stocker ou mettre en cache le contenu du serveur à partir de divers emplacements à proximité immédiate de l'utilisateur final. Le chapitre suivant introduit la mise en cache coopérative, qui est un concept élémentaire pour construire notre système de mise en cache.

CHAPITRE 3

SYSTÈME DE MISE EN CACHE COOPÉRATIVE MOBILE-EDGE COMPUTING

3.1 Introduction

La mise en cache Edge est l'un des techniques les plus émergentes reconnues comme une solution de récupération de contenu à l'extrémité du réseau. Il a également été considéré comme une technique habilitante de MEC qui présente une opportunité intéressante pour effectuer des services de mise en cache coopérative. En particulier, les serveurs MEC sont mis en œuvre directement au niveau des stations de base qui permettent la mise en cache Edge et garantissent un déploiement à proximité des utilisateurs mobiles. Ce chapitre présente un aperçu général sur la mise en cache coopérative, illustre les principales stratégies de mise en cache et leurs principes, leurs objectifs et les moyens de les atteindre. De plus, il contient une description des mesures de performance ainsi que les critères de base couramment utilisés pour comparer les mécanismes de mise en cache existants. Dans cette section, nous passons en revue quelques notions élémentaires dont la compréhension est nécessaire pour tirer pleinement parti de ce mémoire.

3.2 Aperçu

Répondre aux exigences exigeantes de QoS des services émergents en temps réel, interactifs et multimédias a conduit à des paradigmes d'Edge Computing, où l'accent est mis sur la fourniture d'une QoS élevée, à de faibles coûts opérationnels. En outre, la croissance rapide du nombre d'utilisateurs mobiles utilisant ces services entraîne une croissance significative du trafic sur MNO, qui doit être traitée en fonction de la rentabilité, de la capacité limitée de la bande passante de liaison et des temps de réponse souhaitables des applications mobiles. L'utilisation

de la mise en cache coopérative à l'extrémité du réseau améliore la localisation des données tout en aidant à équilibrer la charge de travail du réseau de MNO. Une solution de mise en cache coopérative à l'extrémité du réseau peut améliorer l'expérience QoS et réduire la latence tout en économisant les MNO. La mise en cache coopérative de *Mobile-Edge Computing* fait référence à l'utilisation de la mise en cache coopérative sur les stations de base intelligentes de MNO qui sont équipées de capacités de serveur MEC, pour stocker le contenu au tout premier saut du réseau des utilisateurs finaux. Par exemple, si un utilisateur mobile demande un contenu en cache populaire, il recevra le contenu directement de la station de base concernée plutôt que du serveur cloud distant d'origine jusqu'à ce que la cohérence de la session ou de l'enregistrement en cache expire.

Chaque MNO utilise quelques passerelles pour acheminer le trafic des utilisateurs mobiles hors de leur inter-réseau vers la dorsale Internet. Cette architecture d'infrastructure combinée à un nombre croissant d'utilisateurs mobiles entraîne des problèmes de bande passante et de ressources réseau [6][7]. D'autre part, une grande quantité de consommation de trafic est liée à la demande de contenus populaires sur le Web. En fait, de cette manière, les réseaux cellulaires souffrent de l'envoi du même contenu plusieurs fois alors que ce n'est pas nécessaire [9]. De même, cette congestion entraîne une qualité de service et une expérience utilisateur médiocres.

L'augmentation du nombre d'utilisateurs mobiles a conduit à des déploiements omniprésents de stations de base de radiofréquences à haut débit, ce qui a consécutivement entraîné l'apparition progressive de *Mobile-Edge Computing* pour relever les défis mentionnés précédemment. La mise en cache coopérative MEC aide les MNO à équilibrer la charge du réseau et à réduire les demandes de contenu redondantes. De même, cela contribue à réduire le trafic inter et intra-réseau des MNO [7][27]. Pour conclure, l'utilisation de la mise en cache coopérative MEC peut accélérer la rentabilité, améliorer l'expérience utilisateur et la qualité de service.

3.3 Cas d'utilisation d'une demande de contenu mobile

Comme l'illustre la figure 3.1 dans cette partie, nous passons en revue le flux de processus du système mis en cache coopérative MEC après une simple demande de contenu d'un client mobile qui est connecté à l'une des stations de base . Ce diagramme de processus contient tous les principaux processus mis en œuvre dans le système. Une application mobile est implémentée pour se connecter au système mis en cache coopérative MEC et demander un contenu. Cette application envoie l'identifiant de contenu en tant que demande de contenu au programme principal nous appelons ce serveur principal. Le serveur principal vérifie son cache local pour le contenu, s'il s'agit d'un succès, l'utilisateur mobile recevra une copie en cache du contenu, sinon, le serveur principal vérifie si l'identifiant de contenu existe dans cache global . Si oui, cela signifie qu'il y a un nœud de balise à proximité qui a déjà demandé le même contenu, il a donc une copie en cache.

Au contraire, s'il n'y a pas de nœud de balise, cela signifie que la station de base elle-même est la première à demander ce contenu . Ainsi, il demande directement le contenu à la source d'origine et après avoir reçu le contenu et l'avoir renvoyé à l'utilisateur, le serveur principal ajoute les informations de la station de base en tant que nœud de balise pour le contenu . Une copie du contenu sera également ajoutée au cache local. À la fin, il est évident que si la station de base a une copie en cache du contenu dans son cache local, elle sera immédiatement renvoyée au client.

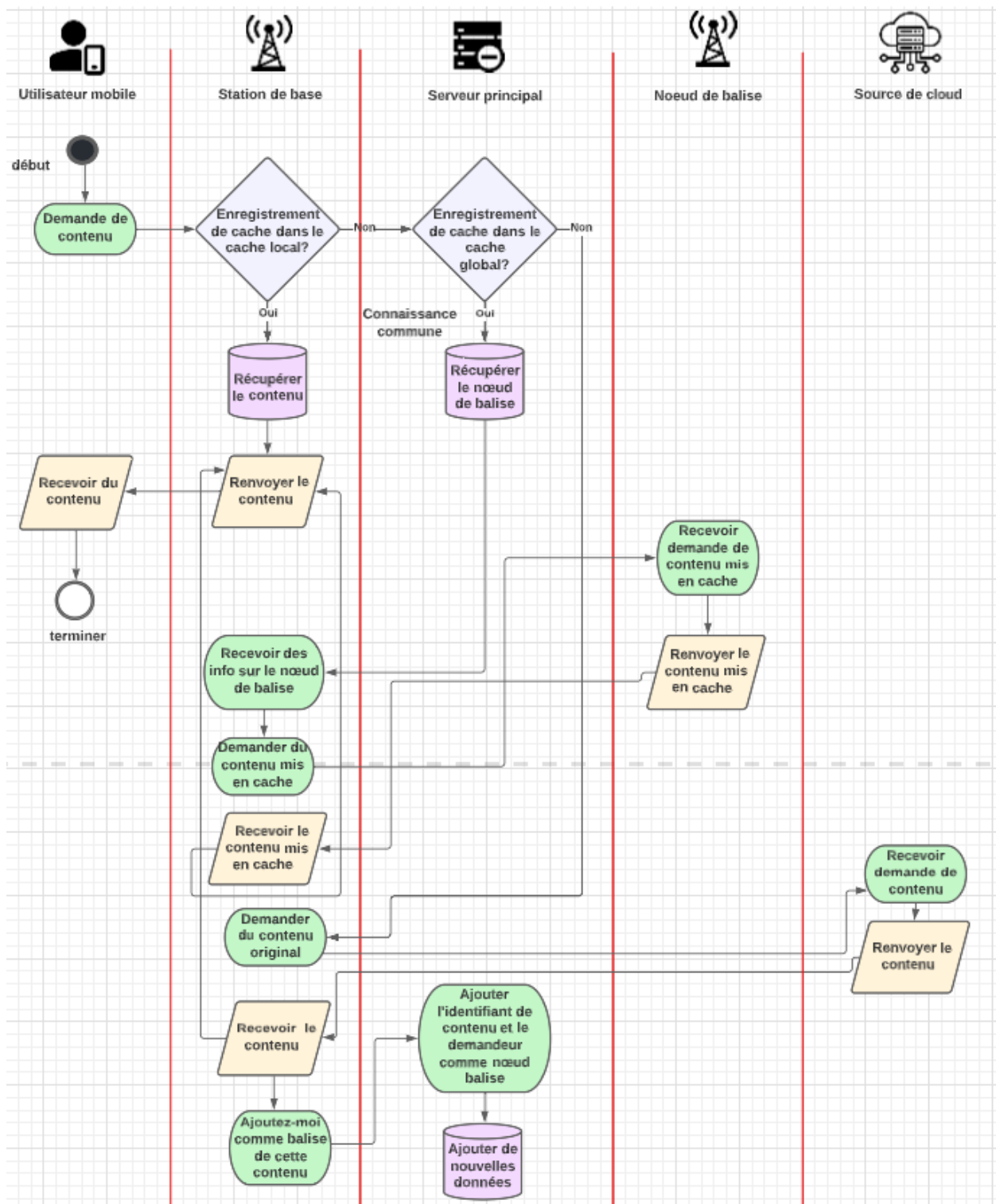


FIGURE 3.1 – Diagramme du processus de demande de contenu

3.4 Conception du système de Co-Caching MEC

Le système de mise en cache coopérative de MEC est un système de mise en cache distribué déployé sur un cluster de stations de base MEC. Toutes les stations de base de ce cluster exécutent leurs propres serveurs MEC afin d'héberger leur part du système. L'idée est que n'importe quelle application de diffusion de contenu arbitraire peut se connecter à ce système avec *Application Interface* (API) d'application et bénéficier de la mise en cache. Certains aspects doivent être pris en compte dans un tel système afin d'atteindre les objectifs souhaités. Nous passons en revue chacun de ces aspects au regard des objectifs suivants : faible latence réseau, expérience utilisateur riche, cohérence du cache, évolutivité et utilisation minimale de la bande passante inter et intra-réseau.

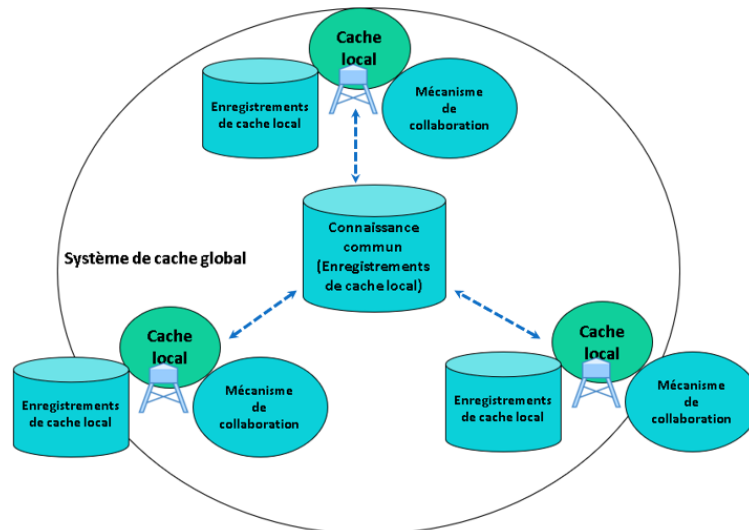


FIGURE 3.2 – Architecture du système de mise en cache coopérative Mobile-Edge Computing

L'architecture globale du système de mise en cache MEC est illustrée à la figure 3.2. Nous reconnaissons deux entités différentes dans notre système. Le premier est le système de cache global qui est formé en coopération sur un groupe de stations de base MEC. La seconde est l'entité cache locale dans chacune des stations de base. Dans le système de cache global, il existe un référentiel de connaissances commun qui est pratiquement l'index de tous les enregistrements de cache existant dans le système. Chaque station de base du cluster garde une trace de tous les enregistrements du cluster via ce référentiel. De plus, dans chaque station de base elle-même, il existe un référentiel d'enregistrements de cache local qui est pratiquement une table de hachage de tout le contenu des données qui est mis en cache directement par la station de base ou demandé à un nœud de balise.

En plus d'un référentiel de cache local dans chaque station de base, il existe un module de mécanisme de collaboration pour synchroniser tous les caches locaux avec le système de cache global. Il existe également un module de cohérence qui s'occupe de la validation et de la cohérence des enregistrements mis en cache. La figure 3.3 illustre ses différents modules du système de mise en cache MEC qui réside dans le serveur MEC de chacune des stations de base du cluster.

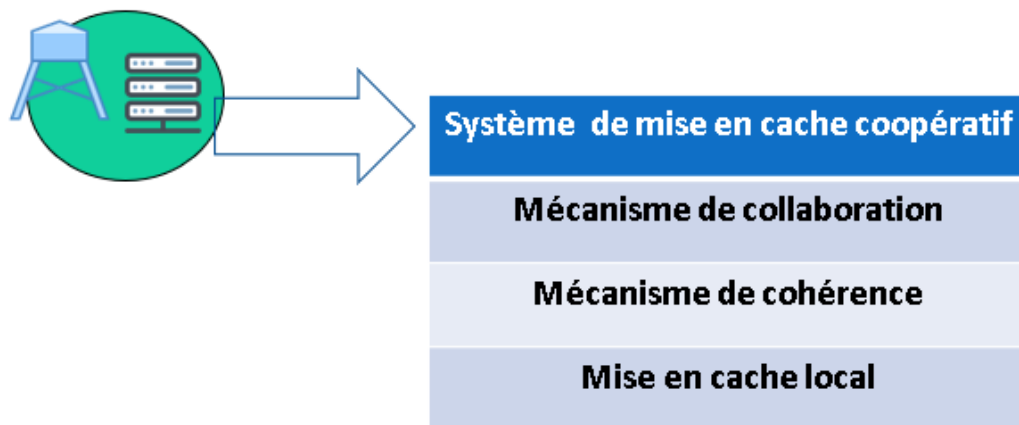


FIGURE 3.3 – Parties du système de cache coopératif MEC sur chaque serveur MEC

3.4.1 Faible latence et expérience utilisateur riche

La mise en cache coopérative MEC est un système distribué qui rapproche le contenu le plus possible de l'utilisateur mobile en utilisant l'approche MEC. Cela signifie que le système est déployé à l'extrémité du réseau sur des stations de base MEC avec une connectivité élevée à Internet. L'utilisateur mobile obtient le contenu mis en cache du système de mise en cache MEC via la station de base connectée. Cette station de base est le premier saut de réseau auquel l'utilisateur est connecté. Cela permet de réduire le délai du réseau pour récupérer le contenu et contribue à améliorer l'expérience utilisateur.

3.4.2 Cohérence

Il existe deux approches générales pour gérer la cohérence d'un système de mise en cache distribué, les méthodes pull et push. Dans l'approche d'extraction, un nœud avec un contenu mis en cache s'occupe de la validité et de la cohérence de l'enregistrement mis en cache. Il sait qui est le nœud de balise dans la hiérarchie supérieure du contenu mis en cache et a certaines

périodes, vérifie si la version du contenu mis en cache est toujours valide ou si elle est modifiée. De cette manière, si le contenu d'origine est modifié, le nœud peut avoir le contenu du nœud balise ou il peut se demander lui-même à partir de la source d'origine. D'autre part, l'approche push fonctionne en soulevant certains événements du côté de la source cloud d'origine vers les propriétaires des copies en cache pour les informer des dernières modifications. De cette manière, les nœuds balises du réseau de distribution de contenu sont chargés d'informer les nœuds abonnés du contenu.

Dans la mise en cache coopérative MEC, nous utilisons l'approche push. Cela signifie que si des modifications surviennent sur un contenu mis en cache, les métadonnées à l'intérieur du référentiel de connaissances commun seront modifiées et automatiquement toutes les stations de base seront informées de ce changement et elles mettront à jour leurs copies en cache ou prendront une action appropriée.

3.4.3 Évolutivité

Le système de mise en cache coopérative de MEC est conçu pour être aussi évolutif que possible. Comme le montre la figure 3.4, cette évolutivité signifie qu'il devrait être possible d'ajouter ou de supprimer de nouvelles stations de base au cluster en cours d'exécution. De plus, il devrait être possible de fusionner deux clusters MEC différents pour en faire un plus grand. Le module de mécanisme de collaboration est chargé de fournir cette flexibilité en permettant à de nouvelles stations de base de rejoindre le système de cache global et d'avoir accès au référentiel de connaissances commun. Les autres stations de base seront informées de l'arrivée de la nouvelle station de base. Le scénario est le même pour la fusion de deux clusters.

3.4.4 Utilisation minimale de la bande passante

Lorsque les utilisateurs mobiles sont connectés au réseau du MNO via des stations de base, les données d'application doivent traverser le réseau du MNO jusqu'à la passerelle du MNO, la passerelle est connectée au Internet. L'utilisation de la bande passante du Internet est l'un des principaux coûts des services de données des MNO, donc si nous réduisons ce coût, nous aidons les MNO à offrir des services moins chers aux utilisateurs finaux. Pour réduire l'utilisation de la bande passante inter et intra-réseau, le système de mise en cache coopérative MEC conserve le trafic de mise en cache à l'intérieur du cluster de cache global en utilisant des nœuds balises. Ces nœuds balises sont les stations de base qui ont obtenu le contenu mis en cache pour la première fois à partir de la source cloud d'origine. Plus tard, si l'un des nœuds

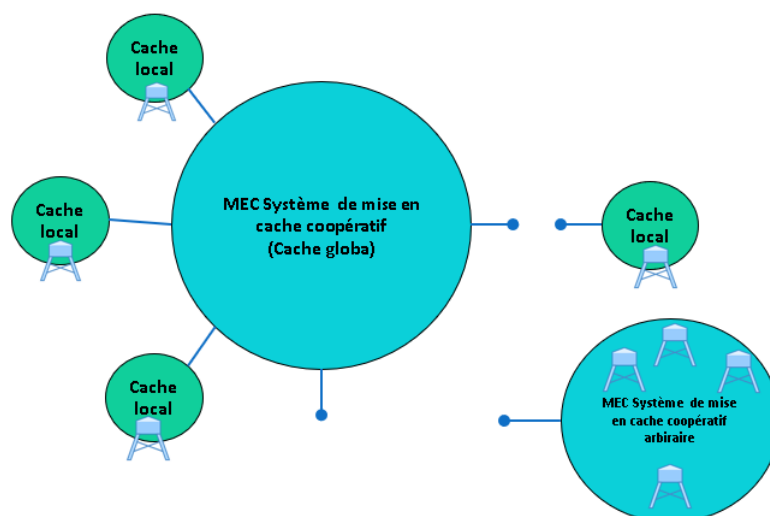


FIGURE 3.4 – Évolutivité de la mise en cache coopérative MEC

MEC doit envoyer le même contenu à un utilisateur, il le demandera au nœud balise au lieu de le demander à la source d'origine.

La figure 3.5 présente le cas d'utilisation que le système de mise en cache coopérative MEC essaie d'éviter pour garder le réseau formidable à l'intérieur du cluster de cache global. Dans ce scénario, l'utilisateur mobile est connecté à une station de base et demande l'objet X. La station de base transmet la demande au cloud distant qui contient le contenu d'origine alors qu'il existe une copie en cache valide de l'objet X dans une station de base à proximité. Respectivement, la figure 3.5 décrit le scénario qui se produit à l'intérieur du système de mise en cache coopérative MEC. Lorsque le nœud MEC reçoit la demande d'objet X, il vérifie d'abord le référentiel de connaissances commun dans le cache global pour voir s'il existe un nœud balise qui a précédemment demandé le même contenu. Si c'est vrai, la station de base récupère les méta-données (contenant l'adresse) du nœud balise spécifique pour se connecter et demande au nœud balise de lui renvoyer la copie en cache de l'objet X. Cette copie sera renvoyée à l'utilisateur par la station de base après réception du nœud balise.

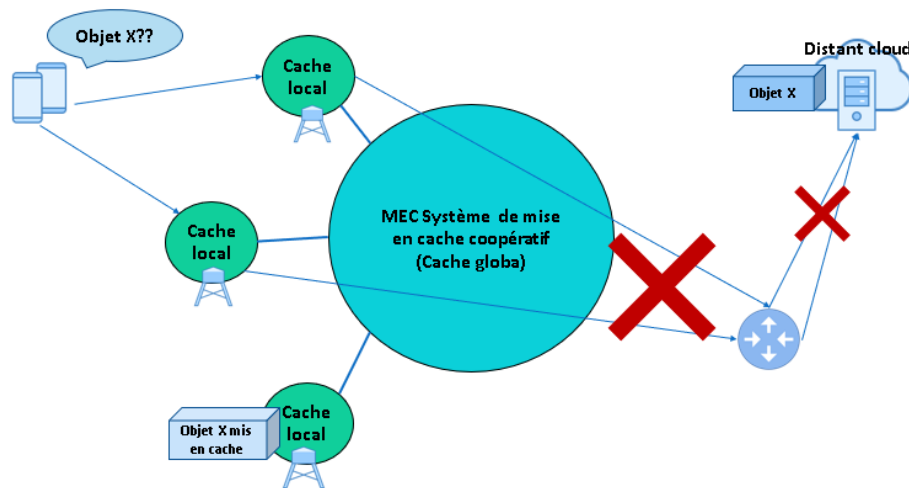


FIGURE 3.5 – Consommation de bande passante intra-réseau

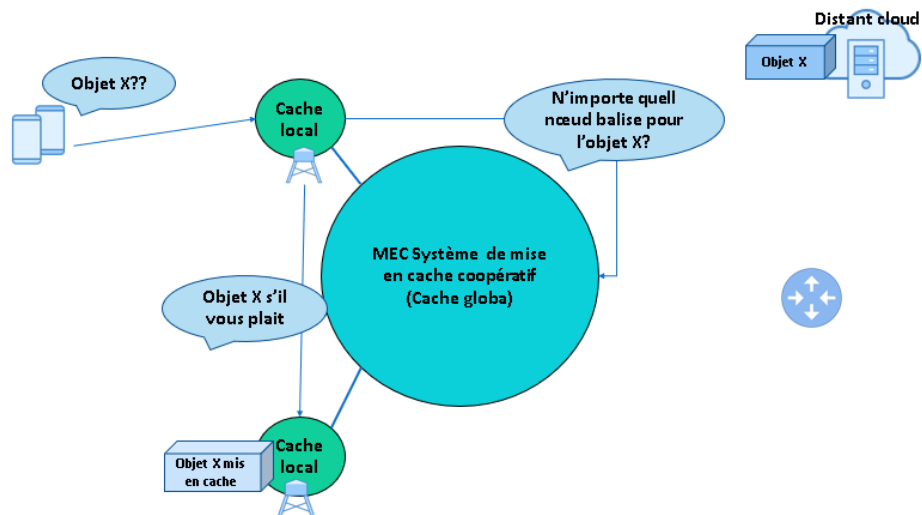


FIGURE 3.6 – Cas d'utilisation d'économie d'utilisation de la bande passante

3.5 Les lignes de base

Les lignes de base sont utilisées pour tester et analyser des indicateurs de performance importants tels que le taux de succès, le taux d'accès, le délai d'accès moyen, l'utilité de l'utilisateur, disponibilité des service, plus précisément, les lignes de base sont les algorithmes nœud de balise, probabilité et sans relais.

3.5.1 Nœud de balise

Un nœud balise est un nœud dans lequel l'enregistrement de cache d'origine d'une contenu dynamique existe. Ce nœud de balise fonctionne comme une référence à ce contenu spécifique. Le reste des nœuds qui a besoin de cet enregistrement particulier dans le groupe

coopèrent avec le nœud de balise de document pour les mises à jour d’invalidation. L’utilisation des nœuds balises permet d’alléger l’équilibrage de charge au sein du groupe et d’être résistant aux pannes. Les nœuds balises de différents clusters peuvent former un groupe pour travailler les uns avec les autres en tant que pairs et faciliter le transfert de documents mis en cache et également se reconnaître mutuellement au sujet de l’invalidation du cache. Un nœud balise dans le système est une station de base qui possède la toute première version du contenu mise en cache à partir de la source de données.

3.5.2 Probabilité

La technique de probabilité est basée sur la sélection des nœuds voisins dans manière aléatoire, en utilisant la fonction du RANDOM ou génération d’arrivées aléatoires via le processus de Poisson (nous utilisons génération d’arrivées aléatoires via le processus de Poisson qui fournit des résultats plus précis), cette technique a été mise en œuvre l’année dernière (2021) pour résoudre des problèmes , mais elle manque encore quelques indications telle que la disponibilité des services.

3.5.3 Sans relais

La technique de sans relais est basée sur l’accès direct à Cloud distant ou source d’origine, c’est une technique traditionnelle, cette technique est apparue avec l’avènement de cloud, ce n’est pas efficace et n’est pas pratique pour le moment car elle a une mauvaise performance.

3.6 Conclusion

Construisent Systeme de mise en cache cooperative MEC est un système de mise en cache distribué déployé sur un cluster de stations de base MEC. Toutes les stations de base de ce cluster exécutent leurs propres serveurs MEC afin d’héberger leur part du système. L’idée de base de mise en cache cooperative est relais de la demande entre les nœuds voisins du réseau au lieu de la source d’origine dans le but de résoudre les problèmes auxquels sont confrontés les utilisateurs finaux. Le chapitre suivant présentera l’implémentation et évaluation des performances des stratégies de mise en cache cooperative.

CHAPITRE 4

L'IMPLEMENTATION ET ÉVALUATION DES PERFORMANCES

4.1 Introduction

Ce chapitre présente les différents aspects de mise en œuvre du système de mise en cache coopérative MEC, et nous présentons également les techniques et les équations utilisés ainsi que leur pertinence pour le système, et nous avons proposé une nouvelle technique qui basé sur l'énergie. Enfin nous avons effectué un grand nombre de tests expérimentaux et d'analyses des indicateurs de performances importants tels que (le taux de succès, le délai d'accès moyen, l'utilité de l'utilisateur, disponibilité des services) pour comparer notre technique par rapport aux lignes de base. En raison de la sélection probabiliste existant dans certaines lignes de base et de la technique proposé, nous avons effectué 10 fois et analysé les résultats des améliorations proposées de la technique.

4.2 Modélisation du système et formulation des problèmes

À partir du modèle collaboratif de mise en cache d'Edge, il peut être divisé en collaboration Cloud-Edge, Edge-Edge et Edge-IoT. Le système proposé dans ce mémoire est basé sur le mode de collaboration Cloud-Edge et Edge-Edge pour étudier les stratégies de mise en cache coopérative. En raison de la capacité de stockage limitée des nœuds Edge, la configuration de l'installation ne peut être effectuée dans le nœud qu'après le téléchargement ou l'acquisition des fichiers source (ou des packages d'installation d'application) depuis le centre de cloud. Généralement, en raison de la capacité limitée des nœuds Edge, après l'installation de nouvelles applications de service, les nœuds Edge suppriment leurs fichiers source (ou packages d'installation d'application).

Dans l'architecture système de la figure 4.1, lorsqu'un utilisateur Internet émet une demande de services d'application qui ont été déployés dans le cloud ou les nœuds Edge, les nœuds Edge répondront à la demande par quatre actions différentes en fonction de leurs accès au cache. Dénotez la demande de l'utilisateur par $r : (f, s, p)$, le service demandé par f , Edge nœud ou serveur par s et la priorité de la demande par p .

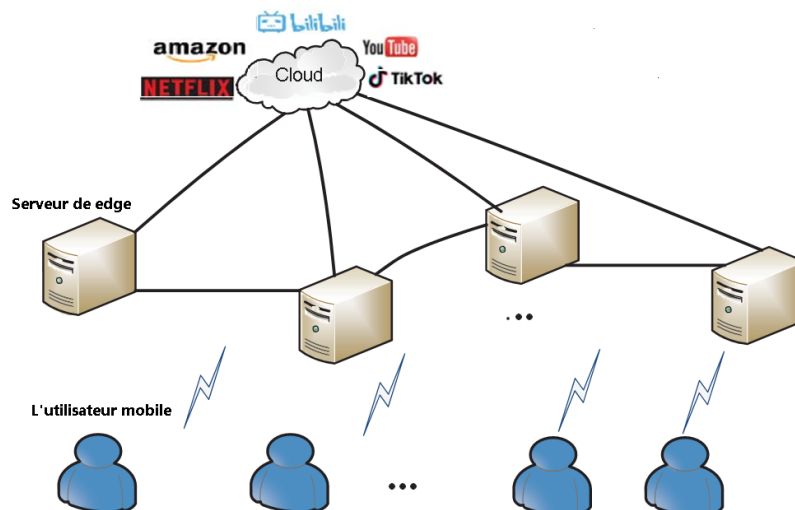


FIGURE 4.1 – Un exemple de système de mise en cache collaboratif en Edge

Tout d'abord, si le nœud Edge local s , tel que la station de base, le routeur et d'autres Edge dotés d'une capacité de stockage et de mise en réseau, a déployé le service f , la demande de l'utilisateur $r : (f, s, p)$ est atteinte localement. Le délai d'accès de la demande de succès local est enregistré comme t_l , d'une manière générale, t_l est petit.

Deuxièmement, si le nœud Edge local s n'est pas succès et que le nœud voisin s' a déployé le service f , s relaie la requête $r : (f, s, p)$ à s' . Le délai d'accès de la demande de succès de relais r est noté t_r , qui est généralement petit.

Troisièmement, si aucun des nœuds Edge n'est pas succès, le nœud local s enverra la requête $r : (f, s, p)$ au centre de données cloud (c'est-à-dire en contournant). La demande de l'utilisateur r n'est pas satisfaite et le délai d'accès à la demande est noté t_b , qui est généralement important.

Quatrièmement, si le nœud local ou le nœud voisin ne se connecte pas plusieurs fois, les nœuds Edge s et s' téléchargent le fichier source de l'application de service ou le package d'installation de l'application depuis le cloud et le configurent sur s ou s' . Le coût ou délai de

temps est t_f , ce qui est généralement important.

La forte simultanéité du service peut facilement entraîner une congestion des demandes. Lorsqu'une requête est bloquée dans un nœud, il y a un délai de mise en file d'attente.

On définit le taux de mise en file d'attente pour la demande de l'utilisateur, appelé le taux de blocage moyen en p_{queue} , comme indiqué dans Eq(1).

$$p_{queue} = \frac{n_{queue}}{n_{request}} \quad \text{Eq(1)}$$

Où n_{queue} représente le nombre de requêtes bloquant dans la file d'attente et $n_{request}$ représente le nombre de requêtes.

On définit Le délai total de mise en file d'attente de la demande dans le nœud, comme indiqué dans Eq(2).

$$\mathcal{T}_{queue} = \sum_{i=1}^k i * t_{avgQ} \quad \text{Eq(2)}$$

Où K est le nombre de tâches en file d'attente et t_{avgQ} est le délai moyen de mise en file d'attente, qui est généralement défini sur 100 millisecondes (ms).

Le taux de succès du cache h_r est un indicateur de performance essentiel pour l'évaluation du système de cache, et sa définition est indiquée dans Eq(3).

$$h_r = \frac{h_{local} + h_{relais}}{n_{request}} \quad \text{Eq(3)}$$

où h_{local} représente le nombre de succès locaux, h_{relais} représente le nombre de succès de relais et $n_{request}$ représente le nombre de requêtes.

De plus, le délai d'accès moyen t_{avg} est également un indicateur de performance essentiel pour l'évaluation du système de cache, et sa définition est indiquée dans Eq(4).

$$t_{avg} = t_l \times hr_{local} + (t_l + t_r) \times hr_{relais} + (t_l + t_r + t_b) \times p_{bypass} + (t_l + t_r + t_b + t_f) \times p_{fetch} + t_q \times p_{queue} \quad \text{Eq(4)}$$

Parmi eux, $hr_{local} = \frac{h_{local}}{n_{request}}$, $hr_{relais} = \frac{h_{relais}}{n_{request}} \cdot p_{bypass}$ fait référence à la proportion des nœuds locaux pour sélectionner le contournement de la demande de l'utilisateur vers le cloud à servir, parmi lesquels le nombre de contournement est $n_{bypass}, p_{fetch} = \frac{n_{fetch}}{n_{request}}$ fait référence

à la proportion du nombre de services d'installation et de configuration par rapport au total des demandes, t_l représente le délai d'accès de l'utilisateur au nœud local, t_r représente le délai d'accès de l'utilisateur au nœud voisin, t_b représente le délai d'accès de l'utilisateur au cloud et t_f représente le temps nécessaire pour configurer l'installation du service ou de l'application.

De plus, nous observons l'équilibrage de charge des nœuds avec la variance de charge du nœud d'Edge va , qui est définie dans Eq(5) et représente la stabilité du délai de réponse du nœud. Du point de vue des utilisateurs, plus la variance de la charge des nœuds est grande, plus la différence de délai de la demande de réponse de service perçue par les utilisateurs est grande (c'est-à-dire que le délai de réponse à la demande de l'utilisateur est parfois grand et parfois petit).

$$va = \sum_{i=1}^m (AVG(hc) - hc(si))^2 \quad \text{Eq(5)}$$

où $AVG(hc)$ est la fonction qui trouve le nombre moyen de charges sur tous les nœuds.

Il est la demande de l'utilisateur $r : (f, s, p)$ obtient la réponse dans le système de cache, comme illustré à la figure 4.1, en fonction du niveau de service et de l'action de réponse obtenue ($u_{niveau} = f * p$), l'utilitaire utilisateur $u_{request}$ est défini intuitivement comme indiqué dans Eq(6).

$$u_{request} =$$

$u_{niveau} * 100$: si réponse locale ;

$u_{niveau} * 10$: si relais au voisin ;

$u_{niveau} * 1$: si contournement vers le cloud ;

Eq(6)

4.3 La technique proposée de mise en cache collaboratif distribué Edge

Ce système a été construit en deux étapes : la première étape consiste à sélectionner un nœud avec énergie, c'est une technique de mise en cache qui assure l'équilibre énergétique et qui améliore la stratégie de mise en cache . Dans la deuxième étape est l'algorithme de la stratégie de mise en cache collaborative Edge qui donnait les indicateurs de performance essentiels pour analyser et évaluer la performance de notre proposition.

4.3.1 La phase de sélection de nœud avec énergie

Nous améliorons une nouvelle technique à l'aide de technique déjà existante, cette technique est la probabilité qui mentionné dans le chapitre 3, nous remarquons que la technique de probabilité est manqué certains indicateurs de performances importants tels que la disponibilité des services, ce qui pose le problème de perdre des nœuds et leur cache local, cela conduit à la perte de données, ceci n'est pas souhaitable et doit être résolu, nous avons proposé une nouvelle stratégie qui améliorer la technique de probabilité pour résoudre leurs problèmes, notre technique proposée, nous l'avons appelée énergie parce qu'il ajoute l'équilibrer d'énergie qui assure étende la durée de vie du système et moins de perdre des nœuds, par conséquent, moins de perte des données, notre technique est basé sur la recherche dans les nœuds voisin a énergie maximale, par exemple s'il y a un nœud qui a une faible énergie nous passons à un nœud voisin a haute énergie, et cela atteint l'équilibrer d'énergie qui assure l'indicateur très important la disponibilité des services.

4.3.2 Stratégie de mise en cache collaborative Edge

Algorithm 1 Stratégie de mise en cache collaborative Edge

Require: $hc(s) = 0, U = 0, r := (f, s, p)$

Ensure: taux d'accès au cache hr , délai d'accès moyen t , utilité de l'utilisateur U

```

for chaque requête  $r : (f, s, p)$  do
  if  $s$  ont un slot  $q$ , et  $f$  dans  $q$  then
     $hc(s)++$ ,  $U += \text{getUtility}(f, p, \text{'local'})$ ,  $s$  réponses  $r$  et son retard est  $t_l$ 
  else if voisin  $s'$  a un slot  $q$ , et  $f$  dans  $q$  then
     $hc(s')++$ ,  $U += \text{getUtility}(f, p, \text{'relay'})$ ,  $s$  réponses  $r$  et son retard est  $t_l + t_r$ 
  else
     $U += \text{getUtility}(f, p, \text{'bypass'})$ ,  $s$  réponses  $r$  et son retard est  $t_l + t_r + t_b$ 
  end if
end for

```

l'algorithme 2 décrit l'action de réponse du nœud Edge ou du nœud voisin, ou du cloud service ou réponse lorsque l'utilisateur demande r l'accès f . $hc(s)$ représente le nombre de requêtes ou charges traitées dans le nœud s , $\text{getUtility}(f, p, \text{'local'})$ représente Eq(6) pour calculer l'utilité d'utilisateur de f lorsque la priorité de demande d'utilisateur est p et que le serveur local est servi au nœud local s . Relay signifie que la demande de l'utilisateur est transmise par le nœud local et répondue par le nœud voisin. Bypass signifie que les demandes des utilisateurs sont contournées par les nœuds locaux et répondues par le cloud.

4.4 Configuration expérimentale

Considérant le système illustré à la figure 4.1, nous définissons quelques paramètres importants et les expliquons comme suit : nous avons défini t_l 1, t_r 10, t_b 100 et l'unité de temps a simplement été définie en millisecondes (ms) . le rapport du coût de fonctionnement de l'extraction et du contournement est de 10, donc réglez $t_f = 1000$ ms. Dans le scénario réel, le délai de mise en file d'attente est étroitement lié à la capacité d'E/S de la machine, au taux d'arrivée des données ou demandes, etc, et il n'existe pas de valeur universelle. Par conséquent, il est possible d'utiliser la valeur approximative dans la longue traîne . La plate-forme expérimentale sélectionnée est présentée dans le Table 4.1.

TABLE 4.1 – Plateforme expérimentale

| Élément | Configuration de plateforme expérimentale |
|---------------------------|---|
| CPU | Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz 2.81 GHz |
| Mémoire RAM installée | 8,00 Go |
| Le système d'exploitation | Windows 10 professional |
| Language | c++ |

4.5 Résultats expérimentale

Sur la base des résultats expérimentaux, comme le montre la figure 4.1, nous pouvons analyser l'impact de la capacité du cache de nœud sur les performances des techniques suivant :

Cette figure 4.2 montrée le taux de succès au fil de la taille du cache, on peut remarque que chaque technique augmente avec l'augmentation de la taille du cache, et courbe d'énergie est très proche de nœud balise et probabilité grâce à propriété de relais entre les nœuds du système, de plus le taux de succès de technique sans relais est minimal car à chaque fois contourné vers le Cloud distant.

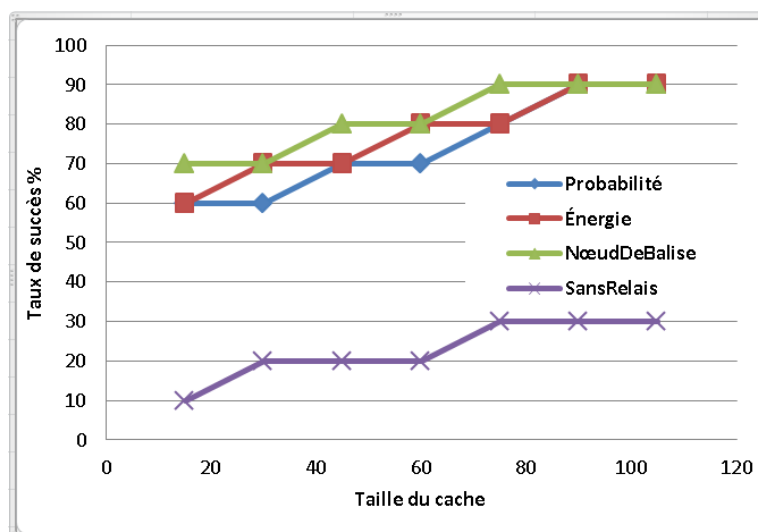


FIGURE 4.2 – Le taux de succès

Cette figure 4.3 montrée le délai d'accès moyen au fil de la taille du cache, on peut remarque que la technique sans relais est considérablement réduite, les avantages apportés par le mécanisme de relais font que les trois autres techniques(énergie, nœud balise, probabilité) ne sont pas significativement modifiés, nous concluons que le délai d'accès moyen inversement proportionnel avec le taux de succès.

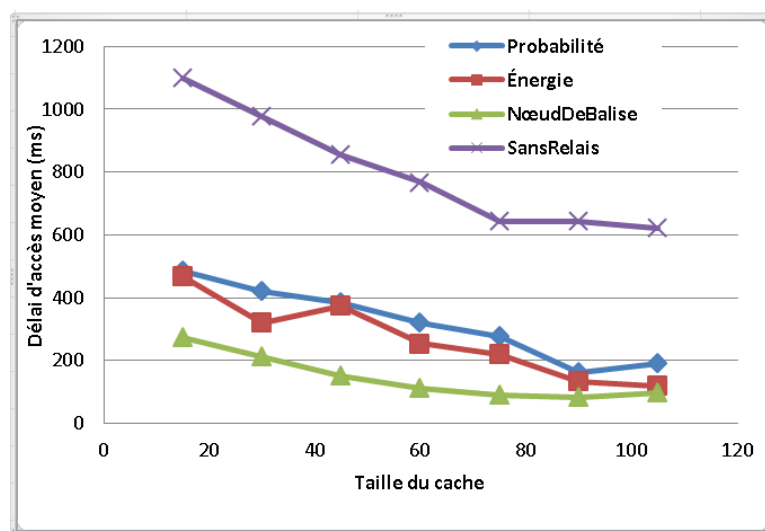


FIGURE 4.3 – Le délai d'accès moyen

Cette figure 4.4 montrée l'utilité de l'utilisateur au fil de la taille du cache, on peut remarque que la technique sans relais est bien inférieure à 0 en petite capacité, elle n'est pas indiquée ici, et on peut voir que la technique de nœud balise est un peu supérieure aux autres techniques(énergie, probabilité), nous concluons que l'utilité de l'utilisateur inversement proportionnel avec le délai d'accès moyen et directement proportionnel avec le taux de succès.

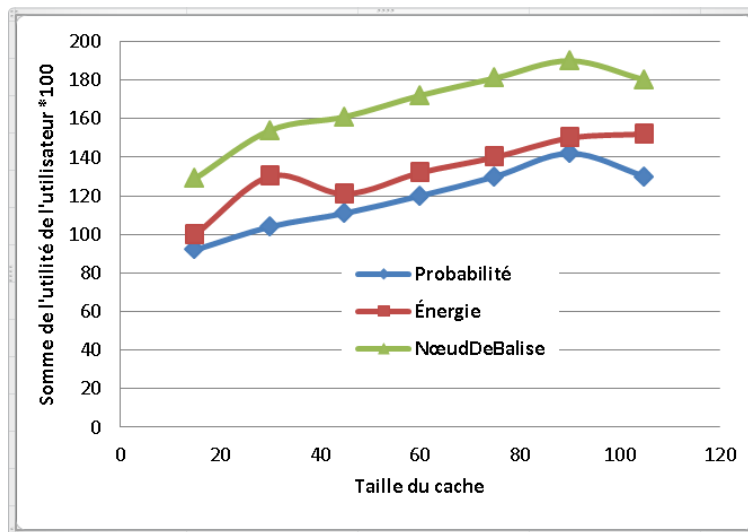


FIGURE 4.4 – L'utilité de l'utilisateur

Cette figure 4.5 montrée la disponibilité des nœuds au fil du temps, on peut remarque que la technique d'énergie assure la meilleure performance par rapport aux autres techniques parce que la chute du nœud se produit presque également, car il contient la propriété d'équilibrer d'énergie, alors que notre technique est donnée un résultat considérable par rapport aux autres techniques, et cela conduit à étende la durée de vie du système.

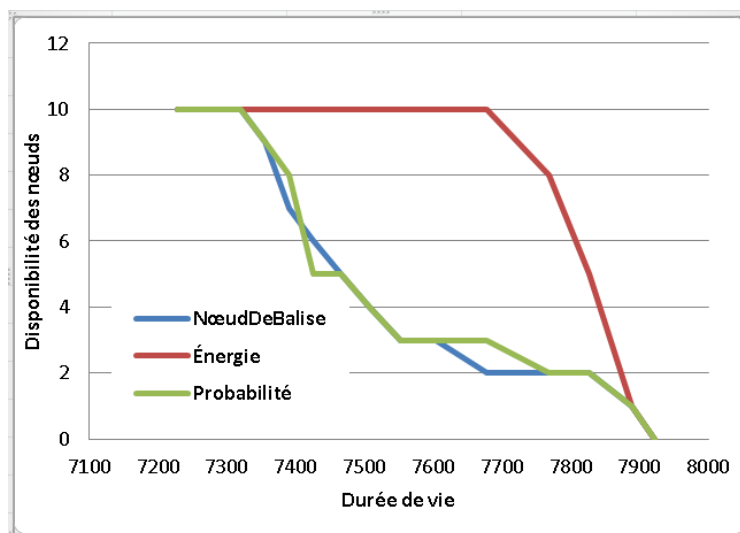


FIGURE 4.5 – Disponibilité des nœuds

4.6 Conclusion

Dans ce chapitre, nous présentons le modèle de sélectionner un nœud avec énergie. Nous avons présenté un nouveau système de mise en cache, Les résultats expérimentaux montrent que notre proposition garantit non seulement que le serveur d'équilibrage d'énergie traite la demande, mais réduit également la latence moyenne d'accès aux demandes des utilisateurs, améliorant ainsi l'utilité des utilisateurs, et surtout, il assure la disponibilité du service.

Dans ce mémoire, nous avons proposé un nouveau scénario de demande dans le système de la mise en cache collaborative d'Edge . Ce système a été construit en deux étapes : la première étape consiste à sélectionner un nœud avec énergie, c'est une technique de mise en cache qui assure l'équilibre énergétique et qui améliore la stratégie de mise en cache . Dans la deuxième étape est la stratégie de mise en cache collaborative Edge qui donnait les indicateurs de performance essentiels pour analyser et évaluer la performance de notre proposition par rapport aux autres techniques existence, nous avons considéré les techniques suivantes : nœud de balise, probabilité et sans relais, ces techniques sont utilisées pour tester et analyser des indicateurs de performance importants tels que le taux d'accès, le délai d'accès moyen, etc. Les résultats expérimentaux montrent que notre proposition garantit non seulement que le serveur d'équilibrage d'énergie traite la demande, mais réduit également la latence moyenne d'accès aux demandes des utilisateurs, améliorant ainsi l'utilité des utilisateurs, et surtout, il assure la disponibilité du service.

Comme indiqué dans les chapitres précédents, la mise en cache d'Edge est considérée comme une technique habilitante de *Mobile-Edge Computing* où les serveurs MEC sont implémentés directement sur les stations de base pour assurer un déploiement à proximité des utilisateurs mobiles. Cependant, l'intégration de serveurs dans un environnement *Mobile-Edge Computing* complique le problème d'économie d'énergie car la puissance consommée par les serveurs MEC est coûteuse, en particulier lorsque la charge change dynamiquement dans le temps. De plus, les utilisateurs d'appareils mobiles présentent leurs demandes, introduisant le défi de gérer ces demandes de contenu mobile en plus de la taille limitée de la mise en cache. Ainsi, il est nécessaire et crucial que les mécanismes de mise en cache prennent en compte les facteurs mentionnés, tandis que la plupart des études actuelles et futures se concentrent sur l'allocation du cache, la popularité du contenu et la conception du cache.

- [1] M. SATYANARAYANAN ,School of Computer Science Carnegie Mellon University. Fundamental challenges in mobile computing. In Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing (1996), ACM, pp. 1–7.
- [2] H. T. DINH, C. LEE, D. NIYATO, AND P. WANG, A survey of mobile cloud computing : architecture, applications, and approaches. *Wireless communications and mobile computing* 13, (2013), 1587–1611.
- [3] M. SATYANARAYANAN, P. BAHL, R. CACERES, N. DAVIES, The case for vm based cloudlets in mobile computing. *Pervasive Computing, IEEE* 8, 4 (2009),pp. 14–23.
- [4] T. Kämäräinen, Aalto University . Design, implementation and evaluation of a distributed mobile cloud gaming system, Espoo 26.5.2014.
- [5] Executive Briefing-Mobile Edge Computing (MEC) Initiative, ETSI - European Telecommunications Standards Institute, September 2014.
- [6] J. ERMAN, A. GERBER, M. HAJIAGHYI, D. PEI, S. SEN, AND O. SkPATSCHECK. To cache or not to cache : The 3g case. *Internet Computing, IEEE* 15, 2 (March-April 2011),pp. 27–34.
- [7] S. WOO, E. JEONG, S. PARK, J. LEE, S. IHM, AND K. PARK , Comparison of caching strategies in modern cellular backhaul networks. In Proceeding of the 11th annual international conference on Mobile systems, applications, and services (2013), ACM, pp. 319–332.
- [8] M. CHA, H. KWAK, P. RODRIGUEZ,Y. -Y. AHN, AND S. MOON I Tube, You Tube, Everybody Tubes : Analyzing the World’s Largest User Generated Content Video System. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (2007), ACM, pp. 1–13.

-
- [9] X. WANG, X. LI, V. C. LEUNG, AND P. NASIOPULOS, A framework of cooperative cell caching for the future mobile networks. *IEEE* (2015) 1530-1605.
- [10] N. FERNANDO, S. W. LOKE, AND W. RAHAYU, Mobile cloud computing : A survey. *Future Generation Computer Systems* 29, 1 (2013), 84 – 106. Including Special section : AIRCC-NetCoM 2009 and Special section : Clouds and Service-Oriented Architectures.
- [11] E. E. MARINELLI, School of Computer Science Carnegie Mellon University , Hyrax : cloud computing on mobile devices using mapreduce. Tech. rep., DTIC Document, 2009.
- [12] L. MEI, W. K. CHAN, AND T. TSE, A tale of clouds : Paradigm comparisons and some thoughts on research issues. In *Asia-Pacific Services Computing Conference, 2008. APSCC'08*. *IEEE* (2008), Ieee, pp. 464–469.
- [13] Mcc forum. <http://www.mobilecloudcomputingforum.com/>. [Consulté le 17/03/2022]
- [14] H. T. DINH, C. LEE, D. NIYATO, AND P. WANG, A survey of mobile cloud computing : architecture, applications, and approaches. *Wireless communications and mobile computing* 13, 18 (2013), 1587–1611.
- [15] P. RANGANATHAN, Recipe for efficiency : principles of power-aware computing. *Communications of the ACM* 53, 4 (2010), 60–67.
- [16] A. P. MIETTINEN, AND J. K. NURMINEN, Energy efficiency of mobile clients in cloud computing. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* (2010), USENIX Association, pp. 4–4.
- [17] Elijah, cloudlet-based mobile computing group.<http://elijah.cs.cmu.edu/>. [Consulté le 22/03/2022]
- [18] http://https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf. [Consulté le 02/04/2022]
- [19] C.-Y. CHOW, H. V. LEONG, AND A. CHAN, Peer-to-peer cooperative caching in mobile environments. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on* (2004), *IEEE*, pp. 528–533.
- [20] C.-Y. CHOW, H. V. CHOW, AND A. CHAN, Peer-to-peer cooperative caching in a hybrid data delivery environment. In *Parallel Architectures, Algorithms and Networks, 2004. Proceedings. 7th International Symposium on* (2004), *IEEE*, pp. 79–84.
- [21] M. TAGHIZADEH, K. MICINSKI, C. OFRIA, E. TORNG, AND S. BISWAS, Distributed cooperative caching in social wireless networks. *Mobile Computing, IEEE Transactions on* 12, 6 (2013), 1037–1053.

-
- [22] YADGAR, G., FACTOR, M., AND SCHUSTER, A. Cooperative caching with return on investment. In *Mass Storage Systems and Technologies (MSST), 2013 IEEE 29th Symposium on* (2013), IEEE, pp. 1–13.
 - [23] J. Cao, Y. Zhang, G. Cao, AND L. Xie, Data consistency for cooperative caching in mobile environments. *Computer* (2007).
 - [24] N. KUMAR, AND J.-H. LEE, Peer-to-peer cooperative caching for data dissemination in urban vehicular communications. *Systems Journal, IEEE* 8, 4 (2014), 1136–1144.
 - [25] S. LIM, W.-C. LEE, G. CAO, AND C. R. DAS, Cache invalidation strategies for internet-based mobile ad hoc networks. *Computer Communications* 30, 8 (2007), 1854–1869.
 - [26] H.SAFA, H. ARTAIL, AND M. NAHHAS, A cache invalidation strategy for mobile networks. *Journal of Network and Computer Applications* 33, 2 (2010), 168–182.
 - [27] M. CHEN, AND A. KSENTINI, Cache in the air : exploiting content caching and delivery techniques for 5g systems. *IEEE Communications Magazine* (2014), 132.