

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche scientifique

Université A/Mira de Bejaia  
Faculté des Sciences Exactes

Département d'informatique  
Ecole Doctorale Réseaux et Système Distribués

*Mémoire de magistère*

En Informatique

Option

Réseaux et Systèmes Distribués



*Thème*

---

---

# Mécanisme de remplacement de cache dans les réseaux Ad Hoc

---

---

Présenté par :

Ali LARBI

Devant le jury composé de :

<b>Président :</b>	Mohand Said Radjef	Professeur, Université de Béjaïa
<b>Rapporteur :</b>	Djamil Aïssani	Professeur, Université de Béjaïa
<b>Examineur :</b>	Abdellah Boukeram	Maître de Conférences, Université de Sétif
<b>Examineur :</b>	Natalia Djellab	Maître de Conférences, Université de Annaba
<b>Invitée :</b>	Louiza Bouallouche	Maître de Conférences, Université de Béjaïa

Proposé par : Pr D. AÏSSANI  
Et MC L. BOUALLOUCHE

Promotion 2007/2008

# Dédicaces

*À mes parents.*

# Remerciements

Je tiens à remercier en premier lieu mon promoteur M<sup>r</sup> D. AÏSSANI, Professeur à l'Université de Béjaïa, de m'avoir honoré avec la proposition de ce sujet. Je remercie également M<sup>me</sup> L. BOUALLOUCHE Maître de Conférences à l'université de Béjaïa, qui m'a co-encadré durant toute cette année, pour ses conseils et orientations. Je les remercie pour la confiance qu'ils m'ont accordée, pour leur disponibilité et tout le temps qu'ils m'ont consacré durant ce travail.

Je remercie tout particulièrement les membres de mon jury, qui ont accepté de juger ce travail. J'adresse mes très sincères remerciements à M<sup>r</sup> M.S. RADJEF, Professeur à l'Université de Béjaïa, qui a bien voulu présider mon jury; à M<sup>r</sup> A. BOUKERAM, Maître de Conférences à l'Université de Sétif et à M<sup>me</sup> N. DJELAB, Maître de Conférences à l'Université de Annaba, pour avoir accepté de juger mon travail. Je les remercie d'avance pour leurs critiques et suggestions.

Je tiens à exprimer toute ma reconnaissance à Nadjette BENHAMIDA pour avoir répondu à mon aide, je la remercie doublement.

Je voudrais remercier toutes les personnes qui contribuent au bon fonctionnement de l'école doctorale.

Je ne pourrai clôturer ces remerciements sans me retourner vers les êtres qui me sont le plus chers, J'adresse mes remerciements à mes parents qui sans eux aucune réussite n'aurait été possible, à toute ma famille en particulier ma petite sœur. Je voudrai remercier profondément ma femme pour avoir toujours répondu présente pour mon aide, je lui suis infiniment reconnaissant.

Je voudrai remercier toutes les personnes qui ont contribué, d'une manière ou d'une autre, au bon déroulement de ce travail.

# Table des matières

<b>Table des matières</b>	I
<b>Liste des figures</b>	V
<b>Abstract</b>	VI
<b>Résumé.</b>	.VI
<b>Introduction générale</b> .....	1
<b>1 Etude des réseaux mobiles Ad Hoc</b>	
1.1 Introduction.....	4
1.2 Définition .....	5
1.3 Les domaines des réseaux mobiles Ad Hoc.....	5
1.3.1 Domaine militaire .....	5
1.3.2 Situation de sinistres .....	5
1.3.3 Réseaux locaux .....	5
1.4 Les caractéristiques des réseaux Ad Hoc .....	5
1.4.1 Bande passante .....	6
1.4.2 Portée radio limitée .....	6
1.4.3 Interférences .....	6
1.4.4 Energie .....	6
1.4.5 Mobilité et changement de topologie .....	7
1.4.6 Sécurité .....	7
1.5 Les normes des réseaux locaux sans fil .....	7
1.5.1 Le Bluetooth .....	7
1.5.2 L' HiperLAN 1 .....	9
1.5.3 L' HiperLAN 2 .....	10
1.5.4 La 802.11 .....	11
1.5.4.1 La couche physique .....	11
1.5.4.2 La couche MAC .....	13
1.5.4.3 Les différentes extensions de la norme 802.11 .....	16
1.6 Le routage dans les réseaux Ad Hoc .....	18
1.6.1 Les approches de routage des réseaux mobiles Ad Hoc .....	18
1.6.1.1 L'architecture de routage .....	19
1.6.1.2 Les techniques utilisées pour le routage .....	19
1.6.1.3 Les familles de protocoles de routage .....	20
1.6.2 Présentation de quelques protocoles de routage des réseaux Ad Hoc .....	21
1.6.2.1 Ad Hoc On-demand Distance Vector (AODV) .....	21
1.6.2.2 Optimized Link State Routing (OLSR) .....	22
1.6.2.3 Dynamic Source Routing (DSR) .....	23
1.6.2.4 Zone Routing Protocol (ZRP) .....	23
1.7 Conclusion .....	24

## 2 Les caches et les réseaux mobiles Ad Hoc

2.1 Introduction.....	25
2.2 Classification des caches.....	26
2.2.1 Organisation interne des caches.....	26
2.2.1.1 Le cache non bloquant .....	26
2.2.1.2 Le cache victime .....	26
2.2.1.3 Le cache en mode pipeline (pipelining).....	26
2.2.2 Localisation des caches.....	26
2.2.2.1 Le cache de navigateur .....	27
2.2.2.2 Le cache de Proxy.....	27
2.2.2.3 Le cache de serveur d'origine .....	27
2.2.3 La fonction des caches .....	28
2.2.3.1 Les caches CPU traditionnels .....	28
2.2.3.2 Les caches Web .....	28
2.2.3.3 Les caches des systèmes mobiles.....	28
2.3 Fonctionnement de cache.....	29
2.3.1 Support HTTP pour le cache.....	29
2.3.2 Cachabilité des documents.....	29
2.3.3 Défaut/succès de cache (cache hits/misses).....	30
2.3.4 Mécanismes de validation/invalidation de documents .....	30
2.3.4.1 Mécanismes de validation.....	31
2.3.4.2 Mécanismes d'invalidation .....	32
2.3.5 Le préchargement (Prefetching) : .....	33
2.3.6 Politique de remplacement de cache.....	34
2.3.6.1 Les métriques des politiques de remplacement .....	35
2.3.6.2 Les algorithmes de remplacement de caches.....	35
2.4 L'apport d'utilisation de cache dans les réseaux Ad Hoc.....	36
2.4.1 Réduction de la latence .....	37
2.4.2 Sauvegarde de la bande passante .....	37
2.4.3 Augmentation de l'accessibilité des données .....	38
2.5 Conclusion .....	38

## 3 Les politiques de remplacement de caches

3.1 Introduction.....	39
3.2 Classification des politiques de remplacement de cache .....	40
3.2.1 Les stratégies de remplacement de cache basées sur la récence.....	40
3.2.1.1 LRU .....	41
3.2.1.2 LRU-Threshold.....	41
3.2.1.3 SIZE .....	41
3.2.1.4 LRU-Min .....	41
3.2.1.5 EXP1 .....	41
3.2.1.6 HLRU .....	41

3.2.1.7 PSS (Pyramidal Selection Scheme) .....	42
3.2.1.8 Avantages.....	42
3.2.1.9 Inconvénients .....	42
3.2.2 Les stratégies de remplacement de cache basées sur la fréquence .....	43
3.2.2.1 LFU .....	43
3.2.2.2 LFU-Aging .....	43
3.2.2.3 LFU-DA .....	43
3.2.2.4 SWLFU (Server-Weighted LFU) .....	43
3.2.2.5 Fréquence relative de Benhamida .....	43
3.2.2.6 Avantages.....	44
3.2.2.7 Inconvénients .....	44
3.2.3 Les stratégies de remplacement de cache basées sur la récence et la fréquence. ....	44
3.2.3.1 SLRU (Segmented LRU) .....	44
3.2.3.2 LRU* .....	45
3.2.3.3 LRU-Hot .....	45
3.2.3.4 HYPER-G .....	45
3.2.3.5 LRU-K .....	45
3.2.3.6 Avantages.....	46
3.2.3.7 Inconvénients .....	46
3.2.4 Les stratégies de remplacement de cache basées sur les fonctions.....	46
3.2.4.1 GD (Greedy Dual)-Size .....	46
3.2.4.2 GDSF .....	46
3.2.4.3 TSP (Taylor Series Prediction) .....	47
3.2.4.4 Stratégie de Bolot et Hoschka .....	47
3.2.4.5 LRV .....	48
3.2.4.6 Avantages.....	48
3.2.4.7 Inconvénients .....	48
3.3 Les stratégies de remplacement de cache dédiées aux environnements mobiles .....	48
3.3.1 La politique TDS (Time and Distance Sensitive) .....	48
3.3.2 La politique ZC (Zone coopérative) .....	49
3.3.3 Politique de remplacement de cache orientée énergie .....	50
3.3.4 La politique de remplacement de cache PIX .....	51
3.3.5 La politique Min-SAUD .....	51
3.4 Proposition d'une politique de remplacement de cache pour les réseaux Ad Hoc .....	52
3.5 Conclusion .....	53

#### **4 Caractérisation de trafic, modélisation et simulation**

4.1 Introduction.....	54
4.2 Les méthodes d'évaluation de performances .....	55
4.2.1 Mesures directes.....	55
4.2.2 Analyse mathématique.....	55
4.2.3 Simulation basée sur les traces .....	55
4.2.4 Simulation basée sur le benchmarking .....	56
4.3 Etat de l'art sur la caractérisation du trafic .....	56

4.3.1 La popularité des documents .....	56
4.3.2 La taille des documents.....	57
4.3.3 La localité des références.....	58
4.3.3.1 La localité temporelle .....	58
4.3.3.2 La localité spatiale .....	59
4.3.4 Les documents référencés une seul fois (one-timers).....	59
4.3.5 Le taux de modification des documents.....	59
4.3.6 Le type des documents.....	60
4.4 Modélisation et génération de trafic .....	60
4.4.1 Modèle de simulation.....	61
4.4.2 La génération de trafic .....	61
4.4.3 Environnement de simulation .....	62
4.4.4 Validation du simulateur.....	62
4.4.5 Les caractéristiques statistiques de trafic généré .....	62
4.4.5.1 la distribution des tailles de documents .....	63
4.4.5.2 la distribution des TTL de documents .....	64
4.5 Résultats de la simulation .....	65
4.5.1 Impact du paramètre de l'âge moyen maximum sur la politique FSDV .....	65
4.5.2 Influence de la taille du cache.....	65
4.5.2.1 Taux de succès de FSDV en comparaison avec LRU et LFU .....	66
4.5.2.2 Taux de succès de FSDV en comparaison avec TDS et ZC.....	66
4.5.2.3 Taux de succès en octets de FSDV en comparaison avec LRU et LFU .....	67
4.5.2.4 Taux de succès en octets de FSDV en comparaison avec TDS et ZC.....	68
4.5.2.5 Taux de succès en distance de FSDV en comparaison avec LFU et LRU .....	69
4.5.2.6 Taux de succès en distance de FSDV en comparaison avec TDS et ZC .....	69
4.5.3 Impact du paramètre de la loi de distribution Zipf .....	70
4.5.3.1 Taux de succès de FSDV en comparaison avec LFU et LRU .....	71
4.5.3.2 Taux de succès de FSDV en comparaison avec TDS et ZC.....	72
4.5.3.3 Taux de succès en octets de FSDV en comparaison avec LFU et LRU .....	73
4.5.3.4 Taux de succès en octets de FSDV en comparaison avec TDS et ZC.....	73
4.5.3.5 Taux de succès en distances de FSDV en comparaison avec LFU et LRU.....	74
4.5.3.6 Taux de succès en distances de FSDV en comparaison avec TDS et ZC .....	75
4.6 Conclusion .....	75
<b>Conclusion générale .....</b>	<b>77</b>
<b>Glossaire.....</b>	<b>80</b>
<b>Bibliographie .....</b>	<b>84</b>

## Liste des figures

1.1	Un scatternet Bluetooth .....	8
1.2	L'organisation générale d'HiperLAN type 1 .....	9
1.3	L'organisation générale d'HiperLAN type 2 .....	10
1.4	Réseau 802.11 en mode infrastructure .....	11
1.5	Réseau 802.11 en mode Ad Hoc .....	11
1.6	Changer de fréquence régulièrement pour réduire l'impact de certaines interférences .....	12
1.7	Etaler le spectre pour réduire l'impact de certaines interférences.....	13
1.8	Le backoff et le defering dans le mode DCF .....	15
1.9	Le problème des nœuds cachés .....	15
1.10	L'échange RTS/CTS .....	16
1.11	Classification des différents protocoles de routage des réseaux Ad hoc.....	21
2.1	Validation de cache en utilisant les TTL.....	31
2.2	L'invalidation par la stratégie de contrat (Lease).....	33
2.3	Réduction de la latence par la technique des caches.....	37
4.1	Le comportement de cache recevant une requête pour un document.....	60
4.2	Changement de la popularité des documents en fonction de rang .....	62
4.3	Distribution des tailles de documents .....	63
4.4	Distribution des TTLs de documents .....	63
4.5	La variation de taux de succès en fonction de l'âge moyen maximum .....	64
4.6	Taux de succès des politiques FSDV, LFU et LRU par rapport à la taille du cache .....	65
4.7	Taux de succès des politiques FSDV, TDS et ZC par rapport à la taille du cache .....	66
4.8	Taux de succès en octets des politiques FSDV, LFU et LRU par rapport à la taille du cache .....	66
4.9	Taux de succès en octets des politiques FSDV, TDS et ZC par rapport à la taille du cache .....	67
4.10	Taux de succès en distances des politiques FSDV, LFU et LRU par rapport à la taille du cache .....	68
4.11	Taux de succès en distances des politiques FSDV, TDS et ZC par rapport à la taille du cache .....	69
4.12	La plage de valeurs des popularités de 6000 documents en fonction de paramètre de la loi Zipf .....	70
4.13	Taux de succès des politiques FSDV, LFU et LRU par rapport au paramètre de la loi Zipf.....	71
4.14	Taux de succès des politiques FSDV, TDS et ZC par rapport au paramètre de la loi Zipf.....	71
4.15	Taux de succès en octets des politiques FSDV, LFU et LRU par rapport au paramètre de la loi Zipf.....	72
4.16	Taux de succès en octets des politiques FSDV, TDS et ZC par rapport au paramètre de la loi Zipf.....	73
4.17	Taux de succès en distances des politiques FSDV, LFU et LRU par rapport au paramètre de la loi Zipf .....	73
4.18	Taux de succès en distances des politiques FSDV, TDS et ZC par rapport au paramètre de la loi Zipf ....	74



## Abstract

Caching frequently accessed documents on the client side is an effective technique for improving performance in a mobile Ad Hoc network, such as the data availability, the user latency and the networks bandwidth. However, the effectiveness of this technique is strongly related to the used cache management mechanism in particular, the cache replacement policies. The objective of this memory is to propose a cache replacement policy which improves the performances of such networks in most cases. So we have implemented a policy called FSDV *Frequency, Size and Distance based Value*, which uses the frequency and the size of document like characteristics of traffic and the distance like characteristic of the Ad Hoc networks. The simulation results showed that FSDV caching policy achieves significant improvements in terms of the hit ratio and distance hit ratio in comparison with other caching policies.

**Keywords** : Mobile Ad hoc networks, Cache's utilization, Cache Replacement, Simulation, performance.

## Résumé

Cacher les documents fréquemment utilisés au niveau du client est une technique très importante pour améliorer les performances des réseaux mobiles Ad Hoc, telle que la disponibilité des données, la latence des utilisateurs ainsi que la bande passante des réseaux. Cependant, l'efficacité de cette technique est fortement liée au mécanisme de gestion de cache utilisé, en particulier les politiques de remplacement de caches. L'objectif de ce mémoire de Magistère est de proposer une politique de remplacement de cache qui améliore les performances de ces réseaux dans la plupart des cas. Pour cela, on a mis en œuvre une politique appelée FSDV *Frequency, Size and Distance based Value*, qui utilise la fréquence et la taille des documents comme caractéristiques du trafic et la distance comme caractéristique des réseaux Ad Hoc. Les résultats de la simulation ont montré que FSDV donne de meilleures performances en terme de taux de succès de requêtes et de taux de succès en distances en comparaison avec d'autres politiques de remplacement de caches.

**Mot clés** : Réseaux mobiles Ad Hoc, Utilisation de cache, Remplacement de cache, Simulation, performance.

# Introduction générale

Nous pouvons commencer par une première question fondamentale : qu'est ce qu'un réseau et à quoi sert-il ? Nous pourrions définir un réseau informatique comme une infrastructure permettant le partage d'information de façon électronique et transparente pour l'utilisateur final. Les réseaux informatiques ont connu un tel développement depuis quelques dizaines d'années, ils se sont intégrés dans notre vie quotidienne, entraînant ainsi une très forte mutation de nos habitudes. Le premier réseau ne reposant sur aucune liaison filaire (PRNet, *Packet Radio Network*) a été construit par le DARPA (*Defence Advanced Research Projects Agency*) en 1987 [2]. Naturellement, les protocoles radio de l'époque étant limités, la bande passante disponible était faible. Cependant, il constitue l'une des bases des réseaux Ad Hoc actuels.

Les réseaux Ad Hoc [30] représentent l'extension des réseaux radio classiques : ils représentent littéralement des réseaux à usage spécifique. Ils sont souvent appelés les réseaux mobiles Ad Hoc en anglais *Mobile Ad Hoc Networks* ou simplement MANET. Ces types de réseaux doivent fonctionner d'une façon autonome, sans configuration, sans intervention humaine et sans infrastructure dédiée. Les terminaux utilisent des liaisons sans fil, ne requérant aucun câblage. Ainsi, les réseaux classiques voient leur utilisation entièrement mutée : alors qu'auparavant, toutes les fonctions intelligentes remplies par le réseau étaient concentrées dans des équipements dédiés, les routeurs, un réseau Ad Hoc doit collaborer et se répartir les tâches, afin de fournir un service réseau autonome et performant d'une façon totalement distribuée.

Cependant, les contraintes posées dans le contexte particulier des réseaux mobiles Ad Hoc sont différentes, et surtout plus complexes que celles rencontrées dans le monde filaire. C'est pourquoi, il est difficile de garantir une qualité de service à une application temps réel dans un réseau Ad Hoc, car il faut prendre en considération les spécificités de ces réseaux comme la bande passante limitée, l'absence d'infrastructure et la topologie dynamique. Ainsi,

de nombreux problèmes doivent être résolus afin de pouvoir profiter pleinement de ses avantages : routage efficace, gestion de la mobilité, la qualité de service ainsi que les mécanismes de gestion de caches.

L'accessibilité des données constitue l'une des questions la plus importante dans le contexte des réseaux mobiles Ad Hoc, puisque l'ultime but d'utilisation de tels réseaux, est de permettre aux nœuds mobiles une bonne accessibilité aux données [95]. Une des techniques la plus efficace pour améliorer l'accessibilité des données est l'utilisation de caches. En particulier, l'utilisation d'un cache dans un réseau mobile Ad Hoc peut améliorer d'une manière significative les performances de ce dernier :

- Minimiser la latence moyenne perçue par les utilisateurs.
- Réduire la consommation d'énergie, ce qui peut améliorer la durée de vie des batteries des nœuds mobiles.
- Réduire la charge de travail sur les sources de données.
- Réduire la consommation de la bande passante du réseau.

Cependant cette solution, malgré ses avantages, pose un certain nombre de problèmes liés à la gestion des caches, comme la consistance des données cachées, les techniques de remplacement de caches, etc.

Dans ce travail, nous nous intéresserons aux politiques de remplacement de caches dans les réseaux mobiles Ad Hoc. Ces politiques de remplacement sont invoquées à l'arrivée d'un nouveau document et trouvant un cache plein, alors il sera question de libérer de l'espace dans le cache. Pour cela, en se basant sur les caractéristiques des documents, la politique de remplacement sélectionne un des documents pour le remplacement. Cependant, dans le contexte du Web, les documents présentent des caractéristiques très spécifiques, telles que la variété des tailles des documents et des temps de chargement des documents, ce qui rend les politiques de remplacement de cache traditionnelles moins performantes. En plus des caractéristiques des documents Web, les politiques de remplacement de caches conçues pour les réseaux mobiles Ad Hoc, doivent prendre en considération les caractéristiques spécifiques de ces réseaux, telle que la mobilité des nœuds et la contrainte d'énergie.

C'est dans cette optique que notre travail s'inscrit. Il s'agit de proposer une politique de remplacement de cache qui servira à améliorer les performances des réseaux mobiles Ad Hoc, en prenant en considération les caractéristiques de trafic Web ainsi que les caractéristiques spécifiques des réseaux mobiles Ad Hoc.

## Organisation du mémoire

Ce mémoire est organisé comme suit : Le premier chapitre présente une étude des réseaux mobiles Ad Hoc avec ses différentes caractéristiques, ainsi que les différentes normes des réseaux locaux sans fil, tel que le Bluetooth, la 802.11, HiperLAN 1 et 2. Parmi ces normes, nous avons prêté une attention particulière à la norme 802.11 établie par l'IEEE qui fonctionne en deux modes, le mode infrastructure le PCF (*Point Coordination Function*) adapté aux réseaux mobiles cellulaires, et le mode appelé Ad hoc le DCF (*Distributed Coordinated Function*) qui est très adapté aux réseaux mobiles Ad Hoc. Ensuite, nous avons exposé les différentes approches de routage des réseaux mobiles Ad Hoc et quelques protocoles de routage qui leurs sont dédiés.

Le deuxième chapitre présente un état de l'art sur les caches. Ainsi, nous présenterons une classification des caches selon plusieurs critères, leurs organisations internes, leurs localisations et leurs fonctions. Puis, nous présenterons le fonctionnement des caches en décrivant un certain nombre de concepts liés aux caches, ainsi que les métriques de performances des caches et l'apport d'utilisation de cache dans les réseaux mobiles Ad Hoc.

Le troisième chapitre présente une classification des politiques de remplacement de caches qui existent dans la littérature, en donnant pour chaque classe, ses avantages et ses inconvénients. Nous terminons par une proposition d'une politique de remplacement de cache, FSDV *Frequency, Size and Distance based Value* pour les réseaux mobiles Ad Hoc.

Le dernier chapitre commence par une présentation de quelques méthodes d'évaluation de performances, ainsi que quelques caractéristiques du trafic Web. Puis, nous décrivons le modèle de notre système et quelques caractéristiques du trafic. Nous terminons par la présentation des résultats de simulation de notre politique de remplacement de cache FSDV avec quatre autres politiques LFU, LRU, TDS, et ZC. Nous clôturons notre mémoire par une conclusion générale.

# Chapitre 1

---

## Etude des réseaux mobiles Ad Hoc

---

### 1.1 Introduction

L'avènement du calcul omniprésent et la prolifération des dispositifs de calcul portatifs ont suscité un réel intérêt, particulièrement dans les réseaux sans fil Ad hoc. Les réseaux mobiles Ad Hoc (ou MANETs, Mobile Ad hoc NETWORKs) sont des configurations de réseaux spécifiques constituées d'une collection de nœuds mobiles sans fil et qui forment un réseau temporaire sans aucune administration centralisée. Dans un tel environnement, il peut être nécessaire qu'un nœud mobile fasse appel à d'autres nœuds dans l'expédition d'un paquet à sa destination en raison de la portée de transmission limitée des interfaces réseau sans fil. Chaque nœud mobile fonctionne non seulement comme un hôte mais également en tant que routeur pour acheminer les paquets de deux nœuds communicants qui ne sont pas dans la portée de transmission directe l'un de l'autre. Les contraintes posées dans le contexte particulier des réseaux Ad Hoc sont différentes et surtout plus complexes que celles rencontrées dans le monde filaire. Vu le fait que les modèles filaires ne pouvaient être appliqués dans les réseaux Ad Hoc, d'autres mécanismes plus adaptés à ce genre de réseaux ont vu le jour.

## 1.2 Définition

Les réseaux mobiles Ad Hoc sont des systèmes répartis complexes, qui se composent de nœuds mobiles sans fil qui peuvent librement et dynamiquement s'auto-organiser. De cette façon, ils forment arbitrairement et temporairement une topologie de réseau Ad Hoc, permettant à des dispositifs de s'interconnecter dans une aire géographique avec aucune infrastructure préexistante [1].

## 1.3 Les domaines des réseaux mobiles Ad Hoc

Les réseaux Ad Hoc ont un très large potentiel dans un futur proche et l'intérêt que porte la recherche pour ce domaine s'explique donc très largement.

### 1.3.1 Domaine militaire

Le premier domaine d'application des réseaux Ad Hoc fut le domaine militaire. Dans les années 80, le DARPA (*Defense Advanced Research Projects Agency*) propose Packet Radio Network [2]. L'intérêt des militaires pour une telle technologie s'explique par le caractère particulièrement adapté de ces réseaux aux situations hostiles.

### 1.3.2 Situation de sinistres

Leur utilisation est également possible dans le cadre de sinistres, comme les tremblements de terre ou les incendies, pour permettre aux équipes de sauvetage de communiquer alors que les infrastructures de communication classiques sont détruites ou pour permettre aux survivants d'établir un réseau aidant à leurs localisations. De plus, la possibilité pour les communications d'emprunter plusieurs routes renforce la fiabilité d'un tel réseau ce qui est indispensable dans un tel contexte [3].

### 1.3.3 Réseaux locaux

Les réseaux Ad Hoc peuvent également servir pour former des réseaux locaux. En effet, la mise en place du réseau est plus simple, pas de câble à tirer dans le bâtiment d'où une économie intéressante pour une entreprise.

## 1.4 Les caractéristiques des réseaux Ad Hoc :

La nature des réseaux Ad Hoc due à la mobilité, l'autonomie des nœuds et principalement l'utilisation des transmissions de type radio sont la base de ce qui rend ce type

de réseaux particulièrement difficile à manier. Les principaux problèmes engendrés par les contraintes spécifiques aux réseaux Ad Hoc sont dressés ci dessous :

### **1.4.1 Bande passante**

Le débit des cartes sans fil est plus faible que le débit des cartes filaires, de plus, la bande passante est partagée entre les nœuds du réseau qui appartiennent au même voisinage. De ce fait la bande passante est moins importante, alors que le routage et la gestion de la mobilité génèrent davantage de flux de contrôle et de signalisation que dans une architecture de réseau filaire. Ces flux doivent être traités de façon prioritaire pour prendre en compte rapidement les modifications de topologie. La bande passante devient ainsi une ressource précieuse qu'il faut prendre soin d'économiser lors de la conception des protocoles de gestion d'un réseau Ad Hoc [4].

### **1.4.2 Portée radio limitée**

Portée radio est limitée en raison de plusieurs facteurs, notamment la puissance de signal qui diminue en fonction de la distance, Non seulement elle est rapidement atténuée avec celle ci, mais elle est également limitée par des réglementations très strictes. Un émetteur ne peut donc dépasser une certaine puissance à l'émission.

### **1.4.3 Interférences**

Les interférences peuvent être de natures diverses. Par exemple, des émetteurs travaillant à des fréquences trop proches peuvent interférer entre eux. L'environnement lui-même peut également produire des bruits parasites (certains équipements électriques, certains moteurs, ...) qui interfèrent avec les communications. Ces problèmes font que les taux d'erreurs de transmission dans les réseaux radio sont nettement plus élevés que dans les réseaux filaires.

### **1.4.4 Energie**

Les équipements mobiles disposent de batteries leurs garantissant l'autonomie pour une certaine durée, souvent limitée. Ces équipements intègrent des modes de gestion d'énergie, il est donc important que les protocoles mis en place dans les réseaux Ad Hoc prennent en compte ce problème[6].

### **1.4.5 Mobilité et changement de topologie**

La mobilité des nœuds peut avoir plusieurs effets. Un nœud peut rejoindre un réseau, changer de position et ainsi de voisinage enfin quitter le réseau. Ces déplacements permanents ont comme conséquence le changement fréquent de topologie, et une simple panne d'un routeur ou l'épuisement d'une batterie peut aussi engendrer le même effet. Les protocoles de routage Ad Hoc doivent réagir rapidement aux changements de topologie pour permettre aux paquets d'emprunter les chemins optimaux vers leurs destinations, et évitant surtout leurs pertes.

### **1.4.6 Sécurité**

Si les transmissions ne sont pas protégées, des nœuds malveillants peuvent espionner les communications radio. Un autre problème qui ne se situe pas tant au niveau du support physique, mais principalement dans le fait que tous les nœuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau. Les possibilités de s'insérer dans le réseau sont plus grandes, ces nœuds peuvent engendrer un trafic dans le seul but de consommer toute la bande passante, et d'empêcher les communications, ils peuvent même perturber le routage en émettant des paquets de contrôle déroutant les paquets ou engendrant des pertes. La détection d'une intrusion ou d'un déni de service est donc plus délicate et l'absence de centralisation pose un problème de remontée de l'information de détection d'intrusions [4] [6].

## **1.5 Les normes des réseaux locaux sans fil**

Plusieurs normes pour les réseaux locaux sans fil (WLANs, *Wireless Local Area Networks*) à portée limitée ont été développées (visant des usages à l'échelle du bureau ou du bâtiment). Leur arrivée a soulevé un engouement nouveau pour les réseaux radio multi-sauts, qui étaient jusqu'alors le domaine exclusif des militaires. Parmi ces normes, on peut citer: Bluetooth, HiperLAN (type 1 et type 2) et 802.11 avec ses différentes versions.

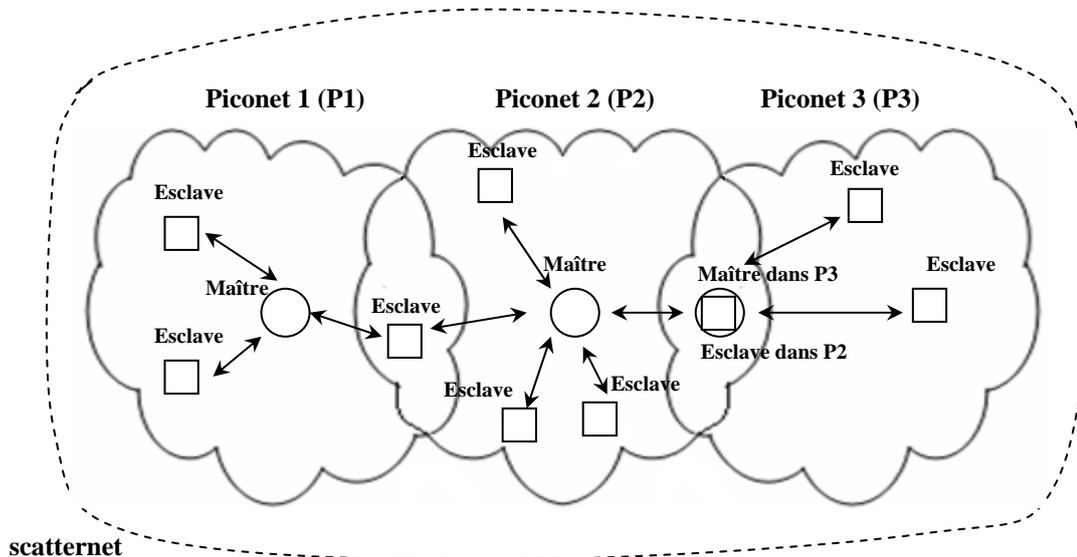
### **1.5.1 Le Bluetooth**

Bluetooth a plutôt pour objectif de faire disparaître les câbles entre les divers équipements numériques (périphériques d'ordinateur tels que clavier, imprimante, modem, ou encore appareil photo numérique, PDA, walkman, etc.). Les équipements bluetooth ont donc des portées et des débits assez modestes, ainsi qu'une consommation électrique en rapport.

C'est le groupe de travail Bluetooth SIG (*Bluetooth Special Interest Group*) qui a élaboré les spécifications de Bluetooth, ils l'ont développé pour permettre la réalisation de réseaux personnels (PAN : *Personal Area Network*). Ces réseaux doivent donc permettre des communications à courte portée, à des débits faibles ou moyens entre toute sorte d'équipements. Ils travaillent dans la bande ISM (Industrial, Scientific and Medical) des 2.4 GHz à des puissances leur permettant d'atteindre des portées allant du mètre à la centaine de mètres environ.

Les réseaux Bluetooth sont construits de manière centralisée. Un maître élu peut prendre en charge jusqu'à huit esclaves et forme ainsi un *piconet*. Dans un *piconet*, c'est le maître qui contrôle toutes les transmissions. Les esclaves ne peuvent émettre des paquets que s'ils y ont été invités par le maître. Ce dernier doit donc les interroger régulièrement pour savoir s'ils ont des données à envoyer (*polling*).

Plusieurs piconets peuvent être reliés afin de former une structure plus grande appelée *scatternet*. A cette fin, les mobiles peuvent quitter temporairement leur *piconet* pour aller s'attacher à un autre. La figure 1.1 donne un exemple de *scatternet*. Un esclave alterne entre les *piconets* 1 et 2 afin d'en assurer la liaison et un mobile esclave du *piconet* 2 peut être aussi maître dans le *piconet* 3 (le processus de dé-association est réversible, il est possible de quitter temporairement un *piconet* puis de le rejoindre à nouveau et de revenir à la situation initiale) [16].



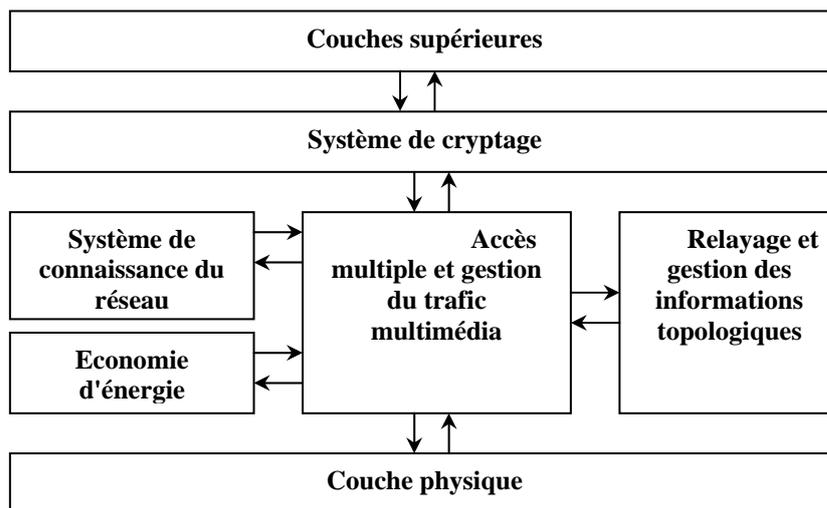
**Figure 1.1 :** Un *scatternet* Bluetooth

### 1.5.2 L' HiperLAN 1 [7]

High Performance Local Area Network type 1 (HiperLan 1) est un standard de l'European Technical Standard Institute (ETSI). Il décrit le fonctionnement d'équipements travaillant dans la bande des 5.15-5.30 GHz et permettant d'atteindre des débits de 23.5 Mbit/s sur une distance d'environ 50 mètres. L'architecture est totalement décentralisée. Il n'y a pas de notion de point d'accès mais les nœuds HiperLAN 1 peuvent cependant avoir des rôles de passerelles. Les caractéristiques les plus marquantes d'HiperLAN 1 sont :

- le mécanisme d'accès au médium évolué, qui gère les priorités. Il est possible d'obtenir des garanties de qualité de service, particulièrement utiles pour les flux multimédias par exemple ;
- la possibilité d'étendre le réseau au delà de la portée radio, par sauts successifs (des nœuds jouant le rôle d'intermédiaires entre ceux qui sont trop loin pour communiquer directement).

Les fonctionnalités d'HiperLAN 1 sont organisées comme présenté sur la figure 1.2. Nous n'entrerons pas dans les détails de cette norme complexe, mais nous pouvons remarquer néanmoins que le mécanisme d'accès au médium est au cœur du système. C'est lui qui permet en particulier la gestion des priorités. Le fonctionnement de ce mécanisme est particulièrement intéressant puisqu'il est prévu pour fonctionner dans un contexte Ad Hoc.

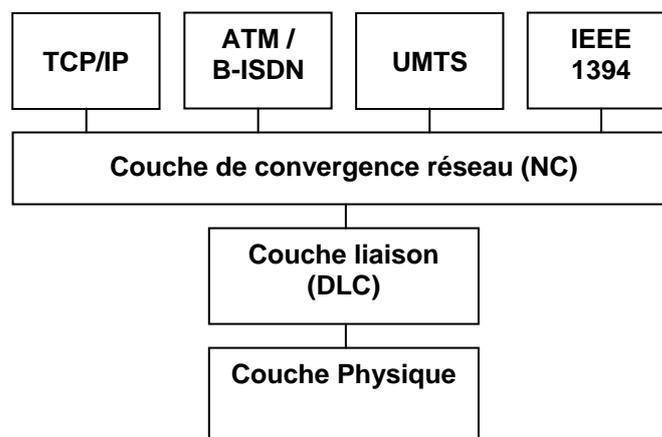


**Figure 1.2 :** L'organisation générale d'HiperLAN type 1

### 1.5.3 L'HiperLAN 2 [8]

L'organisation générale d'HiperLAN type 2 est présentée sur la figure 1.3. On remarque en premier lieu que la norme prévoit la compatibilité avec diverses technologies (TCP/IP bien sûr, mais aussi ATM, UMTS et l'IEEE 1394 connue aussi sous le nom de *Firewire*).

Contrairement au type 1, HiperLAN type 2 est basé sur une centralisation poussée. Les points d'accès sont d'ailleurs indifféremment appelés Access Points (AP) ou *Central Controller* (CC). Les points d'accès sont reliés entre eux par une infrastructure réseau filaire ou non-filaire, les mobiles s'attachent ensuite à ces points d'accès pour accéder aux ressources du réseau. HiperLAN 2 peut aussi fonctionner sans infrastructure fixe. Dans ce cas, un mobile est élu pour jouer le rôle de contrôleur central et les autres vont s'attacher à lui. Bien entendu, dans ce type de configuration il n'est pas question de réseau Ad Hoc au sens où le groupe MANET l'entend. Tout au plus, les mobiles pourront communiquer directement entre eux (un saut), ou alors par l'intermédiaire du contrôleur central (deux sauts). L'accès au médium reflète ces choix architecturaux, et c'est le contrôleur central qui décide de l'ordonnancement des communications dans la zone qu'il gère. Ces communications se font grâce à des trames de durée fixe (2 ms) qui sont elles-mêmes découpées en différents champs de longueur variable et qui portent soit les informations de contrôle du point d'accès, soit les données dans un sens ou dans l'autre. Il faut noter également que la couche physique d'HiperLAN type 2 est très semblable à la couche physique de 802.11a. De part son architecture très ciblée, HiperLAN 2 est donc peu adapté aux réseaux Ad Hoc,

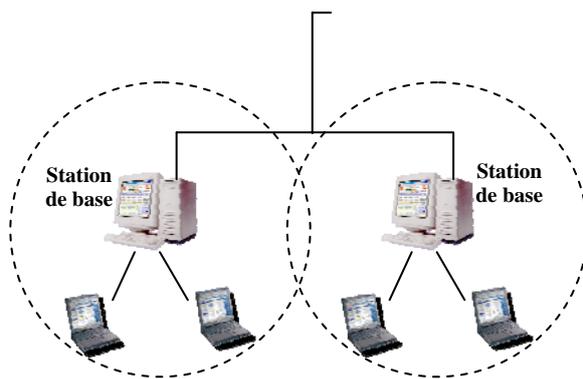


**Figure 1.3 :** L'organisation générale d'HiperLAN type 2

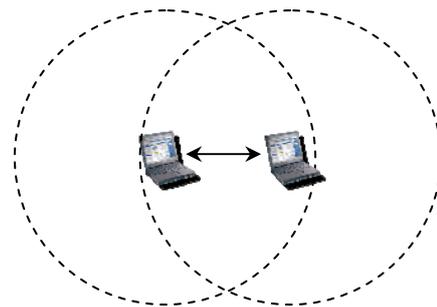
### 1.5.4 La 802.11

802.11 est une norme établie par l'IEEE. Elle décrit les couches physiques et MAC d'interfaces réseau radio et infra-rouge. Les débits possibles varient entre 1 et 54 Mbit/s suivant les techniques et les éventuelles extensions de la norme employées. Les portées prévues varient entre quelques dizaines et quelques centaines de mètres en fonction de la vitesse choisie et de l'environnement. Cette norme cible deux contextes d'utilisation :

- **Le mode "infrastructure"** (l'utilisation privilégiée de 802.11), où des stations de base reliées entre elles par un réseau filaire assurent la couverture d'une certaine zone et prennent en charge les mobiles dans leur voisinage (figure 1.4).
- **Le mode appelé Ad Hoc**, qui consiste en fait simplement à autoriser les communications entre deux mobiles à portée l'un de l'autre, sans intervention de stations ou d'autres mobiles extérieurs (figure 1.5).



**Figure 1.4 :** Réseau 802.11 en mode infrastructure



**Figure 1.5 :** Réseau 802.11 en mode Ad Hoc

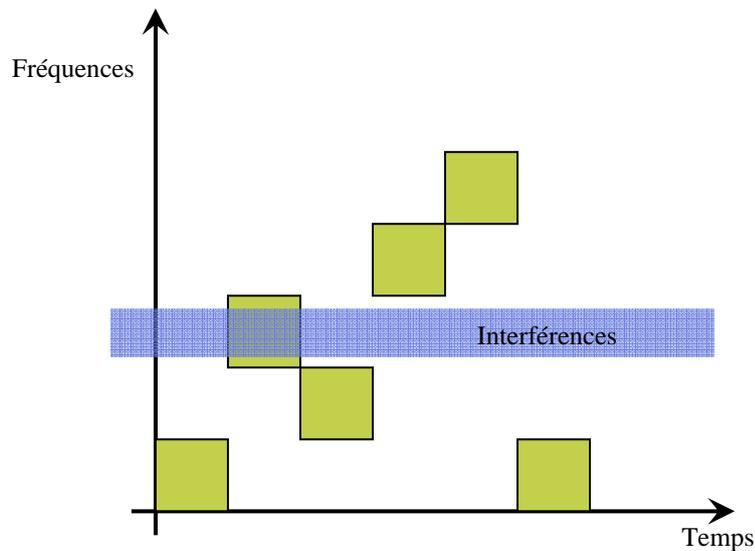
#### 1.5.4.1 La couche physique

Initialement, le standard IEEE 802.11 permet l'utilisation de trois couches physiques différentes (FHSS, DSSS et IR), auxquelles 802.11a a ajouté OFDM :

##### **FHSS : Frequency Hopping Spread Spectrum [29]**

La plupart des interférences nuisibles aux transmissions radio n'agissent en fait que sur des bandes de fréquence assez étroites. Si par malchance, de telles interférences ont lieu au moment où l'on transmet, alors notre signal sera fortement dégradé. Une technique pour protéger notre signal consiste à régulièrement changer de fréquence (figure 1.6). Bien sûr les paquets envoyés sur la bande perturbée seront affectés, mais ils ne représenteront plus qu'une minorité des transmissions et leur retransmission sera moins coûteuse. L'émetteur et le récepteur doivent connaître à l'avance le séquençement des sauts de fréquence, mais des

informations portées par les paquets permettent à un mobile s'attachant au réseau de savoir à partir d'un paquet qu'il reçoit où en est le déroulement de la séquence.



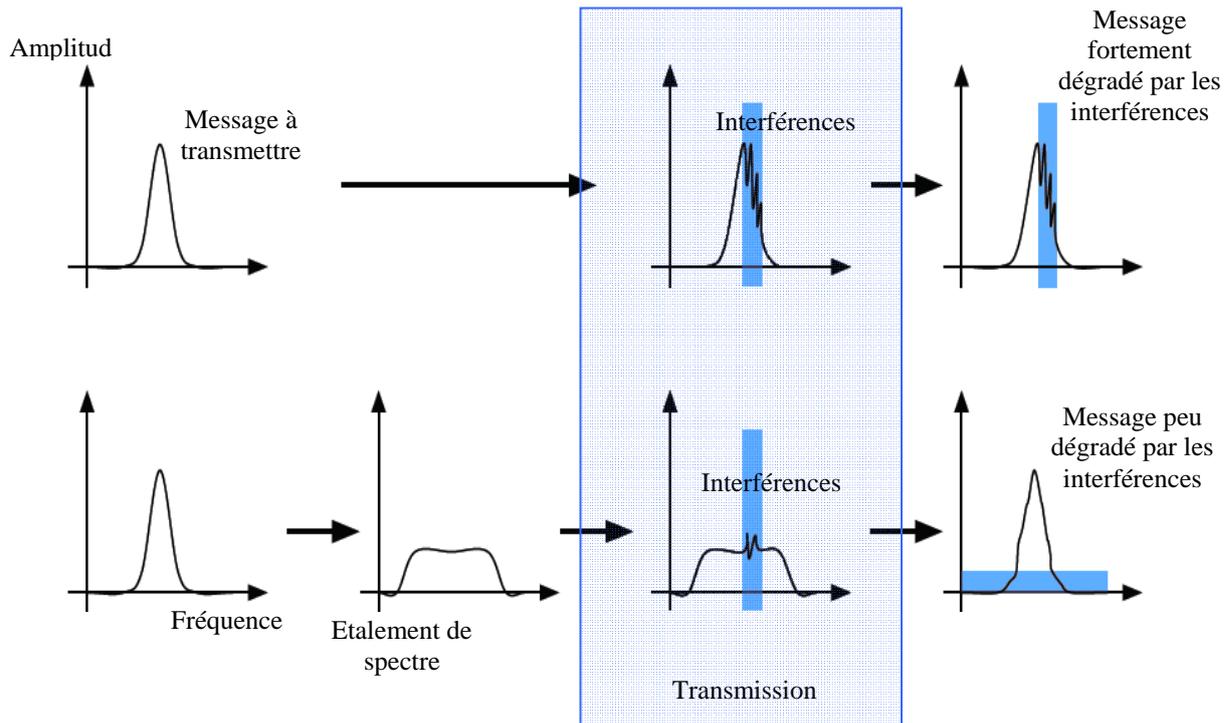
**Figure 1.6 :** Changer de fréquence régulièrement pour réduire l'impact de certaines interférences

### **DSSS : Direct Sequence Spread Spectrum [29]**

Toujours pour lutter contre les interférences importantes mais n'affectant que des plages de fréquences assez étroites, il existe la technique de l'étalement de spectre. Des manipulations sur le signal vont le faire occuper un spectre plus large. A la réception, une manipulation inverse est effectuée (figure 1.7). Cette technique est moins sensible aux interférences dues aux fréquences parasites à faible largeur spectrale.

### **IR : Infra Red**

Deux technologies de réseaux infrarouges existent : les réseaux à ondes infrarouges directes et les réseaux à ondes infrarouges diffusées. La communication s'effectue en plaçant les entités communicantes sur une même ligne de vue et à une distance maximale de 2 m. Les échanges se font entre 1 et 4 Mb/s. Les réseaux à ondes infrarouges diffusés offrent toujours une bande passante de 4 Mb/s, mais une antenne à ondes infrarouges diffusés couvre une zone qui peut atteindre jusqu'à 100 m<sup>2</sup>.



**Figure 1.7 :** Etaler le spectre pour réduire l'impact de certaines interférences

### OFDM : Orthogonal Frequency Division Multiplexing [29]

L'idée de base de l'OFDM réside dans le fait de répartir un canal binaire haut-débit en une multitude de canaux, lesquels étant modulés à bas-débits. Chacun de ces sous-canaux est modulé par une fréquence différente, l'espacement entre chaque fréquence restant constant. Ces fréquences constituent une base orthogonale et le spectre du signal OFDM présente une occupation optimale de la bande allouée.

On peut dire pour résumer succinctement cette méthode, qu'en présence de chemins multiples, à débit total équivalent, l'agrégation d'un certain nombre de canaux lents donne de meilleurs résultats qu'un seul canal très rapide.

#### 1.5.4.2 La couche MAC

L'IEEE 802.11 définit deux modes de fonctionnement de la sous-couche MAC : le mode PCF (*Point Coordination Function*) qui utilise la station de base pour contrôler l'activité de la cellule et le mode DCF (*Distributed Coordinated Function*) qui n'utilise aucune entité de gestion centralisée pour communiquer. Dans les réseaux Ad Hoc multi-sauts, il n'y a pas de stations de base fixes et c'est donc le mode DCF qui sera employé, le mode d'accès PCF n'étant pas applicable dans ces réseaux.

## Description du mode DCF

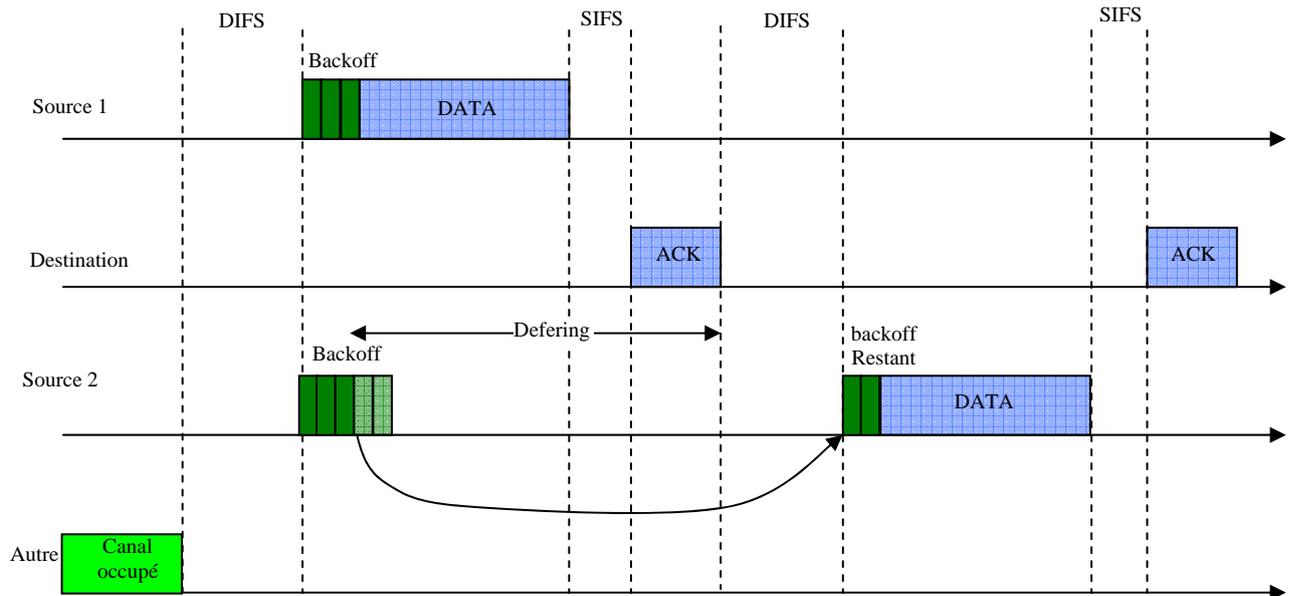
Dans le monde filaire, lorsqu'un émetteur envoie un signal sur le câble, il peut y lire en même temps la valeur qui y est effectivement présente. Si jamais la valeur lue est différente de celle que l'émetteur écrit, c'est qu'un autre émetteur est actif au même moment et qu'il y a collision.

Cette écoute du signal sur le câble au moment de l'émission est à la base de la méthode d'accès CSMA/CD (*Carrier Sense Multiple Access / Collision detection*) bien connue d'Ethernet. CSMA/CD permet de détecter une collision et le cas échéant de retransmettre le paquet après un temps d'attente aléatoire. A la différence de la couche MAC de 802.11 qui utilise des acquittements pour détecter ces collisions et permettre la retransmission des paquets qui ont été perdus (en l'absence d'acquiescement, l'émetteur sait qu'il doit retransmettre).

Avec Ethernet, l'idée est d'observer l'état du canal avant d'émettre. Si le canal est libre, alors nous pouvons envoyer notre trame (et si à ce moment-là nous détectons une collision, nous réémettons la trame un peu plus tard, après une attente de durée aléatoire). Mais avec 802.11 si plusieurs mobiles étaient en attente d'émission, ils détecteraient tous le canal libre et émettraient au même moment. Il y aurait collision au récepteur et il faudrait attendre que le délai imparti pour le retour de l'acquiescement soit écoulé pour s'en rendre compte, ceci pourrait être relativement long.

L'idée retenue pour 802.11 est donc, lorsque le canal devient libre, d'attendre une période de durée aléatoire supplémentaire appelée *backoff* avant d'émettre. Ce mécanisme s'applique lorsque le canal devient libre aussi bien après une de nos propres émissions qu'après toute autre émission. Ainsi, si plusieurs mobiles veulent émettre, il y a peu de chances pour qu'ils aient choisi la même durée. Celui qui a choisi le plus petit *backoff* va commencer à émettre, et les autres vont alors se rendre compte qu'il y a à nouveau de l'activité sur le canal et vont attendre.

Lorsque le canal devient libre, avant toute chose, il faut qu'il le reste pour une période DIFS (*DCF Inter-Frame Space*). Si le canal est resté libre durant toute cette période, alors les mobiles qui veulent émettre tirent un *backoff* aléatoire exprimé en un nombre de time slots d'une durée fixe. Une fois ce tirage effectué, tant que le canal reste libre, les mobiles décrémentent leur *backoff*. Dès que l'un d'eux a terminé, il émet. Les autres mobiles, dès qu'ils détectent le regain d'activité sur le canal stoppent la décrémentation de leurs *backoff* et entrent en période de *defering* Figure 1.8.

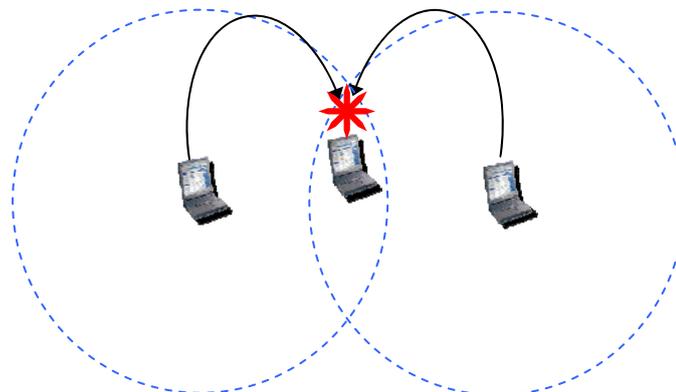


**Figure 1.8 :** Le *backoff* et le *defering* dans le mode DCF

Il faut noter que le temps de pause qui sépare un paquet de données de son acquittement est appelé SIFS (*Short Inter-Frame Space*) et qu'il est plus court que DIFS. Le mobile en période de *defering* ne pourra reprendre la décrémentation de son *backoff* que si le canal est à nouveau libre pendant DIFS. Le fait que SIFS soit plus court empêche que la décrémentation ne reprenne de manière inopportune entre les données et leur acquittement.

### RTS/CTS et le problème des nœuds cachés

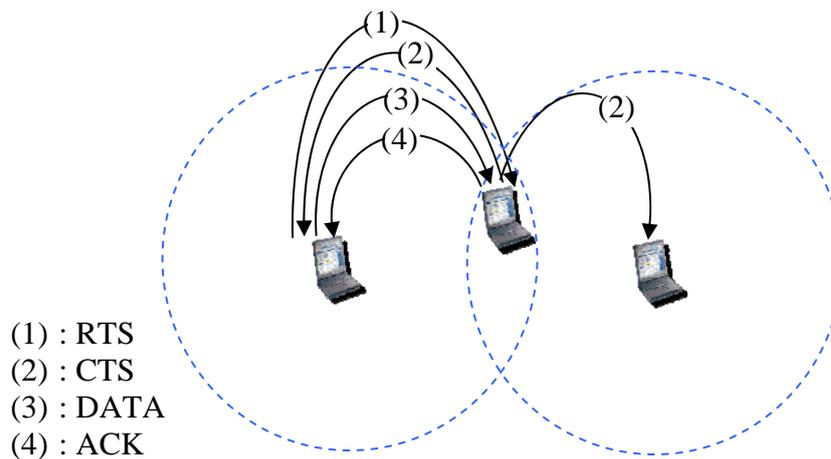
Le mécanisme CSMA/CA est insuffisant quand il s'agit du problème des nœuds cachés (figure 1.9) où deux émetteurs qui ne peuvent pas du tout s'entendre (en général à cause d'un obstacle) veulent atteindre un même récepteur.



**Figure 1.9 :** Le problème des nœuds cachés

Comme dans cette configuration un émetteur ne détecte jamais l'activité de l'autre, il croit que le canal est toujours libre et émet dès qu'il a des données disponibles. Les chances

de collisions à répétition au niveau du récepteur sont très élevées. 802.11 propose un mécanisme utilisant des paquets de contrôle appelés *Request To Send* (RTS) et *Clear To Send* (CTS). Un mobile qui veut émettre ne va plus directement envoyer son gros paquet de données, mais plutôt un petit paquet RTS pour lequel les chances de collision sont plus faibles. A ce paquet RTS, le destinataire va répondre par un petit paquet CTS qu'il diffuse à tout son voisinage comme illustré dans la figure 1.10.



**Figure 1.10** : L'échange RTS/CTS

Les paquets RTS et CTS contiennent des informations qui permettent de réserver le canal pour la durée de transmission des données qui vont suivre. Un mobile qui reçoit un CTS alors qu'il n'a pas envoyé (ni même détecté de RTS) sait que quelqu'un d'autre va émettre et doit donc attendre. Le mobile qui a envoyé le RTS sait, quand il reçoit le CTS correspondant, que le canal a été réservé pour lui et qu'il peut émettre. Au niveau des mobiles, la réservation du canal est implémentée grâce au *Network Allocation Vector* (NAV). Dans chaque nœud, le NAV indique pour combien de temps le canal est utilisé par quelqu'un d'autre.

### 1.5.4.3 Les différentes extensions de la norme 802.11

A l'origine, la bande de fréquence utilisée par cette norme était de 900MHz pour un débit allant jusqu'à 2 Mbit/s en radio. Des extensions ont été publiées depuis qui viennent lui ajouter des améliorations et des modes de fonctionnement plus performants. Les principales extensions sont les suivantes :

- **IEEE 802.11** (version de 1999) passe dans la bande ISM des 2.4 GHz, avec toujours des débits pouvant atteindre les 2 Mbit/s. Cette bande de fréquence est partagée avec d'autres types de réseaux locaux sans fil (Bluetooth en particulier) ainsi que par diverses autres applications [9].

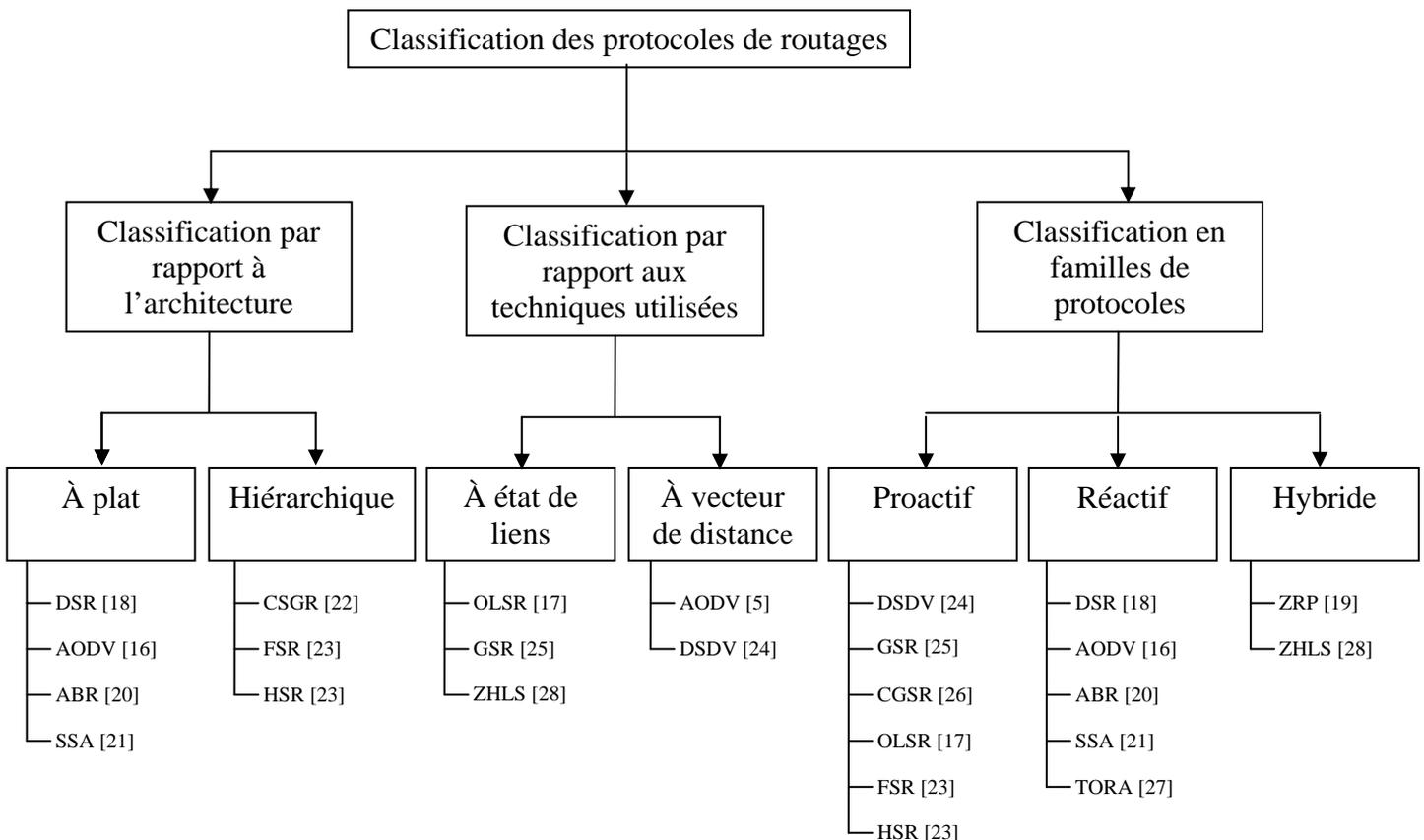
- **IEEE 802.11b** (commercialisée sous le label Wi-Fi) qui travaille toujours dans la bande des 2.4GHz, elle ajoute la description d'une couche physique améliorée (en plus de ceux déjà supportés), la couche DSSS modifiée (*High Rate DSSS*) avec une modulation CCK (*Complementary Code Keying*) et définit ainsi les vitesses de 5.5 et 11 Mbit/s [10].
- **IEEE 802.11a** (commercialisée sous le label Wi-Fi5) ajoute des modes encore plus rapides (jusqu'à 54 Mbit/s) en travaillant dans la bande des 5GHz, mais en utilisant des techniques OFDM (*Orthogonal Frequency Division Multiplexing*) d'accès au canal. [11].
- **IEEE 802.11g** qui travaille toujours dans la bande des 2.4GHz en utilise des techniques OFDM similaires 802.11a et DSSS pour garder la compatibilité avec les équipements 802.11b. Les débits possibles atteignent également les 54 Mbit/s [12].
- **IEEE 802.11e** définit les procédures de la couche MAC (*Medium Access Control*) pour supporter les exigences des applications en terme de qualité de services (QoS). ces procédures incluent le transport de la voix, audio et vidéo à travers les réseaux locaux sans fil (WLANs) IEEE 802.11 [13].
- **IEEE 802.11h** cherche à mieux gérer la puissance d'émission (TPC, *Transmit Power Control*) et la sélection dynamique des fréquences (DFS, *Dynamic Frequency Selection*) dans la bande des 5 GHz suivant si l'on est à l'intérieur ou à l'extérieur de bâtiments. L'objectif est d'être, à terme, mieux en accord avec les diverses législations (notamment européennes) [14].
- **IEEE 802.11i** définit les mécanismes de sécurités d'IEEE 802.11, qui inclue une définition de WEP (*Wired Equivalent Privacy*) avec le standard originelle, IEEE Std 802.11, 1999 Edition. Elle spécifie aussi comment IEEE 802.1X peut être utilisée par les LANs IEEE 802.11 pour effectuer l'authentification [15].
- **IEEE 802.16** (commercialisée sous le label WiMAX) spécifie l'interface d'accès sans file large bande (BWA, *Broadband Wireless Access*) des stations fixes supportant les services multimédias. Il est principalement conçu pour les WMANs (*Wireless Metropolitan Area Networks*), de plus pour pallier aux limitations de portée de IEEE 802.11. Il travaille dans la bande des 2 à 66 GHz et devrait assurer la couverture de réseau pour plusieurs Kilomètre carrés.

## 1.6 Le routage dans les réseaux Ad Hoc

Dans les réseaux Ad Hoc il peut arriver qu'un mobile veuille communiquer avec un autre qui n'est pas dans sa portée de communication directe, dans ce cas les nœuds intermédiaires entre la source et la destination doivent jouer le rôle de relais pour l'acheminement des paquets (principe de réseaux multi-sauts) : c'est ce que l'on appelle le routage.

### 1.6.1 Les approches de routage des réseaux mobiles Ad Hoc

Les protocoles de routage pour les réseaux Ad Hoc peuvent être classés suivant plusieurs critères. La figure 1.11 illustre ces différents critères. Le premier d'entre eux concerne l'architecture de routage qui définit pour chaque mobile le type de vision qu'il a du réseau et les rôles qui lui sont accordés, le deuxième concerne la technique utilisée pour le routage et le dernier concerne le comportement adopté par le protocole pour l'établissement de routes.



**Figure 1.11** : Classification des différents protocoles de routages de réseaux Ad hoc

### **1.6.1.1 L'architecture de routage**

Cette classification est basée sur la vision des protocoles de routage pour le réseau et les rôles qu'ils assignent aux mobiles pour l'établissement de routes, ces protocoles peuvent être vus comme étant hiérarchiques ou plat.

#### **Le routage hiérarchique**

Dans cette architecture, certains nœuds sont élus pour être responsable de fonctions particulières, supervision de différents niveaux de topologie. Avec cette topologie le réseau est divisé en cluster et un nœud est sélectionné à la tête de chaque cluster pour être responsable des transmissions entre clusters. L'utilisation de routage hiérarchique a plusieurs avantages, comme la réduction de la taille des tables de routage, une meilleure scalability et moins de consommation d'énergie. Toutefois dans ce type de protocole, les nœuds superviseurs supportent la majeure partie de la charge du routage et doivent avoir donc de meilleures capacités ce qui n'est pas toujours offert par les environnements Ad Hoc distribués.

#### **Le routage à plat**

Les protocoles de routage plat considèrent que tous les nœuds sont égaux et dans le même niveau, la décision d'un nœud de router un paquet vers un autre nœud ne dépend que de sa position dans le réseau et qui peut changer dans le temps. L'avantage majeur de l'approche de routage à plat est de fournir plus de routes possibles, sans aucune contrainte de nœud critique. Mais une congestion peut avoir lieu dû aux grands nombres de messages de contrôles qui sont transmis entre tous les nœuds.

### **1.6.1.2 Les techniques utilisées pour le routage**

Les protocoles de routage des réseaux mobiles Ad Hoc peuvent être répartis en deux grandes classes selon la technique utilisée, on trouve les protocoles de routage à état de lien et ceux à vecteur de distance.

#### **Les protocoles à état de lien**

Dans les protocoles utilisant cette technique, chaque nœud tient à jour une base de données décrivant la topologie entière du réseau, c'est à dire l'ensemble des nœuds, et leur interconnexion. Pour calculer la distance qui le sépare d'une destination, chaque nœud construit un arbre dont il est la racine, et suit au fur et à mesure toutes les ramifications du réseau tant qu'il n'a pas atteint la destination finale. Un des avantages de ce type de protocole, est leur capacité à pouvoir facilement trouver des routes alternatives lorsqu'un lien est rompu. Il est même possible d'utiliser simultanément plusieurs routes vers une même destination,

augmentant ainsi la répartition de la charge et la tolérance aux pannes dans le réseau. En contre partie, si le réseau est étendu, la quantité d'informations à stocker et diffuser peut devenir considérable.

### **Les protocoles à vecteur de distance**

Plutôt que de maintenir toute une base de données décrivant la topologie entière du réseau (ce qui peut s'avérer extrêmement lourd), cette technique ne conserve dans chaque nœud que la liste des nœuds de destination, et d'associer pour chacun l'interface locale ayant la meilleure distance permettant de l'atteindre. Cette distance est un coût estimé par chaque nœud à partir de messages envoyés par les nœuds voisins.

Un des inconvénients de cette technique est qu'il est plus difficile de conserver plusieurs routes alternatives au cas où celle qui est privilégiée serait rompue (on ne dispose que de l'identité du prochain nœud à atteindre, et on ne sait pas si la suite de la nouvelle route est indépendante de celle qui a été rompue).

#### **1.6.1.3 Les familles de protocoles de routage**

Dans les travaux menés à l'IETF, plusieurs familles de protocoles se sont rapidement dégagées. Chaque protocole peut ainsi être classifié en tant que proactif, réactif, ou hybride.

#### **Les protocoles proactifs**

Un grand nombre de protocoles incluent un comportement périodique, ces protocoles effectuent régulièrement quelques opérations en utilisant des messages de contrôles périodiques durant toute la durée de vie de réseau. L'idée derrière ce comportement est le maintien continu de routes à travers le réseau, de sorte que quand un paquet doit être expédié, la route soit déjà connue et peut être immédiatement utilisée. Ces types de protocoles sont connus sous le nom de protocoles proactifs ou table-driven.

Dans le contexte des réseaux Ad Hoc les nœuds peuvent apparaître ou disparaître de manière aléatoire et la topologie même du réseau peut changer ; cela signifie qu'il va falloir un échange continu d'informations pour que chaque nœud ait une image à jour du réseau. Les tables sont donc maintenues grâce à des paquets de contrôle, et il est possible d'y trouver directement et à tout moment un chemin vers les destinations connues en fonction de divers critères. L'avantage premier de ce type de protocole est d'avoir les routes immédiatement disponibles quand les applications en ont besoin, mais cela se fait au coût d'échanges réguliers de messages (consommation de bande passante) qui ne sont certainement pas tous nécessaires (seules certaines routes seront utilisées par les applications en général).

## **Les protocoles réactifs**

L'alternative à l'approche précédente est l'approche réactive, dans cette famille de protocoles le principe est de ne rien faire tant qu'une application ne demande pas explicitement d'envoyer un paquet vers un nœud distant. Cela permet d'économiser de la bande passante et de l'énergie. Lorsqu'un paquet doit être envoyé, le protocole de routage va rechercher un chemin jusqu'à la destination. Une fois ce chemin trouvé, il est inscrit dans la table de routage et peut être utilisé. En général, cette recherche se fait par inondation (un paquet de recherche de route est transmis de proche en proche dans tout ou partie du réseau). L'avantage majeur de cette méthode est qu'elle ne génère du trafic de contrôle que lorsqu'il est nécessaire. En contreparties l'inondation est un mécanisme coûteux qui va faire intervenir tous les nœuds du réseau en très peu de temps et qu'il va y avoir un délai à l'établissement des routes.

## **Les protocoles hybrides**

Les protocoles hybrides combinent les approches réactive et proactive. Le principe est de connaître notre voisinage de manière proactive jusqu'à une certaine distance (par exemple trois ou quatre sauts) qui définit la zone, et si jamais une application cherche à envoyer quelque chose à un nœud qui n'est pas dans cette zone, elle effectue une recherche réactive à l'extérieur. Avec ce système, on dispose immédiatement des routes dans notre voisinage proche, et lorsque la recherche doit être étendue plus loin, elle en est optimisée (un nœud qui reçoit un paquet de recherche de route réactive va tout de suite savoir si la destination est dans son propre voisinage. Si c'est le cas, il va pouvoir répondre, et sinon il va propager de manière optimisée la demande hors de sa zone proactive). Selon le type de trafic et les routes demandées, ce type de protocole hybride peut cependant combiner les désavantages des deux méthodes : échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un nœud éloigné.

## **1.6.2 Présentation de quelques protocoles de routage de réseaux Ad Hoc**

Dans ce qui suit nous allons présenter quelques protocoles de routage des réseaux mobiles Ad Hoc, à savoir AODV, OLSR, DSR et ZRP.

### **1.6.2.1 Ad Hoc On-demand Distance Vector (AODV) [5]**

AODV est un protocole basé sur le principe des vecteurs de distance et appartient à la famille des protocoles réactifs. En s'appuyant sur le principe des numéros de séquences AODV a su éviter le problème des boucles et le comptage à l'infini connu des protocoles à

vecteur de distances classique. Ces numéros de séquences inclus dans les messages de contrôles lui permettant également de maintenir la consistance des informations de routage dont il dispose. AODV effectue les découvertes de route à la demande en diffusant un paquet RREQ (Route REQuest) qui sera acheminé d'un nœud à un nœud. En réponse à ce paquet, la destination ou un nœud intermédiaire ayant une route valide à la destination, émet un paquet RREP (Route REPlY) suivant le chemin inverse de paquet de la requête jusqu'à la source (c'est pour cela qu'AODV n'utilise que les routes symétrique). Si le nœud destinataire reçoit d'autres paquets RREQ pour la même requête alors le nombre de sauts sera pris en compte pour décider de répondre à ce paquet ou de le supprimer. AODV stocke les routes découvertes dans chaque nœud traversé par la requête RREP et maintient cette route de façon distribuée en utilisant des procédures de réparation locale de la route à son point de cassure. Par conséquent, il est capable de s'adapter aux changements de topologie du réseau en trouvant très rapidement un chemin alternatif sans devoir reconstruire la route dans sa totalité.

### 1.6.2.2 Optimized Link State Routing (OLSR) [17]

OLSR est un protocole proactif et Comme son nom l'indique, c'est un protocole à état de lien optimisé ; il obtient aussi des routes de plus court chemin. Alors que dans un protocole à état de lien, chaque nœud déclare ses liens directs avec ses voisins à tout le réseau, dans le cas d'OLSR, les nœuds ne déclarent qu'une sous-partie de leur voisinage grâce à la technique des relais multipoints. En un nœud donné, cette technique consiste essentiellement à ignorer un ensemble de liens et de voisins directs, et de ne garder qu'un sous-ensemble des ces voisins considérés comme pertinent. Il est choisi de façon à pouvoir atteindre tout le voisinage à deux sauts (tous les voisins des voisins), cet ensemble est appelé l'ensemble des relais multipoints MPR (Multipoint Relay). Ces relais multipoints sont utilisés pour diminuer le trafic dû à la diffusion des messages de contrôle dans le réseau en appliquant la règle suivante: un nœud retransmet un message si et seulement si il ne l'avait pas déjà reçu, et il vient de le recevoir d'un nœud dont il est un relais multipoint. Pour maintenir à jour toutes les informations nécessaires au choix des relais multipoints et le calcul de la table de routage, les nœuds OLSR ont besoin de s'échanger périodiquement deux type de message :

- Le message dit HELLO diffusé par chaque nœud, contenant la liste de ces voisins. Ces messages permettent à chacun de choisir son ensemble de relais multipoints.
- Le deuxième type est le message TC (Topology Control) diffusé périodiquement par chaque nœud et retransmet uniquement par les MPRs pour atteindre tout le réseau. Ce message contient la liste de ces voisins qui l'ont choisi comme un MPR.

Ces informations offrent une carte de réseau contenant tous les nœuds et un ensemble partiel des liens, mais suffisant pour la construction de table de routage par chacun des nœuds. Le routage des données s'effectue saut par saut sans l'intervention d'OLSR dont son rôle s'arrête à la mise à jour de la table de routage

### 1.6.2.3 Dynamic Source Routing (DSR) [18]

DSR est un protocole de routage réactif qui utilise une technique de routage par source dans laquelle les nœuds ne doivent pas nécessairement garder la trace de la route. Chaque paquet contient dans son en-tête la liste complète des adresses des nœuds à traverser vers la destination. Lors de la création des routes si un destinataire est dans le cache d'un nœud source, la route connue est utilisée. Sinon, une procédure de découverte de route est déclenchée par un message "Route Request". Le chemin vers la destination est créé dans le paquet de recherche de route. Chaque nœud qui reçoit ce paquet ajoute à la route préexistante dans ce paquet son propre adresse.

En cas de rupture d'un lien sur un trajet, le nœud situé en amont de cette rupture envoie à la source une indication concernant la rupture de lien. Les nœuds peuvent sur une base volontaire, garder dans leur cache les routes créées par les procédures de recherche de route. Ces routes permettent de répondre plus rapidement à d'autres requêtes de création de route. En cas de rupture de lien, il est possible à un nœud qui possède une route valide vers la destination de détourner par ce nouveau chemin, sauvant ainsi l'acheminement des paquets.

### 1.6.2.4 Zone Routing Protocol (ZRP) [19]

ZRP est un exemple de protocole de routage hybride (réactif/proactif). D'une part, il limite la portée de la procédure proactive seulement au voisinage local du nœud, et d'une autre part, la recherche dans tout le réseau, bien qu'elle soit globale, elle est exécutée de façon efficace en faisant intervenir une partie des nœuds de réseau, par opposition à interroger tout le réseau. ZRP divise le réseau en plusieurs zones de routage où une zone  $Z(k;n)$  pour un nœud  $n$  avec un rayon  $k$ , est définie comme l'ensemble des nœuds à une distance inférieure ou égale à  $k$  sauts de ce même nœud  $n$ . ZRP utilise deux protocoles totalement indépendants l'un à l'intérieur des zones et l'autre entre ces zones. Le premier protocole est l'*Intrazone Routing Protocol (IARP)*, qui opère à l'intérieur des zones. On peut avoir plusieurs protocoles proactifs, où les différentes zones peuvent fonctionner avec IARPs différents. Le deuxième protocole est l'*Interzone Routing Protocol (IERP)*, qui est un protocole réactif utilisé pour le routage entre les zones. Il est utile uniquement quand le nœud destinataire ne se trouve pas dans la zone de routage de nœud source.

Ce découpage du réseau offre plusieurs avantages, notamment le passage à l'échelle. Par exemple, à la réception d'une requête réactive, le nœud est capable d'indiquer immédiatement si cette destination recherchée appartient ou non à sa zone, et pourrait ainsi l'aiguiller vers d'autres zones sans avoir à déranger ses voisins. Hélas, ce protocole présente l'inconvénient de l'absence de coordination entre les nœuds, il en résulte que les zones se chevauchent et un nœud peut être à la fois membre d'une zone et nœud frontière de plusieurs zones.

## 1.7 Conclusion

Dans cette partie, un état de l'art sur les réseaux Ad Hoc avec ses différentes caractéristiques a été présenté, ainsi que les différentes normes des réseaux locaux sans fil, telle que le Bluetooth, la 802.11, HiperLAN 1 et 2. Parmi ces normes, nous avons prêté une attention particulière à la norme 802.11 établie par l'IEEE qui fonctionne en deux modes, le mode infrastructure le PCF (*Point Coordination Function*) adapté aux réseaux mobiles cellulaires, et le mode appelé Ad hoc le DCF (*Distributed Coordinated Function*) qui est très adapté aux réseaux mobiles Ad Hoc. Ensuite, nous avons exposé les différentes approches de routage des réseaux mobiles Ad Hoc et quelques protocoles de routage qui leur sont dédiés.

Le chapitre suivant présente un état de l'art sur les caches. Il commence par une classification des caches selon plusieurs critères, leurs organisations internes, leurs localisations et leurs fonctions. Ensuite, une présentation du fonctionnement des caches en décrivant un certain nombre de concepts liés aux caches, ainsi que les métriques de performance des caches et l'apport d'utilisation de cache dans les réseaux mobiles Ad Hoc.

## Chapitre 2

---

# Les caches et les réseaux mobiles Ad Hoc

---

### 2.1 Introduction

Le cache se réfère au mécanisme de stockage d'une information récemment ou couramment recherchée, pour une future réutilisation. Introduit par WILKES (1965) sous le nom de mémoire esclave (Slave Memory), l'antémémoire ou mémoire cache, ou encore cache, est une mémoire intermédiaire située entre le processeur et la mémoire centrale et dont le temps d'accès est inférieur à celui de la mémoire centrale. L'une des premières réalisations fut mise en œuvre sur les machines IBM 360/85 au cours des années 60, avec un cache de 16 kilo-octets. Un défaut de cache (un *miss* en anglais) peut se produire lorsqu'un document référencé n'est pas présent dans le cache, autrement c'est un succès (un *hit* en anglais). Pour qu'un cache soit bénéfique, il faut que le coût de stockage d'une information soit plus petit que le coût de sa recherche depuis son serveur d'origine.

Le mécanisme d'utilisation de cache a réussi à trouver sa place dans la plupart des systèmes et réseaux informatiques. Les processeurs d'ordinateur utilisent deux types de cache, les caches de données et les caches d'instructions. Les systèmes d'exploitation ont des caches (les buffers) pour les unités de disques. Les systèmes de fichiers distribués tels que le NFS et l'AFS se fondent fortement sur les caches pour de meilleures performances. Les routeurs cachent les routes récemment utilisées. Les serveurs du DNS (Domain Name System) cachent les correspondances hostname et adresse. Dans ce travail nous nous intéressons à l'utilisation des caches dans les réseaux mobiles Ad Hoc. Parmi les principales limites de ces réseaux, on trouve la latence des communications, la consommation d'énergie et la capacité mémoire. Ces caractéristiques peuvent être significativement améliorées par l'utilisation de cache.

## **2.2 Classification des caches**

Les caches peuvent être classifiés selon leurs organisations internes, leurs localisations et leurs fonctions.

### **2.2.1 Organisation interne des caches**

La conception d'un cache reste délicate, celle-ci est basée sur des critères d'efficacité et de performance élevés, beaucoup de paramètres dépendent de son environnement d'utilisation, du type d'application utilisée, et de son emplacement dans la hiérarchie mémoire.

#### **2.2.1.1 Le cache non bloquant**

L'intérêt majeur d'un cache est qu'il doit répondre rapidement à une requête. Les données d'un cache non bloquant continuent à être accédées, même si un défaut de cache est en cours de traitement.

#### **2.2.1.2 Le cache victime**

Dans le but d'augmenter le taux de succès, un cache victime peut être ajouté au cache principal. Celui-ci est utilisé comme second niveau de cache. Lors de la résolution d'un défaut de cache, le document remplacé est mis dans un petit cache associatif appelé cache victime. Quand un défaut de cache se produit dans le cache principal, la résolution de défaut commence par consulter le cache victime pour s'assurer qu'il ne contient pas le document recherché.

#### **2.2.1.3 Le cache en mode pipeline (pipelining)**

Le fonctionnement d'un cache peut être décomposé en plusieurs étapes : lecture et traitement de la référence, comparaison des tags, lecture/écriture des blocs contenant le document, traitement d'un défaut de cache; toutes ces étapes sont exécutées les unes à la suite des autres, chacune dépend des résultats de l'étape précédente. Par contre elles s'exécutent sur des parties de cache indépendantes, d'où l'intérêt du pipeline qui consiste en la mise en parallèle de certaines opérations indépendantes lors de traitement d'une requête.

### **2.2.2 Localisation des caches**

Un cache peut être placé dans différentes localisations, tout le long du chemin reliant le client et la source d'origine. Il peut être intégré au niveau d'un navigateur, ou bien au niveau d'un Proxy. Ce dernier est généralement mis en interaction avec un grand nombre de

clients. Et enfin, il peut être situé au niveau du serveur d'origine, pour cacher les documents populaires.

### **2.2.2.1 Le cache de navigateur**

Les navigateurs tirent un grand avantage d'avoir un cache intégré, lorsque le bouton précédent du navigateur est appuyé, la page qui sera affichée est lue à partir du cache. Généralement, au lieu de maintenir les documents à cacher dans la mémoire, les navigateurs les cachent en tant que fichiers temporaires sur le disque. Un cache de navigateur est limité à un seul utilisateur, ainsi, il obtient des hits seulement quand l'utilisateur revisite une page, à moins que le cache soit utilisé en mode partagé.

### **2.2.2.2 Le cache de Proxy**

À la différence des caches des navigateurs, les caches des Proxy servent beaucoup d'utilisateurs différents en même temps. Puisque beaucoup d'utilisateurs différents peuvent visiter le même document populaire. Plus le nombre d'utilisateur augmente, plus le taux de succès augmente [31]. Un cache de Proxy devrait être localisé pour maximiser le nombre de clients qui peuvent l'utiliser, mais il ne devrait pas être au bout d'un lien lent et ainsi congestionner le réseau.

### **2.2.2.3 Le cache de serveur d'origine**

Les caches des serveurs sont utilisés, pour accélérer les serveurs lents. Naturellement, l'accélération est accomplie simplement en cachant les réponses du serveur. Quelques serveurs Web sont lents parce qu'ils produisent des pages dynamiquement. Par exemple, ils peuvent employer Java pour assembler une page HTML de différents composants stockés dans une base de données relationnelle. Si cette même page est livrée à tous les clients qui la visitent à partir du cache, alors ce dernier accélère le serveur. Si les pages sont personnalisées aux besoins du client à chaque accès, alors il n'accélérerait pas les choses de manière significative.

Les caches sont également souvent utilisés pour cacher les pages issues de décryptage des connexions HTTP/TLS. Un tel décryptage exige une quantité considérable de capacité de traitement. Plutôt que de mettre ce fardeau sur le serveur d'origine à chaque connexion, ce décryptage est fait une seule fois puis, à chaque connexion, la page est servie à partir du cache.

### **2.2.3 La fonction des caches**

Les caches peuvent être répartis, en trois catégories de fonctionnement qui sont, les caches CPU (Central Processing Unit), les caches Web, et les caches mobiles.

#### **2.2.3.1 Les caches CPU traditionnels**

Comme la vitesse de la CPU est très grande, l'algorithme de remplacement de cache doit être très efficace. Dans ce type de fonctionnement LRU (Least Recently Used) est très répondu, puisque son coût de traitement est très bas et de raisonnables performances sont obtenues [33], [34].

Les caractéristiques d'un cache CPU sont comme suit :

- 1) Toutes les données sont de taille égale.
- 2) Le cache est de très petite taille.
- 3) Les mises à jour des données sont effectuées dans le cache.
- 4) Les données sont souvent accédées séquentiellement.

#### **2.2.3.2 Les caches Web**

Contrairement à un cache CPU, la vitesse de communication est très lente pour les caches Web, ainsi, pour ce dernier, nous pouvons utiliser des algorithmes plus compliqués, pour plus de performances [35], [36].

Les caractéristiques d'un cache Web sont comme suit :

- 1) Les tailles de données sont distribuées entre de très petites à de très grandes valeurs.
- 2) Nous pouvons utiliser des disques pour le cache, ainsi la taille de cache peut être assez grande.
- 3) La mise à jour est exécutée sur les pages Web, ainsi un certain contenu de cache peut facilement devenir incohérent.
- 4) Périodiquement, au cours d'une même journée, on peut avoir une grande variation de la charge d'utilisation. Par exemple, la charge d'utilisation à 9 heures du matin, peut être très différente de celle de 5 PM, et même encore de celle de la nuit.

Le grand problème avec ce type de cache, est les tailles des documents qui peuvent être très différentes [36], [37]. En outre, en raison des caractéristiques citées ci-dessus, nous devons ajuster les modèles de gestion de cache traditionnels. Par exemple, pour la politique LRU, en plus de la récence, nous devons considérer la taille des documents.

#### **2.2.3.3 Les caches des systèmes mobiles**

Les systèmes mobiles ont généralement des petits caches, et peu de données à traiter, à comparer avec les deux cas cités ci-dessus [36, 38]. Ces systèmes, à la différence des deux cas

précédents, sont exposés à deux grands problèmes, qui sont : la mobilité, et l'énergie limitée des nœuds. En effet, les documents ayant été populaires dans une aire géographique donnée, et en raison de la mobilité, peuvent ne pas l'être dans une autre, ce qui crée la pollution de cache.

Le concepteur des algorithmes de remplacement de cache, pour les systèmes mobiles, doit prendre aussi en considération le facteur de consommation d'énergie, par exemple en faisant une estimation de coût en énergie nécessaire à la récupération d'un document à distance, si celui-ci doit être remplacé [39].

## 2.3 Fonctionnement de cache

Le cache est un espace mémoire et/ou disque sur une machine. Lorsque celle-ci reçoit une requête, le cache doit appliquer une stratégie pour savoir si le document doit être caché ou pas. En effet, il lui faut déterminer si le document peut servir à une future demande.

### 2.3.1 Support HTTP pour le cache

L'un des plus importants dispositifs de protocole HTTP utilisés dans le contexte des caches, est celui des requêtes conditionnelles, qui recherchent des documents, seulement si un certain nombre de conditions sont satisfaites. Les clients formulent les requêtes conditionnelles, en utilisant les métas informations du document en question. Les serveurs fournissent ces informations avec les documents dans *last-modified* et *ETag* de l'en-tête de leurs réponses. L'en-tête *Last-modified* contient la date et le temps de la dernière modification du document. L'en-tête *ETag*, qui signifie *Entity Tag*, est un identifiant unique pour une instance particulière d'un document.

Un autre groupe d'entêtes est utilisé, pour indiquer quels documents peuvent être cachés, pour combien de temps ils peuvent être cachés, et s'ils peuvent être partagés entre différents clients, etc. En particulier, les en-têtes *expires* et *age* qui sont utilisés par les caches pour expirer les documents cachés. L'en-tête complexe *cache-control* peut être utilisée que ce soit par les requêtes ou par les réponses pour contrôler le comportement du cache en différentes manières, peut-être la plus importante directive de *cache-control* est *no-cache*. Elle permet à une requête d'interdire de donner une réponse à partir du cache, et elle permet à une réponse d'empêcher n'importe quel cache de cacher la réponse.

### 2.3.2 Cachabilité des documents

L'objectif primaire d'un cache est de cacher certains documents, des réponses qu'il a reçues. Un document serait *cachable*, s'il peut être utilisé pour répondre à une future

requête. Il différencie donc entre deux familles de documents, les documents cachables et les documents non cachables. Tous les documents statiques sont considérés comme cachables à l'inverse des documents dynamiques. Les contenus dynamiques sont des pages où les données fournies par le serveur changent dans le temps ou en fonction du client (pages personnalisées). Par exemple, la réponse à un formulaire rempli par le client est personnalisée et unique, elle est non « cachable ». Un cache décide aussi si une réponse particulière est cachable en examinant différents composants de son en-tête. En particulier ce qui suit :

- Le code statut de la réponse
- La méthode de la requête
- Les directives de *Cache-control*
- Le validateur de la réponse
- Authentification de la requête

Ces différents facteurs agissent l'un sur l'autre d'une façon légèrement compliquée. Par exemple, quelques méthodes des requêtes sont non cachables à moins que ce soit permis par une directive de *Cache-control*. Quelques codes statut sont cachables par défaut, mais l'authentification et le *Cache-control* ont la priorité.

### **2.3.3 Défaut/succès de cache (cache hits/misses)**

Quand un cache reçoit une requête pour un document, il vérifie si le document a été déjà caché. Si c'est non, on dit que la requête a provoqué un défaut de cache (cache miss). Un cache miss se produit pour les documents qui n'ont jamais été référencés précédemment, pour ceux qui ne sont pas cachables, ou pour ceux qui ont été supprimés pour faire de la place pour les nouveaux.

Si le document est présent dans le cache, alors nous pourrions avoir un cache hit. Cependant, le cache doit d'abord décider si le document stocké est frais ou périmé. Un document caché est frais si son temps d'expiration n'a pas été écoulé ; autrement, il est périmé. Les documents frais sont donnés au client immédiatement, ils ne provoquent aucune latence et ne consomment aucune bande passante au serveur d'origine. Par contre, les documents périmés exigent la validation avec ce dernier.

### **2.3.4 Mécanismes de validation/invalidation de documents**

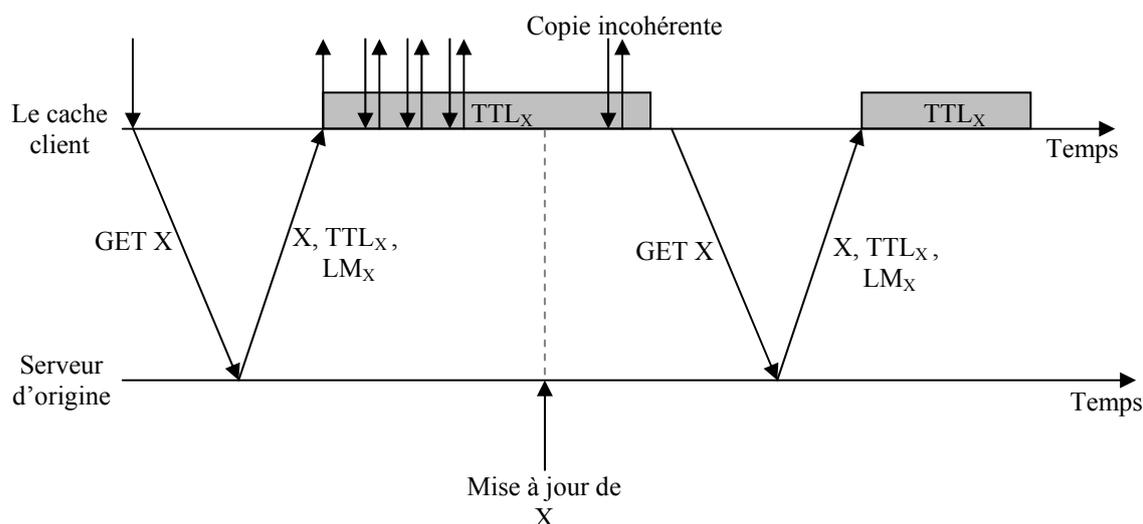
Le mécanisme de validation ou d'invalidation permet de maintenir la consistance des documents dans le cache. Dans le mécanisme de validation, chaque cache interroge périodiquement le serveur origine pour valider les documents périmés. Par contre dans celui

d'invalidation, les caches n'ont pas besoin d'interroger le serveur périodiquement pour la fraîcheur du document, le serveur distribue les documents modifiés aux caches intéressés.

### 2.3.4.1 Mécanismes de validation

Ce mécanisme est réalisé à l'aide de plusieurs stratégies dont on peut citer :

#### 2.3.4.1.1 Validation synchrone



**Figure 2.1** : Validation de cache en utilisant les TTL

Quant un cache reçoit un document cachable envoyé par le serveur d'origine, ce dernier fournit avec ce document une valeur TTL (Time To Live) explicite, en utilisant les champs d'en-têtes *expire* ou *max-age*, ainsi que le *time stamp* de la dernière mise à jour du document dans le champ d'en-tête *last-modified*.

Quand un document est caché, celui-ci peut être utilisé jusqu'à ce qu'il expire. Après cela, et au moment de référencement de ce même document, le cache client doit le valider avant de pouvoir l'utiliser encore. Cela se fait de manière synchrone puisque le client doit envoyer une requête conditionnelle (Par exemple une requête GET avec l'en-tête *if-modified-since* (IMS) contenant la date *last-modified*) comme illustré dans la figure 2.1 au serveur d'origine et attendre sa réponse.

#### 2.3.4.1.2 Validation asynchrone

L'alternative à la validation synchrone est la validation asynchrone qui peut être implémentée de deux manières :

- Avec des threads séparés au niveau du client : ces threads vérifient périodiquement l'âge des documents dans le cache et valident ceux qui sont expirés.

- Avec des déclencheurs de temps (time triggers) : un trigger est armé lorsqu'un document est chargé dans la cache ou validé, et qui sera déclenché lorsque ce document expire. Quand un trigger est déclenché, il validera le document expiré.

La validation asynchrone élimine la latence. Cependant, elle maintient ou même augmente l'échange de message et la consommation de la bande passante, en raison des validations inutiles. Le coût en bande passante est particulièrement considérable quand une validation nécessite le chargement d'une nouvelle version d'un document inutilement.

#### **2.3.4.1.3 Piggyback Cache Validation**

L'utilisation de l'en-tête des requêtes réduit le coût en bande passante de la validation asynchrone mais pas le nombre de messages. Pour réduire le nombre de messages, on peut regrouper les requêtes de validation destinées à un même serveur dans un seul message. Cette approche est appelée *Piggyback Cache Validation* (PCV) la validation de cache par ferroutage et qui a été proposée par Krishnamurthy et Wills [41].

Dans PCV, quand un client doit envoyer une requête HTTP pour un serveur, il vérifie s'il possède d'autres documents de même serveur, et si un quelconque de ces documents a expiré ou sur le point de l'être, le client ajoute ses requêtes de validation sous forme d'en-têtes IMS avec la requête originale.

#### **2.3.4.2 Mécanismes d'invalidation**

Plusieurs stratégies ont permis d'implémenter ce mécanisme dont on peut citer :

##### **2.3.4.2.1 La stratégie de contrat (lease)**

Cette stratégie a été proposée dans [42], dans laquelle le serveur maintient, pour un document donné la liste des clients qui l'ont utilisés, et aux quels, il fait la promesse de les notifier si ce document a été mis à jour pendant une période de temps donnée appelée lease (la durée de contrat). Par conséquent, un serveur ne supprime un client dans une liste que lorsque son contrat expire. De ce fait, une mise à jour peut être retardée par un client inaccessible pendant, au plus la durée de son contrat. La figure 2.2 illustre les interactions entre le cache client et le serveur d'origine avec la stratégie de contrat.

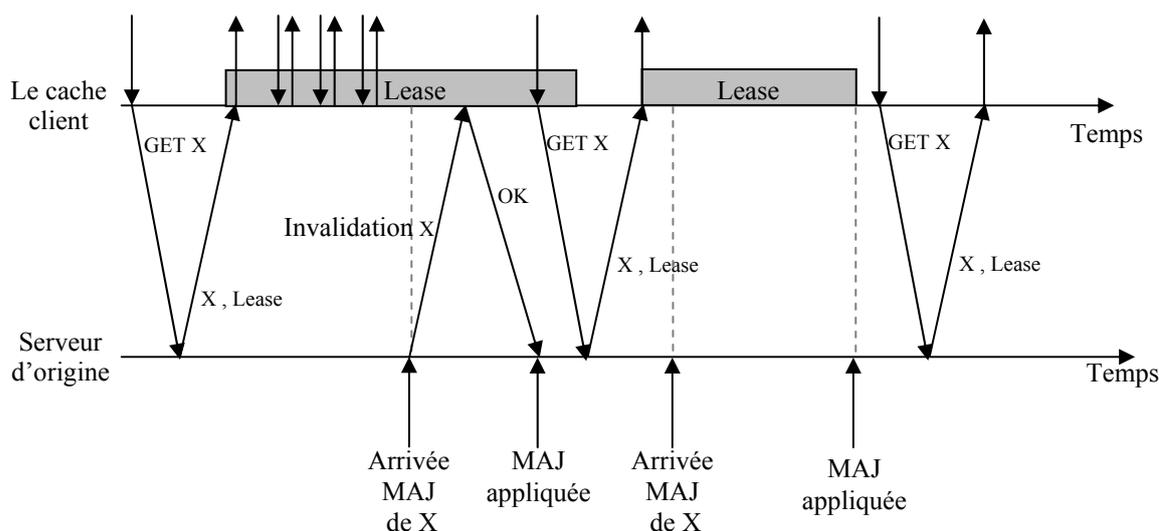


Figure 2.2 : L'invalidation par la stratégie de contrat (Lease)

### 2.3.4.2.2 La stratégie d'abonnement

Dans cette stratégie, les clients demandent explicitement au serveur, de leurs accorder des abonnements à recevoir les invalidations [44]. Les études faites dans [43], ont montré que les documents accédés seulement une fois (one timers) constituent 60 % de tous les documents de la trace étudiée. Ainsi, pour limiter l'invalidation uniquement aux documents qui bénéficieraient le plus, le serveur accorde les abonnements uniquement aux clients qui ont déjà une copie valide du document caché. En d'autres termes, le client peut demander un abonnement seulement avec en-tête *if-modified-since*, et le serveur le lui accorde si le document n'est pas modifié.

Ces abonnements peuvent être implémentés au niveau de la couche application ou la couche IP en utilisant le multicast. Au niveau de la couche application, le serveur d'origine maintient les listes de clients abonnés, et quand un document change, il invalide les clients de la liste correspondante en leur envoyant des messages d'invalidation unicast. Dans le cas de multicast, un groupe multicast est défini pour chaque document et le serveur envoie des messages d'invalidation aux adresses IP multicast correspondant aux documents changés. Les listes des clients sont stockées dans l'état d'adhésion multicast maintenu par les routeurs. L'adresse multicast pour l'invalidation, permet une diffusion efficace des mises à jour du serveur vers les caches clients.

### 2.3.5 Le préchargement (Prefetching) :

Prefetching est le processus de chargement de quelques données avant qu'elles soient réellement demandées, habituellement le temps qui s'écoule entre deux requêtes consécutives

d'un même utilisateur est très long. Il est donc normal de se demander si ce temps pourrait être utilisé pour anticiper le chargement du second document de sorte qu'il puisse être affiché avec peu ou aucun délai.

Le but de Prefetching est d'afficher les documents de Web sur l'écran de l'utilisateur plus rapidement. Par conséquent, la métrique principale qui devrait être utilisée dans l'évaluation des schémas de prefetching est la latence vue par l'utilisateur demandant un document. Certaines méthodes prechargent les documents dans le cache du navigateur, tandis qu'alternativement ou en plus, d'autres prechargent à un point intermédiaire tel que les caches proxy partagés [47].

Le Prefetching représente un compromis entre la latence et la bande passante. Par prédiction, un cache sélectionne des documents au prechargement, en admettant qu'un client les demandera. Une prédiction correcte a comme conséquence une réduction de latence ; cependant, une prédiction incorrecte a comme conséquence une perte de bande passante [48].

### **2.3.6 Politique de remplacement de cache**

Comme la capacité mémoire des caches est fixe, ils ne peuvent contenir tous les documents qui leurs sont soumis. Alors, quand cet espace mémoire se remplit le cache doit choisir un ensemble de documents à expulser pour faire de la place aux nouveaux. La politique de remplacement de cache détermine quels documents devraient être expulsés du cache. Le choix de cette politique doit se baser sur le but d'atteindre une meilleure utilisation des ressources disponibles, comme l'énergie, le disque, l'espace mémoire, et la bande passante de réseau. Puisque ce choix peut avoir un impact significatif sur le trafic global de réseau [45]. Parmi les principales caractéristiques des documents cachés qui peuvent influencer ce choix [46], on peut citer :

- La récence : est le temps écoulé depuis le dernier référencement d'un document ;
- La fréquence : est le nombre de référencements d'un document ;
- La taille : est la taille du document ;
- Coût d'acquisition : est le coût de recherche et de chargement d'un document depuis son serveur d'origine ;
- Temps de modification : est le temps écoulé depuis la dernière modification d'un document ;
- Temps d'expiration (TTL, Time To Live) : est la période de temps pendant laquelle le document restera valide (cohérent).

### **2.3.6.1 Les métriques des politiques de remplacement**

L'une des questions à laquelle il faut répondre quand on fait une évaluation d'une politique de remplacement est, quelle sont les métriques ? Il existe de multiples métriques intéressantes, Certaines d'entre elles sont discutées ci-dessous :

#### **Taux de succès (Hit rate)**

Il représente le pourcentage des documents demandés qui sont servis à partir du cache par rapport au nombre total de documents demandés, les petits documents sont généralement favorisés par les algorithmes de remplacement pour rester dans la cache afin d'augmenter le taux de succès.

#### **Taux de succès en octet (Byte hit rate)**

Il représente le pourcentage d'octets de données servies à partir du cache par rapport au nombre totale d'octets servis. Cette métrique favorise pour le remplacement les petits documents tandis qu'un algorithme de remplacement basé sur le taux de succès favorise les grands.

#### **Économie de bande passante**

Cette métrique vise à quantifier le degré de diminution du nombre d'octets transférés à travers le réseau. C'est une métrique très importante pour beaucoup d'ISPs (Internet Service Provider) pour réduire leurs coûts, qui est directement liée à la bande passante.

#### **Réduction de latence**

La latence se réfère au délai de transmission d'une information d'un point à l'autre. Cette métrique mesure le degré de réduction de cette latence pour une politique de remplacement de cache donnée.

#### **Performance de la CPU**

Une politique de remplacement de cache excessivement compliquée pourrait faire de la CPU un goulot d'étranglement. Il est également important de noter que les politiques de remplacement qui doivent examiner une grande quantité de données pourraient surcharger la mémoire et ainsi donc réduire sensiblement les performances de la CPU.

### **2.3.6.2 Les algorithmes de remplacement de caches**

Après avoir décrit quelques métriques, on présente maintenant un survole sur les politiques de remplacement de cache et qui seront représentées plus en détail dans le chapitre suivant. Généralement, ces algorithmes peuvent être regroupés en trois catégories comme c'est proposé par Aggarwal et al [36].

#### **Les algorithmes traditionnels et leurs extensions directes**

La première catégorie, inclut les algorithmes classiques de remplacement, dont on cite principalement LRU (Least Recently Used) et LFU (Least Frequently Used), comme leurs nom l'indiquent, LRU remplace le document le moins récemment utilisé, tandis que LFU remplace le moins fréquent, on a aussi l'algorithme FIFO (First In, First Out) qui est encore plus simple que LRU et LFU, les documents sont remplacés dans le même ordre dont ils ont été ajoutés au cache.

Parmi les extensions de LRU on a LRU-Threshold et LRU-Min proposées dans [35], EXP1 [49], HLRU [50], PSS (Pyramidal Selection Scheme) [36] et LRU-LSC [51]. LFU possède aussi plusieurs extensions dont on peut citer, LFU-Aging et LFU-DA proposées dans [52],  $\alpha$ -Aging [53], etc.

### **Les algorithmes à base de clefs**

Dans ce type d'algorithmes les documents sont ordonnés à base d'une clef primaire, et une clef secondaire est utilisée pour ordonner les documents ayant une même clef primaire, et si ces mêmes documents ont la même clef secondaire, on utilise une clef tertiaire, et ainsi de suite. Par exemple, la politique de HYPER-G proposée dans [55] utilise la fréquence d'accès comme clef primaire, la récence comme clef secondaire et finalement la taille comme clef tertiaire. On a aussi plusieurs autres politiques dans cette catégorie comme : swLFU (Server-Weighted LFU) [54], SLRU (Segmented LRU) [56] et LRU-Hot [57], etc.

### **Les algorithmes à base de fonction**

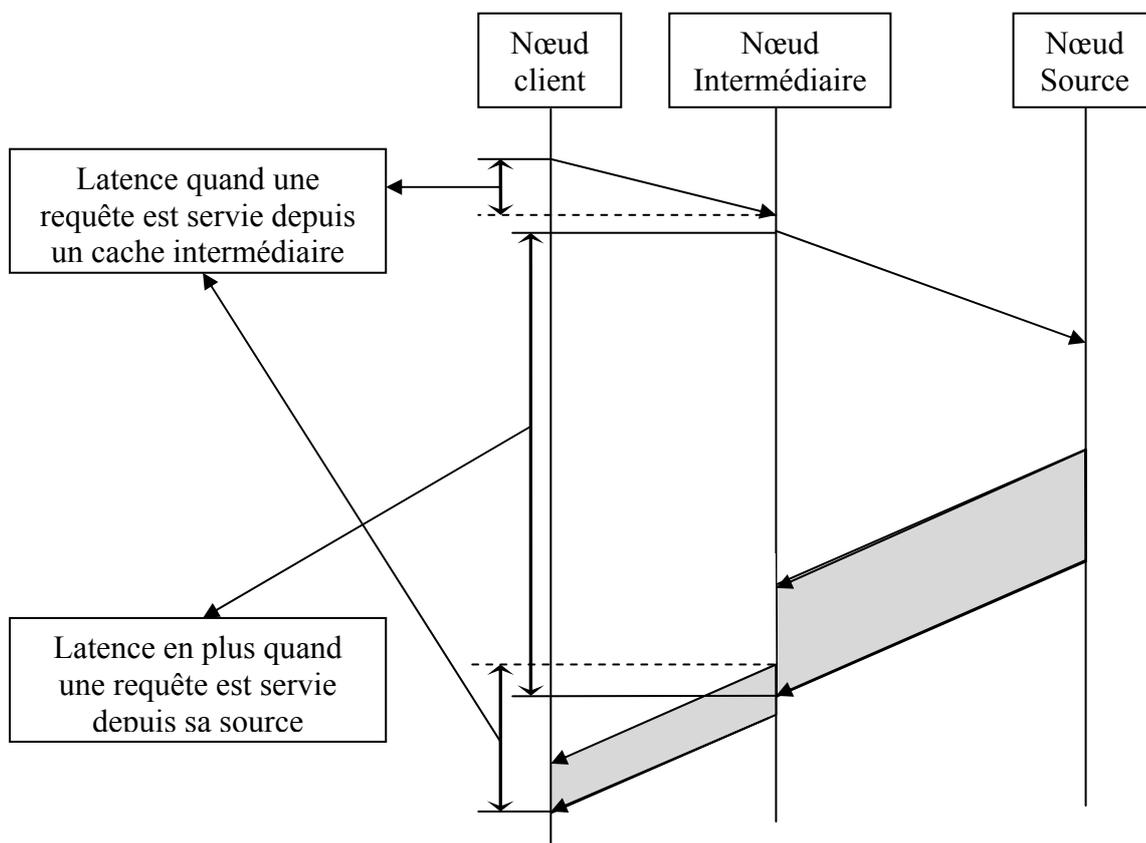
Ces politiques de remplacement ont eu recours à des fonctions dans l'intention d'inclure plusieurs facteurs tels que le temps depuis le dernier accès, le temps d'entrée dans le cache, le coût de transfert, le temps d'expiration (TTL), la taille et ainsi de suite. Par exemple, l'algorithme HYBRID décrit dans [58] utilise une fonction exponentielle basée sur la fréquence d'accès, la taille, la latence au serveur et la bande passante au serveur. On peu citer encore, GD (Greedy Dual)-Size [59] suivie par GDSF [52] et GD\* [60], TSP (Taylor Series Prediction) [61], LRV (Lowest Relative Value) [62], etc.

## **2.4 L'apport d'utilisation de cache dans les réseaux Ad Hoc**

L'utilisation des caches dans un réseau mobile Ad Hoc est une des techniques les plus efficaces pour améliorer ses performances, telle que l'accessibilité des données, la latence moyenne des utilisateurs, la consommation d'énergie et la consommation de la bande passante du réseau.

### 2.4.1 Réduction de la latence

Comme illustré dans la Figure 2.3 le gain en latence peut être considérable pour une requête servie depuis un nœud intermédiaire plutôt que depuis la source du document demandé.



**Figure 2.3 :** Réduction de la latence par la technique du cache

### 2.4.2 Sauvegarde de la bande passante

Un autre avantage pour l'utilisation des caches dans les réseaux mobiles Ad Hoc et dans les réseaux en général, est la réduction de la consommation de la bande passante. Avec cette réduction, l'avantage évident est non seulement la réduction des coûts globaux des transferts à travers le réseau, mais elle réduit également la charge de lien vers la source, et donc elle diminue indirectement la latence.

### **2.4. 3 Augmentation de l'accessibilité des données**

Vu le caractère de topologie dynamique des réseaux mobiles Ad Hoc qui entraîne de fréquentes déconnexions et d'éventuelles partitionnements de réseaux, une source de données peut devenir inaccessible, mais les requêtes qui lui sont destinées peuvent être satisfaites par les caches des autres nœuds accessibles.

## **2.5 Conclusion**

L'utilisation des caches dans les réseaux mobiles Ad Hoc s'est avérée être une technique très importante pour améliorer d'une manière significative les performances de ces réseaux. Néanmoins cette solution pose un certain nombre de problèmes liés à la gestion des caches, comme la consistance des données cachées et les techniques de remplacement de caches.

Nous allons présenter dans le chapitre suivant une classification des politiques de remplacement de caches, qui existent dans la littérature, en donnant pour chaque classe ses avantages et ses inconvénients. Nous terminons, par une proposition d'une politique de remplacement de cache, *FSDV Frequency, Size and Distance based Value* pour les réseaux mobiles Ad Hoc, basée sur une fonction. Cette fonction calcule une valeur pour chaque document en prenant en considération sa fréquence, sa taille, ainsi que la distance en nombre de sauts.

## Chapitre 3

---

# Les politiques de remplacement de caches

---

### 3.1 Introduction

Les caches représentent des mémoires ayant des capacités de stockage finies. Par conséquent, ils ne peuvent contenir qu'un nombre fini de documents. Lorsque l'espace libre devient insuffisant pour contenir un nouveau document, il est nécessaire de remplacer les documents jugés moins utiles, par les nouveaux documents, et cela, suivant une politique de remplacement de cache.

Plusieurs politiques de remplacement de caches ont été proposées, dans le but d'améliorer les performances des caches. Nous allons présenter dans le chapitre suivant une petite synthèse des politiques qui existent dans la littérature, en les classifiant en cinq classes et de donner pour chaque classe ses avantages et ses inconvénients. Nous terminons par proposer une politique de remplacement de cache pour les réseaux mobiles Ad Hoc.

## 3.2 Classification des politiques de remplacement de cache

Pour présenter les différentes propositions de politiques de remplacement de cache qui existent, d'une manière structurée, nous procédons d'abord à leurs classifications, selon les paramètres (caractéristiques) des documents, qui peuvent influencer le processus de remplacement. La plupart de ces paramètres sont décrits dans Krishnamurthy et Rexford. [63] On donne ci après un récapitulatif des paramètres importants :

- 1) **récence** : période (depuis) la dernière référence de document ;
- 2) **fréquence** : nombre d'accès à un document ;
- 3) **taille** : taille de document ;
- 4) **coût de chargement**: coût de chargement d'un document depuis son serveur d'origine;
- 5) **temps de modification** : période depuis la dernière modification ;
- 6) **temps d'expiration (TTL)** : est la période de temps pendant laquelle le document restera valide (cohérent). Quand cette période expire le document devient périmé et peut être remplacé immédiatement.

Ces facteurs peuvent être incorporés à la décision de remplacement. Dans la littérature, la plupart des propositions, utilisent les quatre premiers facteurs.

La récence et la fréquence sont des facteurs importants mais très différents. Par conséquent, les stratégies qui utilisent ces facteurs indépendamment, doivent être mises dans des classes séparées. De plus, certaines stratégies combinent la récence et la fréquence, donc elles doivent être traitées dans leur propre classe.

Certaines stratégies calculent une fonction spécifique incorporant différents facteurs. Dans la plupart du temps des poids sont utilisés et associés à chaque facteur, et donc une classe séparée est nécessaire pour ce genre de stratégies. Et enfin une dernière classe pour les stratégies dédiées aux environnements mobiles. Ces stratégies sont adaptées aux spécificités et caractéristiques de ces environnements. Donc cinq classes de stratégies de remplacement de caches sont mises en évidence, à savoir :

- Les stratégies de remplacement de cache basées sur la récence
- Les stratégies de remplacement de cache basées sur la fréquence
- Les stratégies de remplacement de cache basées sur la récence et la fréquence.
- Les stratégies de remplacement de cache basées sur les fonctions
- Les stratégies de remplacement de cache dédiées aux environnements mobiles.

### 3.2.1 Les stratégies de remplacement de cache basées sur la récence

Ces stratégies utilisent la récence comme facteur principal. La plupart d'entre elles sont plus ou moins des extensions de la stratégie LRU, qui exploite la localité temporelle des références perçues dans les traces de requêtes. La localité temporelle de référence caractérise la capacité de prévoir les futurs accès aux documents à partir des accès passés, elle suppose que les documents récemment accédés sont susceptibles d'être accédés encore à l'avenir.

#### 3.2.1.1 LRU

Cette stratégie remplace le document le moins récemment référencé. Elle est largement utilisée dans différents domaines (par exemple, gestion de buffer de base de données, pagination, les buffers de disque).

#### 3.2.1.2 LRU-Threshold [35]

Un document n'est pas caché lorsque sa taille excède un seuil (threshold) donné. Sinon, elle fonctionne comme LRU.

#### 3.2.1.3 SIZE [55]

Cette stratégie remplace le plus grand document. La stratégie LRU est appliquée aux documents de même taille.

#### 3.2.1.4 LRU-Min [35]

C'est une variante de la stratégie LRU qui essaye de minimiser le nombre de documents à remplacer. Soient  $Lo$  et  $T$  qui dénote, respectivement, une liste et un threshold.

- (1)  $T$  reçoit  $S$ , où  $S$  est la taille de document demandé.
- (2)  $Lo$  reçoit tous les documents qui ont une taille supérieure ou égale à  $T$ . ( $Lo$  peut être vide)
- (3) Appliquer LRU pour  $Lo$  jusqu'à ce que la liste soit vide ou l'espace libre de cache soit au moins  $T$ .
- (4) si l'espace libre de cache n'est pas au moins  $S$ , alors  $T$  reçoit  $T/2$  et aller à l'étape (2).

#### 3.2.1.5 EXP1 [49]

C'est une extension de LRU qui utilise la période de temps entre le temps courant et le temps de dernier accès pour peser l'importance d'un document. EXP1 choisit un document  $i$  si

$\frac{1}{\mu_i} = \min \left\{ \frac{1}{\mu_j} \right\}$  avec  $j = 1..N$  (ensemble de  $N$  documents).  $\mu_i$  est calculé à base des périodes successives entre les différent accès au document  $i$ .

### 3.2.1.6 HLRU [50]

HLRU introduit un schéma, pour supporter l'historique d'un certain nombre de références à un document spécifique. Soient  $r_1, r_2, \dots, r_n$  des requêtes à un document caché à des temps  $t_1, t_2, \dots, t_n$ . La fonction de l'historique est définie comme suit :

$$hist(x, h) = \begin{cases} t_i, & \text{si il y a exactement } h - 1 \text{ référence entre } t_i \text{ et } t_n, \\ x, & \text{si non.} \end{cases}$$

$hist(x, h)$  définit le temps des  $h$  dernières références d'un document caché  $x$ .

HLRU remplace le document ayant la valeur maximale de  $hist$ . Si il y a beaucoup de documents avec un  $hist = 0$ , alors LRU est utilisée pour le remplacement.

### 3.2.1.7 PSS (Pyramidal Selection Scheme) [36]

Cette stratégie fait une classification pyramidale des documents en fonction de leur taille. Tous les documents de la classe  $i$  ont une taille comprise entre  $2^{i-1}$  et  $2^i - 1$ . Chaque classe a une liste LRU séparée. À chaque fois qu'il y a un remplacement, les valeurs LRU des documents de chaque classe sont comparées. PSS choisit le document  $i$  si sa valeur  $S_i \times \Delta T_i$  est la plus grande parmi toutes les valeurs de ces documents.  $S_i$  et  $\Delta T_i$  sont respectivement, la taille de document  $i$  et le nombre d'accès au cache depuis le dernier accès au document  $i$ .

### 3.2.1.8 Avantages

- Elles considèrent la localité temporelle, comme facteur principal. et les traces empiriques des requêtes de Web présentent ce comportement de localité temporelle, ce qui est donc avantageux.
- Elles sont simples à mettre en œuvre. La plupart de ces stratégies utilisent une liste LRU. les nouveaux documents demandés sont insérés à la tête de la liste. quand il y a un hit le document est enlevé de sa position actuelle et inséré à la tête et le remplacement ce fait à la fin de la liste.

### 3.2.1.9 Inconvénients

- Les variantes LRU simples ne combinent pas la récence et la taille. Et comme les documents Web ont de très différente taille, cette dernière devrait être considérée à chaque remplacement.
- Elles ne considèrent pas la fréquence. Ceci a pu être un indicateur très important dans plusieurs environnements.

### 3.2.2 Les stratégies de remplacement de cache basées sur la fréquence

Ces stratégies utilisent la fréquence comme facteur principal. Elles tirent profit de la diversité des valeurs de popularité des documents Web, de ce fait, il en résulte différentes valeurs de fréquence. Donc elles calculent ces valeurs et les utilisent dans les futures décisions.

#### 3.2.2.1 LFU

LFU remplace le document le moins fréquemment demandé.

#### 3.2.2.2 LFU-Aging [52]

Avec LFU, les documents qui étaient très populaires, pendant une période de temps, peuvent demeurer dans la cache, même lorsqu'ils ne sont pas demandés pendant une longue autre période. Ceci est dû à la grande valeur de leur compteur de fréquence, c'est ce qu'on appelle la pollution de cache. Pour éviter cette pollution, une technique de Aging peut être introduite. Le LFU-Aging utilise donc un seuil, si la valeur moyenne de tous les compteurs de fréquence excède ce seuil, tous les compteurs de fréquence sont divisés par 2.

#### 3.2.2.3 LFU-DA [52]

Avec LFU-DA, quand un document  $i$  est référencé, alors sa valeur  $K_i$  est (re)calculée comme suit :

$$K_i = f_i + L .$$

Où  $f_i$  et  $L$  sont respectivement, la fréquence de document  $i$  et le facteur Aging qui est initialisé à zéro. LFU-DA choisit le document avec la plus petite valeur de  $K_i$ . La valeur de ce document est affectée à  $L$ .

#### 3.2.2.4 SWLFU (Server-Weighted LFU) [54]

Cette stratégie utilise pour chaque document  $i$  un poids de fréquence  $W_i$ , ce poids indique l'appréciation du serveur de ce document  $i$  pour qu'il soit caché. Par conséquent, le serveur peut influencer la cachabilité d'un document. La stratégie LRU est appliquée aux documents avec la même valeur  $W_i$ .

#### 3.2.2.5 Fréquence relative de Benhamida [64]

Cette stratégie utilise une fréquence relative  $F_i^r$  au lieu du nombre d'accès pour chaque document  $i$ . Elle est calculée comme suit :

$$F_i^r = F_i / (now - D_i) .$$

Où :

$F_i$  : est la fréquence de document  $i$  ;

$D_i$  : est la date d'entrée du document  $i$  dans le cache ;

*now* : est la date courante ;

### **3.2.2.6 Avantages**

Elles considèrent la fréquence d'accès, qui est un facteur très important dans les environnements statiques où la popularité des documents ne change pas beaucoup sur une période de temps spécifique.

### **3.2.2.7 Inconvénients**

- **Complexité** : Les stratégies basées sur LFU exigent une gestion plus complexe de cache. LFU peut être mise en œuvre, par exemple avec une file d'attente prioritaire.
- **Pollution de cache** : Les documents qui étaient très populaires pendant une période de temps peuvent demeurer dans la cache même lorsqu'ils ne sont pas demandés pendant une longue autre période. C'est pour ça qu'on fait appelle à la technique de Aging qui n'est rien d'autre qu'une technique basée sur la récence.
- **Valeurs similaire** : Beaucoup document peuvent avoir la même fréquence. Dans ce cas-ci, un autre facteur de comparaison est nécessaire.

## **3.2.3 Les stratégies de remplacement de cache basées sur la récence et la fréquence.**

Ces stratégies utilisent la récence et la fréquence et peut-être encore d'autres facteurs pour trouver le document à remplacer.

### **3.2.3.1 SLRU (Segmented LRU) [56]**

La stratégie SLRU divise la cache en deux segments : un segment non protégé et un segment protégé (réservé pour les documents populaires). Sur la première requête pour un document, celui-ci est inséré dans le segment non protégé. Si le document est référencé à nouveau c.à.d. un cache hit, alors le document est déplacé au segment protégé. Cette politique était destinée initialement aux caches disques [88]. Elle a été utilisée pour le Web [56] parce qu'elle prend en considération la fréquence et la récence d'accès dans les décisions de remplacement. En effet, les deux segments sont contrôlés avec la stratégie LRU, mais uniquement les documents du segment non protégé qui sont remplacés. Les documents du segment protégé sont déplacés vers le segment non protégé suivant la stratégie LRU. SLRU

exige utilisation d'un paramètre qui détermine quel pourcentage de l'espace de cache est alloué au segment protégé.

### **3.2.3.2 LRU\* [81]**

Dans cette stratégie tous les documents demandés sont stockés dans une liste LRU. Chaque document a un compteur de nombre de fois qu'il a été référencé. Quand un document caché est référencé alors il est déplacé au début de la liste LRU et son compteur est incrémenté de un. À chaque fois qu'il y a un remplacement, le compteur de document le moins récemment utilisé est vérifié. S'il est à zéro, le document est supprimé du cache. Autrement le compteur est décrémenté de un, et le document est déplacé en tête de la liste.

### **3.2.3.3 LRU-Hot [57]**

LRU-Hot contrôle deux listes LRU : une pour les documents chauds (populaires) et une pour les documents froids (pas aussi populaires). Un document est chaud, si sa fréquence d'accès au serveur original est au-dessus d'un certain seuil. Cette information est envoyée avec le document au client. Selon cette information le document est inséré dans la liste correspondante. Ces listes sont traitées différemment. LRU-Hot utilise deux compteurs de référence : un compteur de base et un compteur pour les documents chauds. Les deux compteurs sont initialisés à zéro. Chaque requête augmente le compteur de base de un. Le compteur chaud est incrémenté de un après chaque  $\alpha$  requêtes ( $\alpha > 1$ ). Quand un document est reçu, il est stocké au début de sa liste correspondante, et une valeur d'accès lui est affectée qui est égale à la valeur actuelle de compteur base. Sur un remplacement, le cache recalcule les valeurs des deux documents se trouvant dans les queues des deux listes.

Le document avec la plus petite valeur est enlevé de cache. L'algorithme est réitéré jusqu'à ce qu'il y ait assez d'espace pour le nouveau document entrant.

### **3.2.3.4 HYPER-G [55]**

Cette stratégie combine LRU, LFU, et elle utilise aussi la taille des documents. Au début le document le moins fréquemment utilisé est choisi. S'il y a plus d'un document qui répond à ce critère, HYPER-G choisit le moins récemment l'utilisé. Si ceci ne donne toujours pas un document unique à remplacer, le plus grand document est choisi.

### **3.2.3.5 LRU-K [89]**

LRU-K garde les informations historiques des documents même s'ils ont été remplacés, notamment elle garde les dates de leurs K derniers accès. Quand cette politique de remplacement de cache est invoquée, la stratégie LRU est utilisée pour sélectionner le document à remplacer parmi les documents ayant une fréquence inférieure à K. RP est un autre

paramètre utilisé par LRU-K pour résoudre le problème qu'on peut rencontrer avec cette politique, lié à la forte croissance de la taille de l'historique. Pour chaque document absent du cache, si sa durée d'absence dépasse RP alors il est supprimé de l'historique.

### **3.2.3.6 Avantages**

Ces stratégies présentent comme avantage le fait de combiner la récence et la fréquence, et par conséquent, de même pour leurs deux avantages. Ces stratégies peuvent aussi nous éviter les problèmes rencontrés dans l'utilisation uniquement de l'une de ces deux approches à part.

### **3.2.3.7 Inconvénients**

L'inconvénient de ces stratégies est dû aux procédures spéciales utilisées dans les mécanismes de remplacement de cache. En effet la plupart d'entre elles introduisent une complexité additionnelle. Mise à part LRU \* qui essaye de combiner une implémentation simple de LRU avec un compteur de fréquence. Cependant, elle ne prend pas en considération la taille des documents.

## **3.2.4 Les stratégies de remplacement de cache basées sur les fonctions**

Ces stratégies utilisent une fonction pour calculer une valeur à base d'un certain nombre de caractéristiques spécifiques d'un document, ensuite cette valeur lui sera affectée. Les documents seront triés en fonction de cette valeur, et le document ayant une valeur minimum ou maximum (selon la stratégie) est choisi pour le remplacement.

### **3.2.4.1 GD (Greedy Dual)-Size [59]**

La GD-Size maintient pour chaque document une valeur caractéristique  $H_i$ . Une requête pour un document  $i$  exige de recalculer  $H_i$  comme suit :

$$H_i = \frac{C_i}{S_i} + L .$$

Où  $C_i$  est le coût de chargement de document  $i$  depuis son serveur

$S_i$  est la taille de document  $i$

$L$  est le facteur courant de vieillissement qui est initialisé à zéro. La GD-Size choisit le document avec la plus petite valeur de  $H_i$ . La valeur de ce document est affectée à  $L$ .

### **3.2.4.2 GDSF [52]**

GDSF calcule  $H_i$  comme suit

$$H_i = f_i \frac{C_i}{S_i} + L .$$

Où  $f_i$  est la fréquence de document  $i$

Une forme généralisée de GDSF a été décrite dans [82]. Elle calcule  $H_i$  comme suit

$$H_i = \frac{f_i^\alpha}{S_i^\beta} + L .$$

$C_i$  reçoit 1 ;

$\alpha$  et  $\beta$  sont des poids paramétrable qui indiquent l'importance des facteurs utilisés.

### 3.2.4.3 TSP (Taylor Series Prediction) [83]

TSP calcule  $H_i$  comme suit

$$H_i = \frac{f_i \times C_i}{S_i \times T_T} .$$

et,

$$T_T = t_p - t_c .$$

Où  $C_i$  est le coût de chargement de document  $i$  depuis son serveur ;

$S_i$  est la taille de document  $i$  ;

$f_i$  est la fréquence de document  $i$  ;

$T_T$  décrit "l'accélération" temporelle de la requête vers le document  $i$  ;

$t_p$  est une prédiction de temps prévu de la prochaine requête pour ce document  $i$  ;

$t_c$  est le temps courant. le  $t_p$  est déterminé avec la série de Taylor de second degré.

### 3.2.4.4 Stratégie de Bolot et Hoschka [84]

Cette stratégie utilise la fonction suivante pour calculer la valeur de document  $i$  :

$$f(i) = W_1 l_i + W_2 S_i + \frac{W_3 + W_4 S_i}{T_i} .$$

Où  $W_1$ ,  $W_2$ ,  $W_3$ , et  $W_4$  sont des paramètres de réglage. Les valeurs des ces paramètres dépendent de la métrique de performance qui devrait être optimisée.

### **3.2.4.5 LRV (Lowest Relative Value)[62]**

Cette stratégie choisit le document avec la plus petite valeur relative qui est calculé en fonction de la probabilité qu'un document soit accédé encore. Cette probabilité  $P_r$  est calculée comme suit :

$$P_r(f_i, T_i, S_i) = \begin{cases} P(1, S_i)(1 - D(T_i)), & \text{si } f_i = 1, \\ P(f_i)(1 - D(T_i)) & \text{sinon.} \end{cases}$$

Pour  $i$ ,  $P(f_i)$  est la probabilité pour qu'un document soit accédé pour la  $f_i + 1$  fois sachant qu'il est accédé  $f_i$  fois. Pour des documents ayant été accédés uniquement une fois, cette probabilité dépend de la taille de document et elle est calculée par  $P(1, S_i)$ .  $D(T_i)$  dénote la distribution des temps inter arrivés des requêtes.

Rizzo et Vicisano [62] ont montré comment calculer  $P(f_i)$  et  $P(1, S_i)$  de manière efficace ainsi que les paramètres qui influent sur le  $D(T_i)$ .

### **3.2.4.6 Avantages**

Cette classe inclue la plupart des stratégies de remplacement de cache proposées dans la littérature. Parmi les avantages de ces stratégies on a :

- Ces stratégies n'exigent pas une combinaison fixe des facteurs utilisés, à travers un choix approprié des paramètres, on peut optimiser n'importe quelle métrique de performance.
- Elles prennent en considération un certain nombre de facteurs qui permettent de manipuler différentes situations de charge de travail.

### **3.2.4.7 Inconvénients**

Certaines propositions, associent des poids aux paramètres qu'elles utilisent pour le remplacement, mais le choix des poids appropriés pour un critère de performance est une tâche difficile.

## **3.3 Les stratégies de remplacement de cache dédiées aux environnements mobiles**

### **3.3.1 La politique TDS (Time and Distance Sensitive) [76]**

Une politique de remplacement de cache, dédiée aux réseaux Ad Hoc, a été proposée dans [76], appelée TDS (Time and Distance Sensitive) qui prend en considération les facteurs de la distance et un facteur temporel pour le remplacement.

TDS se base sur deux facteurs pour choisir une victime à remplacer. Le premier est la distance  $\delta$ , mesurée par le nombre de sauts par rapport aux points d'accès ou terminaux

mobiles, qui ont la donnée demandée. Puisque  $\delta$  est étroitement lié à la latence, si la donnée élémentaire avec un  $\delta$  plus élevé est choisie en tant que victime, alors la latence d'accès serait haute. Par conséquent, la donnée avec la moindre valeur  $\delta$  est choisie en tant que victime. En raison de la mobilité des terminaux, la topologie de réseau peut changer fréquemment, et les valeurs  $\delta$  peuvent devenir obsolètes. Par conséquent un deuxième paramètre  $\tau$  est utilisé, qui capture le temps écoulé depuis la dernière mise à jour de  $\delta$ . La valeur de  $\tau$  est obtenue par  $1/t_{cur} - t_{update}$  où  $t_{cur}$  et  $t_{update}$  sont respectivement le temps courant et le temps de la dernière mise à jour de  $\delta$  pour un document dans le cache. Trois variantes de TDS sont présentées, TDS\_D, TDS\_N et TDS\_T qui choisissent comme victime le document ayant respectivement le minimum de  $(\delta + \tau)$ ,  $(\delta \times \tau)$  et  $\tau$ .

L'inconvénient de cette politique de remplacement est qu'elle ne prend pas en considération la récence et la fréquence des accès.

### 3.3.2 La politique ZC (Zone coopérative) [77]:

Dans [77] un schéma de gestion de cache a été présenté, appelé ZC (Zone coopérative) et dans le quel une politique de remplacement de cache basée sur le calcul d'une valeur a été proposée, où le document ayant la valeur la plus basse, est celui qui est enlevé du cache. Pour chaque document dans le cache, quatre facteurs sont considérés pour le calcul de cette valeur :  
**La popularité** : Elle est la probabilité d'accès d'un document se trouvant dans le cache. Un document avec une petite probabilité d'accès devrait être choisi pour le remplacement, donc le serveur enregistre pour chaque document  $d_i$  une probabilité d'accès  $A_i$  initialisée à zéro. Pour chaque accès à ce document, cette probabilité est mise à jour suivant cette formule :

$$A_i^{new} = \alpha \frac{1}{t_c - t_l} + (1 - \alpha)A_i^{old} .$$

Où  $t_c$  est le temps courant ;

$t_l$  est le temps de dernier accès ;

$\alpha$  est un facteur constant pour peser l'importance de la récence d'accès.

**La distance  $\delta$**  : Elle est mesurée en nombre de sauts entre le nœud demandeur du document et le nœud répondant (la source ou à partir d'un cache). Les documents avec une petite distance sont favorisés pour le remplacement. Ce paramètre est particulièrement important pour la conservation de la bande passante et la réduction de la latence.

**Cohérence** : Un document  $d_i$  est valide pour une durée de vie limitée, qui est connue à partir de champ  $TTL_i$ , les documents avec des petits  $TTL$  sont favorisés pour le remplacement.

**La taille S :** Les documents avec une plus grande taille sont choisis pour le remplacement. Dans ce cas le cache peut contenir plus de documents et ainsi peut satisfaire plus de demandes d'accès.

Basée sur les facteurs cités ci-dessus, la fonction qui calcule la valeur associée à chaque document est donnée par la formule suivante :

$$w_1 A_i + w_2 \delta_i + w_3 TTL_i + w_4 / S_i .$$

où  $w_1, w_2, w_3$  et  $w_4$  sont des poids associés à chaque facteur tels que :

$$\sum_{j=1}^4 w_j = 1 .$$

Le document avec la plus petite valeur de cette fonction est choisi comme victime pour le remplacement.

### 3.3.3 Politique de remplacement de cache orientée énergie [78]

Une autre politique de remplacement de cache orientée énergie est proposée dans [78] dédiée à un schéma de caches coopératifs. Cette politique tente à minimiser la consommation d'énergie des accès de tous les nœuds mobiles à un document donné.

Pour cela, cette politique calcule pour chaque document  $d_j$  un *benefit*( $d_j$ ) comme suit :

$$benefit(d_j) = \sum_{i=1}^Q P_{ij} E_{ij} .$$

$Q$  est le nombre de nœuds mobiles ;

$P_{ij}$  est la probabilité qu'un nœud  $i$  accède au document  $d_j$  ;

$E_{ij}$  est le coût en énergie pour qu'un nœud  $i$  accède au document  $d_j$ .

Quant un document  $d_x$  doit être caché et que l'espace dans le cache est insuffisant cette politique sélectionne un ensemble  $V$  de documents à remplacer dans le cache telle que :

$$\sum_{j \in V} taille(d_j) \geq taille(d_x) .$$

et que

$$\sum_{j \in V} benefit(d_j) .$$

soit le minimum possible.

Le gain en énergie après le remplacement sera :

$$gain(d_x) = benefit(d_x) - \sum_{j \in V} benefit(d_j) .$$

### 3.3.4 La politique de remplacement de cache PIX [79]

Acharya a proposé une politique de remplacement de cache appelée PIX [79], elle est basée sur une fonction du coût (a cost-based caching algorithm) dans la quelle le document avec le plus petit coût est choisi comme victime pour le remplacement.

Soit  $d_i$  un document se trouvant dans le cache, son coût est calculé selon la formule suivante :

$$pix(d_i) = p_i/x_i.$$

où

$P_i$  est la probabilité d'accès ;

$x_i$  est la fréquence de diffusion de document  $d_i$ .

La motivation pour le remplacement basé sur *PIX* est que la victime devrait être le document qui a la plus basse utilité locale (c.à.d. la plus basse probabilité d'accès) et le plus bas coût de re-acquisition (c.à.d. la fréquence de diffusion la plus élevée).

### 3.3.5 La politique Min-SAUD (Minimum Stretch integrated with Access rates, Update frequencies, and cache validation Delay) [80]:

Une autre politique de remplacement de cache a été proposée dans [80] appelée Min-SAUD, (*Minimum Stretch integrated with Access rates, Update frequencies, and cache validation Delay*) pour les environnements mobiles sans file où la consistance doit être vérifiée avant l'utilisation de document caché. Elle est basée sur une fonction de gain qui prend en considération plusieurs paramètres qui affecte le remplacement d'un document dans le cache, à savoir, la probabilité d'accès, la fréquence de mise à jour, la taille de document, le temps de recherche, et le coût de validation de cache. Cette fonction de gain est calculée pour chaque document  $i$  selon la formule suivante :

$$gain(d_i) = \frac{p_i}{s_i} \left( \frac{b_i}{1 + x_i} - v \right)$$

Où

$p_i$  est la probabilité d'accès au document  $i$  ;

$s_i$  est la taille de document  $i$  ;

$x_i$  est le rapport de taux de mise à jour sur le taux d'accès au document  $i$  ;

$v$  est le délais de validation de cache.

L'idée de cette politique est de maximiser le gain total pour les documents maintenus dans le cache. Ainsi, pour trouver l'espace pour un document  $d_j$  au  $k^{iem}$  accès, la politique Min-SAUD identifie comme victime un ensemble optimal de document  $V_k^*$  tel que :

$$V_k^* = \min_{V_k \subseteq (cache)} \sum_{i \in V_k} gain(d_i)$$

et

$$taille(cache) \geq s_j + \sum_{d_i \in cache} s_i - \sum_{d_i \in V_k} s_i$$

### 3.4 Proposition d'une politique de remplacement de cache pour les réseaux Ad Hoc

La conception des politiques de remplacement de caches destinées aux réseaux mobiles Ad Hoc ou aux environnements mobiles en général doit prendre en considération les caractéristiques du trafic des réseaux Ad Hoc (comme la fréquence des documents et leurs tailles), ainsi que les caractéristiques spécifiques des réseaux mobiles Ad Hoc comme la variation des distances qui séparent les nœuds les uns des autres par le fait de la mobilité.

Dans tous les travaux qu'on a vus, la plupart des politiques proposées dans le cadre des réseaux mobiles Ad Hoc sont basées sur la distance, vu son lien direct avec la latence des clients. La politique proposée dans le schéma de coopération de cache appelé ZC (Zone Cooperative) [77] a essayé d'inclure tous les paramètres par sommation et en associant pour chacun un poids  $W_i$ , mais dans cette politique la difficulté réside dans la détermination optimale des poids  $W_i$  pour le remplacement. La politique TDS utilise la distance en nombre de sauts et un paramètre temporel, mais elle ne prend pas en considération la fréquence et la taille des documents qui sont des paramètres très importants et influent considérablement sur les performances des caches.

Dans ce qui suit, nous allons proposer une politique de remplacement de cache basée sur le calcul d'une valeur pour chaque document présent dans le cache qu'on appellera FSDV *Frequency, Size and Distance based Value*. La fonction qui calcule cette valeur inclue quatre paramètres, la fréquence d'accès, la distance en nombre de sauts séparant le nœud qui répond à une demande d'un document de celui qui le demande, le temps écoulé depuis la dernière mise à jour de cette même distance et la taille de document. Donc pour chaque document d'identifiant  $id$  on calcule la fonction  $f(id)$  puis on choisit pour le remplacement le document ayant la plus petite valeur.

$$f(id) = \frac{\text{frequence}(id) \times \text{distance}(id)}{\text{taille}(id) \times \text{age\_distance}(id)}$$

Mais l'utilisation de la fréquence d'accès pose le problème de la pollution de cache. Ce problème est posé lorsqu'un document était très populaire dans une période de temps, alors il peut demeurer toujours dans le cache même s'il reste non référencé pour une longue autre période de temps.

Pour remédier à ce problème, on utilise un autre paramètre, Age\_Max qui représente le seuil maximum pour la moyenne de l'âge des documents présents dans le cache. Quand la moyenne dépasse ce seuil, toutes les fréquences des documents seront divisées par deux.

### **3.5 Conclusion**

La problématique principale des politiques de remplacement de caches concerne le choix des documents qui doivent être remplacés quand l'espace libre dans le cache devient insuffisant. Plusieurs politiques ont été proposées dont l'objectif est de bien choisir les documents inutiles et il s'est avéré que les politiques qui combinent deux ou plusieurs paramètres donnent de meilleures performances.

Dans ce chapitre, nous avons présenté la majorité des politiques de remplacement. Ensuite nous les avons classées selon les paramètres utilisés. Nous avons également proposé une politique de remplacement de cache pour les réseaux mobiles Ad Hoc dans le but d'améliorer les performances de ce dernier.

Dans le chapitre suivant nous allons présenter quelques méthodes d'évaluation de performances, ainsi que quelques caractéristiques du trafic Web. Ensuite, nous passons à la description du modèle de notre système et de quelques caractéristiques du trafic. Nous terminons par la présentation des résultats de simulation de notre politique de remplacement de cache FSDV avec quatre autres politiques LFU, LRU, TDS et ZC.

## Chapitre 4

---

# Caractérisation du trafic, modélisation et simulation

---

### 4.1 Introduction

Toute étude sérieuse des techniques et mécanismes utilisés dans le cadre des caches Web doit commencer par la compréhension des propriétés et caractéristiques des documents Web à cacher. Les questions que l'on doit se poser sont : Quelle est la taille de ces documents? Combien de fois sont-ils reaccédés? À quelle fréquence changent-ils? Toutes ces questions ont un effet profond sur la conception des caches et particulièrement sur les politiques de remplacement de cache.

Dans ce chapitre, nous allons présenter les différentes méthodes d'évaluation de performances existantes, ainsi que les principales caractéristiques du trafic. Nous prêtons une particulière attention aux aspects pertinents liés aux caches. Ensuite nous allons présenter le modèle général du système à étudier, l'environnement de simulation ainsi que les résultats de simulation de notre politique de remplacement de cache FSDV (Frequency, Size, and Distance Value-based).

## **4.2 Les méthodes d'évaluation de performances**

Avant d'entamer la caractérisation du trafic, il sera utile de décrire les principales méthodes d'évaluation de performances qui sont utilisées pour évaluer les nouvelles technologies. Ces méthodes d'évaluation de performances sont réparties en quatre grandes catégories : mesures directes, analyse mathématique, simulation basée sur les traces, simulation basée sur le benchmarking.

### **4.2.1 Mesures directes**

Les mesures directes concernent les méthodes d'observation et d'auscultation du comportement de Web en temps réel. Elles sont particulièrement utiles pour évaluer l'exécution réelle du Web. Mais il est souvent très difficile de mettre en application ces méthodes. Par exemple, observer l'exécution bout en bout comme vue par de vrais utilisateurs exigerait d'équiper leurs navigateurs. Les mesures directes sont généralement non répétables, puisqu'il est impossible de reproduire entièrement les conditions dans lesquelles les mesures initiales ont été prises.

### **4.2.2 Analyse mathématique**

L'évaluation analytique des performances consiste à représenter le système par un modèle mathématique. Il existe plusieurs formalismes et méthodes souvent utilisés pour évaluer les performances de réseau, on peut citer, entre autres : les méthodes de graphes, les réseaux de Petri et les files d'attente. L'avantage principal de ces techniques est qu'elles sont très efficaces, surtout lorsque la solution analytique existe. Toutefois, la modélisation analytique implique un niveau d'abstraction considérable, on ne peut pas utiliser ces méthodes sans faire beaucoup de suppositions simplificatrices qui peuvent compromettre la qualité des résultats.

### **4.2.3 Simulation basée sur les traces**

Les traces de Web sont des enregistrements des informations telles que le temps d'arrivée des requêtes, les URLs demandés, la taille des réponses, et les informations d'entêtes. Les fichiers logs des accès collectés par les serveurs Web représentent un exemple de ces traces. Ces fichiers logs peuvent être aussi collectés auprès des navigateurs ou proxys.

Les différentes propriétés de trafic, telles que la distribution des tailles de documents, ou les intervalles de temps inter arrivés, peuvent être inférés depuis les traces collectées de différentes manières, c'est ce qu'on appelle le processus d'analyse de traces. D'autres

expériences utilisent ces traces d'accès pour simuler le comportement de diverses techniques et mécanismes de Web. Ces expériences s'appellent les trace-driven simulations.

Généralement n'importe quelle trace manifeste des propriétés particulières qui lui sont propres [73]. Une trace des accès de cache d'un proxy d'un laboratoire ou d'une université de recherches peut présenter des propriétés tout à fait différentes d'une trace semblable des utilisateurs résidentiels ; les traces obtenues pendant les jeux olympiques peuvent être différentes des traces des turbulences de marché boursier. Donc la question qui se pose est comment essayer de reproduire une trace pour un comportement plus général.

#### 4.2.4 Simulation basée sur le benchmarking

Benchmarking est l'utilisation d'un trafic synthétique, artificiellement généré pour tester les aspects spécifiques des équipements de Web ainsi que leurs comportements. L'ultime but de la synthétisation de trafic Web est d'imiter le comportement réel du Web aussi étroitement que possible. Ce trafic est généré en analysant de vraies traces, puis extraire les caractéristiques du trafic, et reproduire ces caractéristiques dans un trafic synthétique.

Cette reproduction de trafic synthétique peut exiger de faire des suppositions au sujet du comportement du Web, par exemple en plus de la distribution de popularité, on peut également avoir besoin de la corrélation des références avec les autres caractéristiques du Web; c'est pour cela qu'il est difficile de décider quand cesser de considérer de plus en plus des aspects du comportement de Web.

Parfois, il est souvent nécessaire d'évaluer les systèmes complets dans des conditions de surcharge. Par exemple, pour découvrir comment un server Web se comporte quand il reçoit plus de requêtes que ce qui est fournie par des traces disponibles. Dans ces situations, l'utilisation du trafic synthétique devient inévitable.

### 4.3 Etat de l'art sur la caractérisation du trafic

Dans la littérature, plusieurs travaux ont été consacrés à la caractérisation du trafic Web [33], [66], [67], [70] ainsi qu'à l'étude des performances des caches [78], [80]. Parmi les caractéristiques du trafic Web identifiées on a :

- La distribution de la popularité des documents Web suit une loi Zipf-like ;
- La distribution de la taille des documents suit une loi *heavy-tailed* (Pareto) ;
- La localité de références manifestées par les flux de requêtes.

- Le pourcentage élevé des documents référencés une seule fois (les one-timers).
- Le taux de modification des documents

### 4.3.1 La popularité des documents

Le comportement des politiques de remplacement est très influencé par la distribution de la popularité des documents, puisque ceux qui sont populaires tendent à rester dans les caches. Plusieurs chercheurs ont observé que la fréquence relative avec laquelle les pages Web sont demandées suit la loi de Zipf [65]. La loi Zipf est une loi statistique très célèbre qui est observée dans le comportement de plusieurs systèmes complexes de nature différente. C'est une description du rapport entre la fréquence de l'occurrence d'un événement et son rang.

Surnommé par le professeur George Kingsley Zipf (1902-1950) de Harvard, cette loi a été à l'origine appliquée au rapport entre les mots dans un texte et leur fréquence d'utilisation. Elle affirme que la probabilité relative d'une requête pour le  $i^{\text{ieme}}$  plus populaire document est proportionnelle à  $1/i$ , alors cette relation est exprimée de la façon suivante :

$$P_i = k i^{-1} .$$

Où  $k$  est une constante de proportionnalité.

Glassman [68] était peut-être le premier à utiliser la loi Zipf pour modéliser la popularité des pages Web, puis, plusieurs d'autres auteurs ont également appliqués cette loi [67], [66], [69], [70]. Cependant, plusieurs études sur l'applicabilité de la loi Zipf aux documents Web, ont conclues que la distribution de la popularité des documents Web suit généralement une distribution Zipf-like [71], [85], [72] dont la probabilité relative d'une requête pour le  $i^{\text{ieme}}$  plus populaire document est proportionnelle à  $1/i^\beta$  avec  $\beta$  compris entre 0 et 1. Donc la distribution de la popularité des documents Web ne suit pas la loi Zipf strict (pour la quel  $\beta = 1$ ), mais plutôt une loi plus général Zipf-like, avec le paramètre  $\beta$  qui varie d'une trace à l'autre mais qui est proche de 1. Selon cette loi, la popularité  $P_i$  d'un document de rang  $i$  est donnée par la formule suivante :

$$P_i = k i^{-\beta} \text{ avec } 0 < \beta < 1 .$$

Où  $k$  est une constante de proportionnalité.

### 4.3.2 La taille des documents

Dans l'évaluation de performance des caches Web, et contrairement au cache mémoire (où les documents sont de même taille), la distribution des tailles des documents Web joue un rôle important dans la mesure où elle influe directement sur les performances des politiques de remplacement de cache. En effet, un cache d'une taille donnée aura la capacité de contenir plus de documents de petite taille, ce qui implique un taux de succès élevé [69]. Cependant il ne peut contenir que moins de documents de grande taille ce qui implique un petit taux de succès mais éventuellement un grand taux de succès en octets.

Plusieurs études des caractéristiques du trafic ont montré que cette distribution est de type *heavy-tailed* [43], [86], [56]. Une distribution est dite *heavy-tailed* si :

$$P[X > x] \approx x^{-\alpha} \quad x \rightarrow \infty, \quad 0 < \alpha < 2.$$

Si la forme asymptotique d'une distribution est hyperbolique alors elle est *heavy-tailed*, indépendamment de son comportement pour les petites valeurs de la variable aléatoire. Une forme simple de la distribution *heavy-tailed* est la distribution de Pareto. sa fonction de densité de probabilité est donnée par :

$$P(x) = \alpha k^\alpha x^{-x-1} \quad \alpha, k > 0, \quad x \geq k.$$

et sa fonction de distribution est donnée par :

$$F(x) = P[X \leq x] = 1 - (k/x)^\alpha.$$

Le paramètre  $k$  représente la plus petite valeur possible de la variable aléatoire  $x$ , donc  $k$  est la plus petite taille que peut prendre un document, alors si  $k$  augmente les petites valeurs de taille de documents sont ignorées.

### 4.3.3 La localité des références

La localité des références caractérise la capacité de prévoir de futurs accès aux documents à partir des accès passés, elle est exploitée pour développer les politiques et les mécanismes liés aux caches, pour décider quels sont les documents à garder ou enlever du cache, quels sont les documents à précharger, c'est-à-dire, les charger dans le cache avant qu'il soient accédés. Il y a deux types de localité : temporelle et spatiale.

#### 4.3.3.1 La localité temporelle

La localité temporelle se réfère à la tendance qu'un même document soit re-référencé fréquemment dans un petit intervalle de temps, autrement dit, un document qui vient d'être référencé est plus probable qu'il le soit encore dans le futur proche [69].

La localité temporelle est modélisée à l'aide d'une pile LRU appelé *LRU stack model* ou LRUSM [70], cette pile est utilisée pour stocker les documents les plus récemment accédés, et chaque position dans la pile est associée à une probabilité de re-référencement qui ne dépend que de cette position et non du document qui l'occupe.

#### 4.3.3.2 La localité spatiale

La localité spatiale exprime le fait que certains accès à quelques documents nécessitent fréquemment des accès à d'autres documents [66]. Par exemple, un accès à une page HTML peut nécessiter des accès aux documents encastrés ou hyperliés dans cette page. La localité spatiale implique que le référencement d'un document peut être un facteur prédictif de futur référence.

#### 4.3.4 Les documents référencés une seule fois (one-timers)

Ce sont les documents qui ne sont accédés qu'une seule fois, dans la littérature ils sont communément appelés les *one-timers*. Des études ont montrés que ces documents constituent un grand pourcentage du trafic Web (jusqu'à 70 %) [70], [72], [69], [43]. Il est évident que ces documents ne doivent pas être cachés car en réalité ils ne seront jamais accédés dans le futur, et ils ne font que réduire les performances des caches. Par conséquent, les politiques de remplacement de cache doivent se doter de mécanisme de contrôle d'admission au cache et avoir la capacité de distinguer ces documents parmi d'autres.

Du point de vue positif, les one-timers peuvent être considérés comme une conséquence logique et inévitable de l'efficacité des mécanismes et techniques utilisés dans le cadre des caches Web comme les politiques de remplacement, le prefetching, la réplication, les proxys etc. L'un des objectifs de ces techniques est de réduire la charge des proxys et des serveurs originaux, donc les one-timers s'explique par le fait que ces derniers ne sont pas atteints par les requêtes répétées pour le même document parce qu'elles étaient servies au niveau des caches.

### **4.3.5 Le taux de modification des documents**

Une autre caractéristique des documents Web dont dépend l'efficacité du cache est le taux de mise à jour des documents Web. Le taux de mise à jour est important parce qu'il affecte les mécanismes qui assurent qu'un cache ne sert pas un contenu obsolète aux clients.

Pour caractériser l'effet des changements de documents sur les caches, Douglis et al dans [74] ont étudié le taux de modification, la fraction d'accès ayant retourné des documents qui ont été modifiés depuis leurs derniers accès. Parmi toutes les réponses sur les quelles il était possible de déterminer si les documents ont été modifiés, le taux de modification était autour de 15 %. En d'autres termes, 15 pour cent de requêtes ne pourraient pas être satisfaites depuis le cache par les vieilles copies de documents car ces derniers ont été changés.

### **4.3.6 Le type des documents**

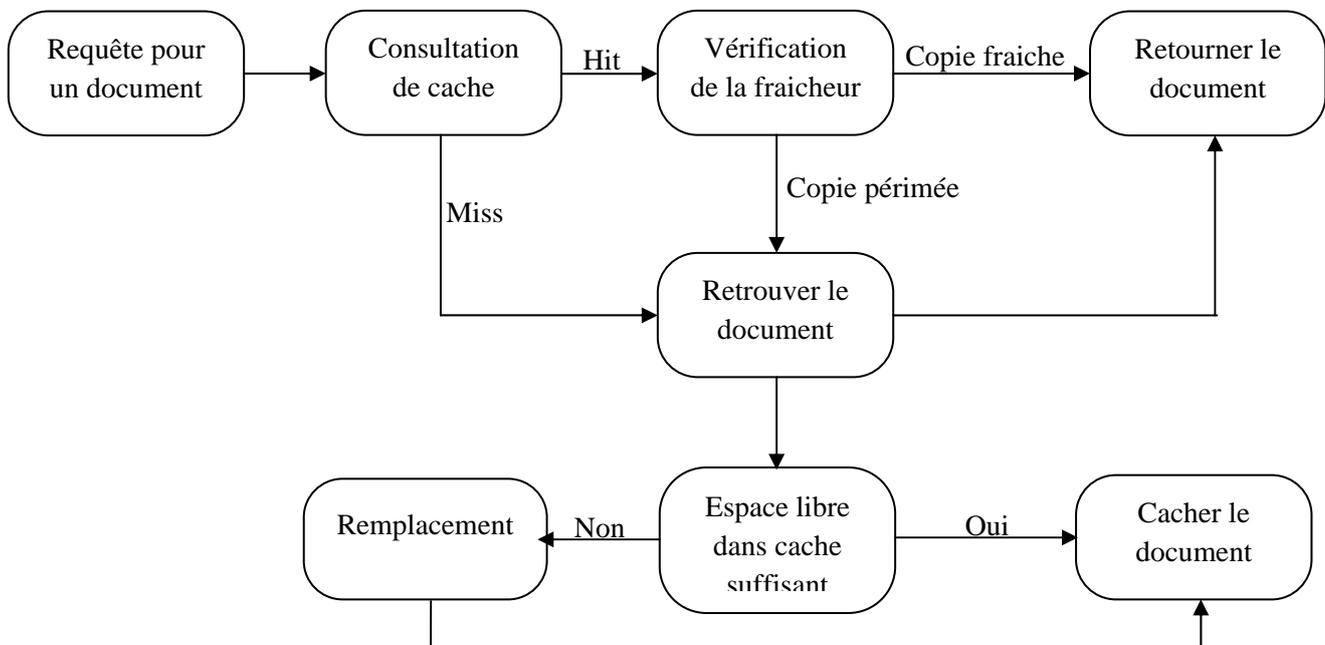
Actuellement la majorité écrasante de tous les téléchargements, en termes de type et de nombre de documents accédés, impliquent des pages HTML et des images. Ensemble, elles comptent autour de 90 % de tous les accès au Web, et plus de 70 % des tailles téléchargées, [74], [55], [75], [43]. Les documents multimédias forment très peu d'accès mais un pourcentage en rapport considérable des tailles téléchargés : 14 % observé dans [55], 24 % dans [43], et jusqu'à 6 % dans [74].

## **4.4 Modélisation et génération de trafic**

Parmi les étapes de toute étude de performance, la modélisation constitue l'étape la plus importante et la plus cruciale. En effet elle permet de représenter les différents aspects à étudier en faisant abstraction du système réel. Dans notre cas, il s'agit de modéliser une politique de remplacement du cache dans le cadre des réseaux mobiles Ad Hoc. Ensuite d'utiliser la simulation pour prouver l'exactitude et l'efficacité de notre proposition en la comparant aux politiques de remplacement de cache traditionnelles et à d'autres politiques qui ont été proposées dans le cadre des environnements mobiles, et ce, en terme de taux de succès (hit rate) des accès aux caches, de taux de succès en octets (byte hit rate) et de taux de succès en distance (distance hit rate).

#### 4.4.1 Modèle de simulation

Notre modèle de simulation fonctionne comme suit. À l'arrivée d'une requête pour un document au niveau d'un nœud, ce dernier consulte son cache, si le document existe alors pour assurer la consistance, il vérifie la fraîcheur de ce document grâce au TTL qui lui est associé. Si le document est frais (c'est un hit), le document est retourné directement en réponse au demandeur. Si le document n'existe pas dans le cache ou bien il n'est pas frais alors la requête est acheminée vers la source de ce document. Quand un nœud reçoit une réponse pour un document, si l'espace libre dans le cache est suffisant pour recevoir ce document, alors il est inséré directement dans le cache, sinon la politique de remplacement de cache est invoquée pour libérer de l'espace pour ce nouveau document. La figure 4.1 montre le comportement du cache recevant une requête pour un document.



**Figure 4.1 :** Comportement du cache recevant une requête pour un document.

#### 4.4.2 La génération du trafic

L'utilisation de la simulation pour l'évaluation de performances des politiques de remplacement de cache nécessite la génération d'un trafic représentatif. Dans la littérature, il existe plusieurs outils dont le but est d'étudier les caractéristiques de trafic Web et de reproduire ces caractéristiques sur un trafic représentatif. Dans notre étude, nous avons utilisé l'outil réalisé par N. Markatchev et C. Williamson en 2002, le WebTraff et plus précisément

le package ProWGen (Proxy Workload Generator) qui est un outil basé sur une approche analytique pour la génération de trafic d'un proxy Web.

Pour adapter ce générateur au trafic des réseaux mobiles Ad Hoc, nous lui avons rajouté principalement deux composants, l'un pour la génération des distances séparant le nœud qui reçoit une réponse de celui qui a répondu à la requête, le paramètre de distance est très important en particulier pour les réseaux mobiles Ad Hoc car il est directement lié à la latence observée par les utilisateurs. L'autre composant est utilisé pour la génération des TTLs (Time To Live) associés aux documents. Ces TTLs sont utilisés par les politiques de remplacement de cache pour assurer la consistance des documents.

### **4.4.3 Environnement de simulation**

L'implémentation de notre simulateur a été faite en C++, sous le système d'exploitation Linux fedora 8. Nous avons utilisé un fichier de trace généré par le générateur ProWGen adapté aux réseaux mobiles Ad Hoc. Afin de tester notre politique de remplacement de cache FSDV (Frequency, Size and Distance based Value), nous l'avons comparée à quatre autres politiques de remplacement de cache que nous avons implémentées, deux politiques classiques LRU (Least Recently Used) et LFU (Least Frequently Used), et deux politiques proposées dans le cadre des réseaux mobiles Ad Hoc à savoir TDS (Time and Distance Sensitive) [76] et ZC (Zone cooperative) [77]. La comparaison a été faite en termes de trois paramètres, le taux de succès, le taux de succès en octets et le taux de succès en distance.

### **4.4.4 Validation du simulateur**

Pour valider les simulateurs des politiques de remplacement de cache cités ci dessus nous avons utilisé des fichiers de courtes traces sur lesquels il était possible de suivre l'exécution des politiques de remplacement. Ensuite, pour chaque politique, on lance son exécution tout en affichant pour chaque remplacement effectué l'état du cache avant et après le remplacement, puis on vérifie si le document remplacé était vraiment celui qui faudrait être choisi par la politique de remplacement simulée. Cette procédure a été répétée plusieurs fois pour chacune des politiques de remplacement de cache.

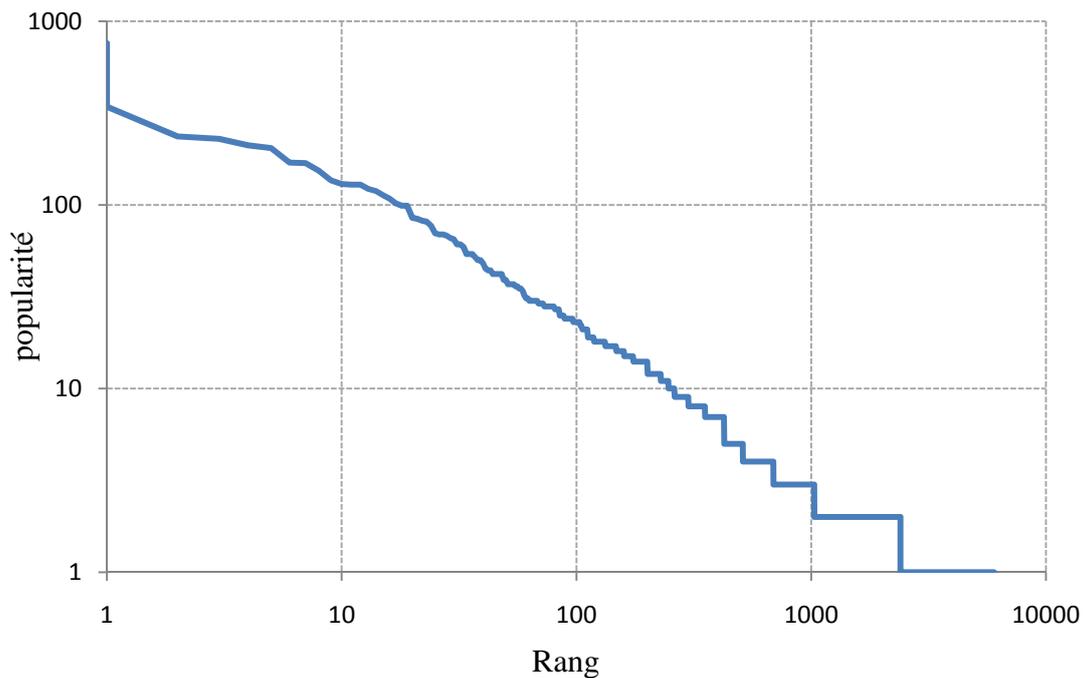
### **4.4.5 Les caractéristiques statistiques du trafic généré**

Dans ce qui suit, nous allons présenter quelques caractéristiques statistiques du trafic généré. Pour cela, nous avons pris un échantillon de 6000 documents pour lesquels nous

avons mis en évidence le changement de leurs popularités en fonction de leurs rangs, la distribution de leurs tailles, et enfin la distribution des TTLs qui leur sont associés.

#### 4.4.5.1 La popularité des documents

La figure 4.2 montre le changement des popularités des documents en fonction de leurs rangs et qui confirme bien que la popularité suit une loi zipf. Nous avons choisi de présenter nos graphes avec l'échelle algorithmique pour mieux visualiser les résultats.



**Figure 4.2 :** Changement de popularité des documents en fonction du rang.

#### 4.4.5.1 La distribution des tailles de documents

Comme nous l'avons mentionné précédemment, la distribution des tailles des documents de trafic suit une loi heavy-tailed. La figure 4.3 montre la variation des tailles de documents de l'échantillon choisi.

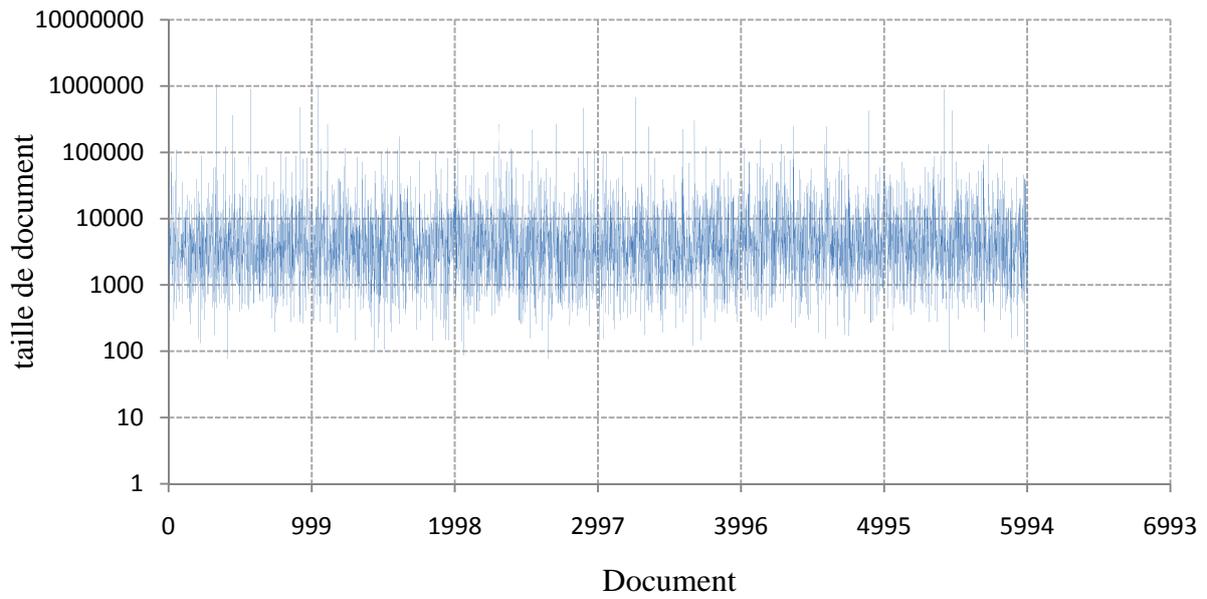


Figure 4.3 : Distribution des tailles de documents.

#### 4.4.5.2 la distribution des TTL de documents

La figure 4.4 montre la distribution des TTL de documents qui ont été générés selon une loi exponentielle de paramètre  $\mu = 1000s$ .

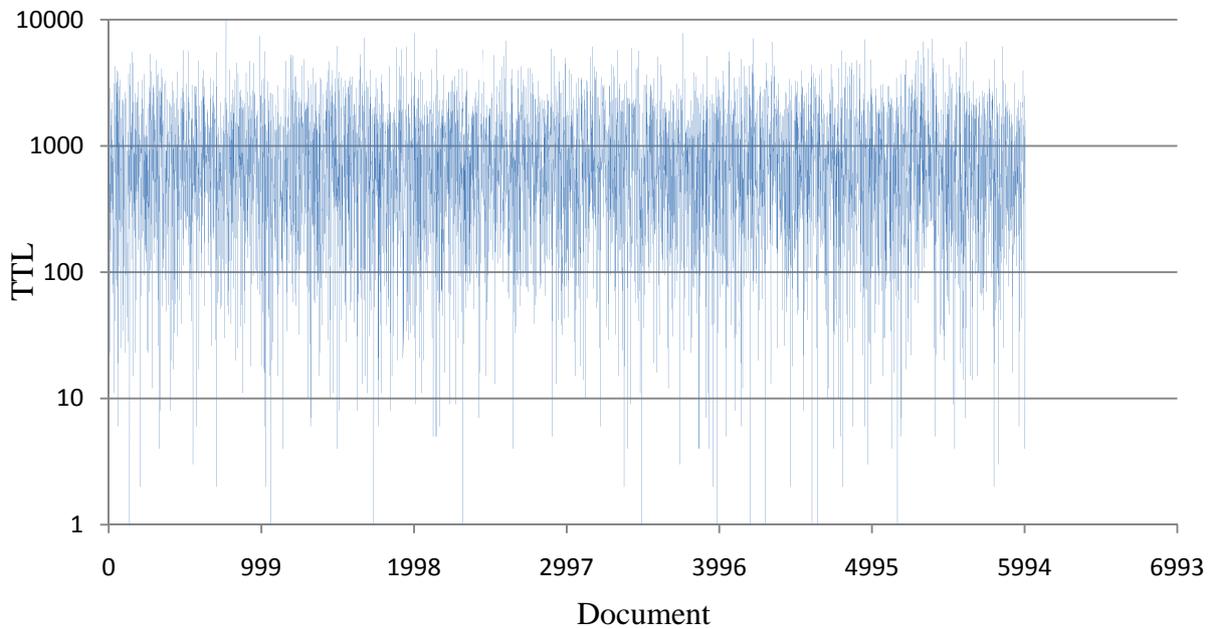


Figure 4.4 : Distribution des TTL de documents.

## 4.5 Résultats de la simulation

Nous allons présenter les résultats de simulation de notre politique de remplacement FSDV et la comparaison avec les résultats obtenus avec les quatre politiques citées précédemment (LFU, LRU, TDS et ZC). On commence par présenter l'impact de paramètre de l'âge moyen maximum sur la politique FSDV, puis la comparaison des performances des politiques en faisant varier la taille du cache. Et enfin, nous terminons par la présentation des résultats de la simulation en faisant varier le paramètre de loi de distribution zipf.

### 4.5.1 Impact du paramètre de l'âge moyen maximum sur la politique FSDV

Le paramètre de l'âge moyen maximum est utilisé pour contourner le problème de la pollution de cache. En effet, quand l'âge moyen en nombre de références des documents présents dans le cache dépasse un certain seuil âge max, toutes les fréquences des documents seront divisées par deux. La figure 4.5 montre la variation des paramètres de taux de succès en fonction du maximum de l'âge moyen.



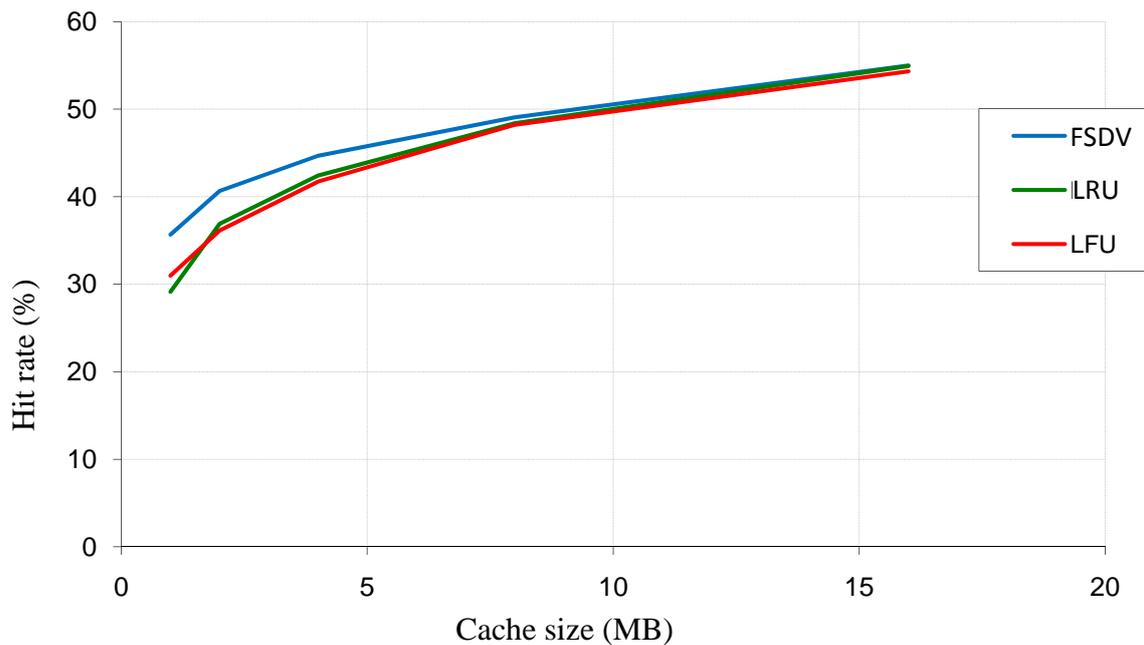
**Figure 4.5:** La variation du taux de succès en fonction de l'âge moyen maximum.

### 4.5.2 Influence de la taille de cache

L'augmentation de la taille des caches permet de cacher plus de documents et par conséquent, permet d'avoir de meilleures performances. Nous allons faire varier la taille du cache de 1MB à 16 MB et voir les performances qui en résultent.

#### 4.5.2.1 Taux de succès de FSDV en comparaison avec LRU et LFU

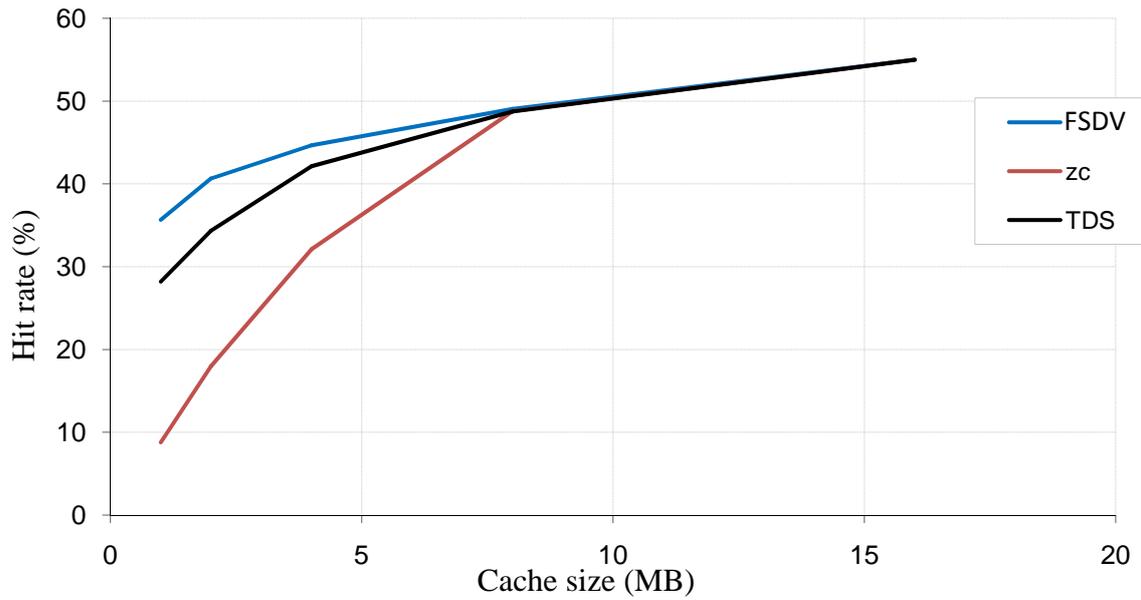
La figure 4.6 montre le Taux de succès de requêtes obtenu avec les politiques de remplacement LRU, LFU et FSDV. On peut constater que FSDV donne de meilleures performances pour toutes les tailles de cache, en particulier pour les petites. mais on remarque que les taux de succès tend vers l'égalité à chaque fois qu'on augmente la taille du cache. Ceci peut être expliqué par la réduction du nombre de remplacements qui différencie FSDV des autres politiques.



**Figure 4.6 :** Taux de succès des politiques FSDV, LFU et LRU par rapport à la taille du cache.

#### 4.5.2.2 Taux de succès de FSDV en comparaison avec TDS et ZC

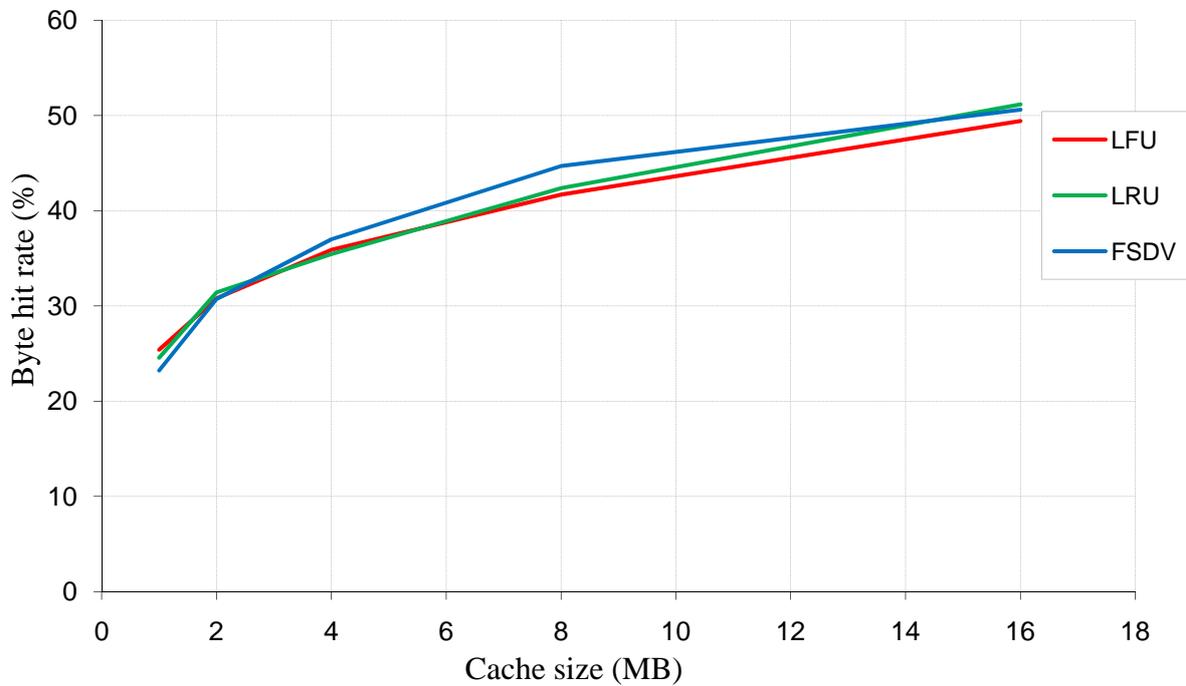
La figure 4.7 montre le taux de succès de requêtes obtenues avec les politiques FSDV en comparaison avec celui donné par TDS et ZC. On peut remarquer que FSDV donne toujours de meilleures performances jusqu'à la valeur 8 MB de la taille du cache. Ensuite, les taux de succès sont presque égaux. En effet, quand la taille du cache augmente les politiques de remplacement du cache opèrent moins de remplacement et leurs performances convergent.



**Figure 4.7 :** Taux de succès des politiques FSDV, TDS et ZC par rapport à la taille du cache.

#### 4.5.2.3 Taux de succès en octets de FSDV en comparaison avec LRU et LFU

La figure 4.8 représente les taux de succès en octets de la politique FSDV et les deux politiques classiques LFU et LRU.

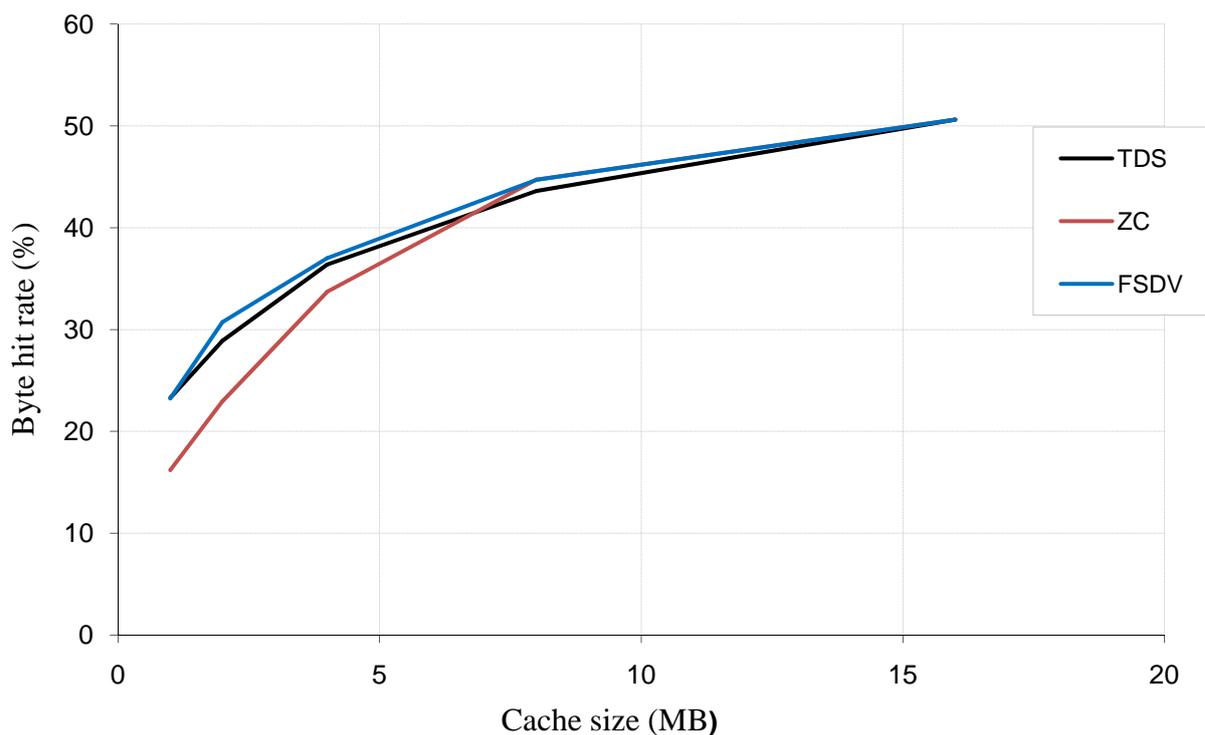


**Figure 4.8 :** Taux de succès en octets des politiques FSDV, LFU et LRU par rapport à la taille du cache.

On remarque que pour la taille du cache inférieure à 2 MB, la politique FSDV fournit de moins bonne performances en terme de taux de succès en octets, parce que FSDV favorise les petits documents et remplace les plus grands. Ceci fournit de bonne performance en termes de taux de succès de requêtes et moins bonne pour le taux de succès en octets. Néanmoins, FSDV fournit de meilleures performances pour les tailles supérieures à 2MB en particulier entre 5 et 10 MB.

#### 4.5.2.4 Taux de succès en octets de FSDV en comparaison avec TDS et ZC

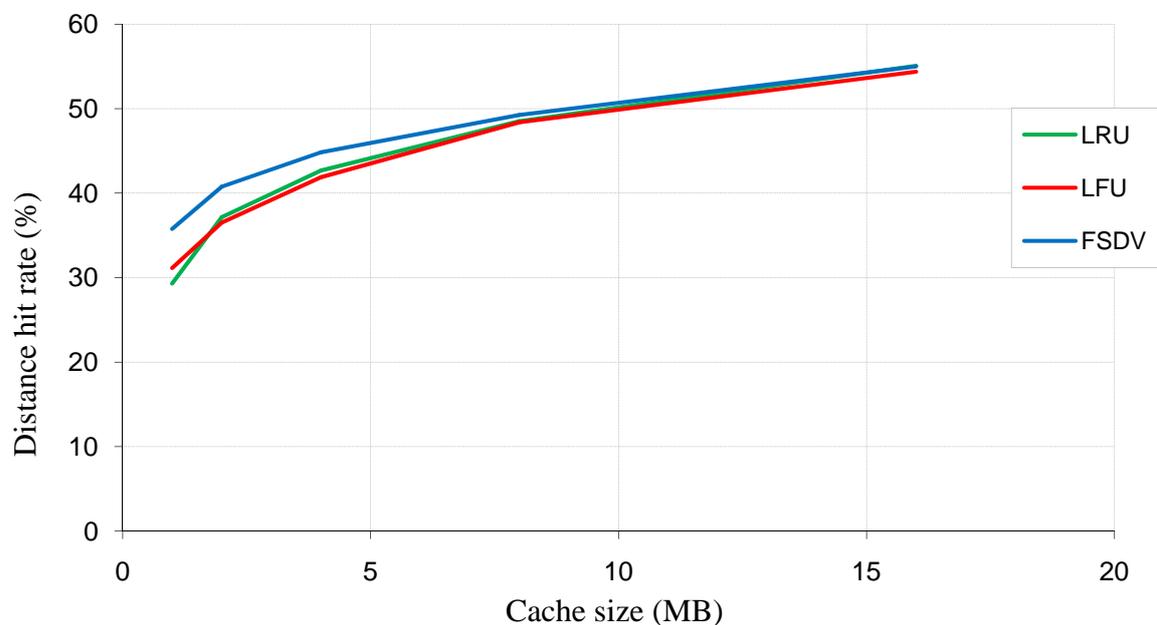
La figure 4.9 montre le résultat de performance en terme de taux de succès en octets de la politique FSDV en comparaison avec les politiques TDS et ZC. On peut remarquer que FSDV donne légèrement plus de performances que TDS et cela uniquement entre 1 et 8 MB. Ensuite, leurs deux taux deviennent presque égaux. Cependant, la politique ZC donne de moins bonnes performances en terme de taux de succès en octets parce qu'elle ne prend pas réellement la taille des documents en considération.



**Figure 4.9 :** Taux de succès en octets des politiques FSDV, TDS et ZC par rapport à la taille du cache.

#### 4.5.2.5 Taux de succès en distance de FSDV en comparaison avec LFU et LRU

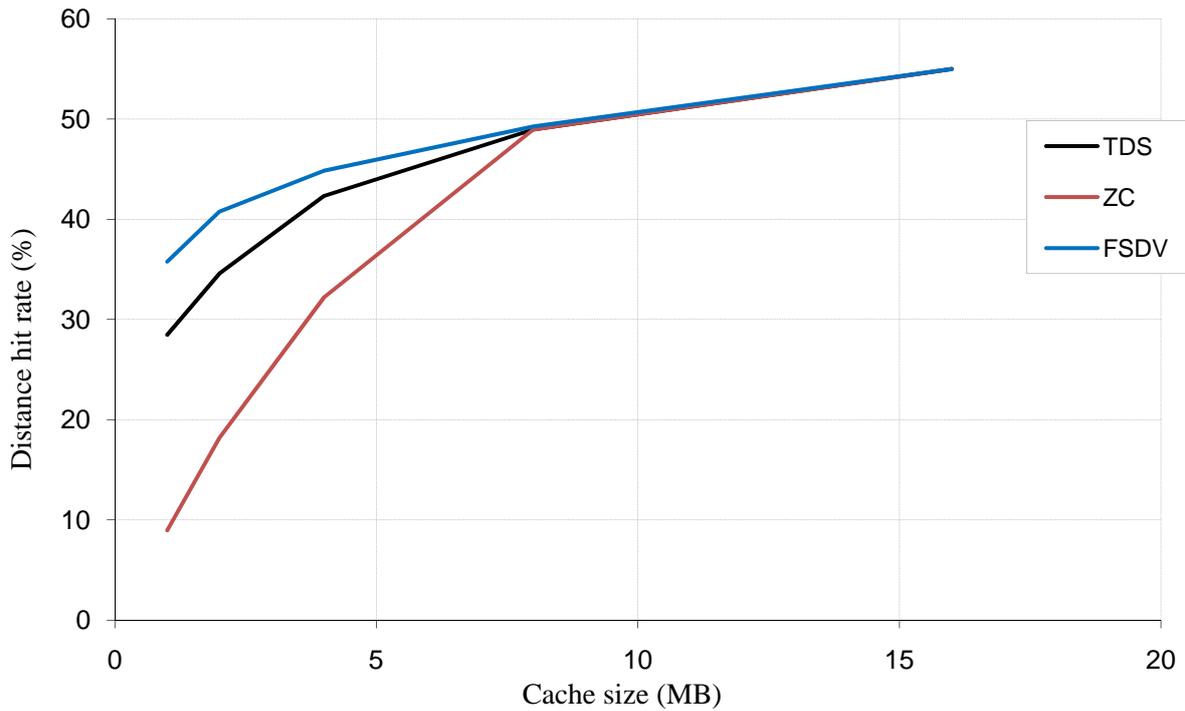
Le taux de succès en distance constitue une métrique de performance particulièrement importante pour les réseaux mobiles Ad Hoc parcequ'elle est liée directement à la latence perçue par l'utilisateur. En effet, cacher les documents dont la source est accessible en longue distance (en terme de saut) depuis le client aurait évité à ce dernier la latence de chargement de ces documents. La figure 4.10 montre les différents taux de succès en distance des politiques FSDV, LFU et LRU et on peut remarquer que FSDV fournit un taux plus grand que les deux autres politiques. Ceci peut être expliqué par le fait que LFU et LRU ne prennent pas en considération le facteur de distance.



**Figure 4.10 :** Taux de succès en distance des politiques FSDV, LFU et LRU par rapport à la taille du cache.

#### 4.5.2.6 Taux de succès en distance de FSDV en comparaison avec TDS et ZC

La figure 4.11 montre que ZC fournit un mauvais taux de succès en distance pour les caches de petite taille sur lesquels le remplacement de cache est fréquent. Pourtant ZC utilise la distance mais avec un poids paramétrable qu'il faut bien choisir.

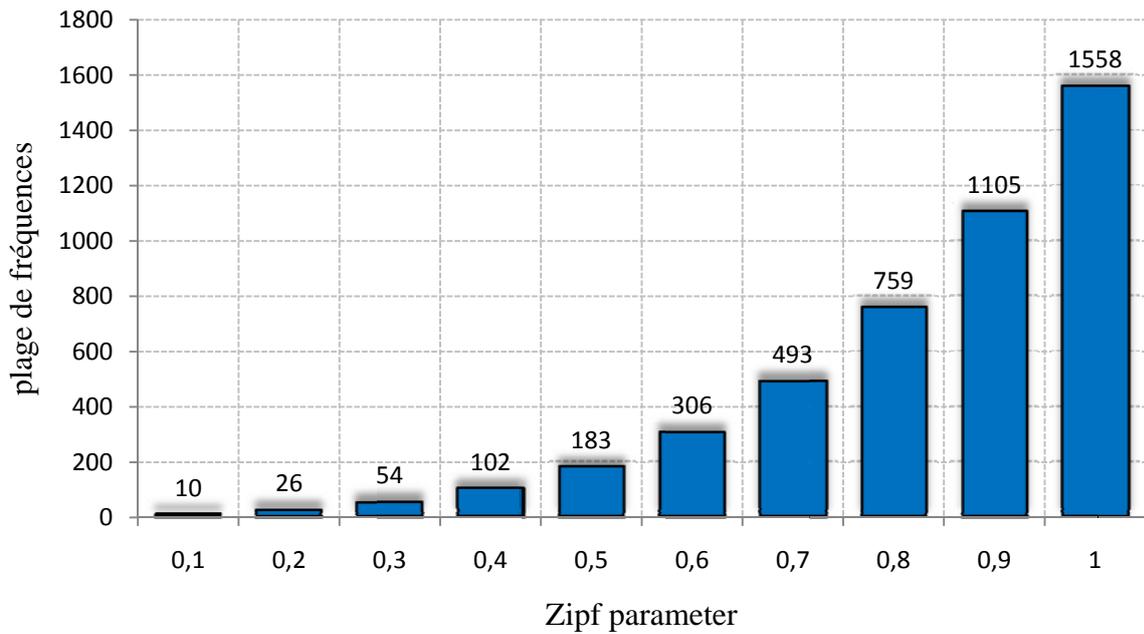


**Figure 4.11 :** Taux de succès en distance des politiques FSDV, TDS et ZC par rapport à la taille du cache.

### 4.5.3 Impact du paramètre de la loi de distribution Zipf

La fréquence d'accès aux documents a un effet sur toutes les métriques de performances du cache. Plus les fréquences d'accès sont décentrées, plus les requêtes sont concentrées sur quelques documents, et par conséquent les taux de succès augmentent. Pour les petites valeurs de paramètre de la loi de distribution Zipf, les fréquences sont moins décentrées c.à.d. qu'elles appartiennent à des plages de valeurs restreintes. La figure 4.12 montre la variation de la plage de valeurs de popularité de 6000 documents en fonction du paramètre de la loi Zipf.

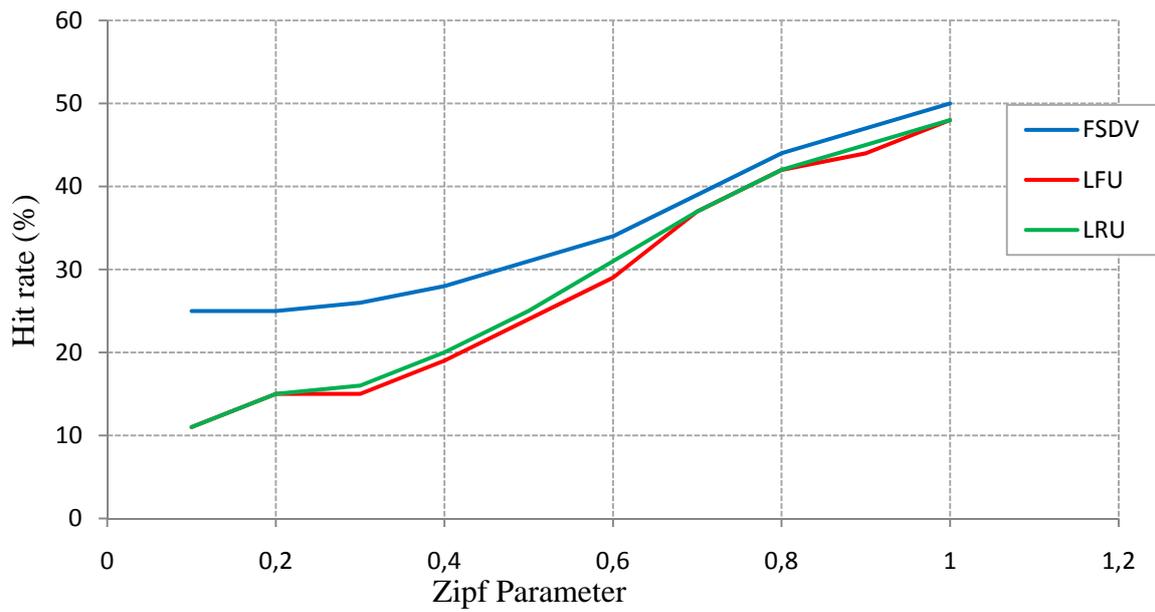
La distribution de popularité des documents Web suit une loi Zipf avec un paramètre entre 0.64 et 0.83 [87]. Nous allons présenter les résultats de simulation des performances des politiques FSDV, LRU, LFU, TDS et ZC avec une taille de cache de 4 MB et en changeant le paramètre de la loi Zipf de 0.1 à 1.



**Figure 4.12 :** Plage de valeurs des popularités de 6000 documents en fonction du paramètre de la loi de distribution Zipf.

#### 4.5.3.1 Taux de succès de FSDV en comparaison avec LFU et LRU

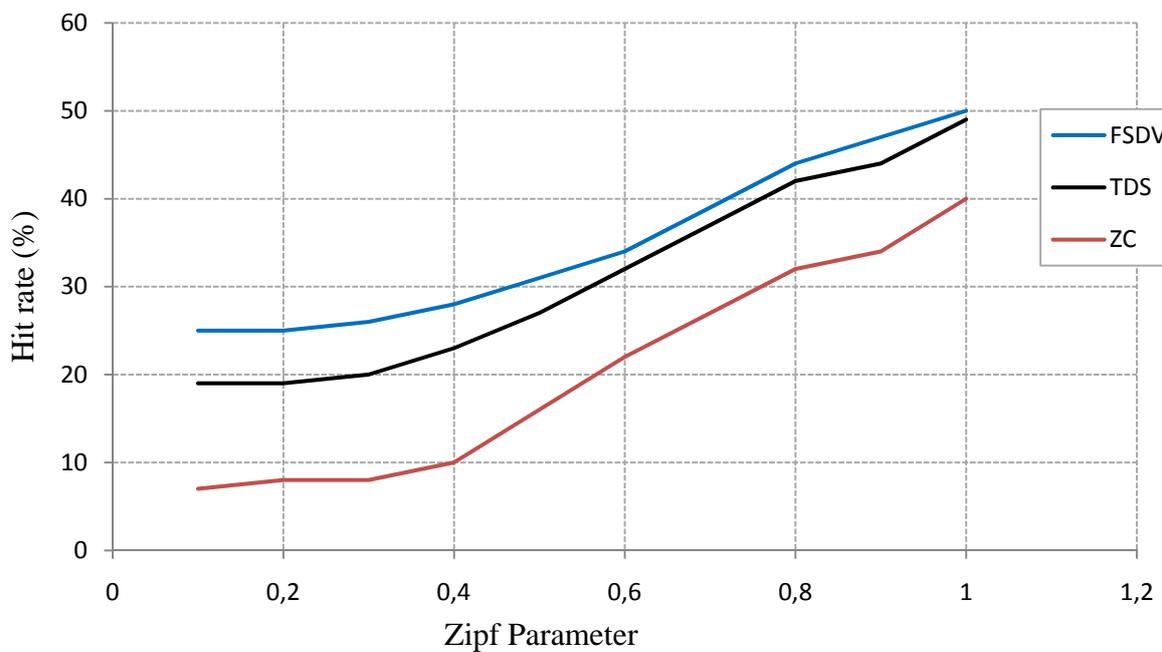
La figure 4.13 montre que la politique de remplacement FSDV fournit un meilleur taux de succès par rapport aux deux autres politiques LFU et LRU en particulier pour les petites valeurs de paramètre de la loi zipf où les fréquences sont réparties sur une plage de valeurs restreinte. Par conséquent, les politiques basées uniquement sur les fréquences donnent moins de performances.



**Figure 4.13 :** Taux de succès des politiques FSDV, LFU et LRU par rapport au paramètre de la loi Zipf.

#### 4.5.3.2 Taux de succès de FSDV en comparaison avec TDS et ZC

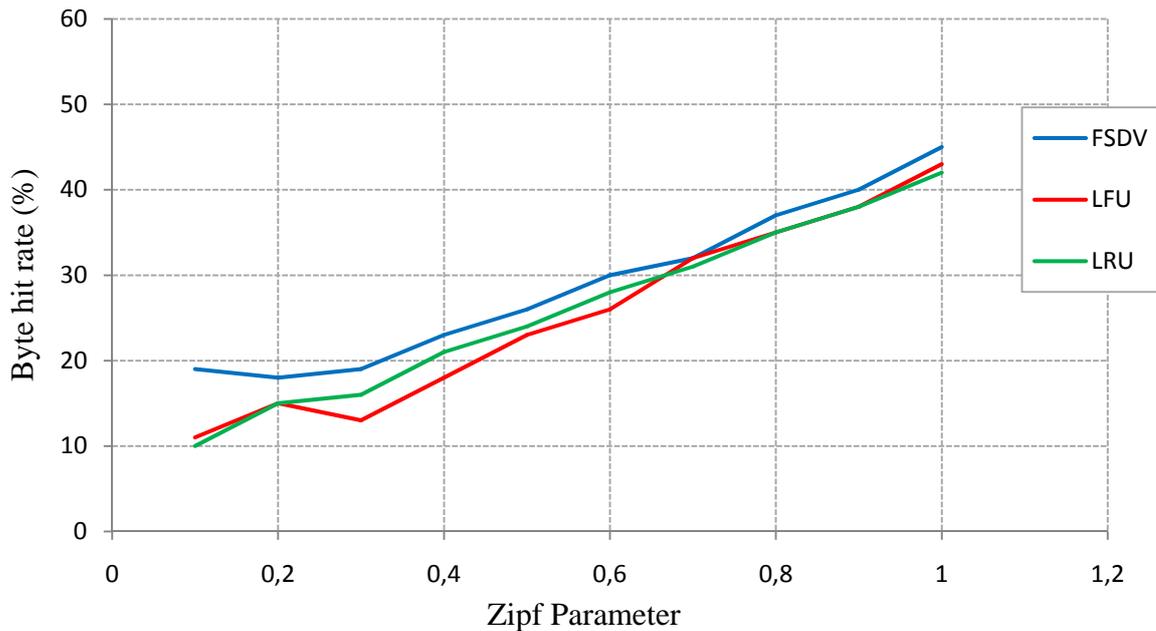
Dans le graphe de la figure 4.14, TDS donne un meilleurs taux de succès par rapport à ZC, car TDS ne se base pas sur la fréquence. Par conséquent, elle n'est pas influencée par les variations des popularités de documents, néanmoins FSDV donne un taux plus élevé.



**Figure 4.14 :** Taux de succès des politiques FSDV, TDS et ZC par rapport au paramètre de la loi Zipf.

#### 4.5.3.3 Taux de succès en octets de FSDV en comparaison avec LFU et LRU

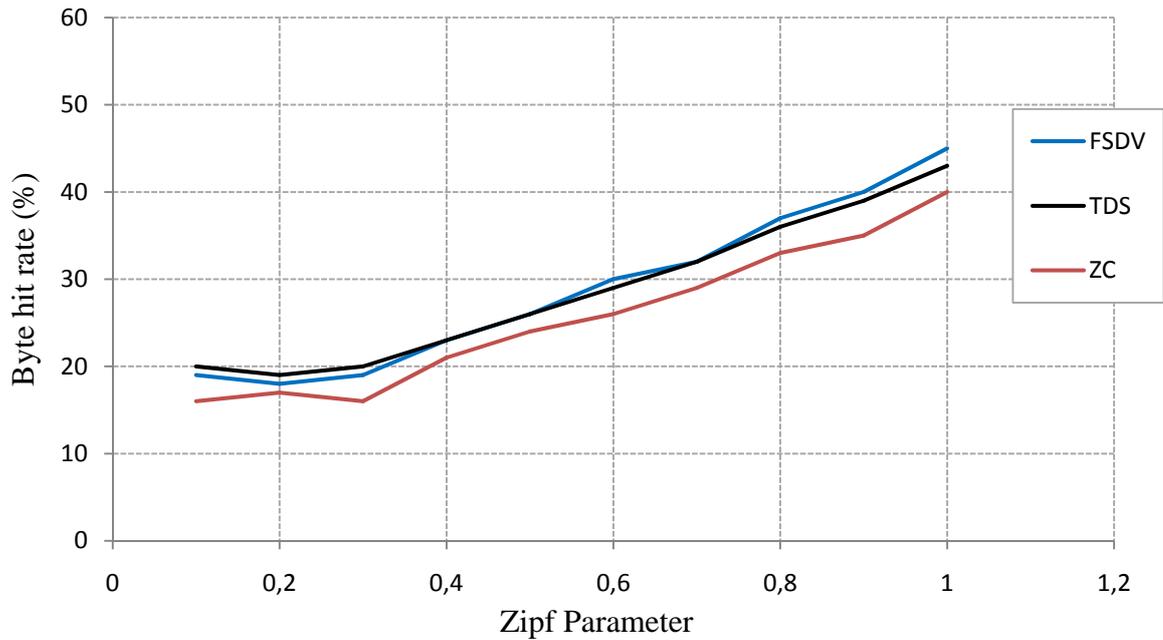
Dans le graphe de la figure 4.15, bien que FSDV favorise les grands documents pour le remplacement, on peut constater qu'elle fournit un taux de succès en octets meilleur que celui de LFU et LRU. En effet, remplacer un grand document par de multiples petits documents augmente le taux de succès. Par conséquent, cela augmente en rapport le taux de succès en octets.



**Figure 4.15 :** Taux de succès en octets des politiques FSDV, LFU et LRU par rapport au paramètre de la loi Zipf.

#### 4.5.3.4 Taux de succès en octets de FSDV en comparaison avec TDS et ZC

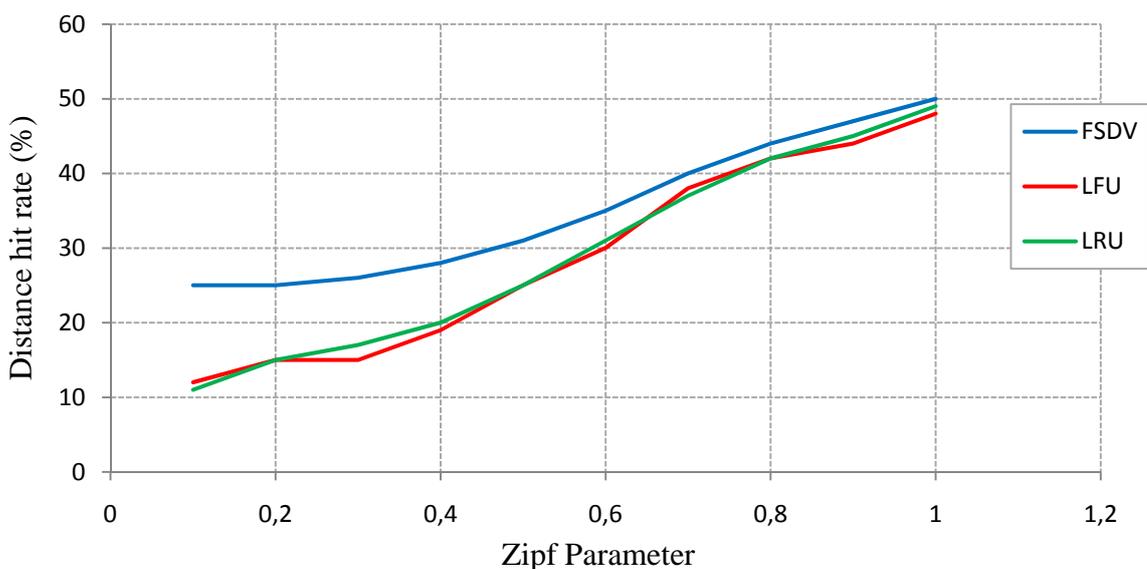
La figure 4.16 montre que pour les valeurs comprises entre 0.1 et 0.4 de paramètre de la loi zipf, la politique de remplacement FSDV donne un taux de succès en octets plus petit que celui donnée par TDS. Ceci peut être expliqué par le fait que TDS n'utilise pas la fréquence dans sa décision de remplacement. Cependant, FSDV est très influencée par les valeurs de popularité centrées (les popularités appartiennent à une plage de valeurs restreintes) générées par la loi de distribution Zipf pour les petites valeurs de son paramètre. En effet, dans ce cas les documents auront des popularités très proches. Par conséquent, les politiques de remplacement basées sur les fréquences donnent moins de performances.



**Figure 4.16 :** Taux de succès en octets des politiques FSDV, TDS et ZC par rapport au paramètre de la loi Zipf.

#### 4.5.3.5 Taux de succès en distance de FSDV en comparaison avec LFU et LRU

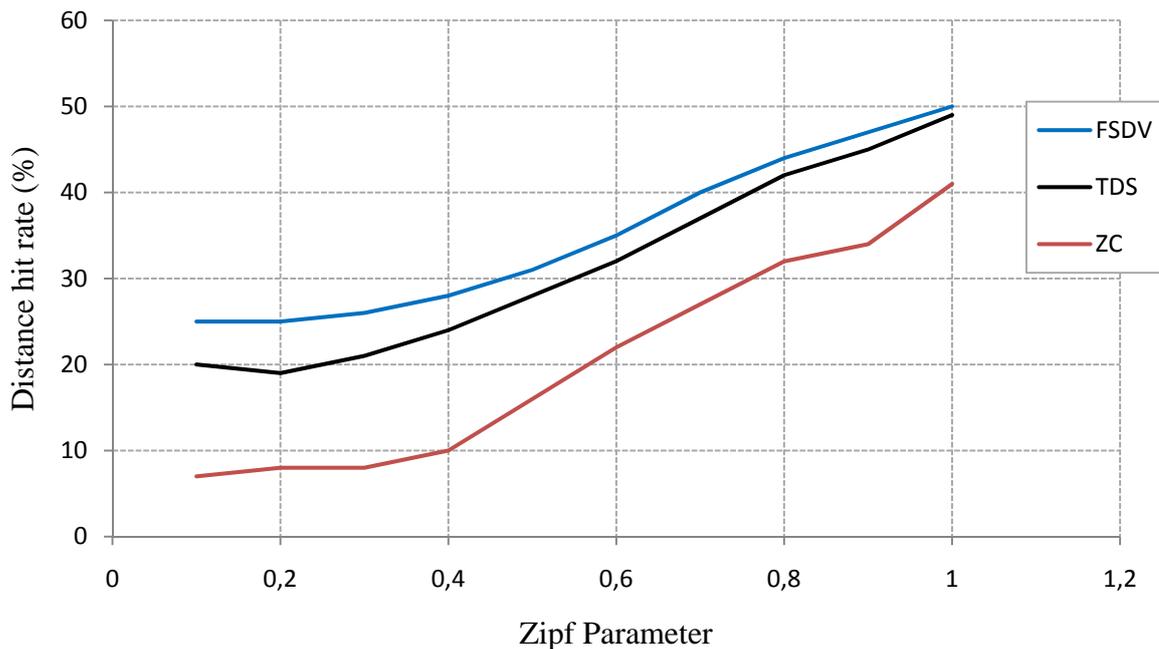
Les deux taux de succès en distance donnés par les deux politiques de remplacement de caches LFU et LRU dans la figure 4.17 sont très proches, mais ils sont plus petits que celui donné par la politique FSDV. Ceci peut être expliqué par le fait que les deux politiques classiques LFU et LRU ne prennent pas en considération le paramètre de distance.



**Figure 4.17 :** Taux de succès en distance des politiques FSDV, LFU et LRU par rapport au paramètre de la loi Zipf.

#### 4.5.3.6 Taux de succès en distance de FSDV en comparaison avec TDS et ZC

La figure 4.18 montre que la politique de remplacement de cache FSDV donne un meilleur taux de succès en distance par rapport aux deux politiques de remplacement de cache TDS et ZC et ce pour toutes les valeurs de paramètre de la loi de distribution Zipf, et cela malgré que TDS inclue le facteur de distance tout comme FSDV. Mais en plus de la distance, FSDV utilise la taille des documents qui est un facteur très important pour l'augmentation des taux de succès en général.



**Figure 4.18 :** Taux de succès en distance des politiques FSDV, TDS et ZC par rapport au paramètre de la loi Zipf.

## 4.6 Conclusion

Après l'étude de performance des Mécanismes de remplacement de caches des réseaux mobiles Ad Hoc, on peut conclure que les politiques de remplacement de caches classiques fournissent de moins bonnes performances et elles sont inadaptées pour ce type de réseaux, et ce, par le fait qu'elles se basent généralement sur un seul facteur dans leur décision de remplacement, comme le cas de LFU qui se base uniquement sur la fréquence.

Par conséquent, pour plus de performances, la conception des politiques de remplacement de caches destinées aux réseaux mobiles Ad Hoc ou aux environnements mobiles en général doit prendre en considération deux catégories de facteurs. La première

catégorie concerne les facteurs qui sont liés aux caractéristiques du trafic comme la fréquence des documents et leurs tailles. La deuxième catégorie concerne les caractéristiques spécifiques des réseaux mobiles Ad Hoc comme la variation des distances qui séparent les nœuds les uns des autres par le fait de la mobilité.

# Conclusion générale

Les problématiques posées dans le contexte particulier des réseaux Ad Hoc sont différentes et plus complexes que celles rencontrées dans le monde filaire. Pour cela, plusieurs travaux ont été réalisés ces dernières années dans le but d'apporter des solutions à ces problématiques. La plupart des recherches ([60], [91], [92], [93], [94], effectuées dans le domaine des réseaux Ad Hoc), se sont focalisées sur le développement des protocoles de routages dynamiques, qui peuvent améliorer la connectivité des nœuds mobiles. Cependant, d'autres questions telles que la disponibilité des données sont également très importantes puisque l'ultime but d'utilisation de tels réseaux est de garantir aux nœuds mobiles l'accès à toutes les informations désirées. Une des techniques les plus efficaces pour améliorer la disponibilité des données est l'utilisation de caches.

Dans le contexte des réseaux mobiles Ad Hoc, la technique d'utilisation de cache s'avère être une technique très efficace. Cependant, cette technique pose un certain nombre de problèmes liés à la gestion des caches, comme la consistance des données cachées et les techniques de remplacement de caches.

Dans ce travail, nous avons proposé une politique de remplacement de cache pour les réseaux mobiles Ad Hoc, que nous avons appelée *FSDV Frequency, Size and Distance based Value*, qui se base sur quatre paramètres, la fréquence d'accès, la distance en nombre de sauts séparant le nœud qui répond à une demande d'un document de celui qui le demande, le temps écoulé depuis la dernière mise à jour de cette même distance et la taille du document. Les résultats de simulation ont montré que FSDV donne de meilleures performances pour la plupart des métriques de performances, en particulier pour le taux de succès des requêtes et le taux de succès en distance. De même, nos résultats montrent que la meilleure politique pour les réseaux mobiles Ad Hoc est celle qui prend en considération les caractéristiques spécifiques de ces réseaux, ainsi que celles du trafic Web.

## **Perspective**

Vu le caractère dynamique de la topologie des réseaux mobiles Ad Hoc, qui entraîne de fréquentes déconnexions, et d'éventuelles partitionnements de réseaux, il sera intéressant, comme perspective, d'étendre notre proposition pour prendre en considération la coopération des caches. Cette technique se base sur le partage et la coordination des données cachées par des nœuds d'une même zone. De ce fait, si une source de données devient inaccessible, il sera possible de satisfaire les requêtes qui lui sont destinées par les caches des autres nœuds. Cependant, avec l'intégration de cette technique, plusieurs questions seront posées, comme la localisation des documents cachées et les remplacements coordonnés.

Pour justifier encore l'efficacité de cette proposition en terme de performances, il sera intéressant de la comparer à d'autres politiques de remplacement qui ont été proposées dans le cadre des environnements mobiles en général.

Il est également intéressant d'étendre cette proposition à la situation où chaque cache aura un état des informations des caches de sa zone, de telle sorte que la gestion des caches soit unifiée plutôt que locale ; autrement dit tous les caches d'une zone seront traités comme étant un même cache de grande taille.



# Glossaire

<b>ABR</b>	Associativity-Based Routing
<b>AODV</b>	Ad hoc On-demand Distance Vector
<b>AP</b>	Access Points
<b>BSIG</b>	Bluetooth Special Interest Group
<b>BWA</b>	Broadband Wireless Access
<b>CA</b>	Collision Avoidance
<b>CCK</b>	Complementary Code Keying
<b>CSMA/CA</b>	Carrier Sense Multiple Access / Collision Avoidance
<b>CSMA/CA</b>	Carrier Sense Multiple Access / Collision Detection
<b>CTS</b>	Clear To Send
<b>DARPA</b>	Defence Advanced Research Projects Agency
<b>DCF</b>	Distributed Coordinated Function
<b>DFS</b>	Dynamic Frequency Selection
<b>DIFS</b>	DCF Inter-Frame Space
<b>DSDV</b>	Dynamic destination-Sequenced-Distance Vector
<b>DSR</b>	Dynamic Source Routing
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>ETSI</b>	European Technical Standard Institute
<b>FHSS</b>	Frequency Hoping Spread Spectrum

---

<b>FSDV</b>	Frequency, Size and Distance based Value
<b>GDS</b>	Greedy Dual Size
<b>GSR</b>	Global State Routing
<b>IARP</b>	Intrazone Routing Protocol
<b>ICP</b>	Inter Cache Protocol
<b>IERP</b>	Interzone Routing Protocol, 21
<b>IP</b>	Internet protocol
<b>IR</b>	Infra Red
<b>ISM</b>	Industrial, Scientific and Medical
<b>LAN</b>	Local Area Network
<b>LFU</b>	Least Frequently Used
<b>LRU</b>	Least Recently Used
<b><i>LRUSM</i></b>	<i>LRU stack model</i>
<b>LRV</b>	Lowest Relative Value
<b>MAC</b>	Medium Access Control
<b>MANET</b>	Mobile Ad hoc NETWORKS
<b>NAV</b>	Network Allocation Vector
<b>MPR</b>	Multipoint Relay
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>OLSR</b>	Optimized Link State Routing
<b>PAN</b>	Personal Area Network
<b>PCF</b>	Point Coordination Function
<b>ProWGen</b>	Proxy Workload Generator
<b>PSS</b>	Pyramidal Selection Scheme
<b>QoS</b>	Quality of Service

<b>RREP</b>	Route REPlY
<b>RREQ</b>	Route REQuest
<b>RTS</b>	Request To Send
<b>SIFS</b>	Short Inter-Frame Space
<b>SLRU</b>	Segmented LRU
<b>SSA</b>	Signal Stability-based Adaptive routing
<b>SWLFU</b>	Server-Weighted LFU
<b>TDS</b>	Time and Distance Sensitive
<b>TPC</b>	Transmit Power Control
<b>TSP</b>	Taylor Series Prediction
<b>TTL</b>	Time To Live
<b>URL</b>	Uniform resource locator
<b>WEP</b>	Wired Equivalent Privacy
<b>WLAN</b>	Wireless Local Area Network
<b>WMAN</b>	Wireless Metropolitan Area Networks
<b>ZC</b>	Zone coopérative
<b>ZRP</b>	Zone Routing Protocol



# Bibliographie

- [1] Al Hanbali, A. Evaluation des performances des réseaux sans-fil mobiles, Thèse de Doctorat en Sciences, Université de Nice - Sophia Antipolis, INRIA 2006.
- [2] Jubin, J. and Tornow. J. D. The darpa packet radio network protocols. In Proceedings of the IEEE, Volume. 75, Janvier 1987.
- [3] Zussman G. and Segall, A. Energy efficient routing in ad hoc disaster recovery networks. In Proceedings of IEEE INFOCOM, San Francisco, USA, pages. 405-421, 2003.
- [4] Miraihi Naima, A. Délai et routage dans les réseaux ad hoc 802.11, Thèse de Doctorat, Université de Versailles Saint-Quentin-En-Yvelines 2005.
- [5] Das, S. Perkins, C. Belding-Royer, E. RFC 3561, Ad hoc On-demand Distance Vector (AODV) Routing [en ligne], Memo. The Internet Society. 2003. Disponible sur : <http://www.ietf.org/rfc/rfc3561.txt>.
- [6] Percher, J. Jouga, B. Détection d'intrusions dans les réseaux ad hoc. SSTIC'03, 1er Symposium sur la Sécurité des Technologies de l'Information et de la Communication. Rennes, juin 2003.
- [7] Broadband Radio Access Networks (BRAN). High Performance Radio Local Area Network (HIPERLAN) type 1. functional specification. Technical Report EN 300 652 ref. REN/BRAN-10-01, ETSI, 1998.
- [8] Broadband Radio Access Networks (BRAN). High Performance Radio Local Area Network (HIPERLAN) type 2. requirements and architectures for wireless broadband access. Technical Report TR 101 031 V2.2.1 (1999-01) ref. RTR/BRAN- 0022001, ETSI, 1999.
- [9] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical Report ANSI/IEEE Std 802-11- 1999 Edition, IEEE, 1999.
- [10] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 GHz band. Technical Report IEEE Std 802.11b-1999 (R2003), IEEE, 2003.

- [11] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed physical layer in the 5 GHz band. Technical Report IEEE Std 802-11a-1999(R2003), IEEE, ISO/IEC 8802- 11:1999/Amd 1:2000(E). 2003.
- [12] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 4: Further higher data rate extension in the 2.4 GHz band. Technical Report IEEE Std 802.11g-2003, IEEE, 2003.
- [13] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. Technical Report IEEE Std 802.11 e™-2005, IEEE, 2005.
- [14] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications : Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz band in Europe. Technical Report IEEE Std 802.11h™-2003, IEEE, 2003.
- [15] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) security enhancements. Technical Report IEEE Std 802.11i-2004, IEEE, 2004.
- [16] D. Dhoutaut, Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation. Thèse Doctorat, Institut National des Sciences Appliquées de Lyon. 2003.
- [17] Jacquet, P. and Clausen, T. RFC 3626, Optimized Link State Routing Protocol (OLSR), Memo. The Internet Society. 2003. Disponible sur : <http://www.ietf.org/rfc/rfc3626.txt>.
- [18] Johnson, D. B., Maltz, D. A. and Hu, Y. The Dynamic Source Routing Protocol for Mobile Ad Hoc Net- works (DSR). Draft IETF, juillet 2004. Disponible sur : <http://tools.ietf.org/id/draft-ietf-manet-dsr-10.txt>.
- [19] Haas, Z. and Pearlman. M. The Zone Routing Protocole (ZRP) for Ad Hoc Net- works. IETF MANET Draft, Juin 1999. Disponible sur : <http://tools.ietf.org/id/draft-ietf-manet-zone-zrp-04.txt>.
- [20] Toh, C-K. Associativity-Based Routing for Ad hoc Mobile Networks. Wireless Personal Communications Journal, Volume. 4, no. 2, pages. 1-36, Mars 1997. Disponible sur : <http://monet.knu.ac.kr/~cktoh/data/ABRjj.pdf>
- [21] Dube, R. Rais, C. D. Wang, K. Y. and Tripathi, S. K. Signal Stability-Based Adaptive Routing for Ad hoc Mobile Networks. IEEE Personal Communications Magazine, pages. 36-45, février 1997. Disponible sur : <http://pdos.csail.mit.edu/decouto/papers/dube97.pdf>
- [22] Chiang, C. Wu, H. K. Liu, W. and Gerla, M. Routing in Clustered Multi- Hop Mobile Wireless Networks with Fading Channel. Proceedings of IEEE SICON 1997, pages. 197-211, Avril 1997. Disponible sur : <http://www.eecs.wsu.edu/~smedidi/teaching/Spring05/CSGR.pdf>

- [23] Iwata, A. Chiang, C. Pei, G. Gerla, M. and Chen, T. Scalable Routing Strategies for Ad hoc Wireless Networks. IEEE Journal on Selected Areas in Communication, Volume. 17,no. 8, pages. 1369-1379, Août 1999. Disponible sur : <http://citeseer.ist.psu.edu/cache/papers/cs/8276/http:zSzzSzwww.cs.ucla.eduzSzNRLzSzwirelesszSzPAPERzSzjsac99.pdf/iwata99scalable.pdf>
- [24] Perkins, C. Bhagwat, P. Highly dynamic destination-sequenced-distance vector routing (DSDV) for mobile computers. ACM SIGCOMM, Volume. 24, no. 4, Octobre 1994, pages. 234-244. Disponible sur : <http://www.cs.virginia.edu/~cl7v/cs851-papers/dsdv-sigcomm94.pdf>
- [25] Chen, T. and Gerla, M. Global State Routing: A New Routing Scheme for Ad hoc Wireless Networks. Proceedings of IEEE IC 1998, pages. 171-175, Juin 1998. Disponible sur : <http://citeseer.ist.psu.edu/cache/papers/cs/696/http:zSzzSzwww.cs.ucla.eduzSzNRLzSzwirelesszSzPAPERzSzgsr.pdf/chen98global.pdf>
- [26] Chiang, C. Wu, H. K. Liu, W. and Gerla, M. Routing in Clustered Multi-Hop Mobile Wireless Networks with Fading Channel. Proceedings of IEEE SICON 1997, pages. 197-211, Avril 1997. Disponible sur : <http://www.eecs.wsu.edu/~smedidi/teaching/Spring05/CSGR.pdf>.
- [27] Park, V. and Corson, M. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proceedings of IEEE InfoCom, pages. 1405-1413, Avril 1997. Disponible sur : <http://cs-www.cs.yale.edu/homes/arvind/cs425/doc/tora.pdf>
- [28] Joa-Ng, M. and Lu, I. A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad hoc Networks. IEEE Journal on Selected Areas in Communications, Volume. 17, no. 8, pages. 1415-1425, Août 1999.
- [29] Jangeun J., Peddabachagari, P. and Sichitiu, M. Theoretical maximum throughput of IEEE 802.11 and its applications. Network Computing and Applications, 2003. Second IEEE International Symposium on. North Carolina State Univ, USA. Pages. 249- 256. Avril 2003.
- [30] Perkins. C. E. Ad hoc networking. Addison-Wesley, 2001.
- [31] Bradley, D. Marwood, D. and Feely, M. The Measured Access Characteristics of World-Wide-Web Client Proxy Caches. USENIX Symposium on Internet Technologies and Systems, Décembre 1997.
- [32] Sailhan, F., and Issarny, V. Cooperative Caching in Ad Hoc Networks. International Conference on Mobile Data Management, pages.13-28. 2003.
- [33] Silberschatz, A. and Galvin, P. B. Operating Systems Concepts. Addison Wesley, Reading, MA, fourth edition, 1994.

- 
- [34] Leey, R. Goshimay, K. Kambayashiy, Y. and Takakuraz, H. Caching Schema for Mobile Web Information Retrieval. Proceedings of the 2nd International Workshop on Web Dynamics, Part of WWW 2002, Mai 2002.
- [35] Abrams, M. Standridge, C. R. Abdulla, G. Wiliams, S. and Fox, E. A. Caching Proxies: Limitations and Potentials. Proceedings Of the 4th International WWW Conference, Boston, December 1995.
- [36] Aggarwal, C. Wolf, J.L. and Yu, P.S. Caching on the World Wide Web, IEEE Transactions on knowledge and data engineering, 11(1), 1999.
- [37] Ren, Q. and Dunham, M.H. Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. The 6th Annual International Conference on Mobile Computing and Networking. août 2000.
- [38] Zheng, B. and Lee, D. L. Semantic Caching in Location-Dependent Query Processing. The 7th International Symposium on Spatial and Temporal Databases, Springer-Verlag, LNCS 2121, pages. 97-113, 2001.
- [39] Sailhan, F. and Issarny, V. Cooperative Caching in Ad Hoc Networks, in: Proceedings of 4th International Conference on Mobile Data Management (MDM), pages. 13–28. 2003.
- [40] Víctor J. Sosa, Leandro Navarro, Influence of the Document Validation/Replication Methods on Cooperative Web Proxy Caching Architectures. 2002.
- [41] Krishnamurthy, B. and Wills, C.E. Study of piggyback cache validation for proxy caches in the World Wide Web. In Proceedings of the USENIX Symposium on Internet Technologies and Systems, pages. 1–12, 1997.
- [42] Gray, C. G., and Cheriton, D. R. (Leases: An efficient fault-tolerant mechanism for distributed file cache consistency. In Proceedings of the 12th ACM Symposium on Operating System Principles, pages. 202–210. 1989.
- [43] Arlitt, M., Friedrich, R., and Jin, T. Workload characterization of a Web proxy in a cable modem environment. Technical Report HPL-1999-48, Hewlett Packard Labs 1999.
- [44] Dilley, J., Arlitt, M., Perret, S., and Jin, T. The Distributed Object Consistency Protocol. Technical Report HPL-1999-109, Hewlett-Packard Labs, 1999.
- [45] Arlitt, M. Dilley, J. Improving Proxy Cache Performance-Analyzing Three Cache Replacement Policies. Internet Systems and Applications Laboratory HP Laboratories Palo Alto HPL-1999-142 Octobre, 1999.
- [46] Podlipnig, S. Boszormenyl, L. A Survey of Web Cache Replacement Strategies. ACM Computing Surveys, Volume.35, no.4, pages.374-398, 2003.
- [47] Rabinovich, M. Spatscheck, O. Web Caching and Replication. Addison-Wesley, Reading, MA. 2002.
- [48] Wessels, D. Web Caching. O'Reilly, Sebastopol, CA. 2001.

- [49] Reddy, M. and Fletcher, G. P. Intelligent Web caching using document life histories: A comparison with existing cache management techniques. In Proceedings of the 3rd International Web Caching Workshop. Manchester, Angleterre. Juin 1998.
- [50] Vakali, A. LRU-based algorithms for Web cache replacement. In International Conference on Electronic Commerce and Web Technologies. Lecture Notes in Computer Science, Volume. 1875. Springer-Verlag, Berlin, Allemagne, pages. 409–418. 2000.
- [51] Hosseini-Khayat, S. Investigation of generalized caching. Ph.D. dissertation. Washington University, St. Louis, MO. 1997.
- [52] Arlitt, M. F., Cherkasova, L., Dilley, J., Friedrich, R. J., and Jin, T. Y. Evaluating content management techniques for Web proxy caches. ACM SIGMETRICS Perform. Eval. Rev. 27, 4 (Mar.), pages. 3–11. 2000.
- [53] Zhang, J., Izmailov, R., Reininger, D., and Ott, M. Web caching framework: Analytical models and beyond. In Proceedings of the IEEE Workshop on Internet Applications. IEEE Computer Society, Piscataway, New Jersey, USA. 1999.
- [54] Kelly, T., Jamin, S., and Mackie-Mason, J. K. Variable QoS from sharedWeb caches: User centered design and value-sensitive replacement. In Proceedings of the MIT Workshop on Internet Service Quality Economics. 1999.
- [55] Williams, S., Abrams, M., Standridg, C. R., Abdulla, G., and Fox, E. A. Removal policies in network caches for World Wide Web documents. In Proceedings of ACM SIGCOMM. ACM Press, New York, USA, 293–305. 1996.
- [56] Arlitt, M., Friedrich, R., and Jin, T. Y. Performance evaluation of Web proxy cache replacement policies. Tech. rep. HPL-98-97(R.1), Hewlett-Packard Company, Palo Alto, Santa Clara, USA. 1999.
- [57] Menaud, J.-M., Issarny, V. and Banatre, M. Improving effectiveness of Web caching. In Recent Advances in Distributed Systems. Lecture Notes in Computer Science, Volume. 1752. Springer-Verlag, Berlin, Allemagne, 375–401. 2000.
- [58] Wooster, R.P. and Abrams, M. Proxy caching that estimates page load delays. In Proceedings of the 6th InternationalWorldWideWeb Conference. Santa Clara, USA, 1997.
- [59] Cao, P. and Irani, S. Cost-aware WWW proxy caching algorithms. In Proceedings of the USENIX Symposium on Internet Technologies and Systems. Pages. 193–206. 1997.
- [60] Jin, S. and Bestavros, A. GreedyDual\*: Web caching algorithms exploiting the two sources of temporal locality in Web request streams. In Proceedings of the 5th International Web Caching and Content Delivery Workshop. Volume.24, pages.174-183, 2000.

- [61] Yang, Q. Zhang, H. H. and Zhang, H. Taylor series prediction: A cache replacement policy based on second-order trend analysis. In Proceedings of the 34th Hawaii International Conference on Systems Sciences. IEEE Computer Society, Piscataway, New Jersey, USA. 2001.
- [62] Rizzo, L. and Vicisano, L. Replacement policies for a proxy cache. *IEEE/ACM Trans. Netw.* 8, 2 (Apr.), 158–170. 2000.
- [63] Krishnamurthy, B. and Rexford, J. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*. Addison-Wesley, Reading, MA. 2001.
- [64] Benhamida, N. Sur les politiques de gestion du cache d'un serveur web. Mémoire de Magistère en Informatique de l'Université A/Mira de Bejaia. 2007.
- [65] Kingsley, G. Zipf. Relative frequency as a determinant of phonetic change. Reprinted from the *Harvard Studies in Classical Philology*, XL, 1929.
- [66] Almeida, V. Bestavros, A. Crovella, M. and Oliveira, A. Characterizing reference locality in the WWW. In *IEEE International Conference in Parallel and Distributed Information Systems*, Miami Beach, Florida, USA, Décembre 1996. Disponible sur : <http://www.cs.bu.edu/groups/oceans/papers/Home.html>
- [67] Cunha, C. Bestavros, A. and Crovella, M. Characteristics of WWW client-based traces. Technical Report TR-95-010, Boston University, Computer Science Dept., Boston, MA 02215, USA, Avril 1995. Disponible sur : <http://www.cs.bu.edu/groups/oceans/papers/Home.html>.
- [68] Glassman, S. A caching relay for the world wide web. In *First International Conference on the World Wide Web*, CERN, Genève, Suisse, Mai 1994. Disponible sur : <http://www1.cern.ch/WWW94/PrelimProcs.html>.
- [69] Busari, M. and Williamson, C. On the sensitivity of web proxy cache performance to workload characteristics. In *Proceedings of IEEE INFOCOMM*, pages. 1225-1234, avril 2001.
- [70] Busari, M. and Williamson, C. Prowgen : A synthetic workload generation tool for simulation evaluation of web proxy caches. *Computer Networks* Volume. 38, No. 6, pages. 779-794, Juin 2002.
- [71] Philippe, G. Breslau, L. Fan, L. Cue, P. and Shenker, S. Web caching and zipf-like distribution : Evidence and applications. In *Proceedings of IEEE INFOCOMM '99*, 18th Annual Joint Cnf. of the IEEE Computer and Communications Societies, pages.126-134, Mars 1999.
- [72] Mahanti, A. Williamson, C. and Eager, D. Traffic analysis of a web proxy caching hierarchy. *IEEE Network*, Volume. 14, no. 3, pages. 16-23, Mai / Juin 2000.
- [73] Paxson, V. and Floyd, S. Why we don't know how to simulate the Internet. In *Proceedings of the 1997 Winter Simulation Conference*, pages. 1037–1044. 1997.

- [74] Douglis, F., Feldmann, A., Krishnamurthy, B. and Mogul, J. Rate of change and other metrics: A live study of the World Wide Web. In Proceedings of the USENIX Symposium on Internet Technologies and Systems, pages. 147–158. 1997.
- [75] Wolman, A., Voelker, G., Sharma, N., Cardwell, N., Brown, M., Landray, T., Pinnel, D., Karlin, A. and Levy, H. Organization-based analysis of Web-object sharing and caching. In Proceedings of the USENIX Symposium on Internet Technologies and Systems, pages. 25–36. 1999.
- [76] Lim, S. Lee, W.-C. Cao, G. and Das, C. R. A novel caching scheme for improving internet-based mobile ad hoc networks performance. in ICCCN' 03:IEEE International Conference on Computer Communications and Networks, Dallas, TX, U.S.A., pages. 38–43. 2003.
- [77] Chand, N. Joshi R. C. and Misra, M. Cooperative cache managment in mobile ad hoc networks, Indian Institute of Technology Roorkee, India. 2005.
- [78] Li, w. Chan E. and Chen, D. Energy-efficient Cache Replacement Policies for Cooperative Caching in Mobile Ad Hoc Network. IEEE Wireless Communications and Networking Conference, 2007.
- [79] Acharya, S. “Broadcast Disks: Dissemination-Based Data Management for Asymmetric Communication Environments. PhD dissertation, Brown Univ., Mai 1998.
- [80] Xu, J. Hu, Q.L. Lee, D.L. and Lee, W.C “Performance Evaluation of an Optimal Cache Replacement Policy for Wireless Data Dissemination“, IEEE Transactions On Knowledge and Data Engineering, Volume. 16, NO. 1, Janvier 2004.
- [81] Chuang, J. and Sirbu, M. Adding quality of service to network storage. In Proceedings of the Workshop on Internet Service Quality Economics. 1999.
- [82] Cherkasova, L. and Ciardo, G. Role of aging, frequency and size in Web caching replacement strategies. In Proceedings of the 2001 Conference on High-Performance Computing and Networking (HPCN'01). Lecture Notes in Computer Science, Volume. 2110. Springer-Verlag, Berlin, Allemagne, 114–123. 2001.
- [83] Yeung, K. H., Wong, C. C. and Wong, K. Y. A cache replacement policy for transcoding proxy servers. In Proceedings of World Multiconference on Systemics, Cybernetics and Informatics. Volume. 12, 234–237. 2001.
- [84] Bolot, J. and Hoschka, P. Performance engineering of the World Wide Web: Application to dimensioning and cache design. In Proceedings of the 5th International World Wide Web Conference. Elsevier, Amsterdam, Pays-bas. 1996.
- [85] Almeida, V. Cesario, Fonseca, M. R. Meira, W. and Murta, C. Analysing the behavior of a proxy server in light of regional and cultural issues. In Proceedings of the Third International WWW Caching Workshop, Manchester, Angleterre., Juin 1998.

- [86] Mahanti A. and Williamson, C. Web proxy workload characterization. technical report, Department of Computer Science, University of saskatchewan, Février 1999.
- [87] Wang, J. Z. Pal, A. and Srimani, P. K. New Efficient Replacement Strategies for P2P Cooperative Proxy Cache Systems. Proc. Designs, Analysis, and Simulation of Distributed Systems, pages. 97-103, 2004.
- [88] Karedla, R. Love, J. and Wherry, B. Caching strategies to improve disk system performance. IEEE Computer, Volume. 27, No. 3, pages. 38-46, Mars 1994.
- [89] O'Neil, E. O'Neil, P. and Weikum, G. The Iru-k page replacement algorithm for database disk buffering. In Proceedings of SIGMOD '93, pages. 297-306, Washington, DC, Mai 1993.
- [90] Frodigh, M., Johansson, P. and Larsson, L. Wireless Ad Hoc Networking – The Art of Networking Without a Network, Ericsson Review, No. 4. 2000 ;
- [91] Das, S., Perkins, C. and Royer, E. Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks , IEEE INFOCOM, pages. 3-12. 2000.
- [92] Johnson, D. and Maltz, D. Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing, pages. 158-181, 1996.
- [93] Perkins, C. and Bhagwat, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, ACM SIGCOMM, pages. 234-244. 1994.
- [94] Perkins, C. and Royer, E.M. Ad Hoc On- Demand Distance Vector Routing , IEEE Workshop on Mobile Computing Systems and Applications, pages. 90- 100. 1999.
- [95] Yin, L. and Cao, G. Supporting Cooperative Caching in Ad Hoc Networks , IEEE INFOCOM, pages. 2537-2547. 2004.