N°d'ordre :02/2007-M/IN

REPUBLIQUE ALGERIENNE DECMOCRATIQUE ET POPULAIRE
MINISTERE D'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TSCHNOLOGIE
« HOURI BOUMEDIEN »
FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER
En : INFORMATIQUE

Spécialité : Programmation et Systèmes

Par : CHEHBOUR Fairouz.

Sujet

Evaluation de Performances de Protocole de Validation Atomique pour les Environnements Mobiles et sans Fils

Soutenu le 17/02/2007,devant le jury composé de :

Mr. N.BADACHE	Professeur, USTHB,	Président
Mr. O.NOUALI	Dr. Maître de Recherche, CERIST	Directeur de thèse
Mr. M .AHEMED NACER	Professeur, USTHB,	Examinateur
Mme M.BOUKALA	Professeur, USTHB,	Examinateur
Mme. N. NOUALI	Chargée de Recherche, CERIST	Invitée

Remerciements

Je tiens à remercier vivement mon Dieu, le tout puissant, de m'avoir donné le courage et la patience jusqu'à l'achèvement de ce travail.

Je remercie le professeur Abdelkader Khelladi, directeur du CERIST, de m'avoir permis de m'inscrire en première année poste graduation, et pour tous les moyens qu'il a mis à ma disposition.

Je remercie le Professeur Najib Badache, qui me fait l'honneur de présider ce jury, Mr Ahmed NacerB et Mme M. Boukala, Professeurs à l'université de l'USTHB, d'avoir accepté d'examiner ce travail, qu'ils trouvent ici l'expression de ma reconnaissance.

J'exprime ma profonde reconnaissance et mes vifs remerciements à Mme Nouali Nadia, chargée de recherche au CERIST, de m'avoir fait confiance en me proposant ce sujet, pour les lectures attentives, les critiques et suggestions qui ont été d'un grand apport pour la réalisation de ce travail, ainsi que Mr Nouali Omar, Maître de recherche au CERIST, qui a accepter d'être mon directeur de thèse.

Je remercie mon mari pour sa patience et ses encouragements tout au long de ce parcours.

Un grand MERCI, aux membres de ma famille, aux membres de ma belle famille et à mes très chers amis pour leurs encouragements, leur patience et leur amour.
Tous ceux qui ont aidé et contribué à la réalisation de ce travail.

Dédicaces

Je dédie ce travail à tous ceux qui me sont chers…

Ma mère;
Mon mari;
Mes frères et sœurs;
Et Tous mes amis.

Sommaire

Introduction générale

Dans les systèmes de bases de données traditionnels, les utilisateurs interagissent avec la base de données en utilisant des transactions. Une transaction est une unité de traitement qui permet d'assurer la cohérence des données et qui doit satisfaire quatre propriétés, appelées propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité) [BHG 87]. L'idée de base d'une transaction est l'indivisibilité, qui annonce que soit toutes les opérations de la transaction sont complètement exécutées soit aucune ne l'est, et les résultats intermédiaires d'une transaction ne sont pas visibles aux autres transactions. On parle de transaction répartie ou distribuée pour qualifier une transaction dont les opérations manipulent des données distribuées sur plusieurs sites. Pour terminer une transaction, tous les sites participants doivent coordonner leurs actions afin de garantir l'atomicité globale de cette dernière. Cette coordination, appelée *validation atomique (Atomic Commitment)*, assure qu'une transaction est validée ou abandonnée de façon uniforme sur l'ensemble des sites même en présence de pannes. La validation atomique est traditionnellement mise en œuvre par le protocole de validation à deux phases (Two Phase Commit ou 2PC) [Gra 87] et ses optimisations.

2PC et ces variantes ou optimisations ont été conçus pour des systèmes fixes plus stables et plus robustes, cependant, le paysage informatique actuel se caractérise par une interconnexion massive et évolutive de calculateurs hétérogènes, géographiquement distants et potentiellement mobiles donnant naissance à un nouvel environnement de calcul appelé environnement mobile. La mobilité implique (1) la portabilités des calculateurs. Cette caractéristique impose aux constructeurs d'alléger le poids des calculateurs mobiles, ce qui désavantage la capacité de traitement (CPU), de stockage (mémoire) et la source d'énergie (batterie), (2) la communication avec un calculateur mobile peut s'effectuer à travers des liens sans fil qui se caractérisent par une faible bande passante offrant une connexion intermittente, augmentant ainsi la latence, le taux d'erreurs et le coût des communications. (3) la mobilité des utilisateurs avec leurs terminaux a induit de nouveaux challenges liés à la localisation des entités communicantes. A cause des caractéristiques d'un tel environnement, le problème de la validation atomique des transactions a motivé de nouveaux travaux de recherche durant ces dernières années. Les nouvelles propositions tentent d'optimiser l'utilisation des ressources des unités mobiles, de surmonter les limitations des réseaux sans fil (déconnexions et faible bande passante) et de pallier aux problèmes induits par la mobilité.

On distingue deux approches de résolution du problème de la validation atomique en environnement mobile. La première repose sur le fait que l'atomicité traditionnelle qualifiée de "stricte" soit considérée comme inadéquate pour le calcul mobile, ce qui implique le rejet du protocole 2PC et la conception de nouveaux protocoles. Les ACPs (Atomic Commitment Protocoles) basés sur cette approche reposent sur la validation optimiste et tolèrent souvent

l'atomicité "sémantique"[1] qui est une notion relâchée de l'atomicité [KDDS 00][KPDS 02][Ser 04]. La deuxième approche propose d'adapter 2PC et ses variantes en palliant aux problèmes de l'environnement mobile [LW 99] [PB 99].

Dans ce mémoire, nous nous intéressons donc à la validation atomique de transactions mobiles. Plus précisément notre travail consiste en l'évaluation d'un protocole à deux phases, Mobile Two Phase Commit ou M-2PC par la simulation. M-2PC a été proposé dans [NDD 05a] comme une adaptation du protocole 2PC à la mobilité. A la différence des autres ACPs qui supposent que la mobilité des clients et/ou des participants est prise en charge par un mécanisme de gestion de mobilité implémenté dans les couches basses de la pile protocolaire, M-2PC gère la mobilité des clients ou des participants de façon intrinsèque.

Nous avons réalisé une comparaison entre M-2PC qui propose d'inclure le mécanisme de mobilité au niveau applicatif avec une implémention du 2PC traditionnel au-dessus du support de mobilité de niveau réseau; à savoir le Mobile-IP. Nous avons réalisé notre évaluation avec NS-2 et les résultats obtenus nous ont permis de mettre en évidence les points forts aussi que les faiblesses d'une telle approche. Précisons que notre analyse a porté sur l'aspect mobilité proprement dite.

La suite de ce document est organisée comme suit :

- Chapitre 1 : présente les concepts de base sur les systèmes transactionnels en mettant l'accent sur le problème de la validation atomique.
- Chapitre 2 : présente les problèmes que rencontre un protocole de validation atomique classique s'il est exécuté dans un environnement de calcul mobile et donne un aperçu de quelques solutions proposées pour résoudre ces problèmes.
- Chapitre 3 : nous présentons dans ce chapitre quelques approches qui permettent de gérer la mobilité au niveau de la couche application. La recherche bibliographique effectuée dans ce domaine, nous a permis de constater la faisabilité de cette approche et d'avoir une idée sur sa performance.
- Chapitre 4 : présente une étude qualitative du protocole M-2PC et présente des propositions afin de le rendre plus performant.
- Chapitre 5 : évalue les performances de ce protocole en terme de latence; ce paramètre affecte significativement la qualité de services des applications.
- Une conclusion générale récapitule notre travail et présente quelques propositions pour l'implémentation du protocole M-2PC et une extension au travail réalisé dans ce mémoire.

[1] On parle d'atomicité sémantique lorsque les résultats d'une sous transaction valide peuvent être consultés par d'autres transactions.

Chapitre 1

La Validation Atomique : concepts de base

Introduction

La notion de transaction a été introduite dans les années soixante dix comme moyen de faire partager aux utilisateurs des informations et pour assurer une gestion de données consistante et des interactions atomiques et isolées entre utilisateurs [Fri 01][Nou 01][WV 02]. Une transaction est une séquence d'opérations qui, lorsqu'elle est exécutée seule dans un environnement sans panne, fait passer le système d'information d'un état cohérent initial à un état cohérent final [Gra 81]. L'idée de base d'une transaction est l'indivisibilité, qui annonce que soit toutes les opérations de la transaction sont complètement exécutées soit aucune ne l'est, et les résultats intermédiaires d'une transaction ne sont pas visibles aux autres transactions. On parle de transaction répartie ou distribuée pour qualifier une transaction dont les opérations manipulent des données distribuées sur plusieurs sites. Les sites accédés sont appelés *participants* à la transaction. Pour terminer une transaction, tout les participants doivent coordonner leurs actions afin de garantir l'atomicité globale de cette dernière. Cette coordination, appelée *validation atomique (Atomic Commitment)*, assure qu'une transaction est validée ou abandonnée de façon uniforme sur l'ensemble des sites même en présence de pannes [AP 98]. Ce chapitre introduit les concepts de base des systèmes transactionnels et de la validation atomique.

1 Systèmes transactionnels

3.1 Propriétés d'une transaction

Une transaction est une unité de programme s'exécutant au sein d'un système qui permet aux utilisateurs d'accéder à des données partagées. Afin d'assurer la cohérence de ces données, même en cas de défaillance, le système transactionnel doit vérifier les propriétés, connues comme propriétés d'ACIDité (Atomicité, Cohérence, Isolation, Durabilité), suivantes [GR 93]:

- Atomicité : si toutes les actions de la transaction sont exécutées, les effets de l'exécution sont traduits sur la base de données et la transaction est dite *validée*. Sinon aucun effet n'est retenu et la transaction est *rejetée.*
- Cohérence ou Consistance : une transaction exécutée dans un environnement sans concurrence et sans pannes, transforme la base de données d'un état cohérent à un autre état cohérent.
- Isolation : les effets intermédiaires d'une transaction en exécution ne sont pas visibles à d'autres transactions *concurrentes*[2].
- Durabilité ou permanence : les modifications faites par des transactions bien terminées sont conservées, même après défaillance.

Si toutes ces propriétés sont vérifiées, on parle de cohérence forte ou stricte. Si une de ces propriétés n'est pas vérifiée la cohérence est dite faible. Si toutes les données du système transactionnel se trouvent physiquement dans une seule base de données, le système est dit centralisé. Sinon il est dit réparti.

3.2 Système transactionnel centralisé

La gestion des données dans un système transactionnel centralisé est accomplie par les modules suivants (figure 1(a)) [BHG 87]:

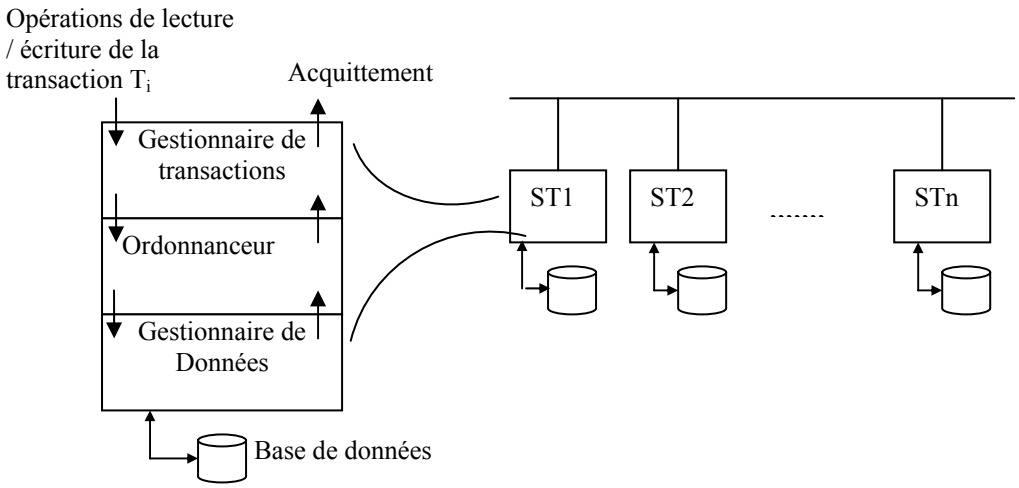
- Le gestionnaire de transactions, qui d'une part reçoit les opérations des transactions des utilisateurs et les soumet à l'ordonnanceur, et d'autre part, acquitte l'exécution de chacune des opérations.
- L'ordonnanceur, qui contrôle l'ordre d'exécution des opérations des transactions. L'ordre d'exécution d'un ensemble d'opérations n'est pas forcement celui de la réception de ces opérations.
- Le gestionnaire de données qui fait en sorte que touts les effets des transactions validées sont traduit sur la base de données. Il traite les opérations de lecture/écriture ordonnancées. De même, les demandes de validation et d'annulation des transactions.
- La base de données.

3.3 Systèmes transactionnels répartis

Un système réparti est défini par un ensemble de processus, où chaque processus est hébergé par un site. Les processus ne communiquent qu'en échangeant des messages par des canaux de communication reliant les sites et ne partagent ni mémoire ni horloge globale. Les données d'un système transactionnel réparti sont distribuées de manière telle que chaque site du système en possède une partie. Une transaction initiée dans un site, peut impliquer l'accès

[2] On considère que deux transactions sont concurrentes s'il existe un instant du temps dans lequel ces transactions sont actives simultanément

aux données hébergées dans les autres sites du système, une telle transaction étant dite *transaction répartie*. L'architecture d'un tel système est représentée dans la figure 1(b).



(a) ST centralisé (b) ST réparti

Figure 1 – Architecture centralisée et répartie des systèmes transactionnels

La gestion d'un système transactionnel, centralisé ou réparti, est garantie par un ensemble d'algorithmes qui permettent de contrôler la concurrence, le recouvrement des exécutions et l'atomicité des transactions.

3.4 Algorithmes de contrôle de concurrence

Le contrôle de concurrence est l'activité d'ordonnancement des accès concurrents à des données partagées de façon à garantir la propriété d'isolation [BG 81] ou de respecter la théorie de sérialisabilité[3]. Deux classes d'algorithmes de contrôle de concurrence sont brièvement présentés : les algorithmes pessimistes et les algorithmes optimistes.

Algorithmes pessimistes

L'objectif des algorithmes pessimistes est d'éviter la production de toute exécution concurrente des opérations en conflit[4]. Dans cette catégorie on trouve notamment le protocole de *verrouillage à deux phases*, ou 2PL (Tow Phase Locking) et le protocole *d'estampillage de transactions*.

2PL est fondé sur le verrouillage des objets de la base de données préalablement à l'exécution des opérations d'une transaction. Deux types de verrous sont associées aux objets

[3] Cette théorie vise à déterminer des ordres d'exécution 'correctes' en forçant la sérialisabilité des exécutions, c-à-d des ordonnancements équivalents à des exécutions en série.
[4] Deux opérations sont en conflit si elles opèrent sur un même objet et si au moins une est une écriture.

de la base de données : les verrous d'écriture et les verrous de lecture. Deux verrous sont *en conflit* si les opérations en question sont en conflit, et appartiennent à des transactions différentes. 2PL doit respecter les trois règles suivantes :

1. Lorsque l'ordonnanceur reçoit une opération d'une transaction, il essaie de poser le verrou qui correspond au type d'opération en question. Si le verrou demandé est en conflit avec un verrou déjà posé, l'opération est mise en attente. Autrement, l'opération est envoyée au gestionnaire de données.
2. Une fois qu'un verrou a été posé pour une transaction, il ne peut pas être supprimé avant que le gestionnaire de données n'ait acquitté l'exécution de l'opération pour laquelle le verrou a été posé.
3. Une fois qu'un ordonnanceur a supprimé un verrou d'une transaction, il ne peut plus poser aucun verrou pour aucune donnée de cette transaction.

Le protocole d'estampillage [CL 99][NP 94] de transactions affecte une estampille à chaque transaction initiée dans le système transactionnel. Si une opération d'une transaction T_i qui arrive est en conflit avec une opération d'une transaction T_j dont l'estampille est supérieure à celle de T_i, l'opération de T_i est rejetée et T_i est abandonnée ; autrement l'opération de T_i est ordonnancée pour l'exécution. Une transaction qui est abandonnée est re-soumise ultérieurement avec une estampille plus grande.

L'estampillage à priori des transactions fait que ce protocole est directement réutilisable dans le cadre réparti, pourvu que les estampilles définissent un ordre global des transactions initiées dans différents sites. Cela peut être assuré par une estampille composée d'un compteur local généré dans chaque site et d'un identificateur unique de site pouvant être utilisé en cas de conflit entre les compteurs locaux [Fri 01].

Les algorithmes optimistes

Dans cette catégorie la détermination des conflits entre les opérations des transactions s'effectue à *posteriori.* Contrairement aux algorithmes pessimistes, les algorithmes optimistes permettent la libre exécution des transactions. Au moment de la terminaison d'une transaction, le protocole exécute une procédure de vérification de conflits : (1) Si l'ensemble des objets écrits par la transaction contient des objets lus ou écrits concurremment par d'autres transactions, ou (2) si la transaction a lu des objets écrits par d'autres transactions. La transaction est annulée si l'on est dans un de ces deux cas, autrement elle certifiée.

Lorsque ce protocole annule des transactions pour assurer l'isolation (ou la sérialisabilité), le 2PL aurait mis ces transactions en attente jusqu'à l'obtention des verrous nécessaires. Sachant que l'abandon et le redémarrage d'une transaction est plus coûteux que la réactivation de celle-ci, on peut conclure que le 2PL s'avère plus performant dans les cas où les conflits sont plus nombreux [Fri 01]. Dans un système transactionnel réparti, le protocole

ci-dessus est utilisé par chaque site du système. Chaque site décide localement quant à la certification ou non des transactions.

2 La validation atomique répartie

Dans les systèmes transactionnels, les transactions peuvent être annulées pour différentes raisons : défaillances, certification non réussie d'une transaction, résolution d'interblocage, etc. Pour satisfaire la propriété d'atomicité des transactions, les systèmes transactionnels répartis doivent assurer que tous les sites accédés par une transaction sont d'accord quand à la décision de la valider ou de l'annuler. Les protocoles qui permettent de résoudre ce problème sont appelés protocoles de la validation atomique ou ACP (Atomic Commitment Protocols).

La *validation atomique* a pour but d'assurer l'atomicité d'une transaction distribuée même en présence de pannes. Chaque site accédé par une branche de transaction distribuée est appelé participant. Dans un ACP, chaque participant révèle sa capacité à valider en émettant un vote. Un vote oui (ou prêt) implique que le participant s'engage à rendre permanentes les mises à jour effectuées par la transaction même s'il tombe en panne. Un vote non (ou abandon) indique qu'il ne pourra pas valider suite à un problème interne.
La validation atomique est définie par les propriétés suivantes [AP 98][Fri 01]:

- Accord uniforme : deux processus ne décident pas différemment.
- Validité uniforme : un processus ne décide de valider que si tout les processus votent pour la validation.
- Non trivialité : si tous les processus sont correctes et si tous les processus votent pour, alors un processus décident de valider.

Un protocole de validation atomique doit être tolérant aux pannes (pannes de sites ou pannes de communications). Pour cela des procédures de reprise après défaillances sont mises en place. Ces procédures assurent qu'une transaction aura les effets de ses opérations sur les objets accédés rendus durables si elle a été validée, sinon, s'il s'agit d'une transaction active (toujours pas validée) qui fait l'objet d'une défaillance, toutes les modifications sur la base de données résultantes de son exécution partielle seront effacées [BCFG+ 97]. Afin de refaire les transactions validées et défaire les transactions exécutées partiellement, les informations sur les opérations des transactions (validées ou actives) et le dernier état cohérent de la base de données doivent être stockées dans des fichiers journaux (log) [BPR 97].

Le protocole le plus répandu est le protocole de validation à deux phases (Tow Phase Commit) [Gra 87]. 2PC (Tow Phase Commit) permet aux serveurs participants à la transaction de se mettre d'accord sur une décision commune sur la validation ou l'annulation de la transaction même en cas de pannes. Les spécifications de 2PC ont été adoptées par une

large gamme de standards tels que ISO [ISO 92], X/OPEN [XOPE+ 96] et CORBA [OMG 94] et même par les SGBDs (Systeme de Gestion de Base de Données) commerciaux.

3.1 Le protocole de validation à deux phases

Dans ce protocole un site nommé *coordinateur* a la responsabilité de diriger les participants dans une transaction distribuée afin d'aboutir à une décision uniforme. Le protocole, comme son nom l'indique, est formé de deux phases (figure 2):

1. Une *phase de vote* : le coordinateur envoie un message de demande de vote, appelé aussi message *prepare*, à tous les participants. Chaque participant répond au coordinateur en votant pour la *validation* ou pour l'*abandon*. Si un participant vote *non*, il envoie son vote et abandonne unilatéralement sa branche de transaction. Par contre, si un participant vote *oui*, il envoie son vote et se met en attente de la décision finale du coordinateur. Le participant est alors dans un état appelé *"prêt à valider"* dans lequel il ne pourra plus ni valider ni abandonner que s'il reçoit la décision du coordinateur.
2. Une *phase de décision* : le coordinateur reçoit les votes, puis informe les sites du résultat de sa décision. Si tous les participants ont voté *oui*, le coordinateur décide de valider la transaction. Si un ou plusieurs participants ont voté *non,* il décide d'abandonner. Les participants qui sont en attente du résultat se conforment à cette décision et envoient un message d'acquittement.

Pour assurer la tolérance aux pannes, les différentes étapes du protocole 2PC sont enregistrées dans des *journaux* maintenus au niveau du coordinateur et des participants. On distingue deux types d'écriture dans un journal:

- Une écriture est dite *non forcée* si l'enregistrement correspondant est mémorisé dans un journal en mémoire puis reporté sur un espace de stockage *stable*[5].
- Une écriture est dite *forcée* si elle impose le report immédiat du contenu du journal sur un espace de stockage stable (n'est jamais perdue en cas de panne).

Pendant l'exécution de la transaction, chaque participant fait une écriture non forcée dans son journal pour chaque opération qu'il exécute. Pendant l'exécution du protocole, un participant doit faire une écriture forcée de son vote ainsi qu'une écriture forcée de la décision reçue du coordinateur. Le coordinateur pour sa part doit faire une écriture non forcées du démarrage du protocole, une écriture forcée de sa décision et enfin une écriture non forcée de la fin de la transaction. L'écriture forcée de la décision garantit qu'une fois la décision prise, elle ne peut être perdue même si le coordinateur tombe en panne. L'écriture non forcée de la fin de la transaction est un moyen pour le coordinateur d'oublier cette transaction et de purger ultérieurement son journal. Cette écriture ne peut avoir lieu qu'après avoir reçu l'acquittement de la décision de la part de touts les participants, qui s'engagent par cette action à ne plus

[5] Un espace de stockage est dit *stable* s'il est *non-volatile* et *fiable*.

jamais interroger le coordinateur sur la décision prise même en cas de panne [AP 98]. La figure 2 détaille les deux phases du protocole ainsi que les informations de journalisation associées.

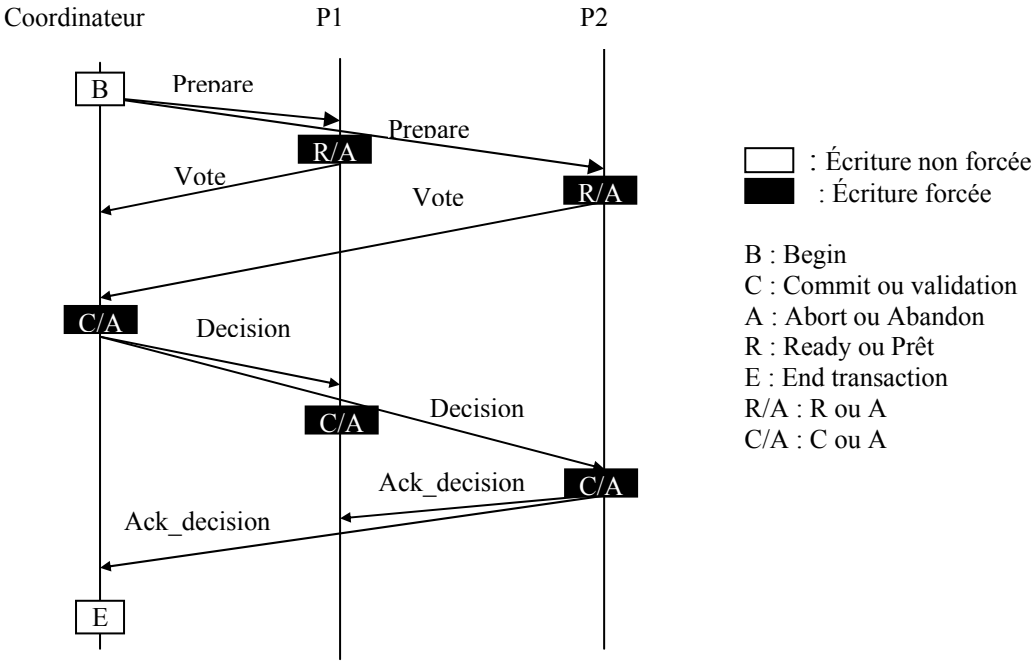


Figure 2 – Le protocole de validation à deux phases (2PC)

Le coût de la validation 2PC d'une transaction répartie sur N participants est de : 2N+1 d'écritures forcées (dans les journaux des participants plus le coordinateur), 4N messages sur le réseau de communication et une latence de trois séquences de messages (demande de vote, vote et décision). Cette latence présente un inconvénient majeur à 2PC. Elle introduit un délai important dans le système même en absence de pannes.

Un autre inconvénient présenté par 2PC est son blocage en cas de panne du coordinateur entre la phase de vote et la phase de décision. Durant cette période (dite fenêtre de vulnérabilité), tous les participants en attente de la décision finale seront bloqués jusqu'à la reprise du coordinateur. Ainsi, une transaction peut détenir des verrous sur des ressources partagées pour une durée indéterminée, pénalisant d'autres transactions du système [Coo 82] [AP 98]. De nombreuses optimisations ont été apportées au protocole 2PC. Ces optimisations ont pour objectifs la réduction du coût de la validation en terme du nombre de messages, nombre d'écritures forcées dans les journaux et en terme de latence.

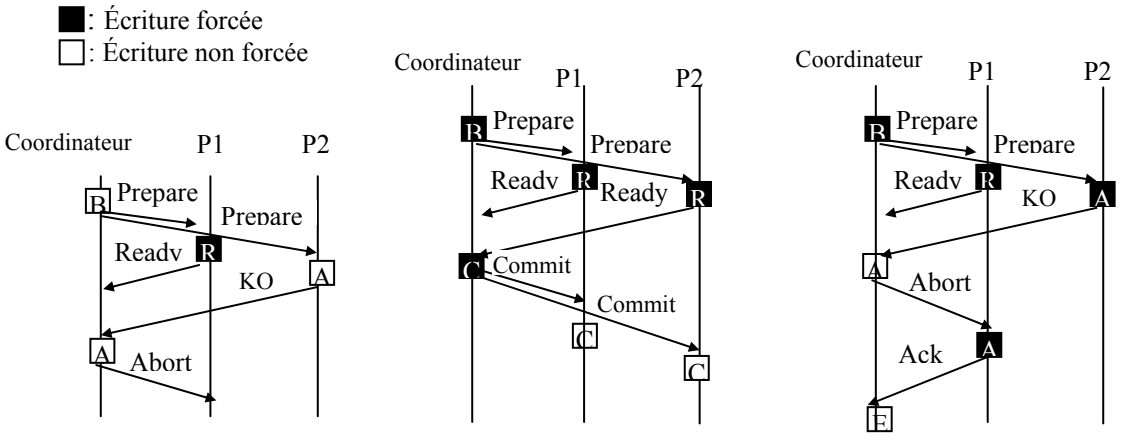
3.2 Optimisations du protocole 2PC

2.2.1 Les protocoles 2PC à abandon présumé et 2PC à validation présumée.

Les protocoles à abandon présumé (Presumed Abort ou PrA) et à validation présumée (Presumed Commit ou PrC) [MLO 86], ont pour objectif de réduire le nombre de messages échangés ainsi que le nombre d'écritures forcées. Le PrA propose (1) d'éliminer les écritures forcées dans les journaux du coordinateur et des participants votant négativement et (2) de ne pas acquitter les décisions d'abandon (voire figure 3 (a)). La trace de la transaction risque d'être perdue en cas de panne du coordinateur à cause de l'écriture non forcée de la décision.

Le protocole PrC est le symétrique du protocole PrA. Il interprète l'absence d'informations concernant une transaction dans le journal du coordinateur comme une présomption de validation. De ce fait, les participants n'ont plus besoin d'acquitter une décision de validation et peuvent la journalisée par une écriture non forcée (voire figure 3 (b)). Cependant la décision d'abandon doit désormais être journalisée par une écriture forcée sur chaque participant (figure 3 (C)). Afin de permettre la reprise suite à une panne avant la validation, PrC impose au coordinateur de mémoriser par une écriture forcée le démarrage de la transaction ainsi que la liste des participants à cette transaction.

L'intérêt de PrC par rapport à PrA est évidant si l'on considère que la majorité des transactions valident. Malgré cela, PrA est souvent préféré à PrC à cause de son écriture forcée supplémentaire dans le journal du coordinateur même pour des transactions n'effectuant que des lectures [AP 98].



(a) le protocole PrA (b) PrC (cas de validation) (c) PrC (cas d'annulation)

Figure 3 – Protocole 2PC à abandon présumé (a) et à validation présumé (b, c)

2.2.2 Les protocoles à une phase (1PC)

Les protocoles à une phase ont pour objectif d'atteindre la borne minimum en nombre de messages échangés pour valider une transaction. Dans ces protocoles la phase de vote est intégrée à l'exécution même de la transaction. Dans cette catégorie on trouve notamment le protocole Early Prepare (EP), le protocole Coordinator Log (CL) et le protocole Implicit-Yes Vote (IYV).

Early Prepare impose à chaque participant de se mettre dans l'état '*prêt à valider'* après l'exécution de chaque opération d'une transaction et avant d'acquitter cette opération (figure 4). Si toutes les opérations sont exécutées avec succès, le coordinateur enregistre la décision de valider par une écriture forcée et diffuse la décision aux participants sans attendre d'acquittement, sinon la transaction est annulée. Ce protocole impose aux participants une écriture forcée après l'exécution de chaque opération ce qui augmente le coût de journalisation.

Le Coordinator Log (CL) [SC 90] exploite l'idée de Early Prepare et évite les entrées/sorties bloquantes au disque sur le participant en centralisant les journaux des participants sur le coordinateur. A chaque fois qu'un participant exécute une opération de la transaction, il renvoie dans le message d'acquittement de cette opération les enregistrements du journal localement créés par l'exécution de cette opération. Le coordinateur enregistre la décision de valider ainsi que l'ensemble des enregistrements de journaux reçus en une seule écriture forcée dans son journal. CL permet de placer l'ensemble des participants dans un état '*prêt à valider*' en une seule écriture forcée. Néanmoins, l'autonomie des participants est violée en leurs forçant de centraliser leurs journaux.

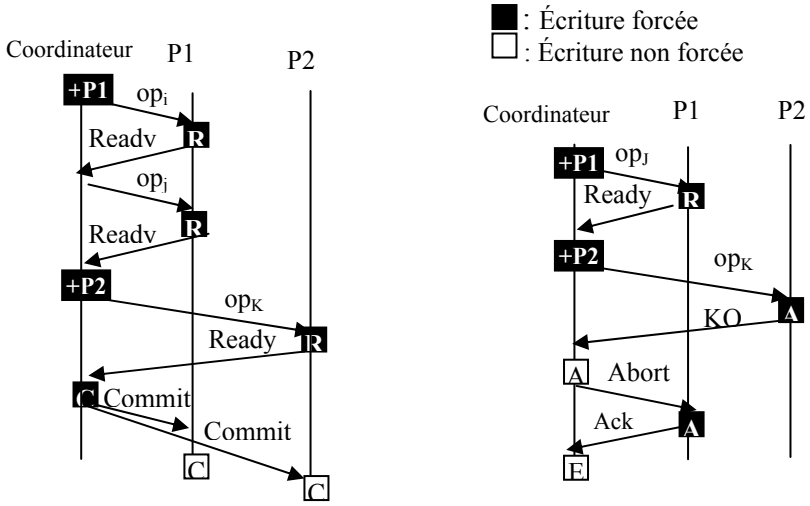


Figure 4 – le protocole Early Prepare

Le protocole Implicite-Yes Vote [AC 95] est directement dérivé du protocole Coordinator Log, bien que sa gestion des journaux diffère quelque peu [AP 98]. Ce protocole

vise à conserver une certaine autonomie aux participants en leur permettant de gérer leurs propres journaux de façon traditionnelle.

Les protocoles à une phase ont une meilleure performance que le protocole à deux phases en terme de réduction du nombre d'écritures forcées et de messages échangés. Cependant, cette performance se paie par des contraintes fortes [Bob 02]. Tout d'abord, ces protocoles supposent que toutes les opérations d'une transaction sont confirmées avant que le protocole ne soit lancé. Ce qui introduit des exigences considérables (taille de journal, messages échangés,…) aux coordinateurs et aux participants. De plus les propriétés CID (Cohérence, Isolation, Durabilité) sont supposées garanties localement, pour chacune des branches de la transaction, au moment de la validation. Enfin, l'autonomie des participants est violée en leur imposant d'exporter tout ou partie de leurs journaux (CL et IYV). D'autre part, ces protocoles étendent la fenêtre de vulnérabilité à toute la durée d'exécution d'une transaction. Du fait que la validation s'effectue en une seule phase, la panne du coordinateur laisse les participants dans un état indécis. Pour exprimer la capacité d'un protocole à tolérer les défaillances, à été défini le problème de la validation atomique non bloquante [Gue 97].

3.3 La validation atomique non bloquante

La validation atomique non bloquante a pour but de garantir en plus de la propriété d'atomicité, la propriété de terminaison. Cette propriété permet à touts les sites corrects[6] d'aboutir à une décision uniforme concernant une transaction malgré la panne d'un ou de plusieurs sites [AP 98]. Formellement, le problème de la validation atomique non bloquante est défini par les propriétés du problème de la validation atomique à savoir : L'accord uniforme, La validité uniforme et La non trivialité [Fri 01]. Auxquelles s'ajoute la propriété de terminaison suivante :

- ***Terminaison*** : tout processus correct décide inéluctablement.

Le protocole de validation en trois phases ou 3PC (Three Phase Commit) vérifie ces propriétés [Ske 82]. En effet, 3PC est le premier protocole qui permet de résoudre le problème de validation atomique non bloquante en ajoutant une phase supplémentaire (appelée phase de pré validation) à 2PC. Cependant, ce protocole suppose que la détection de la défaillance est fiable, une hypothèse qui peut être assurée dans les systèmes synchrones en exploitant la propriété délais des communications bornés, mais qui est irréaliste dans les systèmes asynchrones (à cause de l'absence de borne maximale sur les temps de transfert des messages et sur les temps d'exécutions des pas de calcul). Pour plus de détails sur la validation atomique non bloquante on peut se référer à [FLM 85][Gue 95] [CT 96].

[6] Un site est dit correct s'il n'est jamais tombé en panne durant tout le processus de validation. Autrement, il est dit incorrect.

Conclusion

Le problème de la validation atomique est central dans les applications transactionnelles. La solution de ce problème est mise en œuvre par le protocole de validation à deux phases et ses variantes. Cependant ces solutions, conçues pour des environnements plus stables et plus robustes, nécessitent d'être revisitées dans le contexte de l'environnement de calcul mobile. Le chapitre suivant est donc consacré au problème de la validation en environnement mobile.

Chapitre 2

Le Problème de la Validation Atomique en Environnement Mobile

Introduction

Ce chapitre est consacré au problème de la validation atomique des transactions mobiles. Une transaction mobile est une transaction distribuée sur plusieurs sites dont certains sont mobiles [BLRS 04]. Comme dans le cas fixe, une transaction mobile doit respecter les propriétés d'atomicité, de cohérence, d'isolation et de durabilité. Avant de passer en revue l'impact de l'environnement mobile sur les transactions et en particulier leur validation, nous faisons un rappel des caractéristiques de cet environnement.

1 L'environnement Mobile

Un environnement est dit mobile si les calculateurs (utilisateurs) qui le composent peuvent se déplacer physiquement tout en restant capables d'accéder et d'interagir avec des données indépendamment de leurs positions géographiques [Ade 02]. Les réseaux mobiles et sans fil peuvent être classées en deux catégories [Gra 02] : les réseaux avec infrastructure (réseaux cellulaires) et les réseaux sans infrastructures (réseaux ad-hoc).

Les réseaux cellulaires, plus anciens, sont les plus étudiés dans la littérature. Un système cellulaire est un ensemble d'entités mobiles et d'entités fixes. Chaque station fixe, également appelée station de base, gère une partie du réseau appelée cellule. A tout moment une unité mobile peut changer de cellule. La communication entre entités mobiles s'effectue à travers les stations de bases (voire figure 1). Au sein d'une cellule, la communication entre les entités mobiles et la station de base se réalise à travers des liaisons sans fil. Cette interaction entre un mobile et sa station de base peut s'apparenter à une relation client/serveur classique où le client peut à tout instant changer de serveur. Ce changement engendre des modifications au niveau des tables de routage des stations de base. Le processus qui permet aux unités mobiles de garder leurs connections lors du changement de cellule est appelé *hand off* ou *hand over*. Les étapes de ce processus sont présentées dans la figure 3.

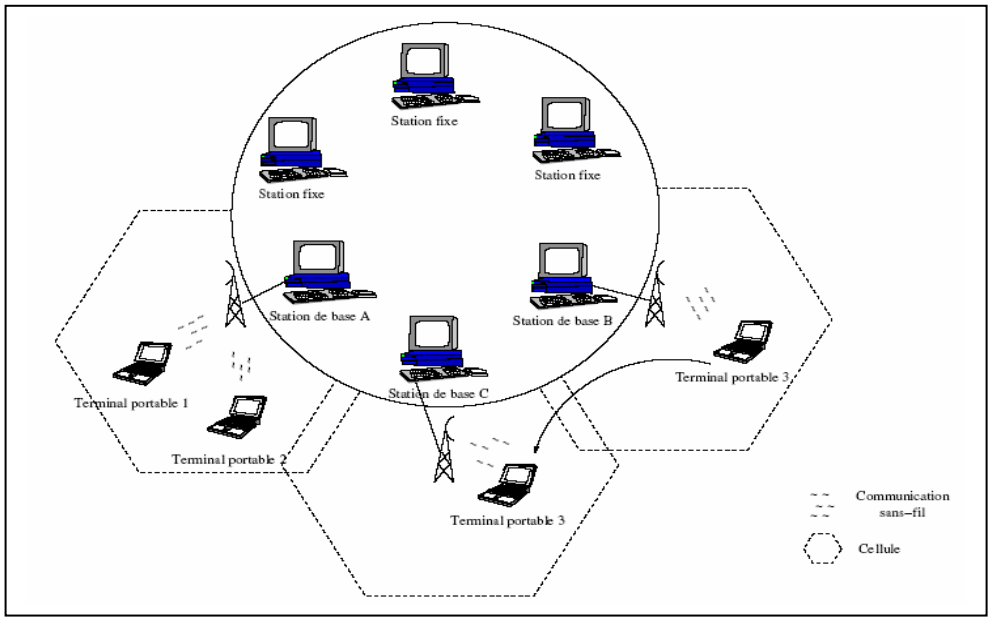


Figure 1- Réseau avec infrastructure

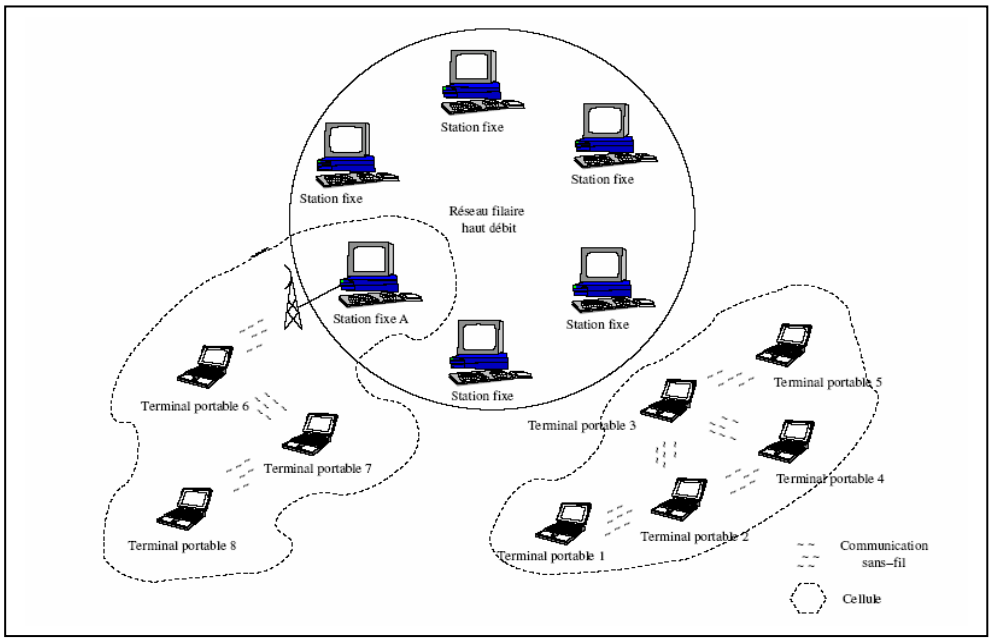
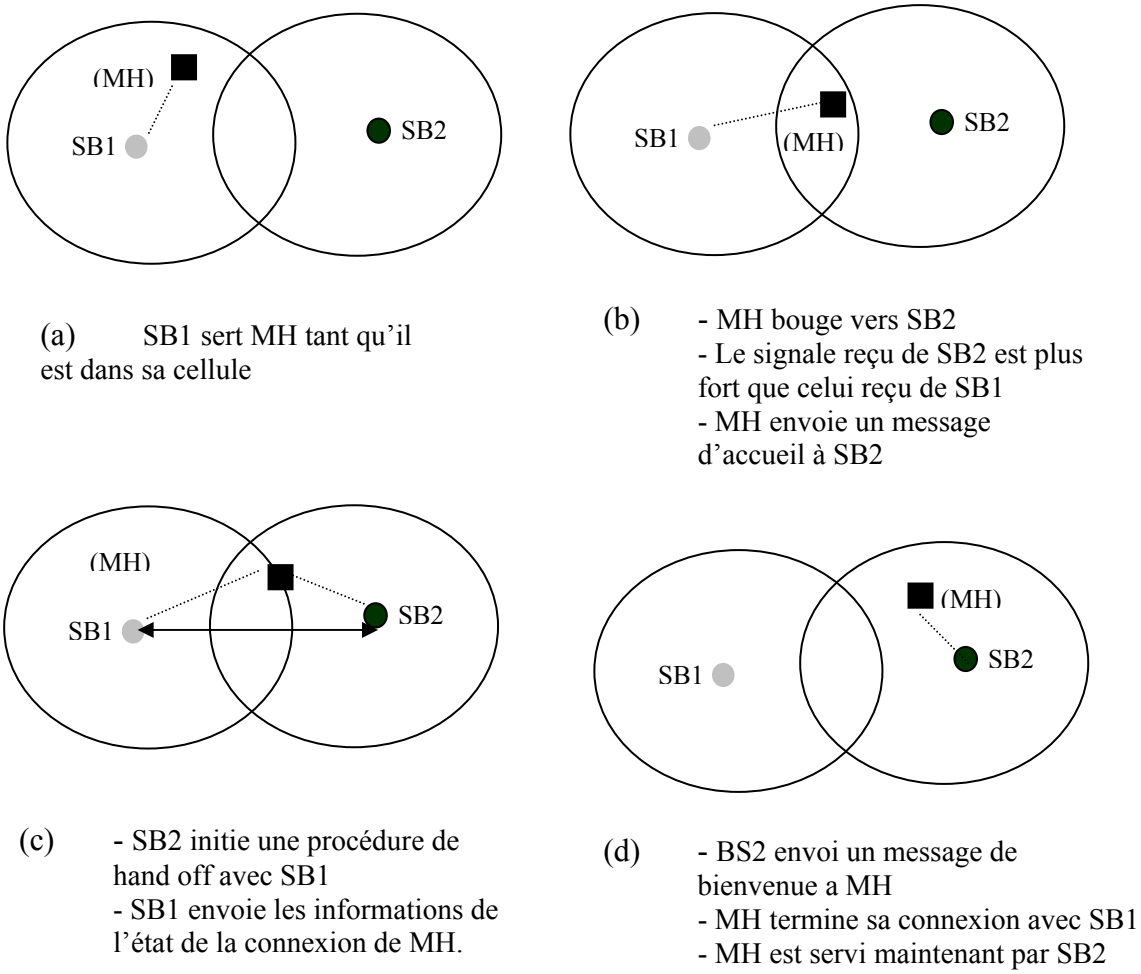


Figure 2- Réseau sans infrastructure

Le développement des systèmes ad-hoc est plus récent. Ces systèmes se caractérisent principalement par le fait qu'ils ne reposent sur aucune infrastructure fixe. Un système ad-hoc est un ensemble d'entités mobiles qui communiquent entre elles principalement via des ondes hertziennes (communication purement sans fil). Afin de pouvoir continuer à acheminer des données entre les différents sites du réseau, les tables de routage, gérées par l'ensemble des participants, doivent systématiquement être mises à jours lorsqu'un calculateur se déplace

physiquement. Dans un tel système, tous les participants ont les mêmes fonctionnalités et jouent le même rôle. En ce sens, ils sont à la fois client et serveurs [Ade 02].

La mobilité engendre de nouvelles contraintes. Ces contraintes sont posées soit par la nature sans fil des communications soit par la portabilité des unités mobiles [TZ 03].



(a) SB1 sert MH tant qu'il est dans sa cellule

(b) - MH bouge vers SB2
- Le signale reçu de SB2 est plus fort que celui reçu de SB1
- MH envoie un message d'accueil à SB2

(c) - SB2 initie une procédure de hand off avec SB1
- SB1 envoie les informations de l'état de la connexion de MH.

(d) - BS2 envoi un message de bienvenue a MH
- MH termine sa connexion avec SB1
- MH est servi maintenant par SB2

Figure 3 - Opérations du Hand off

3.4 La communication sans fil

La communication sans fil est de moindre qualité que celle d'un réseau fixe à cause de l'interaction avec l'environnement qui introduit du bruit et de l'écho [Sag 02]. Le réseau sans fil fournit une bande passante plus faible que celle fournie par un réseau fixe. Cette bande passante est partagée entre utilisateurs d'une même cellule. Avec la mobilité des terminaux, cette bande passante varie dynamiquement. Par exemple, le mobile peut quitter une aire de 10Mb/s de bande passante pour entrer dans une autre de 9600b/s. Les applications s'exécutant sur l'unité mobile doivent s'adapter de manière transparente à la variabilité de la bande

passante. De plus, le signale produit par le réseau sans fil est perturbé par l'environnement. La puissance du signal reçu par le mobile diminue lorsqu'il s'éloigne de la station de base et s'annule lorsqu'il sort de la zone de couverture provoquant la déconnection de l'unité mobile du réseau. Les déconnections peuvent être volontaires, déclenchées par l'utilisateur pour éviter d'être perturbé, préserver la batterie, diminuer le coût de la communication, etc, ou involontaire due à l'épuisement de la batterie ou la sortie de la zone de couverture. Le taux d'erreurs important et la bande passante limitée jouent sur la quantité de données que l'on peut émettre et recevoir, ce qui augmente la latence des applications.

3.5 La portabilité des terminaux

La portabilité impose aux constructeurs d'alléger le poids des terminaux mobiles. Pour qu'un terminal portable soit pratique pour le calcul *anywhere anytime* (n' importe où et n'importe quand), son poids ne doit pas dépasser 100-200 grammes [TVP 00]. Ainsi la commodité de l'unité portable désavantage la capacité de traitement (CPU), de stockage (mémoire), source d'énergie (batterie) et d'interfaces utilisateur (clavier, écran).

3.6 La mobilité

La localisation ou le point d'attache au réseau sans fil des unités mobiles change en fonction de leurs déplacements entre cellules. La configuration d'un système comprenant des unités mobiles, change elle aussi en fonction des changements de localisation des unités mobiles. Par conséquent, le centre d'activité et la charge du système changent dynamiquement et donc la conception d'algorithmes distribués ne peut plus être basées sur une topologie fixe. De plus, le changement de localisation des unités mobiles requiert des structures de données efficaces et des algorithmes appropriés pour la représentation, la gestion et l'obtention de la position des unités mobiles.

La mobilité soulève aussi des problèmes très importants liés à la sécurité et à l'authentification [Bad 98] [Puj 01].

3 Les Transactions Mobiles

Une transaction mobile est une transaction distribuée sur des sites fixes et d'autres mobiles [BLRS 04]. Une transaction mobile peut être lancée par un site mobile aussi bien qu'un site fixe. Un site fixe peut accéder à des données hébergées sur un site mobile [DK 99]. Les sites fixes sont des serveurs qui se caractérisent par une importante capacité de traitement et de stockage, tandis que les sites mobiles sont connectés de façon intermittente, au réseau. Selon la capacité des mobiles, ils peuvent héberger ou non une base de données. Durant leurs mouvements, les utilisateurs mobiles peuvent accéder aux données hébergées dans des sites fixes ou dans des sites mobiles. Les données requises par l'utilisateur mobile peuvent être réparties (1) sur des site fixes et d'autres mobiles ou (2) uniquement sur des sites fixes ou (3)

uniquement sur des sites mobiles ou (4) se trouvant seulement sur l'unité mobile. Par conséquent, l'exécution de la transaction peut suivre l'un des scénarios suivant [Ser 04] :

1. Exécution complète sur des sites fixes : ce scénario utilise les techniques de gestion de transactions traditionnelles. Le résultat de l'exécution est envoyé au site mobile demandant la transaction.
2. Exécution complète sur le site mobile (ou exécution autonome) : dans ce cas, l'optimisation de la consommation des ressources du site mobile (mémoire, CPU, batterie, etc) doit être prise en considération lors de la conception d'une technique de gestion de transactions.
3. Exécution répartie sur des sites mobiles et des site fixes : les techniques de gestion de transaction dédiées à cette exécution doivent prendre en compte les caractéristiques de la communication sans fil en plus de l'optimisation des ressources des unités mobiles.
4. Exécution répartie sur des sites mobiles : ce scénario est dédié aux réseau Ad hoc. Ce modèle demande des techniques de découverte et de localisation des voisins mobiles et la répartition dynamique des rôles et des tâches pour chaque mobile.

Les transactions mobiles opèrent en deux modes [TZ 03] :

- Mode connecté : l'exécution de la transaction se déroule lorsque l'unité mobile est connectée et accède aux données hébergées sur un serveur fixe distant. Les transactions opérant dans ce mode sont dites transactions on-line.
- Mode déconnecté : l'exécution de la transaction s'effectue lorsque l'unité mobile est déconnectée du réseau et accède aux copies locales de données. Les transactions opérant dans ce mode sont dites transactions off-line.

3.1 Exemples de transaction mobile

Exemple 1 : un responsable d'entreprise qui souhaiterait hors de son bureau travailler sur son ordinateur portable. Il accède et met à jours ses données localement et souhaite propager les mises à jours effectuées sur son ordinateur de bureau ou sur la base de données de son entreprise. De façon générale, les fichiers sont copiés dans le portable et sont ensuite restaurés sur l'ordinateur fixe ou sur la base de données de l'entreprise. Le problème qui se pose est que certains fichiers peuvent être partagés par d'autres d'utilisateurs qui sont en mode déconnectés. Ce qui implique un problème de réconciliation des copies divergentes se trouvant au niveau de chaque utilisateur.

Exemple 2 : dans une agence d'assurance d'automobile, un agent doit physiquement examiner chaque dommage et fournir une estimation des coûts de réparation. Fournir cette estimation demande l'accès à différentes informations sur la voiture, le rapport de police, les assurances précédentes prises par la voiture.

Actuellement, le traitement de cette transaction peut être effectué de la façon suivante [DHB 96]: l'agent en recevant l'ordre de traiter une réclamation sous forme écrite ou sous forme électronique de son superviseur doit, avant d'examiner les dommages, rassembler les informations requises définies plus haut. Certaines de ces informations sont obtenues par téléphone, d'autres doivent être demandées en personne aux endroits appropriés et d'autres par Internet. Ensuite l'agent examine physiquement la voiture. L'agent peut au retour dans son bureau procéder au traitement de la transaction. Toutes les informations rassemblées durant la journée sont entrées à travers une transaction et le rapport est envoyé au superviseur (figure 4). Comment cette même opération pourrait être faite dans le future ?

Premièrement, toutes les informations requises sont en lignes. Après avoir examiner les réclamations à traiter du jour, il en détermine l'ordre et commence immédiatement le traitement. Le rassemblement des informations est fait par la transaction. Des sous transactions sont engendrées automatiquement pour obtenir toutes les informations nécessaires. Avant qu'il n'arrive à la voiture endommagée, les informations requises sont disponibles sur son mobile. Et pendant qu'il examine la voiture il peut rentrer toutes les informations requises dans le mobile qui met à jour automatiquement les données présentes et lancer les changements ultérieurement au niveau du réseau fixe (figure 5). Ensuite il peut commencer le traitement d'une autre réclamation pendant qu'il se rend au lieu du dommage.

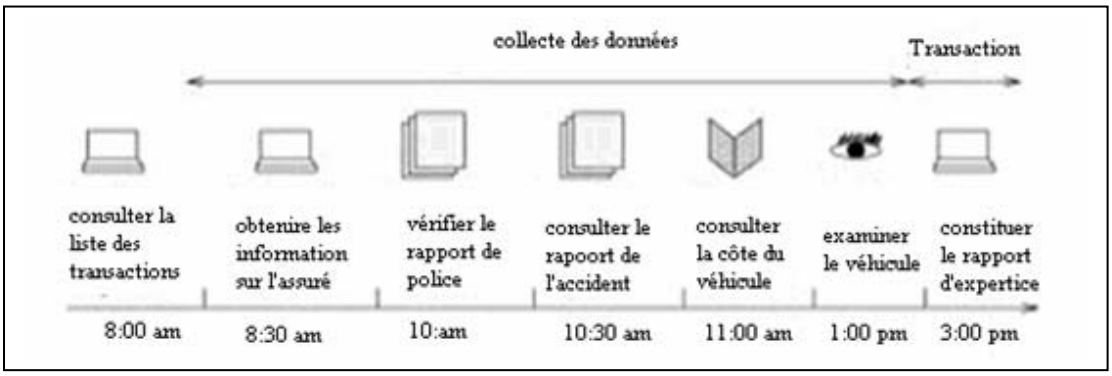


Figure 4 - déroulement d'une transaction classique

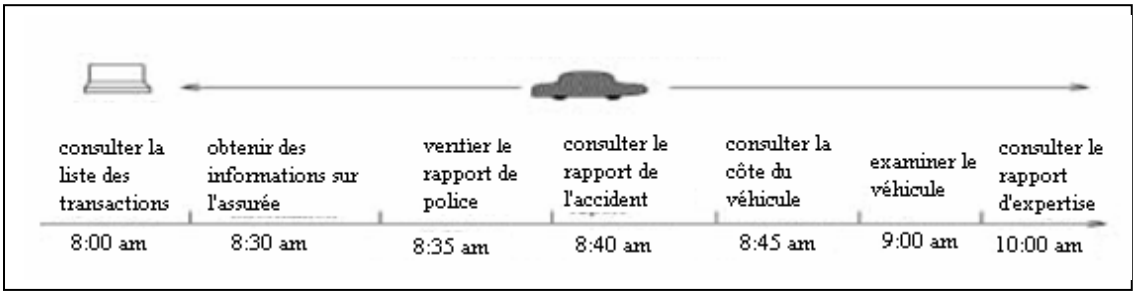


Figure 5 - déroulement d'une transaction mobile

3.2 Les problèmes de l'ACIDité en environnement mobile

L'Atomicité

Il est difficile d'assurer l'atomicité dans un environnement mobile à cause de la longue durée de vie de la transaction induite par les périodes de déconnexion et les limites des terminaux mobiles et du réseau sans fil. La validation atomique serait trop coûteuse à réaliser dans un tel environnement. La décomposition d'une transaction mobile en sous transactions et la définition d'une atomicité au niveau de sous transactions est une solution qui semble nécessaire. L'impact des challenges de l'environnement mobile sur les protocoles assurant cette propriété est traité à part et en détail dans la section 3.

La cohérence

La vérification des contraintes d'intégrité par le mobile entraîne une surconsommation des ressources de ce dernier (espace de stockage, CPU, batterie). Toute fois, si les contraintes d'intégrité sont vérifiées sur la partie fixe du réseau, cela augmente considérablement le coût de la communication sans fil. La cohérence de données doit être aussi garantie en cas d'exécution de la transaction en mode déconnecté. Durant ce mode d'exécution, la transaction utilise des données locales qui peuvent être périmées. A la reconnexion, les techniques de synchronisation doivent être utilisées afin de réconcilier les différentes versions [Bob 02].

L'isolation

Les techniques de contrôle de concurrences pessimiste ne peuvent s'appliquer directement à cause des connexions sans fil sujettes aux pannes et au taux d'erreurs élevé ce qui conduit à des transactions de longues durées. En effet, des verrous sur des données tenus par un mobile en panne ou déconnecté peuvent se faire pour une longue durée provoquant des situations de blocage ainsi qu'une réduction des données disponibles [Sag 02].

La durabilité

Il est difficile voir impossible d'assurer la durabilité sur le mobile à cause de la prédisposition de ce dernier aux pertes, vols et destructions. Cette propriété n'est garantie que sur la partie fixe du réseau [Nou 01] [NDD 04].

3.3 La gestion des transactions mobiles

Il est clair que les modèles transactionnels classiques ne sont pas en mesure de répondre aux besoins des transactions mobiles. Les limites des calculateurs mobiles et des réseaux sans fil ont poussé les chercheurs à réévaluer ces modèles afin de répondre aux exigences du calcul mobile. Un modèle de transaction mobile doit supporter les exigences de base suivantes [Nou 01] [Sag 02]:

- Eviter la duplication des supports de traitement déjà offerts par les systèmes implémentés sur le réseau statique.

- Tenir compte des mouvements et des déconnexions fréquentes des sites mobiles.
- Tenir compte des transactions de longue durée.
- Prendre en charge les besoins des nouvelles applications en tenant compte des modèles transactionnels avancés (imbriqué et multi-base).
- Restructuration flexible des propriétés ACID notamment la validation locale autonome d'un participant doit être permise dans le cas de déconnexion temporaire.

Plusieurs modèles de transactions mobiles ont été proposés. En générale, ces modèles structure une transaction en sous transactions avec une certaine flexibilité sur la cohérence et sur le traitement de la validation [Sag 02]. La gestion de la transaction peut être statique au niveau du mobile ou au niveau de la station de base (ou un autre site fixe tel le serveur de la base de données) ou se déplacer de station de base en station de base. La plupart de ces modèles s'appuient sur un relâchement plus ou mois important des propriétés ACID et tentent de permettre des exécutions non bloquantes en cas de déconnexion. Ces modèles permettent aussi la duplication des données sur le mobile aussi bien que sur la station de base. Des travaux liés aux modèles de transactions sont référencés dans [TZ 03].

4 Problèmes de la validation atomique classique dans l'environnement mobile

2PC est un protocole dédié aux systèmes distribués caractérisés par l'interconnexion d'un ensemble de sites stationnaires équipés par une source d'énergie abondante et des capacités de calcul largement suffisantes. La communication dans les systèmes distribués s'effectue par l'échange de messages via une large bande passante disponible d'une manière permanente, offerte par des canaux de communication filaires. Cependant de telles caractéristiques ne peuvent pas être garanties dans le nouveau environnement de calcul mobile et sans fil pour les simples raisons (1) que la mobilité des terminaux exige leur portabilité ce qui implique la diminution des ressources (CPU, Mémoire et énergie). (2) la communication avec un terminal mobile s'effectue à travers des liens sans fil qui se caractérisent par une faible bande passante, augmentant ainsi la latence, le taux d'erreurs et le coût des communications. (3) la mobilité ou le déplacement des unités mobiles qui provoque un changement de localisations de ces dernières.

3.1 Problèmes liés à la nature des terminaux mobiles

La coordination du processus de validation est généralement le rôle du site initiateur de la transaction. Cette tâche requiert de fortes capacités de calcul et de stockage. Or dans l'environnement mobile, une transaction peut être lancée (ou initiée) par un site mobile. La pauvreté de ce dernier en terme de capacité de calcul et de stockage, le rend incapable d'assurer le rôle du coordinateur.

Le maintien des journaux des opérations des transactions doit être réalisé en utilisant des supports de stockage fiables ce qui n'est pas le cas pour les unités mobiles qui sont facilement sujettes aux pertes et aux vols [Nou 01][NBDD 05].

3.2 Problèmes liés à la nature des liens sans fil

Le nombre de messages échangés dans le processus de validation en utilisant 2PC est de 4N messages où N est le nombre de participants à la transaction [AP 98]. Sachant que le coût de la communication sans fil est beaucoup plus élevé que celui de la communication filaire, la transmission d'un tel nombre de messages sur les liens sans fil entraîne une surconsommation de la bande passante ce qui rend 2PC inadéquat pour le calcul mobile. De plus, la transmission d'un grand nombre de messages par un terminal mobile augmente la consommation de son énergie. Les protocoles à une phase semblent être plus appropriés pour cet environnement car ils permettent une optimisation de la bande passante. Cependant ils imposent des hypothèses très fortes et contraignantes comparées à la capacité des terminaux mobiles.

2PC est formé de quatre séquences de messages (demande de vote, vote, décision et acquittement), ce qui le rend suffisamment lent même en absence de pannes [AP 98]. Cette latence augmente en environnement mobile à cause de *la faible bande passante, des déconnexions et du taux d'erreurs* élevé des liens sans fil.

Les déconnexions des participants mobiles entraînent le blocage des ressources détenues par la transaction mobile exécutant 2PC. Le site mobile peut se déconnecter volontairement afin de conserver sa batterie ou d'optimiser le coût de la communication ou involontairement à cause de l'épuisement de l'énergie ou lorsqu'il traverse une zone non couverte. L'exécution de 2PC en présences des déconnexions conduit à une augmentation du taux des annulations à tors [NDD 05a]. En effet, 2PC traite les déconnexions comme des pannes, ce qui n'est pas acceptable dans l'environnement mobile, car les déconnections sont fréquentes et sont considérées comme un caractère intrinsèque à cet environnement. De plus, 2PC est le protocole qui laisse le moindre d'autonomie aux participants [BLRS 04]. Dans le contexte mobile, les participants (mobiles) doivent avoir un certain degré d'autonomie leur permettant d'annuler une branche transactionnelle de façon autonome. Pour cela les informations nécessaires pour l'annulation doivent être stockées sur le mobile lui même.

3.3 Problèmes induits par la mobilité des sites.

Les protocoles de validation atomique classiques (2PC et ces variantes) sont basés sur l'hypothèse que la location des participants et/ou du coordinateur reste inchangée. Cependant, dans le calcul mobile cette hypothèse n'est plus valide. Les sites migrent en fonction du mouvement de l'utilisateur [Sag 02].

Le placement du coordinateur sur un site fixe permet de minimiser le coût de la communication sans fil et d'améliorer le temps de réponse en minimisant la distance physique entre sites. La station de base à laquelle est attaché le site mobile lors de l'initiation de la transaction est choisie pour assurer la coordination des sites participants à la transaction [Nar 94]. Cette proposition permet (1) de palier au problème des ressources limité du mobile, (2) une réduction significative du nombre de messages échangés sur le réseau sans fil, (3) un stockage fiable des fichiers logs du coordinateur et (4) une réduction des délais de la communication entre le coordinateur et les participants fixes du fait que les messages ne transitent pas via la station de base. Cependant, lors de son mouvement, un site mobile peut changer de cellule et devenir incapable de communiquer avec son coordinateur qui se trouve dans sa cellule mère (l'ancienne cellule). Pour remédier à ce problème deux stratégies de coordination ont été proposées [KDDS 00] [NDD 05a] :

- La coordination statique dans la quelle le coordinateur ne change pas de SB (Station de Base) durant l'exécution de tout le processus de validation. Dans ce cas il est nécessaire d'informer le coordinateur sur la nouvelle position du client mobile à chaque fois qu'il change de cellule. La distance entre le coordinateur et le terminal mobile augmente la latence du protocole ce qui implique l'augmentation de sa fenêtre de vulnérabilité.[7]
- La coordination émigrante dans laquelle le coordinateur se déplace selon les mouvements du site mobile. A chaque fois que le mobile change de cellule, le coordinateur migre vers la SB qui couvre cette cellule. Ce qui représente un inconvénient en cas des déplacements fréquents. La migration du coordinateur n'est autre que le transfert des informations d'états de l'ancien coordinateur vers le nouveau coordinateur qui informe les participants sur ce changement. Cela introduit de nouveaux messages et influe par la suite sur la latence du protocole.

5 Les protocoles de la validation atomique de transactions mobiles

On présente brièvement dans cette partie deux exemples de protocoles de validation atomique pour l'environnement mobile : UCM est une adaptation de protocole 1PC et TCOT est spécialement conçu pour l'environnement mobile.

3.1 Le protocole UCM (Unilatéral Commit for Mobile)

Dans UCM (Unilateral Commit for Mobile and Disconnected Computing) [BPA 00][Bob 02], l'atomicité de la transaction globale est assurée en continu par la journalisation des opérations de la transaction (pour la reprise, si besoin) et par la confirmation de la bonne exécution de chacune de ces opérations (l'envoi d'un ack à l'application après chaque exécution). Dans le cas contraire la transaction est immédiatement annulée. De ce fait, si la transaction termine, la décision est obligatoirement la validation. L'avantage de UCM est

[7] La fenêtre de vulnérabilité est la période du temps qui sépare la phase du vote et la phase du décision.

qu'il permet de décider sur la terminaison de la validation en une seule phase tout en éliminant les risques de décision à tort.

Le déclenchement de UCM s'effectue par l'envoi du journal de l'application au coordinateur. Les bases de données participantes sont représentées par un agent (*PAgent*) pour garantir le respect de leur autonomie et, également, pour assurer le bon fonctionnement de la reprise en cas de panne. Comme pour 2PC, le coordinateur reste bloqué si au moins un acquittement de la décision globale n'arrive pas, mais cette situation n'est pas pénalisante pour le système [BLRS 04].

Pour permettre le traitement off-line, le coordinateur est hébergé sur la station de base. Cependant, le transfert du journal des opérations vers le coordinateur est très coûteux. De plus, la réduction de la latence du protocole de validation (se réduit à une phase) augmente le temps d'exécution global de la transaction (à cause de la journalisation de toutes les opérations de la transaction). Les unités mobiles ne sont pas aptes à supporter les hypothèses exigées par les protocoles 1PC qui comptent sur le contrôle de concurrence pessimiste pour assurer la cohérence et l'isolation.

3.2 Le protocole TCOT (Transaction Commit On Timeout)

TCOT [KPDS 02] est basé sur les mécanismes des timeouts pour décider de la validation d'une transaction mobile. Deux types de timeouts sont définis par TCOT :

1. E_t : qui représente le délai maximal d'exécution d'une branche de transaction par un participant fixe ou mobile. Ce délai dépend de la taille de la branche à exécuter et de la capacité de calcul du participant.
2. S_t : qui représente le délai maximal pour transférer les mises à jour des données du cache du mobile vers le serveur de données localisé sur un site fixe.

TCOT traite le cas des transactions initiées par le site mobile. Ce dernier calcule le E_t et le S_t des opérations de sa branche et envoie le reste des opérations de la transaction au coordinateur qui les distribue sur les participants fixes. Ces derniers calculent leurs E_t et les envoient au coordinateur. S'il s'avère que la durée du traitement des opérations sur un participant est supérieure à E_t, le participant étend la valeur de E_t et informe le coordinateur du changement. Si le coordinateur reçoit les mises à jours transférées par le mobile avant l'expiration de S_t et les demandes de validation des participants fixes avant que leurs E_t expirent, il décide de valider la transaction globale. Autrement, le coordinateur annule la transaction et envoie un message d'abandon global aux participants.

Le fait de disposer d'une copie de données dans le cache du mobile, lui permet d'exécuter sa branche en mode déconnecté de façon autonome. Cependant, il doit se reconnecter avant l'expiration de son E_t pour valider ou étendre la valeur de son timeout.

Bien que TCOT essaie d'optimiser l'usage de la bande passante du réseau sans fil en supprimant la phase du vote , le transfert du journal des mises à jour vers un support stable avant la validation unilatérale augmente considérablement la consommation de la bande passante.

Le protocole TCOT relâche la propriété d'atomicité par le fait de permettre l'autonomie de décision sur les branches de transaction. Si la décision locale n'est pas conforme à la décision globale, une transaction de compensation devra être exécutée sur le site ayant validé de façon prématurée. Une transaction de compensation a pour rôle d'annuler sémantiquement les effets de la transaction validée. TCOT offre alors ce que l'on appelle l'atomicité sémantique.

Conclusion

Nous avons présenté dans ce chapitre le problème de la validation atomique en environnement mobile. Les protocoles de validation atomiques classique (2PC et ces variantes) soufrent de problèmes lorsqu'ils sont exécutés dans l'environnement mobile. Les unités mobiles ne sont pas aptes à coordonner une transaction à cause des ressources limitées dont elles disposent. La faiblesse et parfois l'indisponibilité des communications sans fil conduisent à un taux considérable d'annulation de transactions. De nouveaux protocoles ont été proposés spécifiquement pour terminer des transactions impliquant des mobiles. Ces protocoles assurent souvent l'atomicité sémantique comme le fait TCOT. UCM assure l'atomicité stricte mais impose des hypothèses pénalisantes aux unités mobiles en ce qui concerne l'exigence d'un contrôle de concurrence stricte. Cependant, l'atomicité stricte est exigée dans plusieurs types d'applications transactionnelles mobiles (par exemple : les transferts bancaires, le M-commerce et les applications de télémédecine …). Dans ce cas l'adaptation du protocole de validation atomique (stricte) à deux phases (2PC) ou de ces variantes aux contraintes du calcul mobile, semble être une voie à ne pas écarter complètement.

Les chapitres 4 et 5, sont consacrés justement à l'évaluation de l'une de ces adaptations à savoir le protocole Mobile Two Phase Commit (M-2PC). Étant donné que la particularité de ce protocole réside dans la façon dont il gère la mobilité et que notre étude est consacrée à cet aspect, nous avons consacré le chapitre 3 qui suit aux approches abordées dans la littérature pour résoudre le problème de la gestion de mobilité au niveau applicatif. La présentation du chapitre 3, permet de mieux situer le mécanisme de gestion de mobilité introduit dans le chapitre 4.

Chapitre 3

La Gestion de la Mobilité au Niveau de la Couche Application

Introduction

Le calcul mobile a connu des travaux de recherche intensifs durant les dernières années. La gestion de la mobilité constitue un des principaux challenges auxquels ont été confrontés les chercheurs du domaine. De nombreux protocoles de gestion de la mobilité ont été proposés au niveau des différentes couches de la pile protocolaire. Mais chacune des solutions proposées exige des conditions spécifiques et présente des limites telles que le déploiement de nouvelles entités, un routage pas très performant, la latence du hand off, etc. Dans ce chapitre nous examinons la gestion de la mobilité au niveau de la couche application. En effet, une telle approche offre, à priori, un avantage très important ; la facilité de mise en œuvre et ceci pour deux raisons : (1) la non nécessité du déploiement d'une architecture très élaborée ou sophistiquée de réseau. (2) cela ne devrait pas exiger des modifications sur le noyau du terminal mobile de l'utilisateur ou sur les serveurs. Notre objectif est d'étudier la faisabilité et l'efficacité d'une telle approche en mettant en évidence ses avantages et ses inconvénients par rapport aux autres solutions. Deux classes de solutions peuvent être définies : des solutions basées sur l'utilisation des serveurs ou des 'Proxies' de mobilité et des solutions basées uniquement sur la modification des bibliothèques S*ockets*[8].

1 Support de mobilité à base de serveur (ou proxy) de mobilité

3.3 SIP (Session Initiation Protocol)

Le protocole SIP (Session Initiation Protocol) est, comme son nom l'indique, un protocole d'initialisation de session multimédias. SIP qui a été conçu pour être un protocole de pure

[8] Interface permettant la communication entre un système unique et un réseau utilisant le protocole TCP/IP.

signalisation, a été spécifié à l'origine par le groupe de travail MMUSIC[9] (Multiparty Multimedia SessIon Control) de l'IETF (Internet Engineering Task Force) dans la RFC 2543 en 1999. Le protocole SIP est bâti sur une architecture Client/Serveur et utilise des messages textuels. L'architecture SIP (figure 1) comprend deux types de composants : le « User Agent » (Agent utilisateur) et les Serveurs :

- *Le user Agent :* c'est l'application utilisateur, elle se décompose en une partie cliente et une partie serveur. La partie cliente, appelée User Agent Client (UAC), envoie les requêtes SIP, et la partie serveur, appelée User Agent Server (UAS), les reçoit. Le principal objectif de SIP est de permettre l'établissement de sessions entre User Agents. Les utilisateurs, dans un environnement SIP sont identifiés par des URLs (*Uniform Resource Locators*) SIP dont le format est similaire à celui d'une adresse e-mail (*user_infos@domain*).

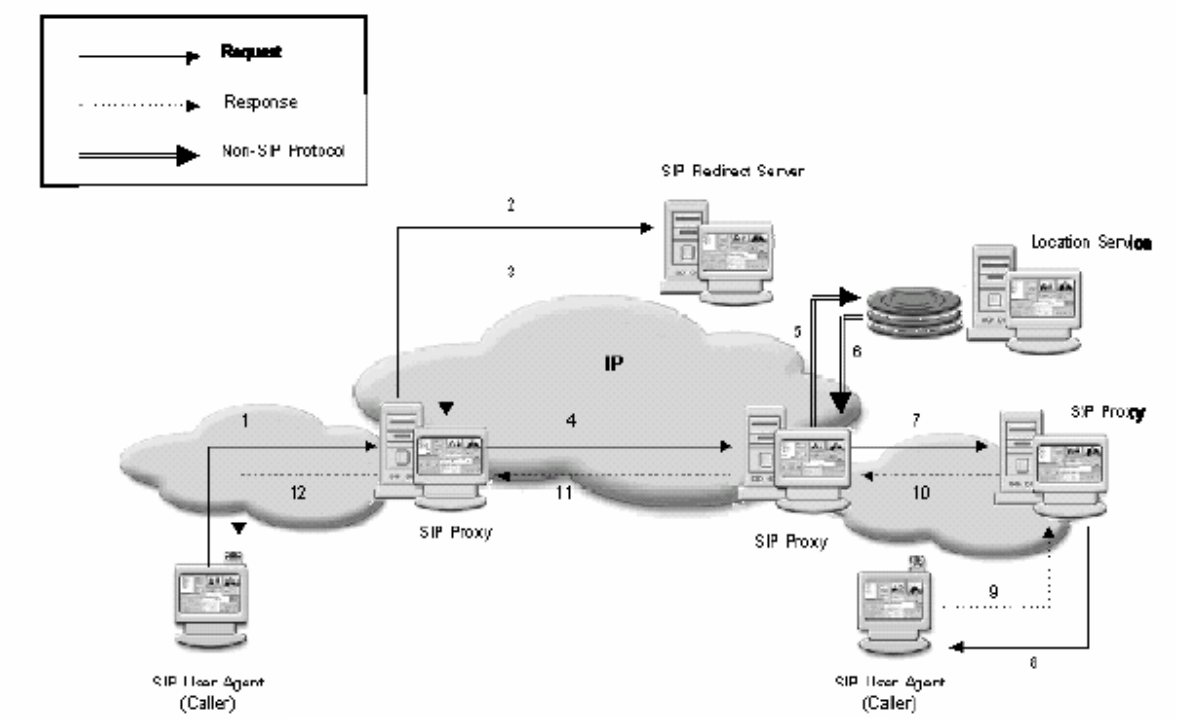


Figure 1 - Model Client/Serveur du protocole SIP

- Les serveurs : e serveur : (1) le serveur d'enregistrement (registration server), auprès de lequel s'enregistre les unités mobile lors du changement de point d'attache. Ce serveur maintient à son niveau la correspondance entre les adresses IP et les adresses SIP. (2) le serveur de localisation

[9] http://www.ietf.org/html.charters/mmusic-charter.html

(location server), qui indique l'adresse IP courante d'une unité SIP. Cette information est mise à jour par le serveur d'enregistrent (figure 2). (3) le serveur Proxy, qui joue le rôle d'un routeur de niveau application). Lorsqu'un *serveur proxy* recevra une invitation à une communication destinée à un utilisateur (user), le *Location Server* lui indiquera l'adresse IP de l'utilisateur et le *Proxy Server* routera le message vers l'adresse IP appropriée (étapes 5, 6 et 7 sur la figure 1). (4) serveur de redirection (redirect server), ce serveur aide à localiser les User Agent SIP en fournissant une adresse alternative à laquelle l'utilisateur appelé peut être joint. Lorsqu'une requête lui parvient, il retourne une réponse de redirection contenant la ou les prochaines adresses à contacter pour joindre le destinataire final (étape 3 sur la figure 1).

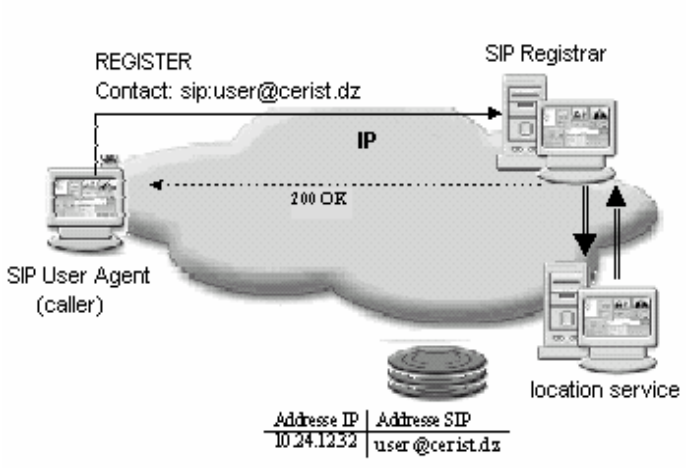


Figure 2 - Exemple d'enregistrement SIP

La localisation des terminaux mobiles dans un environnement SIP s'effectue en permettant aux terminaux mobiles de se re-enregistrer auprès de leurs *Home Registrar* (serveur d'enregistrement du réseau mère) chaque fois qu'ils changent de réseau. Et lorsque le correspondant envoie une requête INVITE à un mobile, le serveur de redirection le redirige vers la localisation courante du site mobile comme nous l'avons indiqué précédemment. Le cas le plus simple est celui où le mobile change de réseau avant de participer dans une session de communication. Dans ce cas le mobile acquiert une nouvelle adresse IP bien avant qu'il établisse une nouvelle communication. Ce cas est appelé *Pre-call Mobility* [SW 99] (voir figure 3).

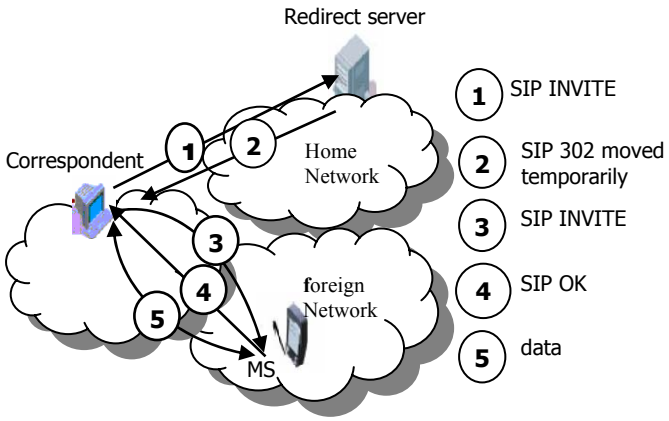


Figure 3 - Localisation avant appel, dans un environnement SIP

Si le mobile change son point d´attache au réseau pendant la session, il informe son correspondant sur sa nouvelle localisation en lui envoyant une requête SIP INVITE (figure 4), en mettant à jour bien sûr la description de la session avec sa nouvelle adresse. Notons que cette session aura le même identificateur que celui de la session précédente (avant le hand off). De cette façon le correspondant sera au courrant de la nouvelle localisation du mobile juste après acquisition d'une nouvelle adresse.

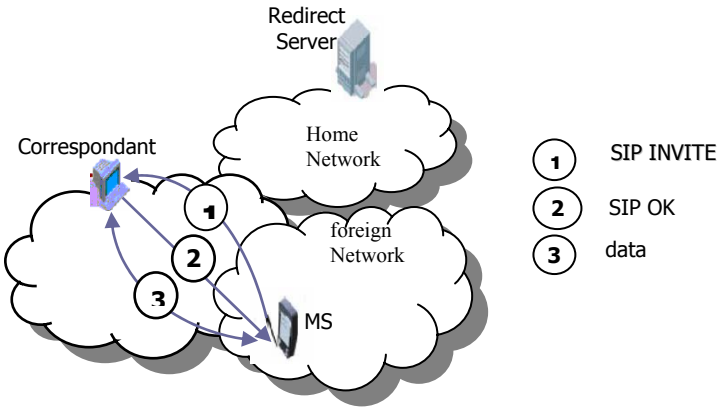


Figure 4 - hand off dans un environnement SIP. (Mid-call Mobility)

3.4 MITP (Mobile Internet Telephony Protocol)

Mobile Internet Telephony Protocol [Lia 99] est un protocole proposé pour les services de la téléphonie mobile dans les réseaux IP. Il est basé sur le modèle client serveur et se trouve au niveau de la couche application. La mobilité des utilisateurs est supportée dans MITP à l'aide d'un ensemble de messages requêtes / réponses échangés entre le client et le serveur de mobilité appelé serveur MITP (*MITP Server*). La syntaxe des messages MITP est similaire à celle des messages HTTP1.1 et SIP 2.0 permettant ainsi la réutilisation du code et la facilité d'intégration des serveurs Web et des Serveurs SIP. MITP définit deux types d'agents de mobilité. Le serveur MITP mère (*Home MITP Server*) se trouvant dans son réseau d'attache et le serveur MITP étranger (*foreign MITP Server*) se trouvant dans le réseau visité

actuellement par le terminal mobile. Les primitives de base utilisées pour la gestion de la mobilité sont les suivantes :

- *discover* pour la découverte des agents de mobilité,
- *register* utilisé par le client pour s'enregistrer auprès des agents de mobilité MITP.
- *call* pour établir, joindre ou quitter un appel en cours.
- *Bind* utilisé par le I-phone Mobile pour informer son serveur MITP mère de sa disponibilité pour la redirection de la conversation.

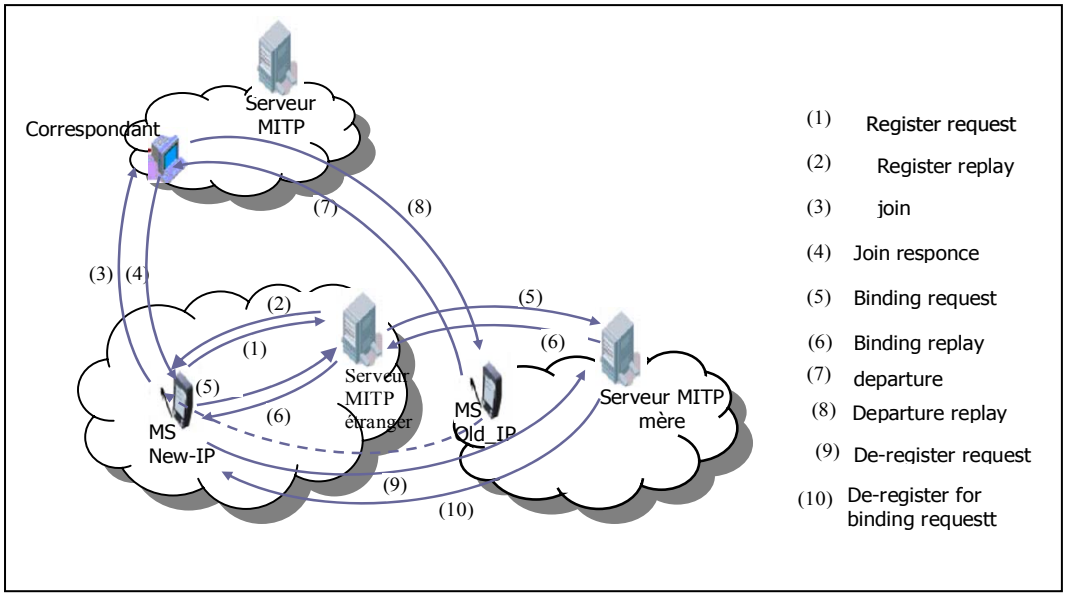


Figure 5 - Les opérations MITP pour la gestion de la mobilité

Dans MITP, lorsque le mobile change son point d'attache au cours d'une communication, il doit d'abord s'enregistrer auprès du serveur MITP du réseau étranger (étape 1 sur la figure 5) et recevoir la confirmation d'enregistrement (étape 2), puis il rejoint la conférence comme étant un nouveau participant (étapes 3 et 4). Après avoir rejoint l'appel avec succès, le mobile met à jour ces informations de localisation au niveau de son serveur MITP mère (les étapes 5 et 6). Le terminal mobile (ancienne adresse) quitte alors la conférence (les étapes 7, 8, 9 et 10).

3.5 ALMP (Application Layer Mobility Proxy)

ALMP [HDS 03] est une approche qui permet de gérer la mobilité au niveau de la couche application des applications temps réel. ALMP utilise un serveur de mobilité de niveau application. Ce dernier repose sur les fonctionnalités de:

1. SIP REGISTRAR pour le stockage des informations de localisation du site mobile, ce qui exige l'installation d'un agent utilisateur SIP au niveau du terminal mobile pour que ce dernier puisse envoyer les messages d'enregistrement au serveur REGISTRAR dés qu'il change de sous réseau et acquiert une nouvelle adresse IP. Et d'un
2. Proxy de mobilité pour capter les données destinées à l'ancienne adresse du terminal mobile et les transférer vers sa nouvelle adresse (figure 5). Les fonctionnalités de Proxy de mobilité (Mobility Proxy) peuvent être tout simplement assurées par les services *ipchains/iptables* de linux. *ipchains* est un Firewall implémenté dans la version 2.1.102 du noyau linux et qui peut être configuré pour supporter le IP masquerade (*IP Masquerading*), qui est une fonction de gestion de réseaux utilisée très souvent par les routeurs et les firewalls commerciaux pour assurer le transfert de données. *iptables* est un descendant direct de *ipchains,* implémenté dans la version 2.4 de linux.

Le principe d'un système à base de Proxy de mobilité est le suivant :
Dés que le terminal mobile change de sous réseau, il envois un message SIP REGISTER au serveur SIP REGISTRAR pour mettre à jours ses informations de localisation. Le SIP REGISTRAR envois à son tour un message au Proxy de mobilité pour lui informer que l'ancienne adresse du terminal mobile n'est plus valide et pour lui communiquer la nouvelle. Dans cette méthode, le correspondant n'est plus au courant du changement d'adresse effectué. Il continu à envoyer les paquets vers l'ancienne adresse du terminal mobile. A la réception du message de mise à jours envoyé par le SIP REGISTRAR, le Proxy de mobilité crée une adresse virtuelle en utilisant l'ancienne adresse du terminal mobile. Il peut donc recevoir les données destinées à cette adresse et les transférer vers la nouvelle. Le SIP REGISTRAR et le Proxy de Mobilité peuvent être hébergés dans le même serveur.

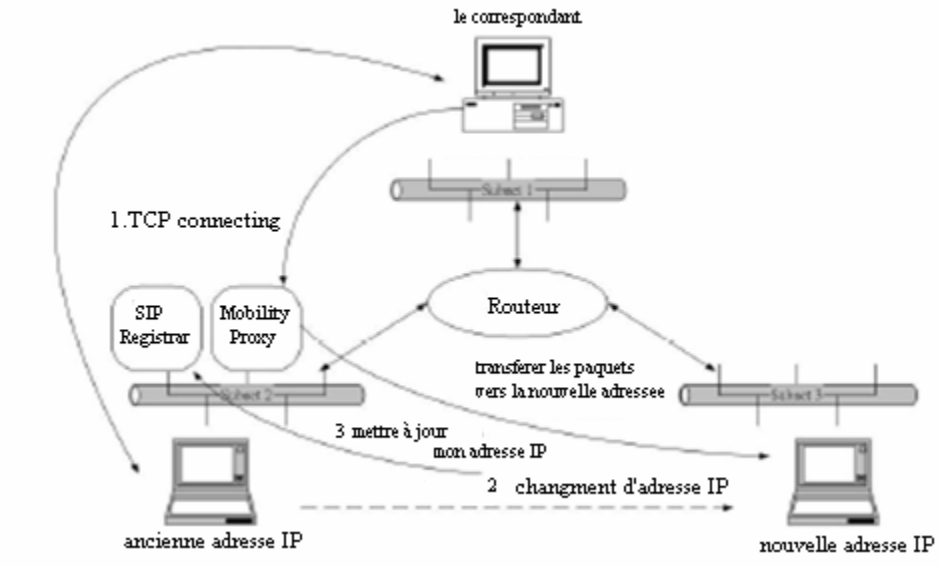


Figure 6 – La gestion de la mobilité dans ALMP

Cette méthode a été testée d'abord en utilisant *ipchains*, puis son ascendant *iptables*. Les résultats obtenus indiquent qu'avec *ipchains* on peut assurer la portabilité et avec *iptables* on peut assurer la continuité d'une connexion TCP ou RTP/UDP dans un environnement mobile et sans fil.

6 Solutions sans serveur ou proxy de mobilité

3.1 Solutions basées sur les sockets

Dans cette partie nous allons décrire quelques approches qui permettent de gérer la mobilité des utilisateurs sans l'intervention d'une tierce partie tels que les serveurs ou les agents de mobilité.

6.1.1 Mobile Socket

Mobile Socket [OMTT 99] est une amélioration de la bibliothèque socket (java .net.socket) dans le but d'offrir aux applications java un support de mobilité. Bien que *Mobile Socket* soit classée parmi les approches de la couche session, elle peut être considérée comme une approche de la couche application en utilisant la pile protocolaire TCP/IP. L'avantage de Mobile Socket est qu'elle n'exige aucune modification ni recompilation aux applications java déjà existantes en leur fournissant un mécanisme de *redirection de connexion implicite*. Ce mécanisme permet de rediriger automatiquement une connexion sans que l'application ne soit au courant du changement du point d'attachement du mobile. C'est à la bibliothèque *MobileSocket* de détecter la déconnexion du nœud Mobile et de faire appel à la redirection implicite.

Une connexion '*Mobile Socket'* peut être aussi maintenue à l'aide du mécanisme de la *redirection explicite* par lequel les programmeurs des applications java spécifient d'une manière explicite où la redirection peut avoir lieu. Les méthodes '*MobileSocket#Suspend'* et '*MobileSocket#Resume'* sont appelées dans ce cas afin de suspendre ou reprendre une connexion (voire figure 7).

```
public class MobilityEvent extends AWTEvent{}

public interface MobilityListener
                              extends EventListener{
  public void ConnectionSuspended(MobilityEvent e)
  public void ConnectionResumed(MobilityEvent e)
}
```

Figure 7 - l'événement de mobilité de la classe Mobile Socket

Pour maintenir une session de communication avec un mobile, *MobileSocket* fournit aux applications java le mécanisme DSS (Dynamic Socket Switching). Ce mécanisme permet de créer une nouvelle connexion (socket) entre le correspondant et le terminal mobile chaque fois que ce dernier change son point d'attachement.

Comme toute approche de gestion de mobilité, *Mobile Socket* propose un mécanisme de *'buffering forwarding'* afin de faire face au problème de la perte de données pendant les périodes de déconnexions. Mobile socket propose d'implémenter ce mécanisme tout simplement au niveau de la couche application. L'idée de base est la suivante : lors de l'établissement d'une connexion '*Mobile Socket*', les données à envoyer seront stockées au niveau d'un bufeur ALW (Application Layer window) se trouvant au niveau de l'émetteur. De son côté, le récepteur stocke le nombre d'octets lus dans le compteur 'ALW_COUNTER' (figure 8). Lorsque la valeur de ce dernier devient égale à celle de la taille du buffeur ALW, un accusé de réception est envoyé à l'émetteur. Ainsi en cas d'une déconnexion, et pendant la phase de la redirection, les deux bibliothèques s'échangent le nombre d'octets écrits et le nombre d'octets lus. Dans le cas ou le nombre d'octets lus diffère de celui du nombre d'octets écrits, l'émetteur constate qu'il y a eu une perte de données, il re-envoie alors les données stockées dans le buffeur 'ALW'.

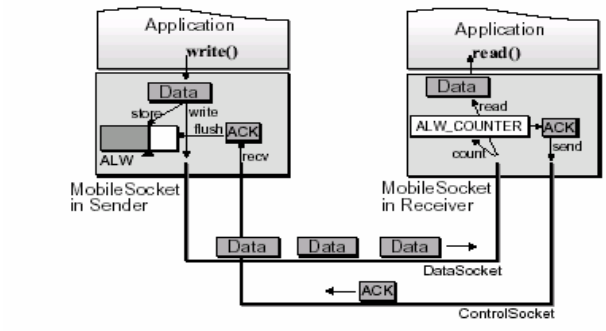


Figure 8 - Application Layer Window

6.1.2 RMS (Resilient Mobile Socket)

RMS (Resilient Mobile Socket) [KP 04], est une plateforme développée par la division Media Technology de l'université de Lulea dans le but de permettre aux applications multimédia temps réel de préserver leurs communications en cas de déconnexions causées par un hand off ou par un changement d'interface réseau. RMS est basée sur le principe d'encapsulation de plusieurs sockets internes représentant chacune un point d'entrée au réseau (figure 9). Ainsi en cas de déconnexion, la socket interne active pendant la communication sera remplacée par une des sockets internes disponibles dans RMS et donc l'application ne

sera perturbée que si aucune socket interne n'est disponible. Dans ce cas la session de communication est suspendue en appelant la procédure suspend. RMS supporte les deux types de hand off à savoir : le hard hand off[10] et le soft hand off[11]. Dans le hard hand off (HHO dans la figure 9), RMS supprime d'abord la socket active avant de créer une nouvelle socket interne. Et dans le soft hand off (SHO dans la figure 9) RMS utilise simultanément plusieurs sockets internes pour l'envoi et la réception des données. RMS traite le hand off comme étant une migration des flux de données entre les différentes sockets internes.

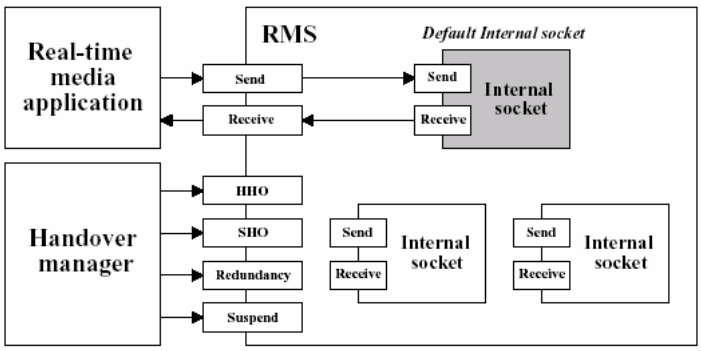


Figure 9 - Architecture RMS (Resilient Mobile Socket)

Pour cacher à l'application l'effet de la commutation des sockets internes ainsi que pour garantir le routage des paquets, RMS propose le mécanisme de translation de paquets. L'idée est de re-timbrer tous les paquets entrants de manière à ce que l'application les considère comme venant de leur emplacement d'origine (on cache à l'application que son correspondent a changé d'adresse) et de re-timbrer tous les paquets sortants de manière à ce qu'ils soient redirigés vers la nouvelle localisation du terminal mobile (réseau visité).

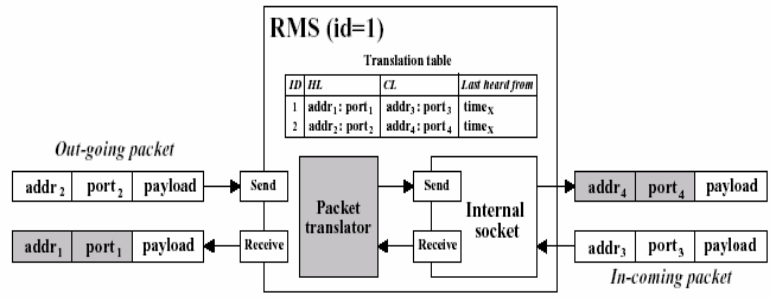


Figure 10 - Processus de translation de paquet.

[10] Le *hard hand off* est une approche simple qui, au prix d'un certain taux de perte de paquets, minimise le trafic de signalisation.

[11] Le *soft hand off* exploite le fait que les hôtes mobiles peuvent recevoir des paquets simultanément de la nouvelle et de l'ancienne station de base durant le *hand off*, minimisant ainsi la perte de paquets et fournissant ainsi des performances améliorées pour les protocoles.

La figure ci-dessous représente un exemple de translation de paquet ; un RMS a changé son point d'attachement de addr1 :port1 à addr3 : port3 (le terminal mobile sur lequel s'exécute l'application a changé son point d'attachement de : adresse1 à adresse3). Son correspondant se déplace lui aussi de addr2 :port2 à addr4 :potr4. Pour préserver la communication après le mouvement des deux terminaux mobiles, la translation suivante doit être effectuée.

$$Send(p) : T(p.addr_2 : p.port_2) \longrightarrow p.addr_4 : p.port_4$$

$$Receive(p) : T(p.addr_3 : p.port_3) \longrightarrow p.addr_1 : p.port_1$$

Afin d'effectuer des translations correctes, chaque RMS doit avoir à son niveau une table de translation qui contient la position originale (home location) et la position courante (curent location) des deux points de la communication.

3.2 MVOIP (Mobile Voice Over IP)

MVOIP [AK 02] est un protocole de la couche application qui supporte la mobilité des terminaux dans les applications VOIP (Voice Over IP). MVOIP propose un mécanisme de gestion de mobilité similaire à celui proposé par SIP et par MITP, sauf que MVOIP n'exige aucune intervention des serveurs ou des agents de mobilité. Le principe de base est le même dans le sens où c'est au mobile de communiquer sa nouvelle position à son correspondant. Cependant, le correspondant peut bondonner la communication s'il découvre la coupure de la connexion avant la réception du message de mise à jour de localisation (IP Update dans MVOIP). Afin d'éviter une telle situation, le terminal mobile envoie un message d'alerte de mobilité (Mobility Alert) à son correspondant juste après la découverte du changement du point d'attachement et avant l'acquisition d'une nouvelle adresse IP. De cette façon le correspondant met en attente la communication et à la réception du message 'IP Update' il met à jour la connexion ce qui permet de poursuivre la communication. Bien que MVOIP peut être utilisé par n'importe quelle application temps réel, son implémentation courante est basée sur le standard multimédia H.323 (Figure 11).

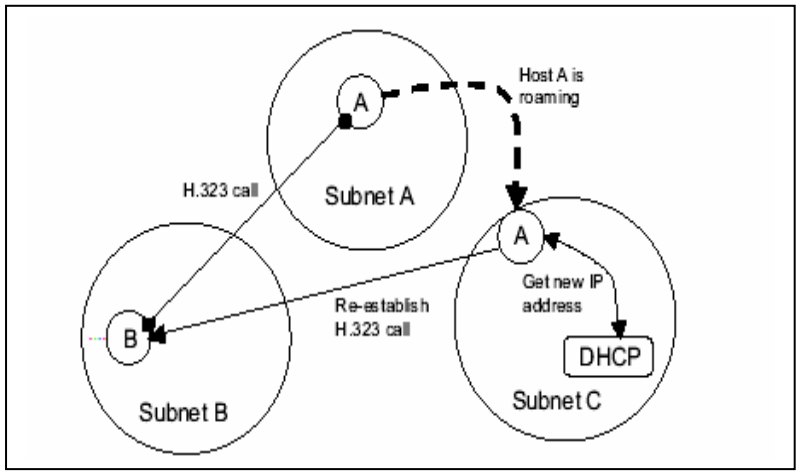


Figure 11 - Changement de sous réseau en utilisant MVOIP

Conclusion

Le problème de gestion de la mobilité est un problème complexe qui peut être traité selon différents points de vu et aux différents niveaux de la pile protocolaire. La gestion de la mobilité au dessous de la couche application résout les problèmes de mobilité mais requiert la modification de la pile protocolaire et de l'infrastructure réseau. Pour surmonter cette barrière la gestion de la mobilité au niveau de la couche application a été envisagée. L'avantage majeur de cette approche est la facilité de mise en œuvre et la non nécessité de déploiement d'une architecture très sophistiquée de réseau.

D'après les solutions rencontrées dans notre recherche bibliographique, on peut constater la faisabilité de la gestion de la mobilité au niveau de la couche application. Mais quand aux performances d'une telle approche on ne peut pas se prononcer avec exactitude car chaque approche est évaluée dans un environnement de test différent de celui des autres. De plus chaque approche est proposée pour un type spécifique d'applications. Le tableau 2 résume les résultats expérimentaux des évaluations de performances des approches décrites dans ce chapitre. Bien que la solution ALMP supporte tous les types d'applications (TCP et UDP), elle reste toujours liée aux services développés par le système d'exploitation linux.

Avec *Mobile Socket*, le taux de perte des paquets est nul et la solution n'exige aucun proxy de mobilité et même si elle est développée initialement pour les applications java, il est envisageable de l'adapter aux autres types applications.

Le chapitre 4 suivant est consacré à l'étude qualitative d'un protocole de validation atomique qui adapte le protocole 2PC à la mobilité. Mobile-2PC a été conçu avec l'idée de pouvoir s'exécuter sur les réseaux n'offrant aucun support de mobilité.

		Mise en oeuvre	La latence du handover	Taux de perte	Spécificité
Solutions utilisant un Proxy de mobilité	**SIP**	Architecture déjà déployée, mais nécessite des mécanismes d'encapsulation en cas d'une connexion TCP	Considérable environs 1.5 seconds. Ce qui est pénalisant pour les applications multimédia temps réel.	Moins de 8%	Solution dominante Pour les applications temps réel
	ALPM	Simple : basé sur la configuration de quelques outils applicatifs de linux	Temps du handoff niveau 2 + temps d'acquisition d'une nouvelle adresse IP + temps nécessaire pour la redirection des medias vers la nouvelle localisation.	Nombre de paquets perdu= ($T_{\text{changement du réseau}} + T_{\text{enregistrement}} + T_{\text{transfert de données}}$) * Taux génération de paquets au niveau du correspondant	- Applications temps réel (UDP) et non temps réel (TCP) - Linux comme OS
	MITP	Nécessite l'implémentation des agents mère et des agents étrangers. Cette tâche repose sur la réutilisation du code http et SIP	_	_	La voix sur IP
Solutions sans Proxy de mobilité	**MVOIP**	Simplicité de mise en oeuvre	3.7 seconds dont 3.1 sont passés dans l'obtention d'une adresse IP à partir d'un serveur DHCP.	_	Applications voix sur IP (transport UDP)
	Mobile Socket	Simple : implémentation de niveau utilisateur	C'est le temps écoulé entre la suspension et la reprise d'une session. Estimé par 316..95 ms /* cette valeur dépend du compilateur java et de la machine virtuelle java*/	0 % car mobile socket fournit la continuité du circuit virtuel	Applications java.
	RMS	simple	_	Entre 10% et 14 %	Application multimédia temps réel

Tableau 2 – Résumé des solutions proposées pour la gestion de la mobilité au niveau de la couche application

Chapitre 4

Évaluation de performances qualitatives du protocole M-2PC.

Introduction

De façon générale, le choix d'un protocole dépend (1) des propriétés que celui-ci doit assurer, (2) des hypothèses faites et (3) du degré d'autonomie que l'on désire laisser aux participants [BLRS 04]. Comme tout protocole de validation atomique dédié à l'environnement mobile, M-2PC doit garantir les propriétés qualitatives et quantitatives d'un ACP (Atomic Comittment Protocol) à savoir l'atomicité, la tolérance aux pannes et l'autonomie, tout en optimisant l'utilisation des ressources de l'environnement mobile et sans fil et en palliant aux problèmes de cet environnement.

Ce chapitre présente une étude qualitative du protocole M-2PC. Une description en détail de ce protocole fait l'objet de la section 1. La section 2 présente une analyse qualitative, alors que la section 3 est consacrée aux modifications proposées à M-2PC.

2 Le protocole Mobile Two Phase Commit (M-2PC)

Le protocole M-2PC [NDD 05a][NDD 05b] est, comme son nom l'indique, un protocole de validation à deux phases pour l'environnement mobile. Son objectif est de valider de manière globale une transaction mobile T_m qui s'exécute sur plusieurs sites dont certains sont mobiles. Le principe de base du protocole M-2PC est le suivant :

Une transaction mobile T_m est lancée par un hôte mobile (HM) appelé ***Home-MH***, attaché à une station de base appellée ***Home-BS***. Lors de son mouvement, HM peut changer de cellule.

Cette nouvelle cellule est appelée ***Current-BS.*** Le processus de validation est déclenché par le ***Home-MH*** qui en lançant une requête de validation (*Commit-Request*), sa station de base courante (***Current-BS***) prend la responsabilité de coordonner les sites participants dans le processus de la validation. Elle joue le rôle de ***Commit-BS***. Ce rôle peut être joué par le ***Home-BS*** dans le cas où HM se déplace tout en restant dans la même cellule initiale.
De façon similaire à 2PC, la décision de la validation dans M-2PC est le résultat de la coopération de trois principales entités à savoir :

- Le déclencheur du processus de la validation, le client mobile initiateur de la transaction.
- Les participants, ensemble de sites fixes ou mobiles accédés par la transaction $T_{m.}$
- Le coordinateur, responsable de la coordination des participants dans le processus de la validation.

M2PC prend en compte la mobilité du client et celle des participants. Le scénario présenté dans la figure1 suppose que seulement le client est mobile tandis que les participants sont fixes.

3.3 Le cas d'un client mobile et des serveurs fixes

Supposons qu'une transaction mobile T_m est lancée par un client mobile HM. Supposons aussi que toutes les données accédées par T_m se trouvent dans des serveurs de bases de données fixes. Au moment de la validation de la transaction T_m, le client mobile délègue à sa station de base courante (*Current-BS*) la coordination des participants dans le processus de validation. Ceci permet à ce dernier de s'exécuter entièrement sur la partie fixe du réseau. Ce qui rend trivial le choix de 2PC pour la validation de la transaction. La stratégie de coordination adoptée par M-2PC est la coordination statique, c'est-à-dire le coordinateur s'exécute dans la même station de base (*Commit-BS*) durant tout le processus de la validation même si HM change de cellule. Le choix d'un tel type de coordination est basé sur le fait que M-2PC s'adresse à la mobilité du terminal durant la période d'exécution de l'ACP (Atomic Commitment Protocol). Cette période est très courte et ne nécessite pas la migration du contrôle pour la raison suivante: soit HM se déplace lentement et donc il y a une forte probabilité que l'ACP se termine dans la même cellule. Soit il se déplace rapidement, et dans ce cas la migration fréquente du contrôle augmente la latence du protocole ainsi que sa vulnérabilité [NDD 05a].

Lors du déclenchement du processus de validation, le client mobile envoie au coordinateur son journal en même temps que la requête de validation. Le processus de validation se déroule ensuite sur la partie fixe en utilisant le 2PC traditionnel (figure 1) en permettant au client mobile de se déconnecter volontairement du réseau. Après la réception

des acquittements des participants, le coordinateur informe le client sur le résultat de la validation et attend son acquittement avant de libérer les ressources détenues par $T_{m.}$

Durant l'exécution du processus de validation sur la partie fixe, HM peut effectuer un hand off et se connecter à une nouvelle cellule. Le coordinateur doit être au courant de la nouvelle localisation de HM afin de lui envoyer le résultat de la validation. M-2PC remédie à ce problème par l'ajout d'un nouveau message à l'ensemble des messages de 2PC. Le client mobile (HM) envoie un message 'I-M-HERE' au coordinateur pour l'informer de sa nouvelle localisation. Ainsi la mobilité des terminaux est traitée d'une manière intrinsèque par M-2PC et donc par l'application transactionnelle.

Le client mobile a besoin de stocker à son niveau l'identité et les informations de localisations du coordinateur pour les utiliser lors de son enregistrement avec la nouvelle station de base. Pour éviter de perdre ces informations, suite à une panne ou à une déconnexion soudaine, la sauvegarde de ces informations s'effectue par une écriture forcée juste avant l'envoi de la requête de validation. Du fait que le coordinateur est statique, l'écriture forcée des informations du coordinateur ne s'effectue qu'une seule fois.

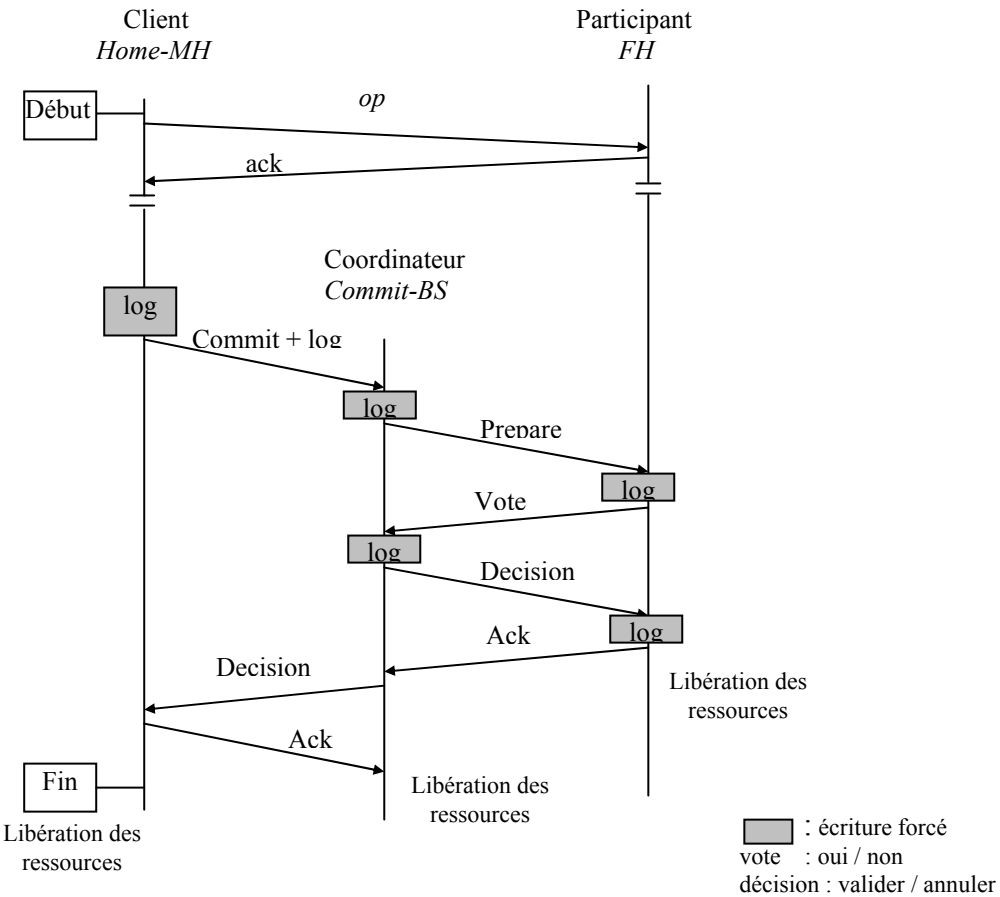


Figure 1 - Le protocole M-2PC avec seulement des participants fixes

3.4 Cas d'un client mobile avec au moins un serveur mobile.

La figure 2 décrit le fonctionnement de M-2PC dans le cas où au moins un serveur participant à la transaction, autre que l'initiateur, est mobile. De façon similaire au client mobile, le participant mobile dispose d'un agent délégué appelé ***Agent-Participant*** (P-Agent) qui se situe au niveau de la station de base à laquelle est attaché le participant lors de la réception du message '*prepare'*. Cet agent joue le rôle du représentant du participant mobile dans le processus de validation. À la réception du message '*prepare'*, Le participant mobile joint ses fichiers logs au message '*vote'*. Le message '*vote'* continue son chemin vers son destinataire (le coordinateur) alors que les fichiers logs sont utilisés par le P-Agent dans le processus de validation. Le participant mobile est libre de se déconnecter à partir du moment où il délègue son agent représentant. Le processus de validation se déroule selon les principes du 2PC traditionnel, et la décision est envoyée au participant mobile par son Agent représentant qui attend son acquittement avant de libérer les ressources. L'agent participant est aussi responsable de la sauvegarde des fichiers logs et du recouvrement dans les cas de pannes.

Le participant mobile peut effectuer un hand off durant le déroulement du processus de la validation sur la partie fixe. Ainsi et d'une façon similaire au client mobile, le participant mobile doit informer son délégué sur sa nouvelle localisation en lui envoyant le message *'I-M-HERE'* qui contient les informations de localisation du participant mobile. Même si T_m implique des sites mobiles pour son exécution, M-2PC réussi à la valider en utilisant uniquement les ressources de la partie fixe du réseau, ce qui permet de préserver l'énergie et les ressources de la communication en minimisant le trafic sur la partie sans fil du réseau.

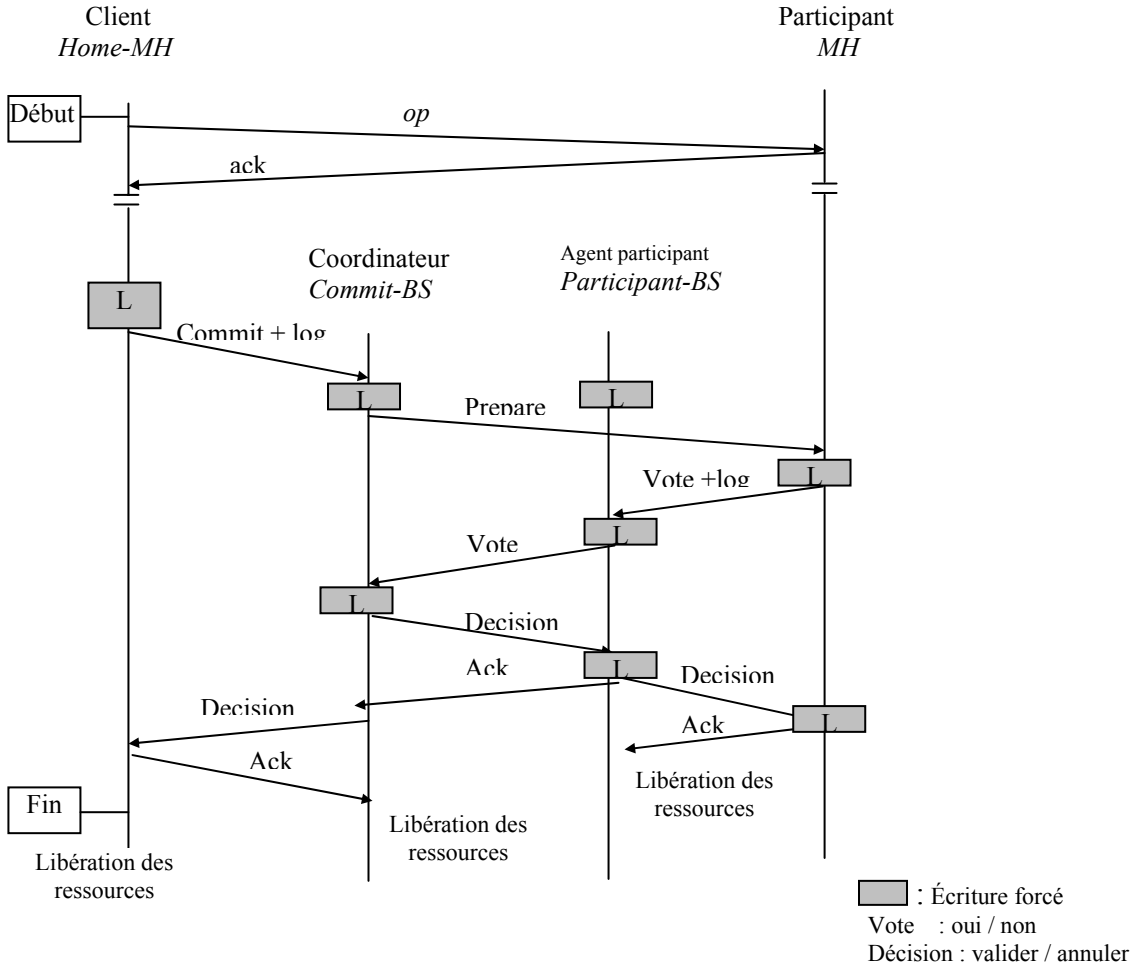


Figure 2 - Le protocole M-2PC Avec des serveurs mobiles

7 Analyse qualitative du protocole M-2PC

Dans cette section nous analysons le protocole M-2PC en termes de propriétés atomicité, tolérance aux panes, niveau d'autonomie imposé aux SGBD sous-jacents, optimisation des ressources de l'environnement mobile et sans fil et la gestion de la mobilité et des déconnexions des unités mobiles.

3.1 Atomicité

Les sites mobiles impliqués dans une transaction T_m (client ou participants) délèguent leurs contributions dans le processus de validation à leurs stations de bases courantes. La validation ou l'annulation d'une transaction se déroule alors en utilisant le 2PC traditionnel. Le résultat de la validation est ensuite envoyé aux sites (client ou participants) mobiles. Qui selon la décision du coordinateur traduisent ou non les effets de l'exécution de la transaction sur leurs bases de données locales. La propriété du tout ou rien (Atomicité) est donc assurée par M-2PC du moment qu'elle est assurée par 2PC. Cependant des contraintes sont imposées au système tels que le blocage des données (hérité du 2PC) et le transfert des journaux.

3.2 Tolérance aux pannes.

L'idée de base de M-2PC est de déléguer les stations de bases courantes des terminaux mobiles dans le processus de validation, ainsi la panne d'un site mobile n'affecte pas le déroulement de l'ACP.

3.3 Autonomie.

L'autonomie indique le degré d'indépendance des composants du système. Ceci concerne, entre autres, la liberté de choisir le type de données utilisées, la manière de les manipuler, la liberté de décider du type d'information à partager avec le système global, l'indépendance de l'exécution ainsi que la non imposition de changements au système [OV 99]. 2PC est le protocole de validation le plus répandu [BLRS 04], bien qu'il laisse le moins d'autonomie aux participants. Ce qui pour les sites mobiles est pénalisant. M-2PC allège ce problème en déléguant la contribution des sites mobiles dans le processus de validation aux sites fixes.

3.4 Optimisation des ressources de l'environnement mobile et sans fil.

M-2PC vise une optimisation accrue du point de vue communications sans fil. En délégant la contribution des unités mobiles dans le processus de validation à leurs stations de bases courantes, la validation effective sera réalisée sur la partie fixe du réseau. Seules la demande de validation (et / ou le vote dans le cas d'un participant mobile), le journal, et la décision seront transmis à travers le support sans fil.

Cependant, M-2PC introduit deux écritures forcées du côté du mobile. Une pour stocker de façon permanente les informations concernant le coordinateur/l'agent participant. Et l'autre pour sauvegarder la décision de la validation. De plus le journal de la transaction mobile est d'abords réalisé sur le mobile avant d'être transféré vers le Commit-BS ou le Participant-BS. La taille du journal doit être réduite au maximum afin de réduire le coût du transfert sur le lien sans fil, pour cela des techniques de compression peuvent être utilisées [Bob 02].

3.5 Les Déconnexions

A partir du moment où le mobile délègue sa station de base dans le processus de validation, il aura la liberté de se déconnecter ou de rester connecté au réseau. Dans le cas d'une pane ou d'une déconnexion soudaine, M-2PC suppose que le recouvrement du site mobile s'effectue en une période finie, durant laquelle les résultats seront gardés au niveau du coordinateur/l'agent participant. Ce dernier se charge de les transmettre au site mobile juste après sa reconnexion. M-2PC est donc capable de valider une transaction mobile même en cas de déconnexions inattendues des sites mobiles minimisant ainsi le risque d'abandon.

3.6 La mobilité (hand off et localisation).

Afin de contourner le coût considérable du développement de nouvelles solutions ou de modification des infrastructures réseaux existantes ou des systèmes d'exploitation opérants, M-2PC propose de gérer la mobilité des sites impliqués dans la transaction. M-2PC est supposé s'exécuter au dessus de la pile protocolaire TCP/IP sans aucune politique de gestion de mobilité [NDD 05b] [NDD 05c]. Lorsqu'un hand off se produit, un message 'I-M-HERE' est envoyé par le site mobile au coordinateur/l'agent participant pour l'informer de sa nouvelle localisation. Etant donnée que M-2PC est de niveau application, le message 'I-M-HERE' ne sera pas servi par une haute priorité. La procédure du hand off n'affecte pas l'atomicité du protocole mais peut augmenter la latence de ce dernier.

M-2PC a pour objectif de terminer une transaction mobile qui s'exécute au dessus du réseau Internet natif. M-2PC élimine toute technique de gestion de mobilité quelque soit son niveau d'implémentation dans la pile protocolaire, même s'il s'agit du module de gestion de mobilité ajouté à la couche réseau du modèle OSI, à savoir le Mobile-IP.

Rappelons que le réseau Internet est constitué d'un ensemble de domaines interconnectés via des liaisons filaires, mais permet aux terminaux portables de se déplacer tout en étant connectés à l'aide de son extension sans fil. Un domaine peut contenir plusieurs cellules. Lors de son mouvement, un terminal mobile (le client ou le participant dans le cas de M-2PC) peut effectuer (1) un hand off vers une cellule du même domaine, on parle de *mobilité intra-domaine* ou *micro mobilité* (figure 3), comme il peut (2) traverser les frontières d'une cellule appartenant à un autre domaine administratif, on parlera dans ce cas d'une mobilité *inter domaine ou macro mobilité* (figure 3). Le hand off induit par la mobilité intra domaine, appelé *hand off de niveau 2* ou *hand off L2*, affecte la couche liaison de la pile protocolaire du modèle ISO. Tandis que le hand off déclenché par la mobilité inter domaine, appelé *hand off de niveau 3* ou *hand off L3*, affecte la couche réseau du modèle ISO.

Le changement de cellule n'implique pas forcement le changement d'adresse IP. L'attribution d'une nouvelle adresse IP se fait seulement dans le cas où le mobile se déplace vers une cellule de domaine différent. Dans le cas contraire (mobilité intra domaine), l'adresse IP du mobile est toujours valide. Dans ce cas l'envoi du message 'I-M-HERE' est inutile. L'élimination du message 'I-M-HERE' n'influe pas sur les performances du protocole M-2PC mais requiert l'ajustement de ce dernier pour mieux fonctionner dans les environnements mobiles. Dans ce qui suit nous présentons nos modifications apportées au protocole M-2PC après élimination du message 'I-M-HERE'. Rappelons que ce message n'est éliminé que dans le cas de mobilité intra domaine.

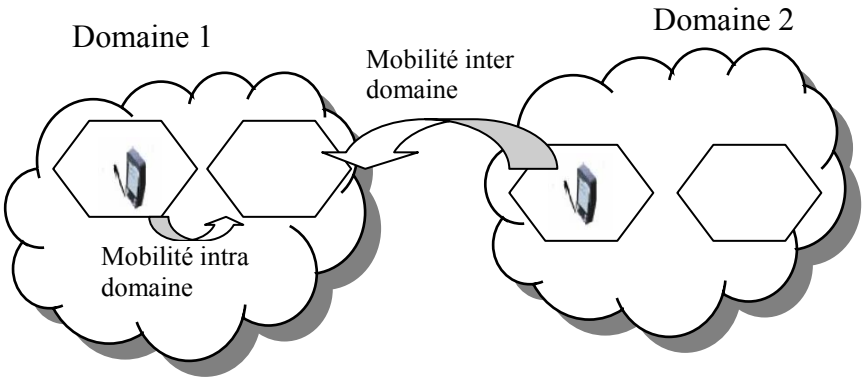


Figure 3 – Mobilité intra et inter domaine

8 Ajustement du protocole M-2PC pour les réseaux IP

3.1 Comportement de M-2PC en présence de hand off L2

Dans ce type de hand off, le site mobile change de cellule mais ne change pas d'adresse IP du moment qu'il s'est déplacé dans le même domaine. Ainsi, à la réception de tout les acquittements, le coordinateur envoie la décision de la validation au site mobile en utilisant son ancienne adresse sans se poser la question "a t- il changé de cellule ou pas ?" Le terminal mobile n'envoie pas dans ce cas le message *'I-M-HERE'* au coordinateur/agent participant pour lui communiquer ses informations de localisation. La communication entre le coordinateur/l'agent participant et le mobile s'effectue le plus normalement possible en utilisant les sockets TCP/IP *((adresse source, port source), (adresse destination, port destination))*. Du côté coordinateur l'adresse destination est l'adresse du mobile dans le cas où il désire lui envoyer la décision de la validation. Cette adresse ne change pas si le mobile se déplace vers une cellule du même domaine. Si le hand off L2 se termine avant la terminaison du processus de validation sur la partie fixe, la technologie sans fil utilisée prend la responsabilité de rediriger le message vers la nouvelle station de base. Si la décision est envoyée au site mobile pendant l'exécution de la procédure de hand off, sachant que durant cette période aucune communication n'est possible avec le terminal mobile, le message *'decision'* risque d'être perdu. Afin de remédier à ce problème nous proposons d'exploiter les techniques de buffuring forwarding [ZWZ 03] proposées par les normes de transmission sans fil. Ces techniques consistent à utiliser des tampons temporaires afin de stoker les paquets de données destinés au site mobile pendant la période du hand off. Ces données seront transférées par la suite vers la nouvelle station de base dés qu'une nouvelle connexion est établie (voire figure 4). Le message 'I-M-HERE' est un message ascendant[12]. Son élimination permet d'économiser l'énergie du site mobile et de réduire la latence du protocole M-2PC sachant que La mobilité intra domaine est plus fréquente que la mobilité inter- domaine.

[12]Un transfert de données est descendant lorsqu'il s'effectue du serveur (hôte) à l'utilisateur local (importation) et ascendant en direction contraire (exportation).

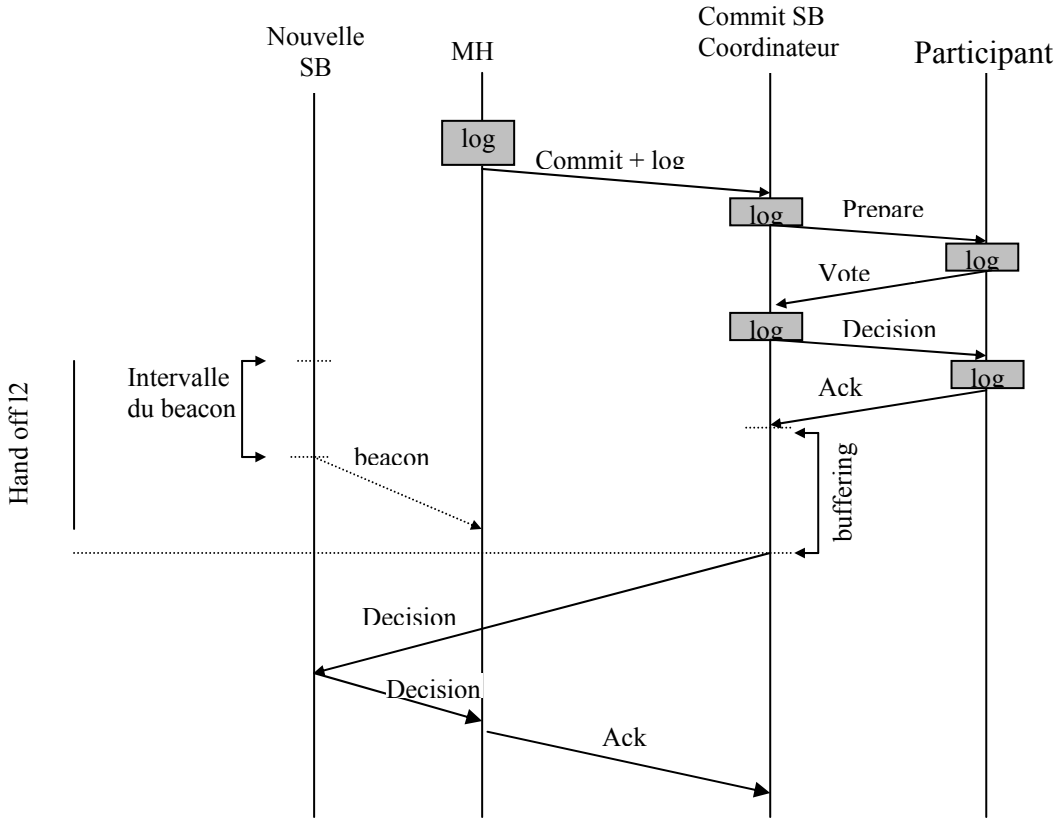


Figure 4 – Comportement de M-2PC dans le cas de mobilité intra domaine

3.2 Comportement de M-2PC en présence du hand off L3

Dans ce type du hand off le mobile change entièrement du domaine et donc son adresse IP ne sera plus valide dans ce nouveau domaine. Le mobile acquiert une nouvelle adresse en utilisant le serveur d'allocation dynamique d'adresse (DHCP), et envoie au coordinateur/l'agent participant le message '*I-M-HERE'* qui contient l'information nouvelle adresse. Le coordinateur/l'agent participant envoie la décision de la validation directement au mobile (voire figure 5).

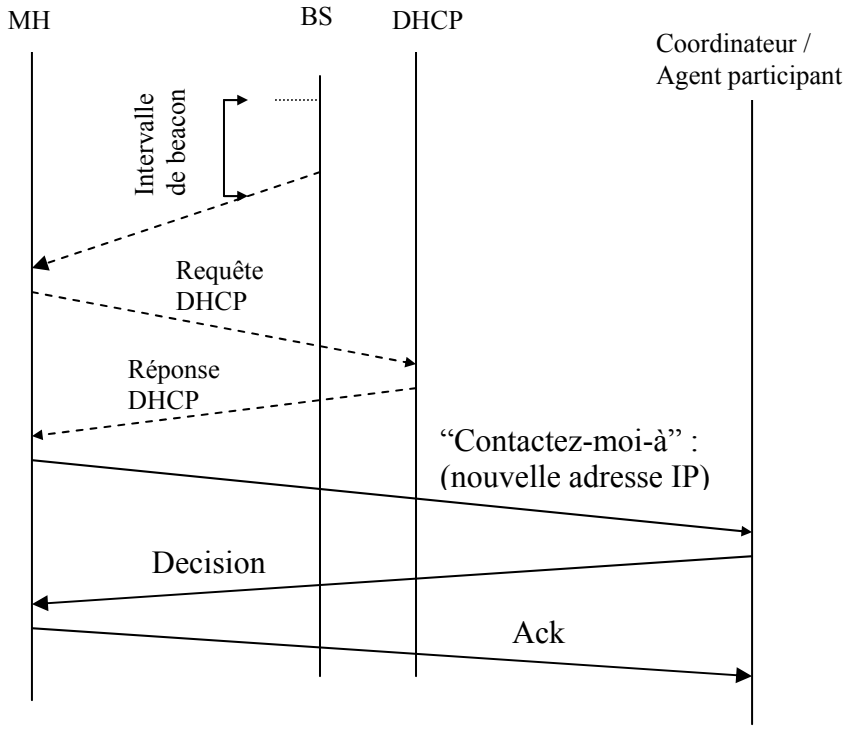


Figure 5 – Comportement de M-2PC dans le cas de mobilité inter domaine

L’envoi de la décision pendant l’exécution du hand off augmente le délai de la réception du ACK. Etant donnée que le protocole M-2PC est de niveau application, les entités coordinateur/agent participant ne peuvent distinguer entre les retards causés par les déconnexions et ceux causés par le déclenchement d’un hand off. Pour résoudre ce problème nous suggérons d’affecter aux délégués des sites mobiles (coordinateur ou l’agent participant) trois temporisateurs (T_1, T_2, T_3). Le premier est initialisé à la durée que peut prendre une déconnexion de courte durée. Le second prend la valeur temps nécessaire d’un hand off L2 plus le délai moyen de transmission du message ACK. La latence d’un hand off de niveau 2 dépend de la technologie sans fil utilisée. Et le troisième prend la valeur du temps nécessaire à la réception du message ‘I-M-HERE’. Cette valeur n’est autre que la somme des valeurs : latence du hand off L2, temps nécessaire pour obtenir une nouvelle adresse IP, temps nécessaire pour la reconfiguration du site mobile et le délai de transmission du message ‘I-M-HERE’.

Si le coordinateur ne reçoit pas l’acquittement de la décision pendant une période qui dépasse le délai T_1, il suppose l’apparition d’une déconnexion de courte durée. Le coordinateur/l’agent participant décide dans ce cas de renvoyer la décision. Si le ack est non encore reçu pendant un délai T_0 (délai moyen nécessaire à la réception d’un message ascendant), le coordinateur assume que le client est en train d’exécuter un hand off de niveau

2 et que l'acquittement peut être reçu avant l'expiration du timer T_2. Si tel n'est pas le cas, le coordinateur constate alors que le mobile a traversé les frontières d'un autre domaine et qu'il doit attendre l'arrivé du message 'I-M-HERE' pour renvoyer la décision à la localisation courante du site mobile. Si la réception du message 'I-M-HERE' ne s'effectue pas pendant un délai T_3, le coordinateur suppose que le client s'est déconnecté volontairement pour une longue durée. Le coordinateur décide alors de sauvegarder la décision à son niveau et de la transmettre au site mobile lors de sa prochaine connexion. La figure 6 décrit le comportement du coordinateur vis à vis des deux types de mobilité et des deux types de déconnexion (volontaire et involontaire).

La technique de gestion de mobilité offerte par M-2PC est classée parmi les approches qui ne se basent pas sur les serveurs ou les agents de mobilité. Plus précisément, elle est similaire à celle proposée par le protocole MVOIP (chapitre 3, section 2.2), sauf que dans MVOIP le mobile doit d'abord alerter son correspondant du mouvement avant de lui communiquer ces informations de localisation. Ceci afin d'empêcher le correspondant d'abandonner la communication en cas où il découvre la coupure de la communication avant la réception du message de mise à jour de localisation. Dans M-2PC, le client n'a pas besoin d'alerter le coordinateur pour les raisons suivantes : (1) le processus de validation est indépendant (parallèle) du processus hand off. (2) l'introduction des timers à M-2PC permet au coordinateur de livrer la décision de la validation dans tous les cas.

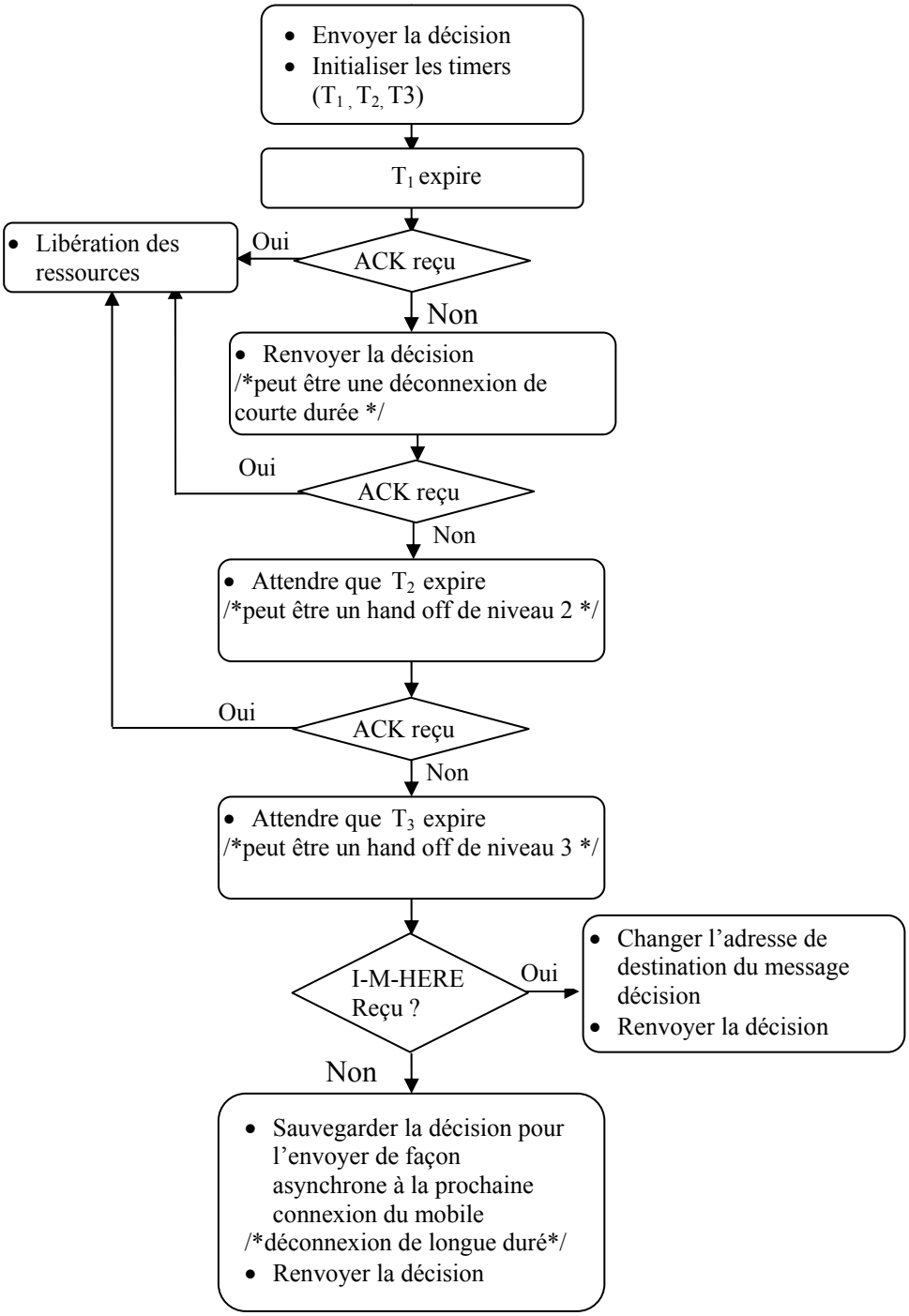


Figure 6 – Gestion des déconnexions et des hand off au niveau du coordinateur

Conclusion

Nous avons présenté une étude de performances d'un protocole de validation à deux phases pour des transactions mobiles. Durant notre étude qualitative nous constatons que M-2PC hérite les avantages et les inconvénients du protocole 2PC. Néanmoins M-2PC est indépendant de toute politique de gestion de mobilité, puisqu'il gère la mobilité des participants ou du client de façon intrinsèque. La mobilité prise en compte par M-2PC est la mobilité inter domaine. L'évaluation de performances de la technique de gestion de mobilité proposée par M-2PC fait l'objet du chapitre suivant.

Chapitre 5

Évaluation de Performances Quantitatives du Protocole M-2PC.

Introduction

Nous présentons dans ce chapitre, les résultats de l'évaluation des performances quantitatives du protocole M-2PC. L'objectif de ce chapitre est de (1) mesurer la latence de ce protocole en présence du hand off induit par la mobilité des clients et/ou des participants à la transaction mobile, (2) comparer les résultats obtenus avec ceux obtenus de l'exécution du protocole de validation traditionnel (2PC) au dessus de Mobile-IP. En effet, 2PC n'étant pas conçu pour l'environnement mobile, son exécution nécessite un support pour la gestion de mobilité qui peut être offert par Mobile-IP.

La micro mobilité, comme indiqué précédemment, peut être gérée au niveau de la couche liaison par l'implémentation de la 802.11 par exemple (voire section 3.1 du chapitre précédent). Et donc c'est le problème de la macro mobilité que doit résoudre le protocole M-2PC. Ce dernier doit être au courant du changement d'adresse IP que peut subir un terminal mobile lors de son mouvement. Dans les applications transactionnelles temps réel, qui sont sensibles au délai de transmission de données, la décision de la validation M-2PC doit être reçu par le site mobile le plus rapidement possible. Pour cela un mécanisme permettant de livrer la décision de la validation au site mobile même en cas de changement d'adresse IP, doit être implémenté au niveau des délégués des sites mobiles (Commit-BS/Agent-participant). Le message 'I-M-HERE' envoyé par le site mobile à son délégué, supposé être

fixe, permet à ce dernier de directement livrer la décision de la validation à la nouvelle adresse du site mobile sans l'intervention d'une tierce[13] partie. Ce qui permet, à première vue, de diminuer la latence du protocole M-2PC. Réduire la latence d'un protocole de validation permet de réduire la durée d'une transaction et de ce fait, de libérer plus rapidement les ressources qu'elle détient sur chaque participant [AP 98].

Nos mesures ont été réalisé dans l'environnement de simulation de réseaux NS (Network Simulator) pour les raisons citées dans la section 1. Cette section fait l'objet d'un bref aperçu sur l'environnement de simulation NS. L'implémentation du protocole M-2PC dans cet environnement est décrite dans la section 2. Dans la section 3 nous définissons les différents paramètres qui peuvent influencer la latence du M-2PC. L'impact de chaque paramètre est discuté dans la section 4. Enfin une comparaison avec l'implémentation du 2PC traditionnel au dessus de Mobile-IP est présentée dans la section 5.

3 L'environnement de simulation NS.

La grande hétérogénéité ainsi que l'évolution rapide du réseau Internet, rendent la simulation de l'Internet difficile [PF 97]. L'hétérogénéité commence avec les liens, comprend les protocoles et finit avec les trafics d'applications diverses et variées. Afin de permettre aux terminaux portables de se déplacer librement tout en étant connecté, une extension sans fil a été introduite au réseau Internet. Cette extension rend la simulation du réseau Internet plus compliquée.

Les environnements NS (Network simulator) [NS 03], Opnet [Opn 04] et GloMoSim [Glo 01] représentent les projets de simulateurs de gestionnaires réseau les plus mûrs et les plus utilisés par la communauté des chercheurs. Ns et GloMoSim sont en libres accès aux chercheurs. L'environnement GloMoSim est destiné uniquement à simuler des réseaux ad hoc. Or le protocole M-2PC est conçu pour des réseaux mobiles avec infrastructure. De ce fait, l'environnement NS est le plus adéquat pour évaluer les performances quantitatives du protocole M-2PC.

Le simulateur NS, développé par le projet VINT[14] (Virtual InterNeworked Testbed) de l'université de Californie, est particulièrement adapté aux réseaux à commutation de paquets et à la réalisation de simulation de petite taille. Il contient les fonctionnalités nécessaires à l'étude des différents protocoles de communication filaire et sans fil. Sa richesse en configurations et en paramètres ouvre le champ à l'étude de nouveaux mécanismes au niveau des différentes couches de l'architecture réseau. Ns est devenu l'outil de référence pour les chercheurs du domaine [AH 99].

[13] La plupart des approches de gestion de mobilité requièrent l'existence d'un serveur ou d'un agent de mobilité afin de router les données à la nouvelle localisation du site mobile.
[14] http:// www.isi.edu.nsnam/ns

3.3 Architecture de NS

Ns est implémenté en C++ et en OTCL (Object oriented Tool Command Language). Le code C++ est utilisé pour implémenter les détails du protocole à étudier, alors que le code OTcl est utilisé pour la configuration et le contrôle des scénarios de simulations. Le package NS inclue une hiérarchie de classes compilées d'objets C++ et une autre interprétée d'objets OTcl. La correspondance entre les objets compilés et les objets interprétés s'effectue à l'aide des classes suivantes :

- La classe Tcl : cette classe comprend les méthodes utilisées par le code C++ pour accéder et communiquer avec l'interpréteur.
- La classe TclObject : cette classe englobe tout les objets (compilés et interprétés) du simulateur NS (voire figure 7). Elle prend la responsabilité d'établir les liens entre les différents types de variables (entier, réel, bande passante, temps, booléen) de la hiérarchie compilée avec ceux de la hiérarchie interprétée.
- La classe TclClass : cette classe est très importante lors de l'implémentation d'un nouveau protocole. Elle offre deux fonctionnalités : (1) construire la hiérarchie interprétée pour correspondre à la hiérarchie compilée ; (2) offrir les méthodes à l'instanciation d'un nouveau objet TclObject.

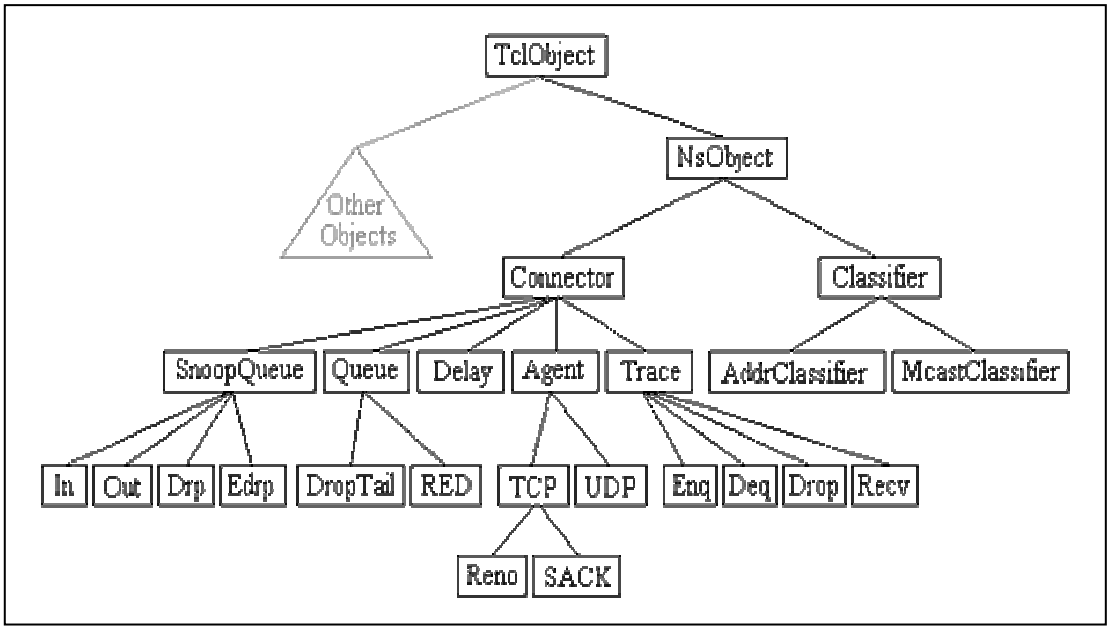


Figure 7 – Une partie de la hiérarchie de classes des objets NS

3.4 Simulation des réseaux mobiles et sans fil sur NS

Le modèle de réseau sans fil de base est constitué essentiellement des nœuds mobiles, implémentés dans la classe MobileNode, et des fonctionnalités nécessaires à la simulation des réseaux ad-hoc et des WLAN (figure 8). Un nœud mobile hérite les fonctionnalités d'un nœud fixe et est enrichi des fonctionnalités de support de mobilité par exemple, Le déplacement dans une certaine topologie, l'émission et la réception des signaux sur un canal sans fil, etc. La pile protocolaire réseau d'un nœud mobile, décrit dans la figure 8, consiste en un ensemble de composants connectés au canal de communication :

- La couche liaison (Link Layer) simule les protocoles liaison du modèle ISO.
- Une file d'attente des priorités (notée ifq) qui donne des priorités aux paquets de routage lors d'une transmission.
- La couche MAC (Medium Access Control) implémente la norme IEEE 802.11 et TDMA (Time Division Multiple Access) pour le contrôle d'accès au canal de communication.
- Les interfaces réseaux qui simulent des interfaces hardwares utilisées par le nœud mobile pour accéder au canal de communication.
- Le modèle de propagation radio. Deux modèles sont définis : le modèle espace-friss pour les courtes distances et le modèle Two Ray Ground pour les grandes distances.
- L'antenne simulée est une antenne omnidirectionnelle.

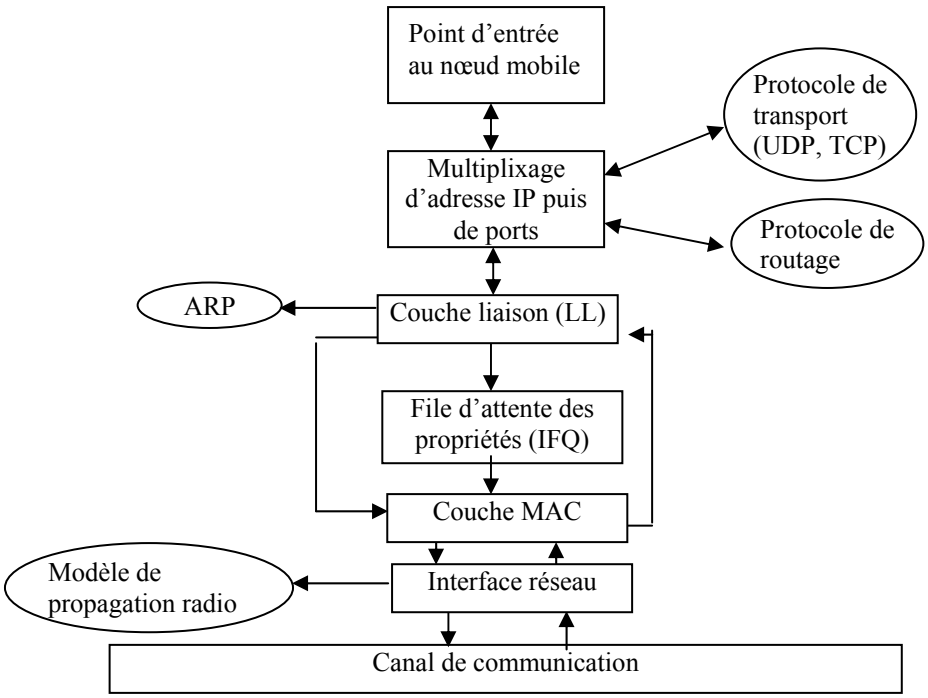


Figure 8 – Les composants réseau d'un nœud mobile dans NS

Afin d'acheminer les paquets de données vers un nœud mobile, le modèle sans fil implémente quatre protocoles de routage à savoir : DSDV (Destination Sequensed Distance Vector), DSR (Dynamic Source Routing Protocl), TORA (Temporally Ordered Routing Algorithm), AODV (Ad hoc On-demand Distance Vector Routing).

Les nœuds mobiles peuvent se déplacer dans une topographie 3D selon les trois axes (X, Y, Z). Le paramétrage des mouvements d'un nœud mobile se fait à l'aide d'un des deux mécanismes suivants :

1. Indiquer le point d'origine, la destination et la vitesse explicitement pour chaque nœud mobile. Les mises à jours sont déclenchées chaque fois que l'on exige la position du nœud mobile à un moment donné.
2. Générer des mouvements aléatoires des nœuds mobiles. Le nœud mobile de déplace aléatoirement en démarrant à partir d'une position aléatoire. Il exécute des mises à jours de routage pour changer de destination et de vitesse.

Rappelons que l'objectif principal de cette partie est d'évaluer la performance de la technique de la gestion de mobilité offerte par le protocole M-2PC. Pour cela nous devant déclencher des hand off de niveau 2 et de niveau 3 pendant le déroulement du processus de la validation. De ce fait, et d'une manière similaire au protocole de gestion de la mobilité Mobile- IP [Gre 03], le paramétrage de mouvements des nœuds mobiles doit être réalisé à l'aide du premier mécanisme.

Le modèle décrit ci-dessus ne permet que la simulation des réseaux Ad hoc et WLAN [NS 03]. Ce modèle a été étendu pour le besoin de simuler des topologies constituées de plusieurs WLAN connectés via des nœuds fixes. L'objectif est donc de relier des réseaux locaux sans fil à des réseaux filaires. Un nouveau type de nœud, appelé station de base (la classe BaseStationNode), est alors créé pour assurer la liaison entre la partie fixe et la partie sans fil.

9 Implémentation de M-2PC dans NS.

Pour implémenter Le protocole M-2PC dans NS, nous avons décidé d'implémenter d'abord le protocole 2PC puis d'introduire le caractère mobilité des terminaux. La structure des messages M-2PC ainsi que les comportements des entités initiateur, coordinateur et participant sont décrits dans ce qui suit.

3.1 Les messages M-2PC

Les messages échangés entres les entités du protocole M-2PC comportent les champs suivants :

<Transaction ID>
Identificateur de la transaction.

<Type>
Le type du message : ce champs prend les valeurs suivantes :
M2PC-Commit-Request : le client envoi la requête de validation plus le journal de la transaction. La taille de ce message varie en fonction de la taille du journal de la transaction.
M2PC-Prepare : envoyé par le coordinateur en multi-cast aux participants à la réception du message Commit-Request.
M2PC-Vote : envoyé par un participant à la réception d'un message prépare, le résultat du vote est indiqué dans le champs Vote décrit ci-aprés.
M2PC-Decison : ce message est envoyé par l'entité coordinateur à la réception de tout les messages votes de la part des participants.
M2PC-ACK : l'acquittement du message décision envoyé par chaque participant.
M2PC-I-M-HERE : envoyé par le site mobile (participant ou client) pour informer son délégué sur sa nouvelle adresse IP.
M2PC-Commit-Responce : c'est le résultat de la décision de la validation envoyé au site mobile. Il a la même valeur que le message *M2PC-Decision,* mais il n'est envoyé qu'après réception des *M2PC-ACK*.
M2PC-ACK-M : l'acquittement du message Commit- Responce.

<Vote>
Ce champ contient le vote du participant. Il prend la valeur 'C' pour commit 'A' pour Abort et 'Null' dans les autres cas (le champ type différent de M2PC-Vote).

<Decision>
Contient la décision de la validation. Il prend la valeur 'C' pour commit, 'A' pour Abort 'Null' dans le cas où le type du message est différent de M2PC-Decision.

<Source>
L'ancienne adresse IP du site mobile émetteur du message 'I-M-HERE', 'Null' pour les autres types de message.

<Contact>
Ce champ contient la nouvelle adresse IP du site mobile dans le cas du message 'I-M-HERE', 'Null' pour les autres types de message.

<Sendtime>
Ce champ contient le temps en seconde d'envoi de la requête de validation 'Commit-Request'. Ce champ est utilisé pour des besoins d'étude de performance (calcul de la latence).

3.2 L'initiateur

L'initiation du processus de la validation s'effectue à partir de l'interface Tcl à l'aide de la commande : **ns at** moment 'client **send** commit request'. Au niveau du client mobile un message M-2PC est créé ayant pour type M2PC_COMMIT_REQUEST, et affecte au champ Sendtime la valeur actuelle de l'horloge (afin de calculer la latence du protocole). Ce message est ensuite envoyé au coordinateur. Le pseudo code de l'entité initiateur est décrit dans la figure 9.

```
{
Allouer_nouveau paquet();
Nouveau_paquet->type = M2PC-Commit-request ;
Joigner_journale() ;
Nouveau_paquet-> sendtime = la valeur actuelle de l'horloge;
send (nouveau_paquet, 0);
}
```

Figure 9 – Initiation de M-2PC

3.3 Le coordinateur

A la réception du message 'C*ommit-Request'*, le coordinateur crée le message '*prepare', et* l'envoie en multi-cast à tous les participants. Les figures 10 et 11 décrivent le pseudo code des processus de création et d'envoi des deux messages '*prepare'* et '*decision'* respectivement par le coordinateur aux participants.

```
{
Allouer_nouveau paquet();
Nouveau_paquet->type = M2PC-Prepare ;
send_multicast (nouveau_paquet, 0);
}
```

Figure 10 – Création et envoi du message '*Prepare'*

```
{
Allouer_nouveau paquet();
Nouveau_paquet->type = M2PC-Decison ;
Nouveau_paquet->Decison = la decison de la validation ;
send_multicast (nouveau_paquet, 0);
}
```

Figure 11 – Création et envoi du message *'Decision'*

3.4 Le participant

Un participant qui reçoit le message '*prepare'*, sauvegarde son état local et envoie au coordinateur son vote. L'opération de sauvegarde de l'état local s'effectue par une écriture forcée. Cette opération dure dans la plus part des cas dans les environs de quelques millisecondes pour les sites fixes et une dizaine de millisecondes pour les sites mobiles [NSB 97][CBML O3]. Durant notre étude nous considérons des participants avec des capacités de calcul différentes. Ainsi le temps nécessaire pour une écriture forcée diffère d'un participant à un autre. A chaque participant, et à la réception du message '*Prepare,* est affecté un *timer* initialisé aléatoirement entre le temps minimum et le temps maximum que peut prendre une écriture forcée. A l'expiration de ce timer, le participant envoie son *vote* au coordinateur. Le même principe est utilisé par l'initiateur et le coordinateur pour toutes les opérations qui nécessitent une écriture forcée. Les figures 12 et 13 décrivent le comportement du participant lors de la réception des messages '*prepare*' et '*decision'*.

```
{
 Si message_reçu == 'Prepare' alors
 vote_timer = Random::uniform(0.008, 0.01); // temps nécessaire pour
   une écriture forcée.
 Allouer_nouveau paquet();
 Nouveau_paquet->type = M2PC-vote ;
 Nouveau_paquet->Vote = le vote du participant;
 Send (nouveau_paquet, 0);
}
```

Figure 12 - Création et envoi du message vote

```
{
  Si message_reçu == 'Decision' alors
ACK_timer = Random::uniform(0.008, 0.01); // temps nécessaire pour
une écriture forcée.
Allouer_nouveau paquet();
Nouveau_paquet->type = M2PC-ACK ;
send (nouveau_paquet, 0);
}
```

Figure 13 - Acquittement de la décision

3.5 Le support de mobilité.

Comme indiqué précédemment, pour implémenter M-2PC dans les réseaux IP nous supposons que la mobilité intra domaine est prise en charge par la couche liaison (la technologie sans fil). Et c'est à M-2PC de gérer la mobilité inter domaine. L'implémentation de ce type de mobilité requiert l'introduction de l'agent DHCP au simulateur NS. L'agent DHCP est responsable de l'allocation de nouvelles adresses IP aux terminaux mobiles. Le pseudo code de l'implémentation du DHCP est présenté dans la figure ci-dessous. En réalité, la duré du processus d'obtention d'une nouvelle adresse IP est entre une centaine de milli secondes (environs 800ms) et plus d'une seconde, selon l'utilisation d'un serveur DHCP ou d'un serveur DRCP [KKLY+ 04]. Des mesures pratiques du délai nécessaire pour l'obtention d'une adresse IP à partir d'un serveur DHCP ont été prises au niveau du département KReSIT, IIT Bombay. Ces expérimentations affirment que le temps nécessaire pour l'obtention d'une adresse IP est égal à 1.76 secondes [Kum 03]. Afin d'obtenir des résultats proches de la réalité, l'agent DHCP ne doit répondre à une requête DHCP qu'après écoulement d'une période du temps prise aléatoirement entre l'intervalle 800ms et 1760ms. Pour cela il utilise le même principe des timers décrit dans le paragraphe précédent.

```
 {
Si message_reçu == 'DHCP_REQUEST' alors
nouvelle_IP = allouer_NouvelleIP();

DHCP_timer = Random::uniform(0.8, 1.76); // temps
nécessaire pour pour obtenir une adresse IP.
Allouer_nouveau paquet();
Nouveau_paquet->type = DHCP_REPLY;
send (nouveau_paquet, 0);
}
```

Figue 14 - Implémentation du DHCP dans NS

A la réception du message DHCP_REPLY envoyé par l'agent DHCP, le terminal mobile récupère l'information *'nouvelle adresse'*, change son adresse puis envoie le message 'I-M-HERE' à son coordinateur pour l'informer de sa nouvelle localisation (voire figure 15).

```
{
Si message_reçu == 'DHCP_REPLY' alors
      changer mon adresse_noeud = nouvelle_IP;
     Allouer_nouveau paquet ();
Nouveau_paquet->type = M2PC-I-M-HERE;
send (nouveau_paquet, 0);
}
```

Figure 15 - Le comportement du terminal mobile après obtention d'une nouvelle adresse

10 Extraction des paramètres affectant la performance de M-2PC

L'objectif principal de cette simulation est de mesurer la latence du protocole M-2PC dans les deux cas de mobilité (intra et inter domaine), et de comparer les résultats obtenus avec l'exécution du protocole 2PC traditionnel au dessus de Mobile-IP. Cette comparaison a permet d'évaluer les performances de l'approche de mobilité offerte par le protocole M-2PC et de répondre à la question suivante : est-il plus intéressant que la gestion de la mobilité soit prise en compte par le protocole de validation de couche application, ou suffit-il de compter sur Mobile-IP en exécutant le protocole de validation traditionnel 2PC ? En fait, la première option si elle offre de bonnes performances peut être mise en œuvre sur les réseaux IP actuels.

La latence d'un protocole de validation est définie par la durée du temps écoulée entre le moment où une application décide de terminer une transaction et le moment où cette terminaison devient effective pour chaque participant [AP 98]. Elle est mesurée par le temps consommé entre la soumission de la demande de validation par l'application et la réception de la décision par les participants [BLRS 04]. Dans M-2PC, le coordinateur n'envoie le résultat de la validation au client mobile qu'après réception des acquittements de la décision de tous les participants. Le processus de validation M-2PC se termine alors à la réception de l'acquittement du client mobile. Ainsi on définit la latence du protocole M-2PC par la durée du temps écoulé entre le moment où l'application décide de terminer la transaction (*la sauvegarde de l'état local*) et le moment où le client mobile acquitte la réception du message résultat (voire figure 1).

La latence du protocole M-2PC ne dépend pas seulement des deux types du hand off effectués (hand off L2 et hand off L3), mais dépend aussi d'autres paramètres. Afin de bien

définir l'ensemble des paramètres qui affectent la latence de M-2PC une formule a été définie pour le calcul de cette dernière. Le processus de la validation atomique M-2PC peut être divisé en deux sous procédures séquentielles. La première s'exécute sur la partie fixe du réseau et commence au moment où l'application transactionnelle mobile décide de terminer la transaction, et la seconde est constituée des opérations envoi et réception des messages *decision* et *ack* respectivement. Ainsi on peut écrire la latence de M-2PC (**Lat**) par la formule suivante.

$$\mathbf{Lat} = \mathbf{T_f} + \mathbf{T_h} + \mathbf{T_{da}} \quad (1)$$

Avec:

$\mathbf{T_f}$: la duré d'exécution de l'ACP sur la partie fixe. T_f est le temps nécessaire pour la réception du dernier ack en commençant par le moment où l'application décide de terminer la transaction. T_f n'est autre que la somme des délais nécessaires aux opérations disques (écritures forcées) et aux transmissions des messages (*commit request, prepare, vote, decison, ack*). Voire figure 1.

Le temps nécessaire pour effectuer une opération disque diffère d'un terminal à un autre, selon la capacité de calcule du terminal. Les messages *'prepare', 'vote, 'Decision' et 'ack'* sont des messages bien définis de taille fixe dont le délai de transmission dépend seulement de la topologie du réseau et de la capacité de la bande passante. Par contre le message *'Commit-Request'* qui contient le fichier log de la transaction est de taille variable et dépend du nombre d'items accédés par la transaction.

$\mathbf{T_h}$: la duré du temps que peut attendre le coordinateur pour envoyer la décision à la position courante du terminal mobile, elle peut être nulle dans deux cas : (1) le hand off effectué est un hand off L2 ; (2) un hand off L3, y compris le délais de transmission du message *'I-M-HERE'*, qui se termine avant que T_f se termine.

$\mathbf{T_{da}}$: délai de transmission du message '*decision'* au client mobile plus le délai nécessaire à la réception de son acquittement.

Supposons qu'un hôte mobile HM se trouvant dans une cellule **C** de rayon (de porté) **R**, à la position **X** et se déplaçant par une vitesse uniforme **V**, effectue un hand off de niveau L2 ou de niveau L3 à un instant **T** du déroulement de l'ACP sur la partie fixe. On donne ci dessous une autre manière pour le calcul de La latence. Pour plus de simplicité nous supposons que le site mobile se déplace horizontalement selon l'axe des abscisses (Y = 0 et Z = 0).

$$\mathbf{Lat} = \mathbf{T} + \mathbf{T_{hl3}} + \mathbf{T_c} + \mathbf{T_{da}} \quad (2)$$

T est le temps écoulé avant le déclenchement du hand off, et c'est la durée nécessaire pour arriver aux frontières de la cellule. NS considère des nœuds mobiles qui se déplacent avec une vitesse uniforme. La valeur de T peut être obtenue par la formule suivante.

$$\mathbf{T = (R - X) / V} \quad (3)$$

T_{hl3} : le temps nécessaire pour effectuer un hand off L3 et qui n'est autre que la somme des temps nécessaires pour effectuer un hand off L2 (link layer hand off) et du temps nécessaire pour obtenir une nouvelle adresse IP.

$$\mathbf{T_{hl3} = T_{hl2} + T_{dhcp}} \quad (4)$$

T_c : délai de transmission du message contact au coordinateur.
En remplaçant (3) et (4) dans (2) on obtient

$$\mathbf{Lat = [(R - X) / V] + T_{hl2} + T_{dhcp} + T_c + T_{da}} \quad (5)$$

T_{dhcp} = 0 dans le cas du hand off niveau 2. En se basant sur les formules (1) et (5), on peut extraire l'ensemble des paramètres qui affectent la latence du protocole M-2PC dans les deux types de mobilité (voire le tableau ci-dessous).

Paramètre	Micro mobilité	Macro mobilité	Description
La taille du fichier log	/	/	Ce paramètre varie selon le nombre d'items accédés par la transaction
R	/	/	Le rayon ou la porté de la cellule
X	/	/	La position du terminal mobile lors Du déclenchement de l'ACP
V	/	/	La vitesse du mouvement du terminal mobile
T_{DHCP}		/	Le temps d'obtention d'adresse IP
T_{hl2}	/	/	La latence du hand off L2, dépend de la norme sans fil utilisée.

Tableau 1 - Les paramètres affectant la latence du M-2PC

11 Simulations et résultats

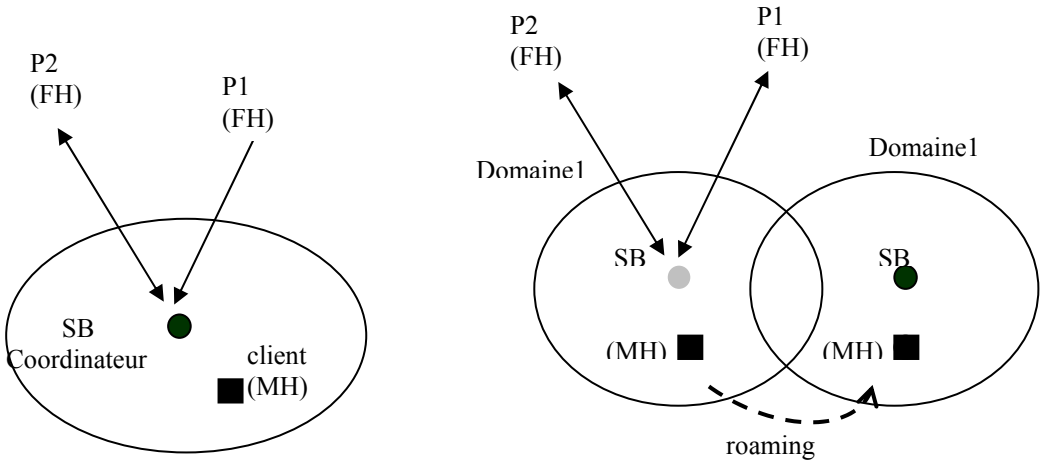
La configuration des paramètres cités précédemment est réalisée sous forme de scénarios de mobilité NS (section 3.1). Un scénario de simulation comporte un scénario de mouvement des unités mobiles. Une unité mobile peut se déplacer à l'intérieur d'une cellule, ou traverser les frontières d'une cellule du même domaine ou d'un domaine différent. Le mouvement des unités mobiles est réalisé grâce à l'outil **Setdest** (disponible dans NS).

3.1 Scénarios de simulations

Etant donné que le protocole M-2PC est conçu pour les applications transactionnelles mobiles opérants dans des réseaux sans fil et mobiles avec infrastructure, nous considérons le scénario de mobilité décrit par :

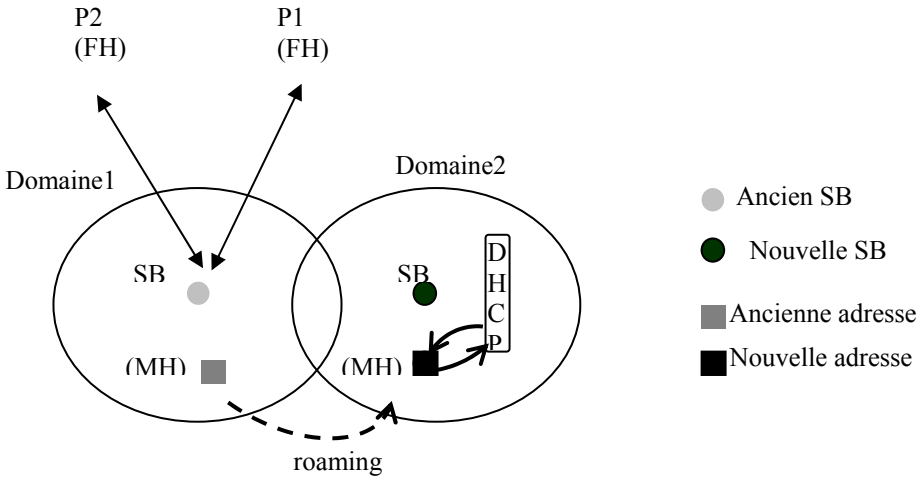
1. Une architecture spécifique de réseau sans fil : le nombre de nœuds mobiles, le nombre de nœuds fixes, le nombre de nœuds de type station de base, l'interconnexion entre les différents nœuds, les dimensions des cellules couvertes par les stations de bases, etc.
2. Les déplacements dans une cellule ou entre différentes cellules de même domaine ou de différents domaines

Les topologies des réseaux sans fil utilisés durant nos simulations sont décrites dans la figure 16. Afin de pouvoir étudier les performances de M-2PC dans le cas de la mobilité intra domaine nous considérons la topologie d'un réseau mobile avec infrastructure comprenant deux stations de base appartenant à un même domaine (figure 16(b)). La figure 16(c) désigne l'architecture d'un réseau mobile avec infrastructure comprenant deux stations de base appartenant à des domaines différents (mobilité inter domaine). Dans les deux scénarios le client mobile quitte sa cellule mère et traverse la nouvelle cellule pendant le déroulement du processus de validation. Nous considérons aussi le cas le plus simple où le client mobile se déplace sans changer de cellule (la figure 16 (a)).



(a) Sans hand off

(b) Mobilité intra domaine



(c) Mobilité inter-domaine

Figure 16 – Topologies des réseaux sans fil à simuler

La technologie de transmission du réseau sans fil utilisé est la IEEE 802.11. Le standard IEEE 802.11 est destiné à fournir une connectivité sans fil à des stations fixes ou mobiles. IEEE 802.11 a été publié en tant que premier standard international pour les réseaux locaux sans fil et WLAN (Wireless Local Area Network) [Puj 01]. La norme IEEE 802.11 (désignée aussi par WIFI) couvre les deux premières couches du modèle OSI. Elle définit une couche MAC Commune à toutes les couches physiques.

Dans cette section nous allons étudier l'effet de chacun des paramètres cités dans le tableau 1 sur la latence du protocole M-2PC dans les deux cas de mobilités. Nous considérons seulement l'architecture avec des participants fixes, nous fixons en premier lieu le nombre de

participants à 2 afin d'étudier les effets des autres paramètres. Puis nous étudierons l'effet du nombre de participants à la transaction sur la latence du M-2PC. Pour ce faire nous devons fixer quelques paramètres nécessaires au bon fonctionnement du M-2PC et à l'obtention des résultats proches de la réalité.

Taille du journal (fichier log)

Une base de données mobile occupe généralement un Méga d'espace mémoire [Vid 01]. Dans une unité mobile une donnée (un objet ou un item) est supposée occuper une page mémoire. Une page mémoire est sur un 1Kb [DCKM+ 94]. Une transaction mobile peut modifier entre 2 et 16 items [KDDS 00]. Donc la taille d'un fichier log varie entre 2 et 16 KO.

Temps des opérations disques (écritures forcées).

Le Temps d'une opération disque d'un nœud mobile est entre 10 ms et 40 ms [CBML 03] et celui d'un nœud fixe est entre 8 ms et 16 ms [NSB 97].

La latence du hand de niveau 2

La durée du hand off L2 varie entre une dizaine et environs 200ms [KKLY+ 04] selon la technologie sans fil adoptée. La latence du hand off dans les réseaux WIFI est dans les environs de 158 ms [MN 02].

Temps de pause

Durant son mouvement, un noeud mobile peut passer par des périodes de pauses (immobilité). L'objectif de nos simulations est de déclencher des hand off pendant le déroulement du processus de validation sur la partie fixe. La durée d'exécution de ce processus est suffisamment courte ce qui rend inutile la considération des périodes de pause.

Temps d'obtention d'une nouvelle adresse

Le temps d'obtention d'une adresse IP à partir d'un serveur DHCP est soit pris aléatoirement entre 800 ms et 1670 ms [Kum 03][KKLY+ 04] soit fixé à 1670 selon nos besoins des simulations (plus de détail dans les sections qui suivent).

Temps de reconfiguration

C'est le temps que peut prendre un site mobile pour reconfigurer son interface et quelques paramètres réseaux pour établir une nouvelle connexion [KKLY+ 04]. Après avoir analysé l'implémentation NS du protocole Mobile-IP, nous constations que ce dernier ne considère pas la période de reconfiguration. Par conséquent et pour les besoins de comparaison entre les deux implémentations nous devons éliminer cette période.

paramètre	**valeur**
Capacité des liens filaires	5Mb ou 10 Mb
Type de liens filaires	Duplex
Topographie de la zone terrestre simulée	670 x 670 m^2
Vitesse de déplacement	10 – 140 m / s

Tableau 2 – les paramètres des fonctions du réseau de communication sans fil

3.2 Effet de la taille du ficher log

Afin d'étudier l'effet de ce paramètre (taille du fichier log) sur la latence du protocole M-2PC, nous avons fixé les paramètres vitesse (V) à 50 m/s, position lors de l'initiation du processus de la validation (X) à 248 m et le rayon de la cellule (R) à 250 m. Nous faisons varier le paramètre taille du fichier log entre 2KO et 16 KO. Le résultat obtenu est décrit dans la figure 17.

On note en premier lieu la différence importante entre les latences de M-2PC dans les deux cas de mobilité. Bien que la duré d'exécution du processus de validation sur la partie fixe augmente en fonction de la taille du ficher log (figure 17), la latence du protocole M-2PC dans le cas de mobilité inter domaine reste stable dans tout les cas. Ceci sous entend que la taille du fichier log n'influe pas sur la latence du M-2PC malgré qu'elle influe sur le temps d'arrivé du message *commit requeste* (figure 17). On peut expliquer ce phénomène par le fait que la latence de l'ACP fixe est beaucoup plus petite comparée au temps nécessaire pour obtenir une nouvelle adresse IP, même lorsque la taille du fichier log atteint son max (255 ms pour 16 KO). Et donc si un hand off L3 s'est déclenché durant cette petite période, le coordinateur ne peut envoyer le résultat de la décision au client mobile qu'après réception du message 'I-M-HERE'. Ce dernier ne peut être envoyé par le mobile qu'après la terminaison du hand off L2 et l'obtention d'une nouvelle adresse. La durée de l'exécution séquentielle de ces deux processus dépasse 1200 ms ce qui rend la latence maximale de l'ACP obtenue pour la taille maximale du fichier log, négligeable par rapport au temps que peut passer le coordinateur dans l'attente de la réception du message 'I-M-HERE'.

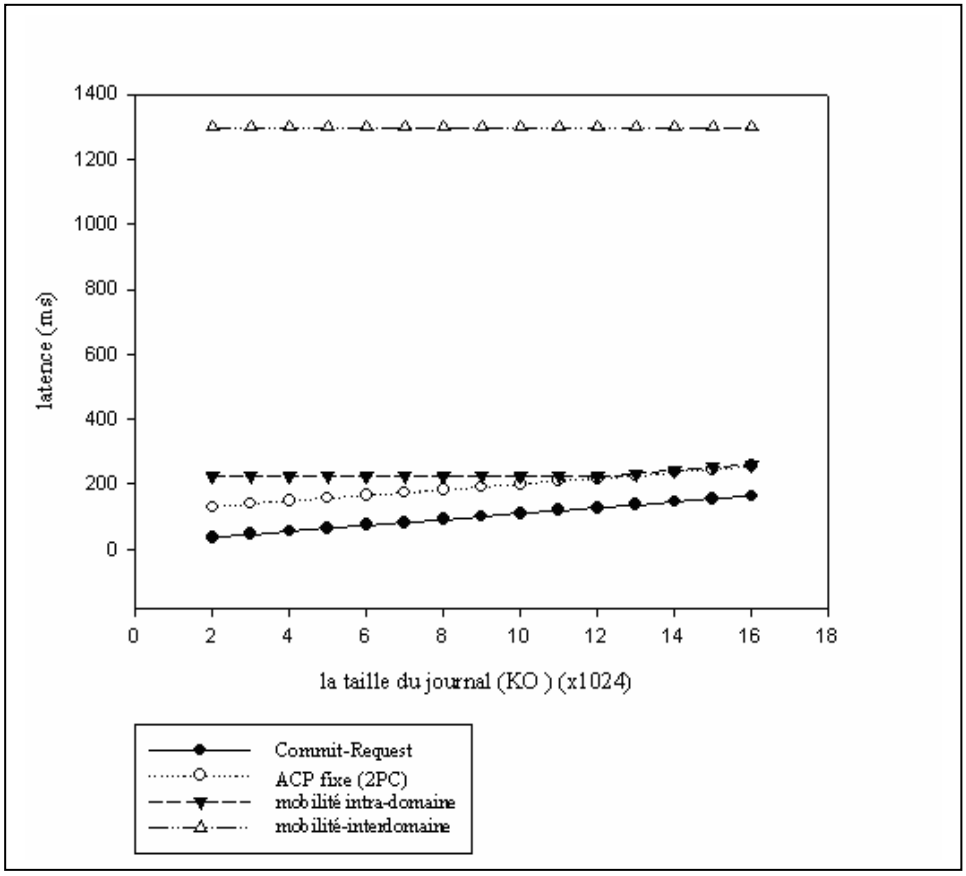


Figure 17 – Impact de la taille du log sur la latence de M-2PC

Dans le cas de la mobilité intra domaine, la latence de M-2PC reste stable pour les tailles du log allant de 2KO à 11KO. Elle s'accroît légèrement à partir de 12KO. Les valeurs 2KO à 11KO permettent d'augmenter la latence de L'ACP fixe mais ne permettent pas à ce dernier de se terminer après la terminaison du hand off. Par conséquent, la latence du processus de validation globale (M-2PC) dépend de la latence du hand off L2. À partir de 12KO le délai d'exécution du processus de validation sur la partie fixe devient suffisamment grand au point où il se termine après le hand off. Dans ce cas la latence globale n'est autre que la latence de l'ACP fixe (cette dernière augmente en fonction de la taille du fichier log) plus le temps d'envoi et de réception des messages *decision et ack* respectivement (équation (1)).

3.3 Effet des paramètres de mouvement (X, V, R)

Les paramètres de mouvement modélisent les situations de mobilité qu'une application mobile doit gérer. Selon le mode d'utilisation d'une application mobile (lors d'un voyage, lors d'une présentation, …), le mouvement de l'utilisateur peut être décrit.

L'information sur le mouvement d'un nœud mobile qui exprime la mobilité du nœud comprend la vitesse, la direction, la localisation, le temps de pause et la portée de la

transmission. La porté de la transmission n'exprime pas le mouvement du nœud mobile mais affecte la mobilité du nœud d'un point de vu connectivité [TMUY 03].

11.3.1 Effet de la vitesse (le paramètre V)

Afin d'étudier l'effet de ce paramètre nous avons fixé la taille du log à 9 KO (la moyenne), le rayon de la cellule à 250 m et la position du mobile lors de l'initiation du processus de validation à 248 m. le résultat obtenu est décrit dans la figure 18.

Les courbes décrivant la latence de M-2PC dans les deux cas de mobilité se comportent de manières identiques pour les valeurs de la vitesse allant de 10m/s à 140m/s. Lorsque le mobile se déplace avec une vitesse de 10 m/s, la latence de M-2PC atteint son minimum dans les deux cas de mobilité, cette valeur est obtenue à cause de l'absence du hand off durant le processus de validation. Cette absence est induite par le déplacement assez lent du nœud mobile. En effet un nœud qui se déplace lentement possède moins de chances de traverser les frontières des cellules voisines ce qui augmente la probabilité que le processus de validation se termine dans la même cellule. Lorsqu'on a augmenté la vitesse du mouvement à 20m/s la latence du protocole augmente significativement dans les deux cas de mobilité. Ce qui explique la présence du hand off avec ces deux types. Dans le cas de la mobilité inter domaine, la latence se décroît légèrement pour les vitesses allant de 30 à 140 m/s. par contre dans le cas de mobilité intra domaine, la latence se décroît légèrement a chaque fois qu'on augmente la vitesse, puis elle se stabilise. En effet plus le mobile est rapide plus le hand off se déclenche tôt, augmentant ainsi le parallélisme entre le processus de validation et celui du hand off. C'est-à-dire qu'une grande portion du hand off s'exécutera en parallèle à l'ACP sur la partie fixe ce qui permet de diminuer la latence de ce protocole. On remarque aussi que, dans le cas de la mobilité intra domaine la latence atteint son minimum même lorsque le mobile se déplace assez rapidement (90 m/s à 140 m/s). Contrairement au cas précèdent, la latence minimum est atteinte cette fois-ci par la présence du hand off et non pas par son absence. Le mouvement assez rapide permet le déclenchement assez tôt du hand off et par conséquent sa terminaison avant l'achèvement du processus de validation. En conclusion, la présence d'un hand off, n'augmente pas forcément la latence du protocole contrairement à ce que l'on attendait. En fait cela dépend à quel moment il se produit.

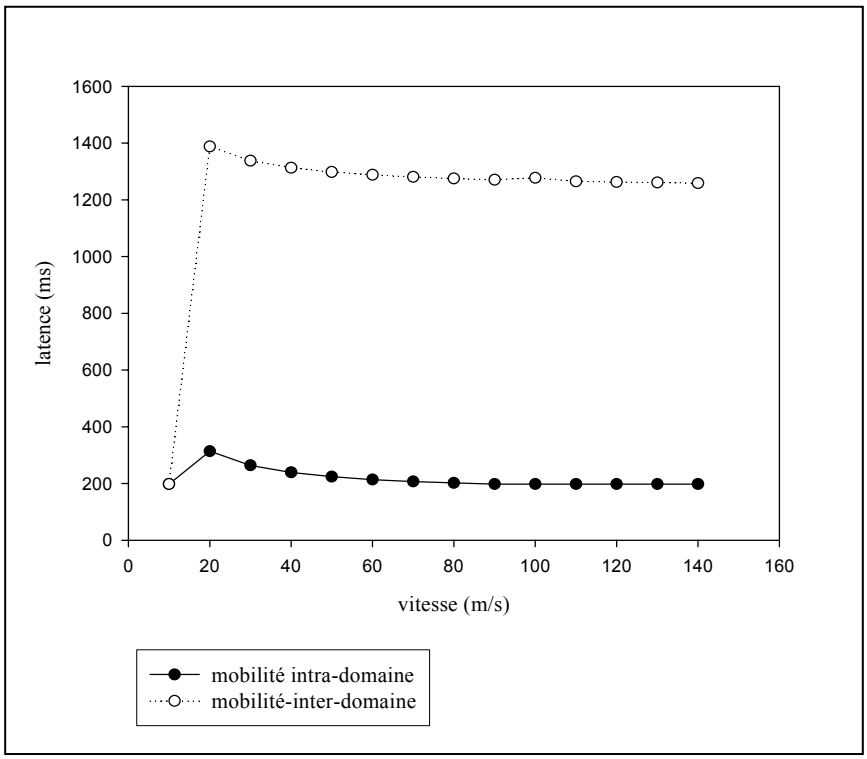


Figure 18 – Effet de la vitesse sur la latence du protocoleM-2PC

11.3.2 Effet de la position du client mobile lors de l'initiation de l'ACP. (Le paramètre X)

Dans ce scénario nous avons choisi une cellule de rayon 250m, nous nous fixons la vitesse de déplacement du mobile à 50 m/s et nous faisons varier sa position initiale à chaque fois qu'on veut lancer le processus de validation. Les résultats obtenus sont présentés dans la figure 19.

On remarque une stabilité de la latence pour les valeurs allant de 5m à 240m pour les deux types de mobilité. Bien que le déplacement avec une vitesse de 50 m/s a provoqué un hand off dans l'expérience précédente, le déplacement avec une telle vitesse, en commençant à une position initiale appartenant à l'intervalle [5m, 240m], ne permet pas au mobile de déclencher un hand off. Ceci est dû au fait que le mobile est loin des frontières de la cellule lors de l'initiation du processus de validation. Ce qui annule la probabilité d'effectuer un hand off, et par conséquent, augmente les chances du mobile de recevoir le résultat de la validation dans sa cellule mère. Un nœud qui se trouve loin de sa station de base (et donc très proche des frontières de la cellule) lors de l'initiation du processus de validation est un nœud qui est sur le point d'effectuer un hand off. Dans le cas de mobilité inter domaine, plus le nœud est proche des frontières de la cellule lors de l'initiation de l'ACP, plus une bonne partie du hand

off s'exécute en parallèle à ce dernier permettant ainsi de diminuer la latence du M-2PC. Dans le cas de la mobilité intra domaine, un nœud mobile qui est sur le point de quitter sa cellule lors de l'initiation du processus de validation a plus de chance de terminer son hand off avant que l'ACP se termine. Ce qui permet d'éviter le stockage de la décision au niveau de l'ancienne station de base. Le résultat de la validation est directement envoyé à la nouvelle localisation du mobile (nouvelle cellule). Ceci permet de diminuer la latence.

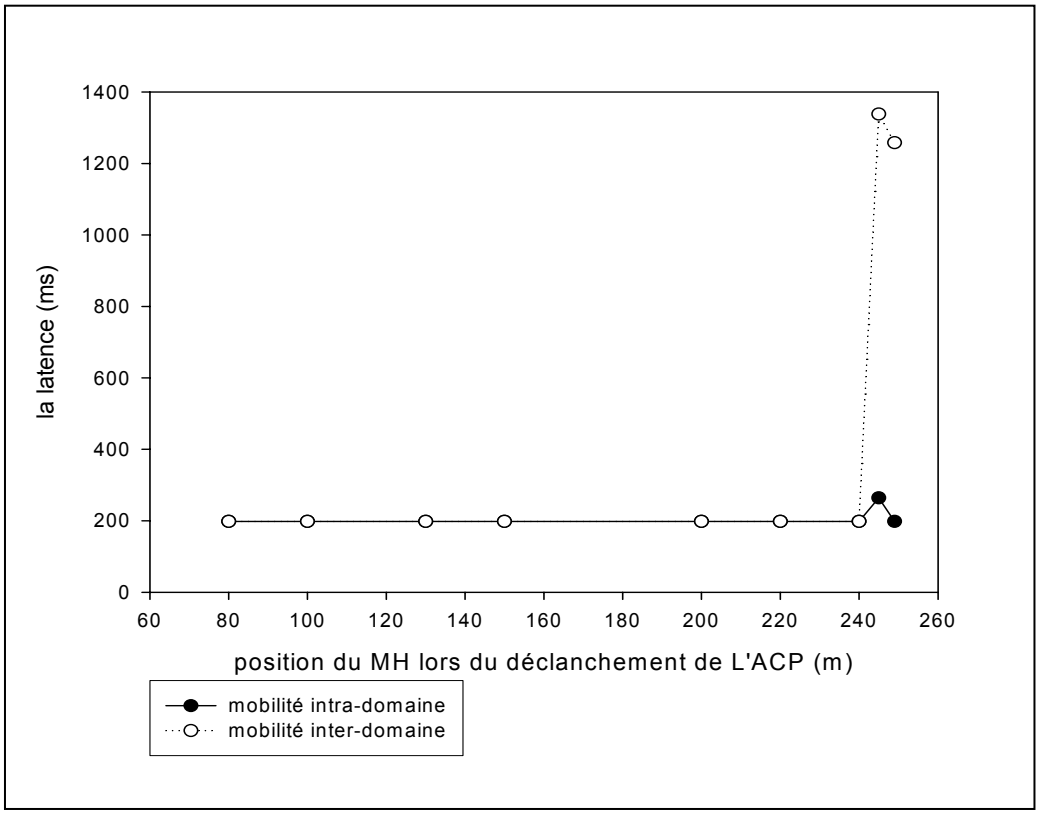


Figure 19 – Effet du paramètre X sur la latence de M-2PC

11.3.3 Effet de la portée de la station de base (le paramètre R)

En générale plus la cellule est de taille petite, plus la probabilité d'effectuer un hand off augmente, et inversement plus la taille de la cellule est grande, plus la probabilité d'effectuer un hand off diminue. On a vu dans les testes précédents que la latence de M-2PC augmente si un hand off se déclenche pendant le déroulement du processus de validation et se termine après la terminaison de ce dernier. Pour voire clairement l'effet de la porté de la station de base (le rayon ou la taille de la cellule) on considère un nœud mobile qui se déplace dans une cellule de porté qui varie entre 50m et 250m. La figure 20 décrit le résultat obtenu.

La latence atteint ces valeurs maxima lorsque le mobile se déplace dans une cellule de taille 50m pour les deux types de mobilité. Ces valeurs diffèrent selon la vitesse du mouvement. Ce paramètre (la vitesse) n'a aucun effet sur la latence du M-2PC lorsque la portée de la cellule dépasse 80m. À partir de cette valeur la latence devient stable à cause de l'absence du hand off. En effet, les vitesses 50m/s, 80m/s et 120m/s ne permettent pas au mobile de traverser les frontière d'une cellule de porté supérieure ou égale à 80m (cette valeur dépend bien sûr de la position du mobile lors de l'initiation du processus de validation)

Les paramètres de mouvements sont étroitement liés, et dépendent l'un de l'autre. La relation : T = [(R – X) / V] définit le temps de déclenchement du hand off relativement au temps d'initiation de la validation. Il existe un certain parallélisme entres ces deux processus. Plus T est petit plus le parallélisme entre ces deux processus augmente permettant la diminution de la latence. Et inversement, plus T est grand plus le parallélisme diminue augmentant ainsi la latence du protocole (voire figure 21).

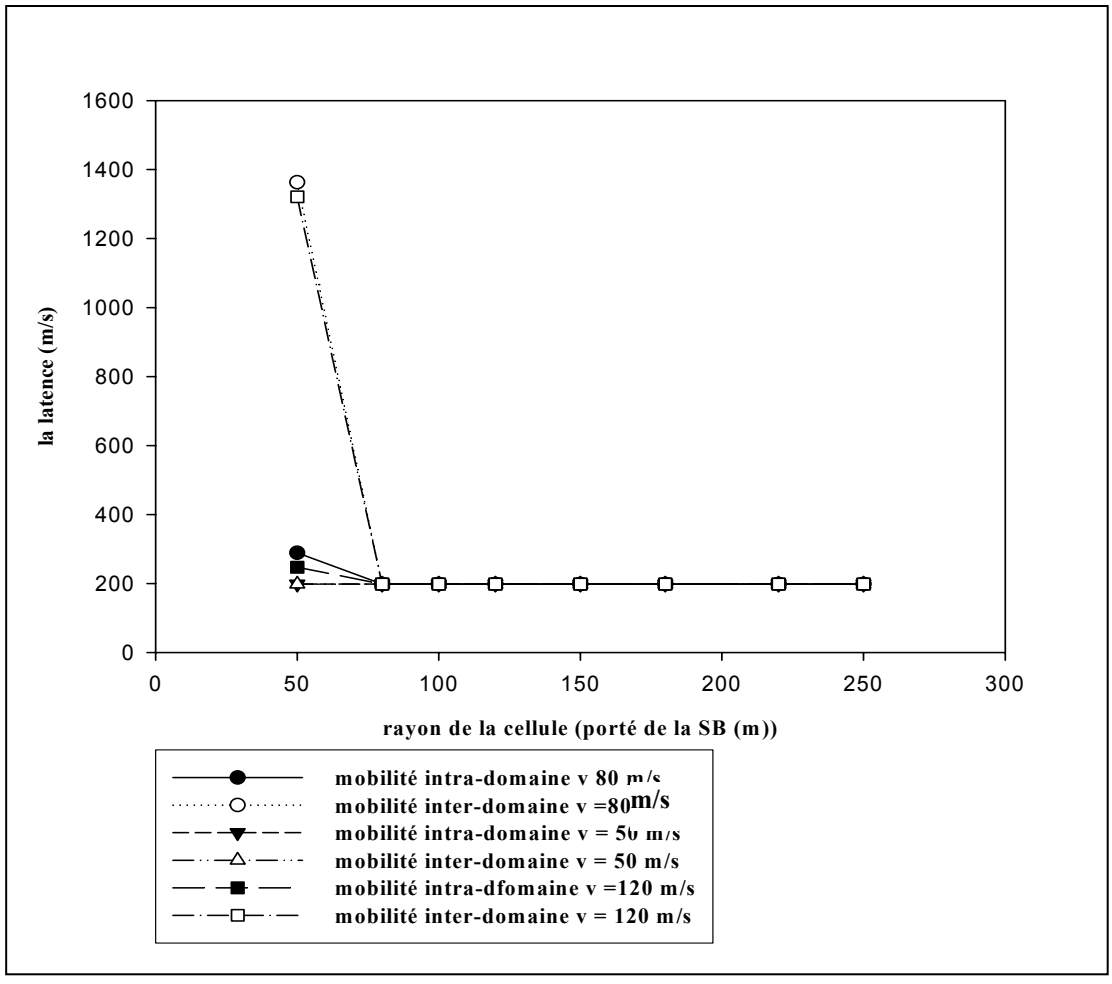


Figure 20 – Effet de la taille de cellule sur la latence de M-2PC

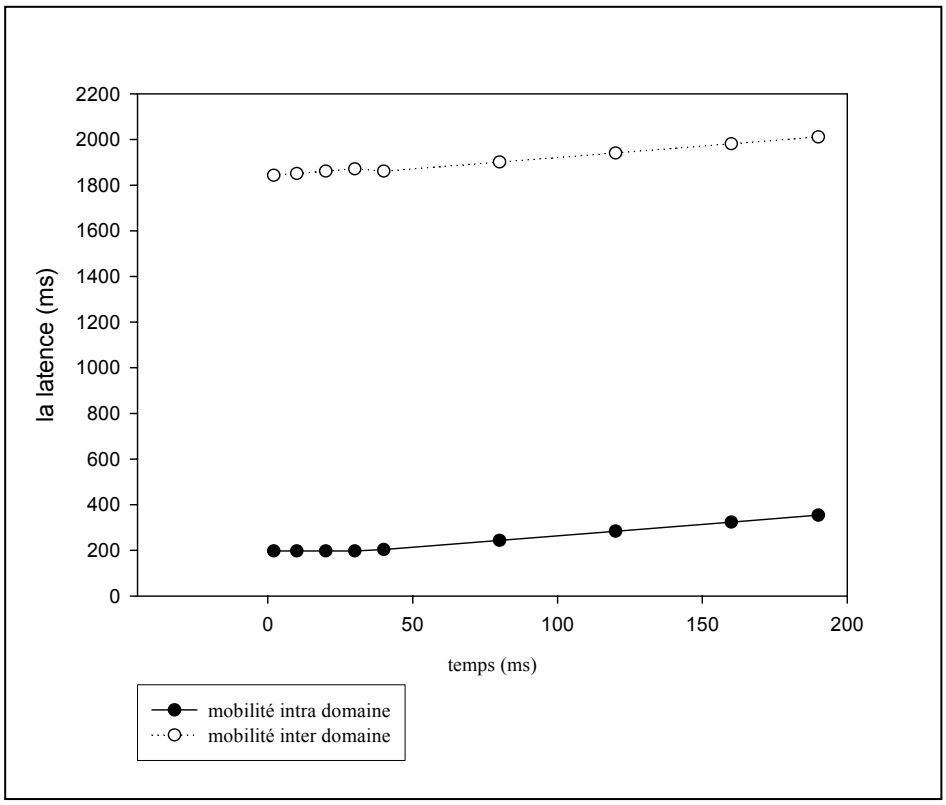


Figure 21 – Temps de déclenchement du hand off par rapport au temps d'initiation de l'ACP

11.3.4 Effet du temps d'obtention d'une nouvelle adresse IP

Dans ce scénario nous avons fixé les paramètres de mouvement X, V, R à 248m, 50m/s, 250m respectivement. Pour la taille du journal nous avons pris la moyenne (9KO). Nous faisons varier le temps nécessaire pour obtenir une nouvelle adresse entre 800 et 1670ms. Le choix de cet intervalle a été déjà commenté dans la section 3.2.5. Le résultat obtenu est présenté dans la figure 22.

Ce résultat montre que la latence du protocole M-2PC est proportionnelle au temps nécessaire à l'obtention d'une nouvelle adresse IP. Ce qui été prévisible. En effet ce temps représente la cause principale de dégradation de la qualité de service des applications temps réel [KKLY+ 04]. Afin d'améliorer la qualité de service des applications temps réel, [Dom+ 01] [KKLY+ 04] [CNZ 05] proposent d'anticiper le processus d'obtention d'une nouvelle adresse IP. L'idée principale est de déclencher la procédure obtention d'une nouvelle adresse, d'une manière proactive, en parallèle de la procédure de hand off L2. Cette technique permet une réduction considérable de la latence de hand off de niveau 3 et par conséquent une amélioration significative de la qualité de service.

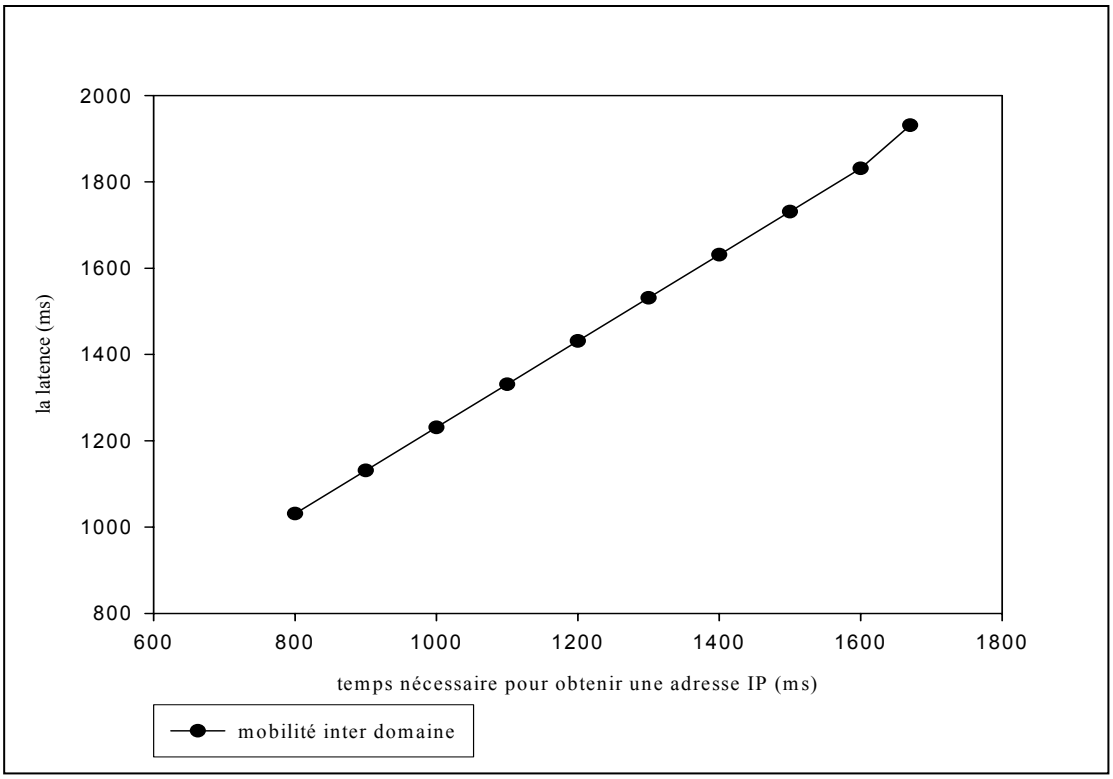


Figure 22 – L'effet du délai d'obtention d'une nouvelle adresse

11.3.5 Effet du nombre de participants fixes.

Dans ce scénario, nous gardons les mêmes valeurs que précédemment pour les paramètres X, V, R et la Taille du journal. Nous augmentons au fur et à mesure le nombre de participants fixes à la transaction en commençant par deux. A chaque fois on ajoute un participant fixe dont le temps des opérations disque est pris aléatoirement entre 8ms et 10ms, La capacité de son lien filaire est soit 5Mb/s soit 10Mb/s, le délai de transmission des messages sur les liens filaires varie entre 5ms et 10ms. Comme le montre la figure 23, les deux courbes sembles être stables pour toutes les valeurs données aux nombres de participants fixes. En vérité le changement est vraiment léger au point où il n'est pas observable. La latence dans le cas de mobilité intra domaine varie entre 224ms et 233ms. Et dans le cas de la mobilité inter domaine elle prend ces valeurs entre 1901ms et 1910ms. Ce changement léger est dû au fait que la décision de valider ou d'annuler une transaction est le résultat de la coopération des participants fixes. Ce processus coopératif est un processus parallèle qui ne dépend pas du nombre du participants mais dépend essentiellement du participant le plus faible en terme de capacité de calcule, capacité de bande passante, et la distance qui le sépare du coordinateur. Ces paramètres influent légèrement sur le processus de validation sur la

partie fixe ; cette influence légère se répercute par la suite sur le processus de validation globale.

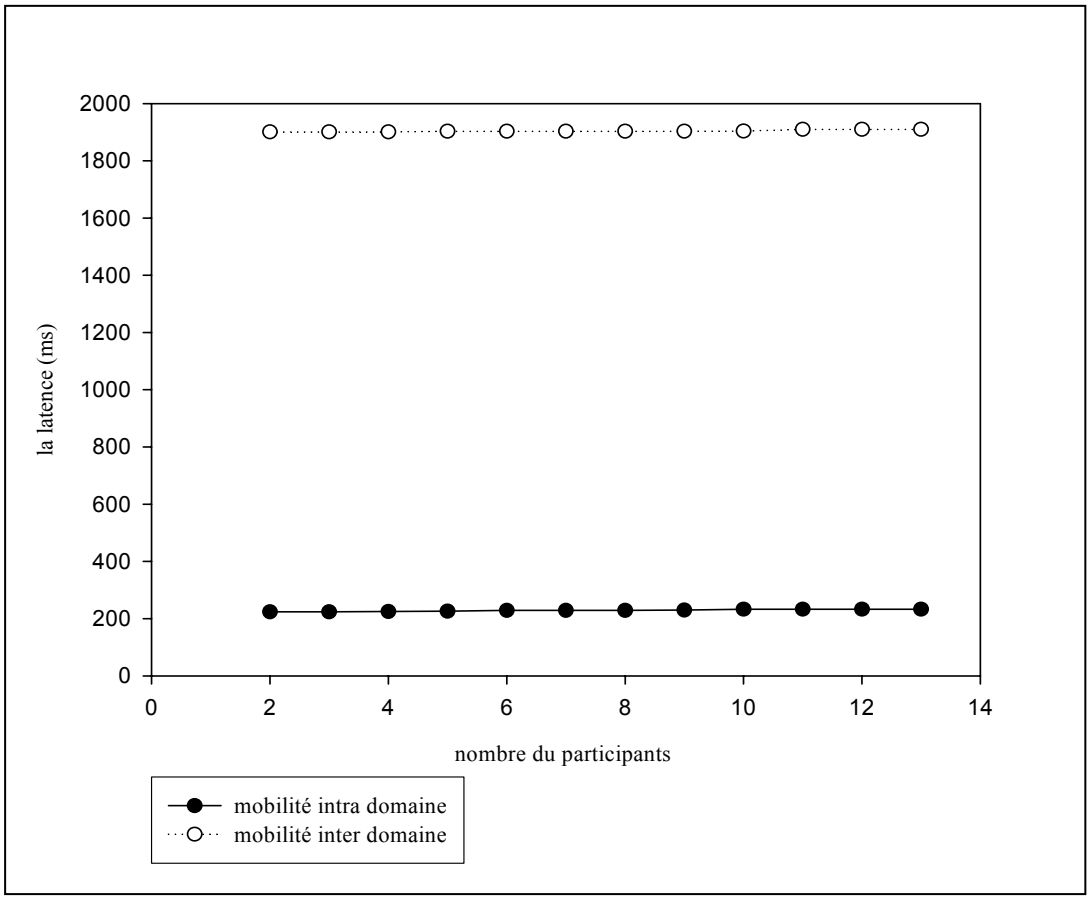


Figure 23 – L'impact du nombre de participants fixes sur la latence du protocole M-2PC

11.3.6 Effet du nombre de participants mobiles.

Afin d'étudier l'impact du nombre de participants mobiles sur la latence du protocole M-2PC, deux configurations ont été testées. Dans la première, les participants mobiles se déplacent dans leurs cellules sans effectuer un hand off (figure 16 (a)). Dans la deuxième configuration les participants mobiles peuvent effectuer un hand off de niveau 2 ou de niveau 3 (figures 16 ((b) et (c))). Au début nous commençons par un système composé seulement de deux participants fixes. Nous ajoutons à chaque fois un participant mobile à l'ensemble des participants du système. La probabilité pour que le participant mobile ajouté effectue un hand off est P_i définit comme suit :

$$\begin{cases} P_i = 1 & \text{le participant mobile ajouté effectue un hand off} \\ P_i = 0 & \text{Sinon} \end{cases}$$

Cette probabilité suit une loi de Bernoulli[15]. La probabilité pour que le hand off effectué soit un hand off de niveau 2 est $P_{iL2.}$ La probabilité pour que le hand off effectué soit de niveau 3 est $P_{iL3} = 1- P_{iL2}$. Le tableau ci-dessous indique les probabilités d'effectuer ou non un hand off. Les résultats obtenus à partir des deux configurations sont présentés dans la figure 24.

participant	1	2	3	4	5	6	7	8	9	10
P_i	0	0	1	1	1	0	1	1	0	1
P_{iL2}	-	-	0	1	0	-	1	0	-	0
P_{iL3}	-	-	1	0	1	-	0	1	-	1

Tableau 3 – Les probabilités du hand off

Dans la première configuration, on observe une augmentation plus ou moins significative de la latence du M-2PC dés qu'on ajoute un participant mobile au système. Dans le cas où aucun participant mobile n'est présent dans le système, la latence est minime (224ms). L'ajout d'un participant mobile au système fait augmenter la latence du protocole à 320ms, ce qui n'est pas le cas pour les participants fixes. Cette augmentation est due au fait que le participant mobile doit envoyer conjointement à son vote le journal des opérations exécutées à son niveau. Comme nous l'avons déjà montré dans la section 3.4.2, cette opération fait retarder le processus de la validation ce qui influe par la suite sur la latence du protocole.
Pour le nombre de participants mobiles allant de 2 à 10, la latence ne se comporte pas de façon régulière, elle augmente et diminue légèrement entre les valeurs 320ms et 425ms. Bien que la taille du journal envoyé par les participants mobiles à leurs délégués soit la même, la localisation et la capacité de calcul des nœuds mobiles diffèrent d'une configuration à une autre.

Dans la deuxième configuration, on observe une augmentation importante de la latence par l'ajout du troisième participant. La cause principale est le hand off de niveau 3 effectué par ce participant. La latence ensuite s'accroît légèrement en passant par des périodes de stabilité même si plusieurs participants exécutent un hand off (voir le tableau des probabilités). On constate que la latence du M-2PC n'est pas proportionnelle au nombre de participants exécutant un hand off. En d'autres termes, le déclenchement de plusieurs hand off n'a pas une grande influence sur la latence du protocole même si les hand off exécutés sont de niveau 3. On peut expliquer ce phénomène, par le fait que M-2PC s'adresse à la mobilité des participants pendant le déroulement du processus de validation, plus précisément entre le moment où le participant mobile délègue sa station de base et le moment où cette dernière

[15] Si X est une variable aléatoire qui vaut 1 s'il y a succès, 0 sinon, alors X suit une loi de Bernoulli..

décide de lui envoyer le résultat de la validation. Le déclenchement de plusieurs hand off pendant cette courte période, permet une exécution parallèle de ces derniers. Les hand off L2 se terminent bien avant que les hand off L3 se terminent. Le protocole M-2PC dépendra alors du dernier hand off L3 déclenché.

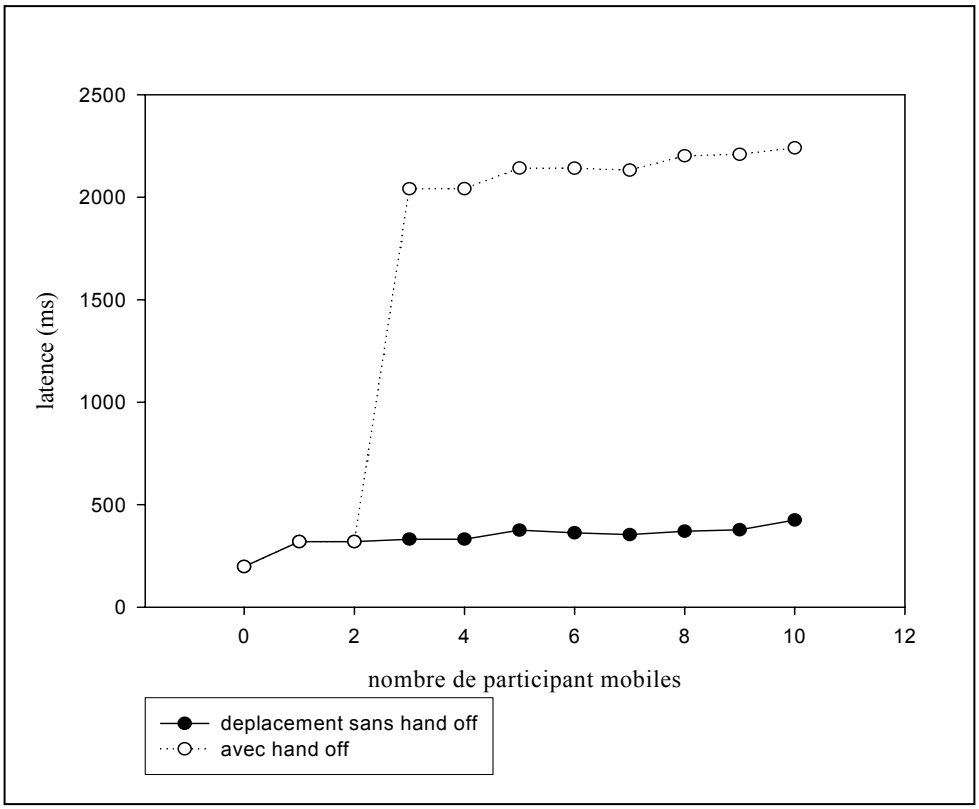


Figure 24 – l'impact du nombre de participants mobiles sur la latence du protocole

11.3.7 Effet de la charge du réseau sans fil.

Les résultats des expériences précédentes ont été obtenus en considérant que le client mobile est seul dans sa cellule. Même dans le cas où on a ajouté des participants mobiles, on les ajoutés dans des cellules différentes. Dans cette expérience nous étudierons le cas où le client mobile n'est pas le seul à émettre ou à recevoir des données dans cette cellule. Le but de cette expérience est d'étudier l'effet de la charge du réseau sans fil sur le protocole de la validation. Pour se faire nous définissons trois scénarios de mobilité, dans le premier on considère que le client mobile se déplace dans sa cellule sans traverser les frontières des cellules voisines (figure 16 (a)). Dans le second le client mobile se déplace en exécutant des hand off de niveau 2 (figure 16 (b)), et dans le troisième scénario le client mobile effectue des déplacements vers des cellules de domaines différents (figure 16 (c)). Nous augmentons au

fur et à mesure, dans chaque scénario, le nombre d'unités mobiles actives dans la cellule du client mobile (Commit-BS). La figure 25 décrit les résultats obtenus.

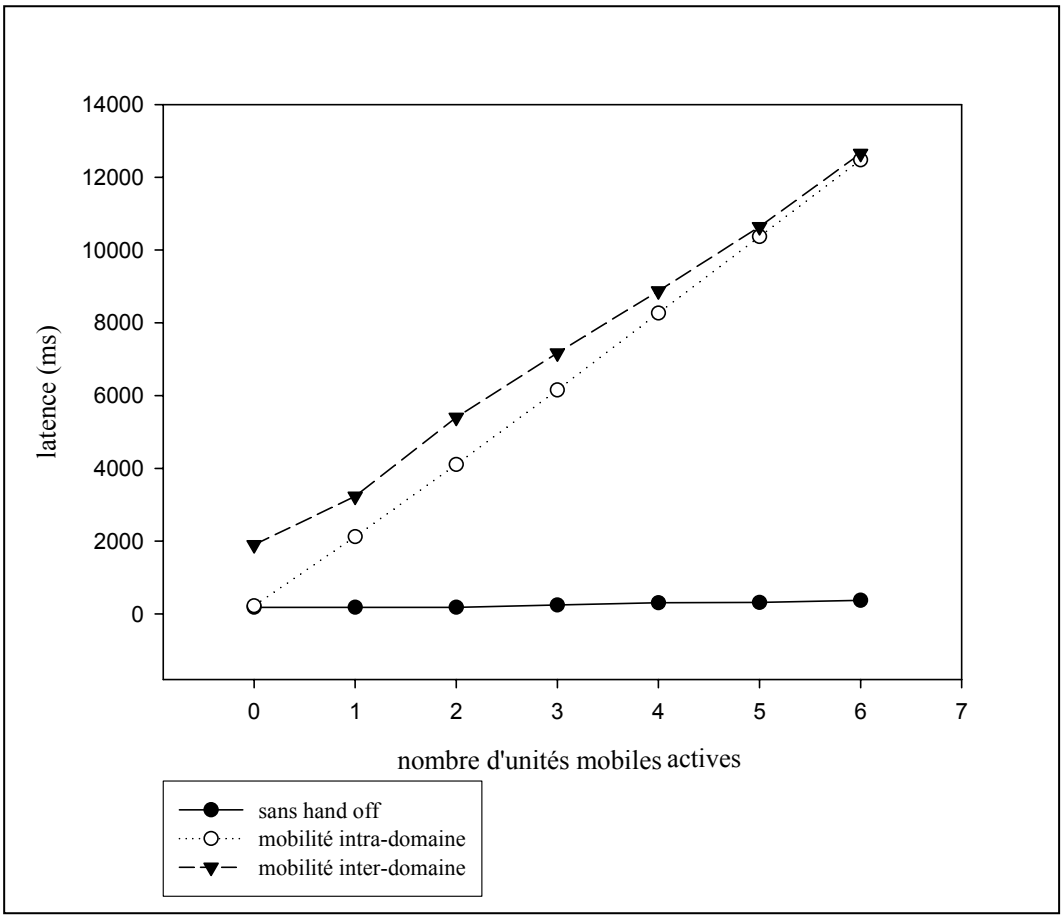


Figure 25 – Effet de la charge du réseau sans fil sur la latence du protocole M-2PC.

Dans le cas des déplacements sans hand off, la latence varie entre 183ms et 184ms lorsque le nombre d'unités mobiles actives dans la Commit-BS est inférieur ou égal à 2, puis elle augmente avec l'augmentation du nombre d'unités actives (elle atteint la valeur 374ms pour 6 unités actives). Nous constatons que même lorsque le client mobile ne se déplace pas vers des cellules voisines, la latence du protocole M-2PC peut atteindre des valeurs plus ou moins grandes. On peut expliquer ces résultats par le fait que l'augmentation du nombre d'unités mobiles actives dans le système, fait augmenter les risques de collision sur le support de communication, ce qui diminue par conséquent la performance de la technique de gestion des canaux de transmission du standard IEEE 802.11 [YV 02][Pha 05]. Le client mobile prend donc plus de temps pour accéder au canal de transmission.

Dans les deux configurations restantes (mobilité intra domaine et mobilité inter domaine), nous remarquons que la latence augmente en augmentant le nombre d'unités mobiles actives dans le système, de plus les valeurs atteintes sont énormes (entre 2124ms et 12479ms dans le cas de la mobilité intra domaine et entre 3231ms et 13649ms dans le cas de la mobilité inter domaine). La norme IEEE 802.11 est toujours en cause dans ce phénomène. Durant le hand off, le site mobile et la station de base échangent un ensemble de messages appelés messages de contrôle. La transmission des messages de contrôle s'effectue à travers le même et l'unique canal de la cellule couverte par cette station de base [MN 02] (commit-BS dan notre cas). L'envoi des trames de contrôle à travers cet unique canal et en présences de plusieurs unités mobiles actives, nécessite le partage de ce dernier par les deux types de trames à savoir les trames de données et celles de contrôle. Par conséquent le client mobile exécutant un hand off prend plus de temps pour se synchroniser avec la station de base cible. Ce qui influe considérablement sur la latence du protocole.

12 Comparaison avec 2PC

Dans cette expérience, nous essayons de comparer la performance du protocole M-2PC à celle du protocole 2PC. Le but de cette comparaison est de voir qu'elle est la meilleure approche pour la validation des transactions mobiles. Le protocole 2PC n'a pas été conçu initialement pour les environnements mobiles, et donc il ne prévoit pas la mobilité des sites participants à la transaction. Pour cela, 2PC doit être utilisé conjointement à une politique de gestion de mobilité. Le protocole de routage en environnement mobile (Mobile-IP [JP 98]) a été choisi afin de gérer les situations de hand off provoquées par la mobilité des participants au processus de validation d'une transaction mobile en utilisant le protocole 2PC.

Le protocole Mobile-IP permet à 2PC de s'exécuter indépendamment de la localisation des participants mobiles. Mobile-IP est basé sur un procédé à deux adresses. Le site mobile possède une adresse permanente (*homme address*) qui correspond à son réseau mère. Et une adresse temporaire (*care of address*) qui correspond au réseau visité. Lorsqu'un site mobile change de réseau d'attache, sa station de base mère (Home Agent) prend la responsabilité de rediriger les données lui dessinées vers le réseau visité en utilisant son adresse temporaire (figure 26). Cette opération nécessite l'enregistrement du mobile auprès de son agent mère.

Nous considérons dans cette expérience un réseau sans fil constitué de deux participants fixes, deux cellules appartenants à deux domaines différents, et un client mobile. Nous définissons à partir de cette topologie trois scénarios de mobilité. Dans le premier nous utilisons le protocole M-2PC. Dans le deuxième scénario nous utilisons le 2PC traditionnel en gardant le coordinateur au niveau du site mobile et dans le troisième scénario nous fixons le coordinateur en le mettant au niveau de la station de base. L'activation de Mobile-IP dans les

deux derniers scénarios est nécessaire. Nous faisons varier dans chaque scénario la vitesse du mouvement du client mobile.

Après avoir analysé le code source de l'implémentation du protocole Mobile-IP dans NS, nous constatons que ni le délai d'obtention d'une nouvelle adresse ni la latence du hand off de niveau 2 ne sont considérés dans cette implémentation. De ce fait, et pour des raisons de cohérence des résultats, nous n'avons pas considéré, dans cette expérience, ces deux délais même pour le protocole M-2PC. Les résultats obtenus sont représentés dans la figure 27.

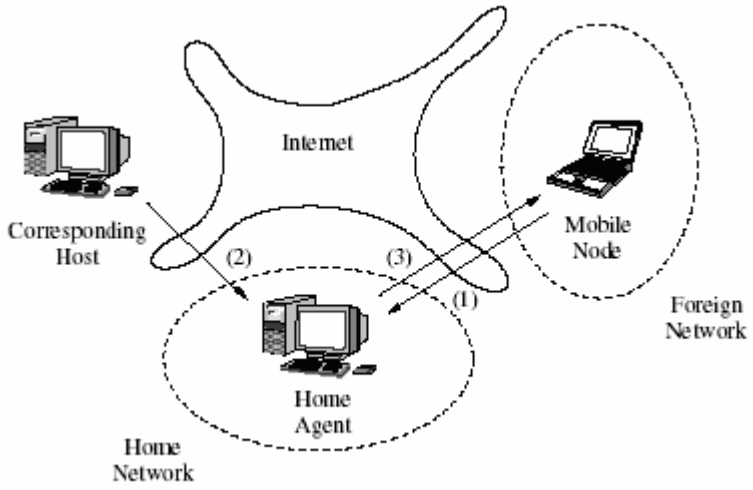
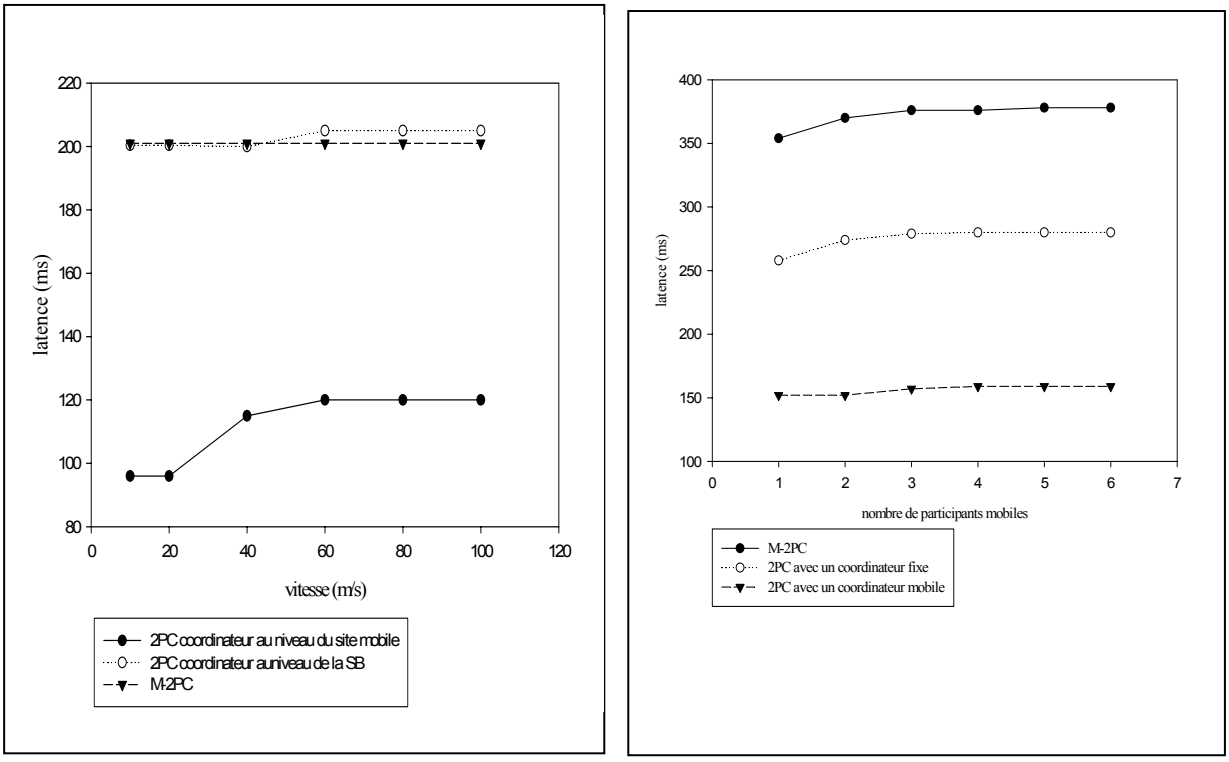


Figure 26 – Gestion de la mobilité avec Mobile-IP

Nous observons une différence significative entre la latence de 2PC traditionnel (coordinateur mobile) et celle de 2PC avec coordinateur fixe et M-2PC. Contrairement à nos prédiction, 2PC avec coordinateur mobile paraît être le plus performent en terme de latence que le 2PC avec coordinateur fixe et M-2PC.

Dans le cas de 2PC avec coordinateur fixe, le client mobile se trouvant dans le réseau visité reçoit la décision de la validation à travers un tunnel établi entre son agent mère (HA) et son agent visité (FA). Ce routage triangulaire de la décision est à l'origine de l'augmentation de la latence de 2PC avec coordinateur fixe. De même pour le cas de 2PC avec coordinateur mobile. Dans cette configuration nous distinguons deux cas possibles. (1) Le coordinateur mobile s'est déplacé après réception des messages votes. Ceci nécessite l'envoi des acquittements des participants à travers le tunnel ce qui explique l'augmentation de la latence pour v = 40m/s. (2) Le coordinateur se déplace assez rapidement ($V \geq 60m$) et quitte sa cellule avant la réception des votes des participants. Dans ce cas les messages votes et les acquittements des participants seront tous envoyés au coordinateur en utilisant le routage triangulaire. Ce qui permet encore d'augmenter la latence. Nous remarquons aussi que M-2PC est plus performant en terme de latence que le 2PC avec un coordinateur fixe. En effet le

coordinateur M-2PC envoie directement la décision au client mobile en se basant sur l'information de localisation indiquée dans le message 'I-M-HERE' tandis que la décision 2PC avec coordinateur fixe doit d'abord passer par le HA (Home Agent) puis se rediriger vers le FA (Foreign Agent) où se trouve actuellement le client mobile. Mais dans le cas où le nombre de participants mobiles augmente M-2PC présente moins de performances que le 2PC traditionnel avec coordinateur fixe (figure 27 (b)). En effet Le transfert des journaux des participants mobiles vers leurs stations de bases augmente la latence du protocole M-2PC contrairement à 2PC avec un coordinateur fixe qui nécessite que le transfert du journal du coordinateur.



(a) Effet de la vitesse

(b) Effet du nombre de participants mobiles

Figure 27 – Comparaison entre 2PC traditionnel et M-2PC.

Le tableau 4 récapitule les performances de chaque protocole en termes de nombre de messages échangés, nombre d'écritures forcées, latence et opérations pénalisantes dans les deux types de mobilité. Le paramètre latence est exprimé en nombre d'étapes de communication entre le moment où l'application décide de terminer une transaction et le moment où la validation devient effective sur chaque participant. P représente le nombre de

participants, M représente le nombre d'unité mobiles (client et participants) et H le nombre d'unité mobiles exécutant un hand off L3.

	Protocole	Messages	Ecritures forcées Coord.+participants	Latence	Opération pénalisante
Mobilité intra domaine	2PC au dessus du mobile IP	4P	2P+1	3	/
	M-2PC	4P +3M	2P+2M+1	6 (0 participants mobiles) 7 (au moins un participant mobile)	Transfert du journal
	2PC avec coordinateur fixe	4P+3	2P+3	6	Transfert du journal
Mobilité inter domaine	2PC au dessus du mobile IP	4P	2P+1	3	Routage indirect
	M-2PC	4P +3M+H	2P+2M+1	6 (0 participants mobiles) 7 (au moins un participant mobile)	/
	2PC avec coordinateur fixe	4P+3	2P+3	6	Routage indirect

Tableau 4 - Comparaison des performances de 2PC et de M-2PC

2PC est le protocole qui nécessite moins d'écritures forcées et moins d'étapes de communication pour la validation, sachant que ces deux paramètres influent sur la latence des protocoles de validation, ce qui le rend plus performant en terme de latence que M-2PC et 2PC avec coordinateur fixe dans les deux types de mobilité. En plus du nombre élevé d'étapes de communication, M-2PC et 2PC avec coordinateur fixe sont pénalisés par l'opération du transfert du journal dans le cas de la mobilité intra domaine. Dans le cas de la mobilité inter domaine, le routage indirect offert par mobile IP pénalise le protocole 2PC et le 2PC avec coordinateur fixe. Malgré cela 2PC reste plus performant que M-2PC et 2PC avec coordinateur fixe à cause du nombre d'étapes de communication nécessaires pour la

validation ainsi que le nombre d'écritures forcées. Rappelons que les écritures forcées ajoutées à M-2PC et à 2PC avec coordinateur fixe sont ajouté du coté des unités mobiles. Sachant que le délais d'une opération disque sur une unité mobile est non négligeable. (Voire section 4.1).

Conclusion

La performance du protocole M-2PC en terme de latence ne dépend pas seulement du type du hand off, mais également des paramètres de mouvement, de la taille du fichier log, du nombre de participants mobiles, et surtout de la charge du réseau sans fil.

Bien que 2PC n'a pas été conçu pour les environnements mobiles et sans fil, on observe une performance assez intéressante en termes de latence en le comparant à M-2PC et à une variante 2PC ayant le coordinateur sur une station fixe. La latence de ces deux derniers protocoles est pénalisée par le nombre d'étapes de communication nécessaires à la validation et le transfert du journal de la transaction s'exécutant sur le terminal mobile vers le coordinateur (en cas de mobilité intra domaine) qui est supposé fixe. Néanmoins, 2PC est le protocole qui nécessite le coût de communication sur le support sans fil le plus élevé à cause du nombre de messages échangés entre le coordinateur mobile et les participants. Nous donnons ci-dessous le résumé des résultats des simulations décrites précédemment:

1. Les paramètres de mouvement (la vitesse, la position du client lors de l'initiation du processus de la validation et la portée de la station de base) sont étroitement liés et définissent ensemble le paramètre T= (R-X)/V. Ce paramètre représente le temps du déclenchement du hand off par rapport au temps de l'initiation du processus de validation.
2. Plus le hand off se déclenche tôt, plus le parallélisme entre le processus du hand off et le processus de validation augmente. Ce qui permet de diminuer la latence.
3. La latence du protocole M-2PC en présence du hand off L2 (mobilité intra domaine) est moins considérable que celle mesurée en présence du hand off L3 (mobilité inter domaine). Cette dernière est pénalisée par la durée d'obtention d'une nouvelle adresse à partir d'un serveur DHCP. L'utilisation du DRCP peut réduire la latence mais les résultats obtenus restent contraignants pour les transactions temps réel.
4. La localisation du coordinateur au niveau de la partie fixe du réseau permet de surmonter les limites du réseau sans fil mais impose le transfert du journal des opérations des transactions ce qui pénalise la latence des protocoles de validation en cas de mobilité intra domaine.
5. L'influence du nombre des participants fixes sur la latence du protocole M-2PC est légère voire négligeable. Par contre l'influence du nombre des participants mobiles est considérable à cause du transfert du journal des opérations de la transaction en même temps que le message *vote*.

6. La latence du protocole M-2PC ne dépend pas seulement des paramètres de mouvement et du nombre de participants fixes et/ou mobiles, mais dépend essentiellement de la charge du réseau sans fil (nombre d'unités mobiles actives dans la cellule du client ou du participant).
7. Malgré le fait que 2PC n'a pas été conçu pour des systèmes impliquant des mobiles, il donne une meilleure performance inattendue en terme de latence en comparaison aux protocoles qui proposent de localiser le coordinateur sur la partie fixe. Ces protocoles sont pénalisés par le nombre détapes de communication nécessaires pour la validation. Ainsi que l'opération du transfert des journaux dans le cas de mobilité intra domaine. Dans le cas de mobilité inter domaine, 2PC avec coordinateur fixe est aussi pénalisé par le routage indirect du résultat de la validation vers le client mobile.

Conclusion générale

L'importance de la validation atomique des transactions est telle que les recherches la concernant n'ont jamais cessé depuis l'apparition du concept de transaction dans les années 70s. Ces recherches connaissent un renouvellement d'intérêt dans le contexte du calcul mobile et sans fil vu les nouveaux problèmes qui ont été introduits. Des solutions ont été proposées pour garantir la terminaison correcte des transactions mobiles. Ces solutions supposent l'existence d'une politique de gestion de mobilité, au dessous de l'application transactionnelle, qui prend en charge la mobilité du client et/ou des participants. Or à ce jour les infrastructures réseau n'intègrent pas encore ces solutions. Partant de ce constat et afin de contourner le coût considérable du développement de nouvelles techniques tels que la modification des infrastructures réseaux existantes ou des systèmes d'exploitation opérants, le protocole de validation atomique à deux phases pour les environnements mobiles (M-2PC), propose de prendre en charge la mobilité du client et/ou des participants à la transaction. Pour cela, il intègre un mécanisme qui permet au client et au participant d'informer le coordinateur sur leur nouvelle localisation.

Dans ce travail, nous avons étudié la performance du protocole M-2PC que nous avons comparé à 2PC. Nous avons d'abord commencé par une étude qualitative puis une étude quantitative en utilisant le simulateur des réseaux NS. D'après les résultats des simulations que nous avons obtenus, nous pouvons dire que :

- L'influence du nombre de participants fixes est négligeable.
- La taille du journal à transférer n'influe pas sur la latence du protocole en cas de la mobilité inter domaine. En effet, le processus hand off est un processus parallèle à celui de validation sur la partie fixe. Et donc la latence du protocole de validation globale dépend du processus le plus lent. Le processus hand off en cas de mobilité inter domaine est plus lent car il est pénalisé par le délai d'obtention d'une nouvelle adresse.
- La localisation du coordinateur au niveau de la partie fixe du réseau est une solution pour surmonter les limites des unités mobiles et du réseau sans fil, mais désavantage la latence du protocole à cause du (1) nombre élevé d'étapes de communication nécessaire pour la validation, (2) nombre d'écriture forcés et (3) le transfert des journaux (mobilité intra domaine). Les protocoles de validation atomique en environnement mobile proposant un coordinateur fixe doivent utiliser des techniques de compression ou d'optimisation des journaux à transférer sur le support sans fil.

- La latence du protocole M-2PC varie entre 1298ms et 2241ms dans le cas de mobilité inter domaine. Cette latence est pénalisée par le processus d'obtention d'une nouvelle adresse qui dure plus d'une seconde. En général, le client/participant mobile n'est pas le seul à être géré par la station de base, il existe d'autres mobiles dans la cellule qui partagent avec le client/participant mobile la bande passante. Dans ce cas, la latence augmente considérablement en fonction du nombre de sites actifs dans la cellule. D'après nos résultats de simulation, nous avons obtenu une latence de 13 secondes pour 6 unités actives. Bien que cette valeur est acceptable pour les applications transactionnelles non temps réel, elle ne l'est pas pour les applications transactionnelles temps réel.

  Dans le cas de la mobilité inter domaine, 2PC au dessus du Mobile-IP présente une meilleure performance que M-2PC, mais ce dernier est plus performant que le 2PC avec le coordinateur fixe. Bien que M-2PC évite le problème du routage indirect (induit par l'utilisation de mobile IP pour la gestion des clients/ participants mobiles) en envoyant la décision directement au mobile, il reste moins performant que 2PC au dessus du mobile IP à cause du nombre élevé d'étapes de communication et du nombre d'écriture forcées ajoutées aux unités mobiles. En fin, le 2PC avec un coordinateur placé sur la partie fixe (cette variante convient aux unité mobiles ne disposant pas de moyen de calcul nécessaire à un coordinateur et aussi afin de minimiser l'utilisation des liens sans fil) réuni le routage indirect et le nombre élevé d'étapes de communication, ce qui le rend moins performant que 2PC avec coordinateur mobile et M-2PC.

Discussion de la gestion de mobilité au niveau de la couche application

D'après la recherche bibliographique effectuée dans le domaine de la gestion de la mobilité au niveau de la couche application, nous constatons la faisabilité de cette approche. Mais quand aux performances, et selon les résultats des simulations obtenus par M-2PC, nous pouvons constater qu'elle est fortement liée aux fonctions du réseau sans fil.

Les approches basées sur l'utilisation d'un proxy/serveur de mobilité permettent de gérer la mobilité des sites sans exiger des modifications aux applications. Cependant, ces approches possèdent l'inconvénient du routage indirect des données destinées au site mobile. Ce qui conduit à une augmentation du temps de réponse de l'application. D'un autre côté, les approches qui ne se basent pas sur un serveur de mobilité permettent d'éviter le routage indirect, mais exigent la modification des applications par l'insertion de quelques lignes de code pour le support de mobilité tel que le mécanisme de redirection explicite de *Mobile Socket*. L'ajout de quelques lignes de codes ou d'un message à un protocole de niveau application tel que M-2PC, n'augmente pas la difficulté de conception de l'application.

Implémentation du protocole M-2PC

L'implémentation du protocole M-2PC peut se faire de la même manière que le protocole 2PC, à l'exception que :

- Le coordinateur doit être localisé sur la station de base ou une autre station fixe.
- Chaque participant mobile doit déléguer sa station de base/une station fixe dans le processus de validation.
- Le client/participant mobile doit disposer d'un mécanisme lui permettant de découvrir le changement du point d'attachement du site mobile sur lequel il s'exécute afin d'informer son délégué sur sa nouvelle localisation. Pour cela, nous proposons l'utilisation de l'une des deux techniques suivantes :
    1. Au démarrage du protocole de validation M-2PC, le client mobile envoie une séquence de ping au coordinateur et calcule le temps de réponse moyen du coordinateur à un ping. Si le site mobile détecte le changement du point d'accès, il envoie un ping à son coordinateur. Dans le cas où il ne reçoit aucune réponse pendant une période supérieure au délais moyen de réponse à un ping. Le client déduit qu'il a changé d'adresse IP, et qu'il doit informer son coordinateur de sa nouvelle localisation.
    2. Une API qui permet de vérifier de manière périodique les paramètres de configuration réseau du site mobile, et de détecter le changement d'adresse IP, peut être développée. L'API peut indiquer ce changement à l'application transactionnelle (le client mobile) qui informe à son tour le coordinateur. Plus la période de vérification est courte, plus la détection du changement d'adresse IP est rapide. Cependant, la vérification périodique peut augmenter la latence du protocole.

L'étude réalisée dans ce mémoire ainsi que les résultats des comparaisons présentées ne concernent que la gestion de la mobilité. Or, pour pouvoir compléter la comparaison entre les protocoles étudiés il faut étendre cette étude en incluant les autres problèmes tels que ceux liés à la consommation d'énergie, l'utilisation de la bande passante sans fil et la gestion des déconnexions notamment; l'impact de ces dernières sur le taux d'abandon des transactions.

Bibliographie

[AC 95] Y.Al-Houmaily, P.Chrysanthis, "Two-phase Commit in Gigabit-Networkerd Distributed Databases", Proceedings of the 8th International Confrence on Parallel and Distributed Computing Systems (PDCS), 1995.

[Ade 02] Équipe ADEPT, "Asynchronous Distributed Environments,Protocols, and Time", rapport d'activité, INRIA, France, 2002.

[AH 99] P.Anelli et Horlait, "NS-2 : Principe de Conception et d'Utilisation", version 1.3, septembre 1999.

[AK 02] G. AyorkorMills-Tettey. D.kotz 'Mobile Voice Over IP (MVOIP) : An Application Level Protocol for Call Hand-off in Real Time Applications'. dans *Proceedings of the Twenty-first IEEE International Performance, Computing, and Communications Conference*, pages 271-279. IEEE Computer Society Press, April 2002.

[AP 98] M.Abdallah, P.Pucheral, "Validation Atomique: Etat de l'Art et Perspectives", Revue Ingénierie des Systèmes d'Informations (ISI), Vol.5, n° 6, 1998.

[Bad 98] N.Badache, "Ordre Causal et Tolérance aux défaillances en Environnement Mobile", Thèse de doctorat d'Etat en Informatique. Présentée à l'USTHB, octobre 1998.

[BCFG+ 97] J.Besancenot, M.Cart, J.Ferrié, R.Guerraoui, P.Pucheral, B.Traverson, "Les systèmes transactionnels", Edition Hermes, Paris, 1997.

[BG 81] P.Bernsein, N.Goodman, "Concurrency controlin distributed database system", ACM Computing Survey, vol 13 N°2, 1981.

[BHG 87] P.A.Bernstein, V.Hadzilacos, N.Goodman, "Concurrency control and recovery in database systems", livre, Addison Wesley Publisher, 1987.

[BLRS 04] C.Beaubino, C.Labbé, C..L.Roncancio, P.serrano-Alvarado, "Performances de protocoles transactionnels en environnement mobile", dans 20ème journée bases de données avancées (BDA), Montpellier, France, Octobre 2004.

[Bob 02] C.Bobineau, Gestion de transaction en environnement mobile", thèse de doctorat, université de Vairsailles Saint-Quentin-en-Yvelines, 20 Décembre 2002.

[BPA 00] C.Bobineau, P. Pucheral, M. Abdallah. "A unilateral commit protocol for mobile and disconnected computing", dans Proc; of the 12th int. conf. on parallel and distributed computing systems (PDCS), Las Vegas, USA, Aug. 2000.

[BPR 97] E.Bertino, E.Pagani, G.P.Rossi, 'Fault Tolerance and recovery in mobile computing systems', dans Recovery Mechanisms in database systems, chapitre 24, Edition Prentice-Hall, 1997.

[CBML O3] Y.Chung, Bhargava, M.Mahoui, L.Lilien, "Autonomous Trannsaction Processing Using Dta Dependency in Mobile", dans the ninth IEEE worksop on Future Trends of Distributed Computing Systems (FTDCS'03). 2003.

[Chr 93] P.K.Cherniak, "Transaction Processing in a Mobile Computing Environment", dans proc of the IEE Workshpo on Advances in Parallel and Distributed Systems, pages 77-82, 1993.

[CL 99] G.C.Chen, S.Y.Lee, "An analytic model for performance analysis of concurrency strategies in mobile environments", the computer journal, vol.42, n°6.1999.

[CNB 05] F.Chehbour, N.Nouali, L.Boukantar, "Gestion de la mobilité au Niveau de la Couche Application: Etat de l'Art", 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, 27-31 mars, 2005- Tunisie.

[CNZ 05] F.Chehbour, N.Nouali et K.Zeraoulia, "Fast handoff for Hierarchical Mobile SIP Networks", dans the 3rd World EnformatiKa Conference. WEC'05 April 27-29, 2005 in Istanbul, Turkey.

[Coo 82] E.C.Cooper, "Analysis of distributed commit protocols", dans the 1982 ACM SIGMOD International conference on management of data, Orlando, Florida, P.175-183, 1982

[CT 96] T.Chandra, S.Toueg, "Unreliable failure detectors for reliable distributed systems", Journal of the ACM, vol.34, n° 1, 1996.

[DCKM+ 94] F.Douglis, R.Caceres, F.Laashoek, K.Li, B.Marsh, J.A.Tauber, "Storage alternatives for mobile computers", dans 3rd Int. WorkShop on Modeling, Analysis and simulation of Wireless and Mobile Systems (*MSWiM*), Boston, USA, 1994.

[DG 98] R.A.Dirckze et L.Gruenwald, "A Toggle Transaction Management Technique for Mobile Multidatabases", dans G.Gardarin, J.French, N.Pissino

[DHB 96] M.H.Dunham, A.Helal, S. Balakrishanan, "A mobile transaction model that captures both the data and movement behaviour", dans the ACM/baltzer journal on special topics in mobile Networks and Applications, 1996.

[DK 99] M.Dunham, V.Kumar, "Impact of mobility on transaction management", dans the 1st ACM International workshop on data engineering for wireless and mobile access, USA, P. 14-21.

[Dom+ 01] G.Dommety et al, "Fast Handovers for Mobile IPv6", IETF draft, draft-ietf-mobileip-fast-mipv6-03.txt, July 2001.

[EGLT 76] K.P.Eswarn, J.Gray, R.A.Lorie, I.L.Triger, "the notions of consistency and predicate locks in a database system", Communications of the ACM (CACM), vol 19, n° 11, 1976.

[FLM 85] M.Fisher, N.Lynch, M.Paterson, "Impossibility of distributed Consensus with one faulty process", journal of the ACM, vol.32, n°2, 1985.

[Fri 01] U.Fritzke, "Les systèmes transactionnels répartis pour données dupliquées fondés sur la communication du groupe", Thèse de doctorat, université de Rennes 1, Janvier 2001.

[HDS 03] P.Y. Hsieh, A.Dutta, H.Schulzrinne "Application Layer Mobility Proxy for Real-time communication" Accepted for 3G Wireless 2003

[Gue 97] R.Guerraoui, "Transactions réparties : Algorithmes, Systèmes et langages", rapport scientifique pour l'obtention de l'habilitation à diriger des recherches en informatique, 1997.

[GHOS 96] J.Gray, P.Hella, P.ONeil et D.Shasha, "The Dangers of replication and a Solution", dans H.V.Jagadish et I.S.Mumick, editors, Proc. Of the 1996 ACM SIGMOD Int. Conf. on Management of Data, Montreal, Quebec, Canada, ACM SIGMOD Record, Vol.25, n° 2, pages 173-182, ACM Press, Juin 1996.

[Glo 01] GLOMOSIM, "Global Mobile Information Systems Simulation Library", UCLA Paralle Computing Laboratory, http://pcl.cs.ucla.edu/projects/glomosim/, 2002.

[GR 93] J.Gray et A.Reuter, "Trans.actions Precessing: Concepts and techniques", San Mateo, California, Morgan Kaufmann, 1993.

[Gre 03] M.Greis, "tutorial for NS simulator", VINT group, 11 Mars 2003 http://www.isi.edu/nsnam/ns/tutorial/

[Gra 02] P.Gralla, "How wireless works", Edition Que, 2002.

[Gra 81] J.Gray, "The transaction Concept: Virtus and Limitations", dans proc of the 7th VLDB, Cannes, pp. 144-154, 1981.

[Gra 87] J.Gray, "Note on database operating systems", Operating Systems: AN Advanced Course, LNCS, vol 60, Springer Verlag, 1978.

[Gue 95] R.Guerraoui, "Revisiting the relationship between non-blocking atomic commitment and consensus", dans proceeding of the 8th International workshop on distributed algorithms, WDAG'95. Springer Verlag., septembre 1995.

[Gue 97] R.Guerraoui, "Transactions réparties : algorithmes, systèmes et langages", rapport scientifique pour l'obtention de l'habilitation à diriger des recherches en informatique, 1997.

[ISO 92] ISO, Open System Interconnection Distributed Transaction Processing (OSI-TP), Model ISO IS 100261, 1992.

[JP 98] D.Johnson, C.Perkins, "Mobility Support in IPv6", Internet Draft, draft-ietf-mobileip-ipv6-08.txt, Mars 1998.

[KD 98] V.Kumar, M.H.Dunham, "Defining Location Data Dependency, Transaction Mobility and Commitement", Technical report 98-CSE-01, Departement of Computer Science and Engineering, Souther Methodist University, Dallas, USA, 1998.

[KDDS 00] V. Kumar, K. Dash, M.H. Dunham, A.Y, Seydim. "A timeout-based mobile transaction commitment protocol", ADBIS-DASEAA 2000, Advances in DB and information systems, in cooperation with ACM SIGMOD, Prague, Czech republic, Sept. 5-8, 2000.

[KK 00] K.I.Ku, et Y.S.Kim, "Moflex Transaction Model for Mobile Heterogeneous Multidatabase Systems", dans RIDE 2000, Proc. Of the 10th Int. Workshop in research Issues in Data Engineering: Middleware for Mobile Business Application and E-Commerce, février 27-28, 2000, San Diego, USA.

[KKLY+ 04] W. Kim, M. Kim, K. Lee, C. Yu and B. L. “Link Layer Assisted Mobility Support Using SIP for Real-time Multimedia Communications” ACM MobiWac, 2004.

[KP 04] J. Kristiansson. P. Parnes “Application Layer mobility Support for Streaming Real Time Media”, dans IEEE Wireless Communications and Networking Conference (WCNC´04), Georgia, Atlanta, 2004**.**

[KPDS 02] V.Kumar, N.Prabhu, M.Dunham, Y.A.Seydim, “TCOT a timeout-based mobile transaction commitment protocol”, IEEE transactions on Computers, Vol.51 (10) P.1212-1218, 2002.

[Kum 03] R.Kumar, “MSIP: A Protocol for Efficient Handoffs of Real-time Multimedia Sessions in Mobile Wireless Scenarios”, Dissertation submitted for the degree of Master of technology, Kanwal Rekhi School of Information Technology Indian Institute of technology, Bombay Mumbai-400076, 2003.

[Lia 99] W.Liao, “Mobile Internet Telephony Protocol: An application layer protocol for mobile Internet telephony services”, Work supported by the National Science Council, Taiwan, NSC 88-2219-E-002-007, IEEE 1999.

[LS 94] Q. Lu et M.Satyanarayanan, “Isolation-Only transactions for Mobile Computing”, ACM Operating Systems Review, P.81-87, 1994.

[LW 99] Y.W. Lin, H.U.Wu., Commit protocol for low powered mobile clients, *IEICE Trans*. INF&SYST, vol. E82-D, n°.8, August 1999.

[MLO 86] C.Mohan, B.Lindsay, R.Obermark, “Transaction management in the distributed data base management system”, ACM Transactions on database systems, vol.11, n° 4, 1986.

[MN 02] N.Montavont, T.Noel, “Handover Management for Mobile Nodes in IPv6 Networks”, IEEE Communications Magazine, August 2002.

[Nar 94] V.R.Narasayya, “Distributed Transactions in Mobile Computing System”, Rapport Technique, Depart. Of Computer Science, University of Washington.

[NBDD 05] N.Nouali, L.Boukantar, A.Doucet et H.Drias, “Transaction Commitment Protocols for Mobile Wireless Environment”, dans International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, 27-31 mars, 2005- Tunisie.

[NDD 04] N.Nouali, A.Doucet et H.Drias, “Executing the Two-Phase Commit Protocol in Mobile Wireless Environment”, dans Special session in wireless computing, International conference on Advances in Computer Science and Technology (ACST 2004). St Thoma, Virgin, Islands, USA, 22-24 Novembre 2004.

[NDD 05a] N.Nouali, A.Doucet, H.Drias, “A Two-Pahse Commit Protocol for Mobile Wireless Environment”, dans Sixteeth Australian Database Conference (ADC2005), Newcastle, Australia.CRPIT, 39.Williams, H.E.and Dobbie, G., Eds. , ACS.135-144, 2005.

[NDD 05b] N.Nouali, A.Doucet, H.Drias, “Mobility management support for transactional applications”, dans 5th Workshop on Applications and Services in Wireless Networks (ASWN2005), PARIS, France, 29 Juin -1 Juillet 2005.

[NDD 05c] N.Nouali, A.Doucet, H.Drias, “Revisiting distributed protocols for mobility at the application layer”, dans World Enfomatika conferences (WEC’05), International conference on parallel and distributed systems (PDS 2005), Istanbul, Turkey, 27-29 April 2005.

[Nou 01] N.Nouali, “Impact de la mobilité sur les modèles de transactions”, rapport interne, CERIST, janvier 2001.

[NP 94] K.Norvag et H.Pedersen, “DBMS scheduling performance”, Technical report, Norwegian institu te of technologie, 1994.

[NS 03] The Network Simulator – ns-2, “The ns Manual”, The VINT Project, collaboration between researchers at UC Berkely, LBL, USC/ISI, and Xerox PARC, Kevin Fall and Kannan Varadhan Editors, 2003, http://www.isi.edu/nsnam/ns.

[NSB 97] K.Norvag, O.Sandsta, K.Bratbergengen, “Concurrency Control in Distributed Object-Oriented Database Systems”, dans Advances in Databases and Information Systems ADBIS’97, édité dans electronic Wirkshops in Computing (eWiC), Springer-Verlag, pages 9-17, 1997.

[OMG 94] OMG Object Management Groupe, Object Transaction Service OMG Document 94.8.4, OMG editor, August 1994.

[OMTT 99] T.Okoshi, M. Mochizuki, Y. Tobe, H. Tokuda. “MobileSocket: Toward Continuous operation for Java Appplications”. IEEE 8th International National Conference on Computer Communications and Networks 1999 (IC3N’99), pages 50-57

[Opn 04] OPNET Technologies, “OPNET Modeler accelerating networks R&D”, 2004, www.opnet.com

[OV 99] T.Ozsu, P.Valduries, “Principals of distributed Database Systems”, Prentice Hall, 2nd edition, 1999.

[PB 94] E.Pitoura et B.Bbhargava, “Building Information Systems for Mobile Environments”, dans N.R. Adam, B.Bhargava, et Y.Yesha, editors, Proc. of the 3rd Int conf on information et Knowledge Management (CIKM 94), Gaithersburg, Maryland, November 29-decembre 2, 1994, pages 371-378, ACM Press, 1994.

[PB 95] E.Pitoura et B.Bbhargava, “Maintaining Consistency of Data in Mobile distributed Environments ”, dans Proc of the 15th Int conf on Distributed Computing systems, pages 404-413, IEEE computer Society Press, 1995.

[PB 99] E.Pitoura et B.Bbhargava, “Data Consistency in Intermittently Connected Distributed Systems”, IEE ETransactions on Knowledge and Data enginnering, Novembre/Decembre 1999.

[Pit 96] E.Pitoura, “A Replication Schema to support Weak Connectivity in Mobile Information Systems”, dans R.R. Wagner and H.Thoma, editors, Database and expert System Applications, Proc of the 7th Int conf, DEXA’96, Zurich, Switzerland, Septembre 1996, lecture Notes in Computer science, vol. 1134, pages 510-520. Springer-Verlag, Berlin, 1996.

[PJA 03] C.Perkins, D.B.John et J.Arkko, "*Mobility Support in IPv6*", Internet-Draft, 2003.

[PF 97] V.Paxon, S.Floy, "Why we dont know how to simulate the internet?", dans Proceedings of the Winter Simulation Conference, December. 1997

Pha 05] P.P.Pham, "Comprehensive Analysis of IEEE 802.11", Mobile Networks and Applications, Vol 10, page 691-703, 2005.

[Puj 01] A.V.Pujolle, "Réseaux de mobiles et réseaux sans fil"

[Sag 02] F.N.Sagna, "Transaction mobiles", mémoire de DEA, Université de Nice sophia antipolis, septembre 2002.

[SC 90] J.Stamos, F.Cristian, "*A Low-Cost Atomic Commit Protocol*", Proceedings of the 9th Symposium on Reliable Distributed Systems, 1990.

[Ser 04] P.Serrano-Alvarado, "Transactions Adaptables pour les Environnements Mobiles", Thèse PhD, université Joseph Fourier, Grenoble, France, Février 2004.

[Ske 82] D.Skeen, "Nonbloking commit protocols", dans ACM International SIGMOD Conference on management of data (SIGMOD' 82), pp.133-147, Orlando, FL, Iuin 1982.

[SW 99] H. Schulzrinne and E. Wedlung. "*Application-layer mobility using SIP*". ACM Mob. Comput. Commun. Rev. 1(5), 1999.

[TMUY 03] J. Tsumochi, K.Masayama, H. Uehara, M.Yokoyama, "Impact of mobility Metric on Routing Protocols for Mobile Ad hoc Networks", dans IEE PacificRim Confrence on Communications, Computers and Signal Processing (PACRIM03), Victoria, p.322-325, 2003.

[TVP 00] A.Tsalgatidou, J.Veijalainen et E.Pitoura, "*Challenges in Mobile Electronic Commerce*", in IeC 2000, 3rd International Conference on Innovation through E-Commerce, November 14th-16th, 2000, Manchester UK.

[TZ 03] C.Turker, G.ZINI, "A survey of academic and commercial approches to transactions support in mobile computing environnements", technical report, ETZ Zurich departement of science, 2003.

[Vid 01] B.Vidal, "Applications mobile avec oracle", Edition Eyrolles, 2001.

[WC 97] G.D.Walborn et P.K.Chrysanthis, "PRO-MOTION: Management of Mobile Transactions", dans Proc. Of the 1997 ACM Symposium on Applied Computing, 28 février- 1Mars, 1997, San Jose, CA, USA, pages 101-108, ACM Press, 1997

[WC 99] G.D.Walborn et O.K.Chrysanthis, "Transaction Processing in PRO-MOTION", dans Proc. Of the 1999 ACM Symposium on Applied Computing, 28 Février-2 Mars, 1999, San Antonio, Texas, USA, Pages 389-398, ACM Press, 1999.

[WV 02] G.Weikum, G.Vossen, "Transactional information systems: teeny algorithms and the practice of concurrency control and recovery"; ed. Morgan Kaufmann, USA, 2002.

[XOPE+ 96] X/OPEN CAE Specification, Distributed Transaction Processing, Reference Model. X/OPEN Guide version 3, G307, X/OPEN Company Limited, 1996.

[YV 02] X.Yang, N.H.Vaidya, “Pipelined Packet scheduling in wireless LANs”, technical report, August 2002.

[YZ 94a] L.H.Yeo, et A.Zaslavsky, “Layered Approch to Transaction Management in Multidatabase Systems”, dans Proc. of the 5th Int. Hong Kong Computer Society Database Workshop: Next Generation Database Systems, pages 179-189, 1994.

[YZ 94b] L.H.Yeo, et A.Zaslavsky, “Submission of Transactions for Mobile Workstations in a Cooperative Multidabase Processing Environment”, dans Proc. Of the 14th IEE Int.Conf. on Distributed Computing Systems, Juin 21-24, 1994 Posnan, Poland, pages 372-379, IEEE Computer Society Press, 1994.

[ZWZ 03] J.L.Zheng, X.P.Wang, G.D.Zhang, “MPSSF: A Buffering & Forwarding Handover Scheme Reducing Out-of-Sequence Packets”, dans International Conference on Computer Networks and Mobile Computing (ICCNMC'03), 2003.