Performances de protocoles transactionnels en environnement mobile

Christophe Bobineau — Cyril Labbé — Claudia Roncancio — Patricia Serrano-Alvarado

Laboratoire LSR-IMAG, BP 72, 38402 St. Martin d'Hères, France
e-mail : Prénom.Nom@imag.fr

RÉSUMÉ. *L'omniprésence d'unités mobiles et le développement des réseaux sans fil motivent des avancées en matière de supports d'exécution pour une grande variété d'applications en environnement mobile. Des efforts importants sont faits pour offrir une bonne gestion des données malgré les caractéristiques limitées de tels environnements. La notion de transaction a été réétudiée pour proposer des modèles et des protocoles permettant d'assurer certaines propriétés transactionnelles. Les algorithmes et protocoles proposés tentent d'optimiser l'utilisation des ressources des unités mobiles et de surmonter les limitations du réseau mobile. Dans cet article nous nous intéressons spécifiquement aux protocoles de validation de transactions réparties sur plusieurs unités dont certaines mobiles. Nous présentons ici les résultats d'un travail qui identifie des propriétés qualitatives et des indices de performance quantitatifs, et étudie ceux-ci sur quatre protocoles (2PC, UCM, CO2PC et TCOT). Trois de ces protocoles sont des propositions spécifiques pour des environnements mobiles et sont représentatifs des propositions actuelles. Nous nous intéressons à la phase de validation des transactions, ainsi qu'à l'influence des différents protocoles sur les performances du déroulement de l'ensemble des transactions.*

ABSTRACT. *This is an abstract. The omnipresence of mobile devices and wireless networks lead to a growing interest in supporting a wide variety of applications in mobile environments. Numerous efforts in providing appropriate data managements for such environments are made. Transaction supports have been revisited to propose adapted transaction models and properties. Proposed algorithms and protocols try to optimise the use of mobile units resources and to overcome wireless network limitations. This work concerns protocols to commit transactions distributed over several mobile and fixed units. Results presented here concern the identification of qualitative properties and quantitative performance indices that are studied on four protocols (2PC, UCM, CO2PC et TCOT). Three of them are specifics propositions for mobile environments and are representative of current proposals. The analysis concerns the transaction validation phase as well as the impact of the protocols on the performances during a transaction execution itself.*

MOTS-CLÉS : *transactions mobiles , evaluation de performances, protocoles de validation.*

KEYWORDS: *mobiles transactions, performances evaluation, validation protocols.*

1. Introduction

A l'heure actuelle, la recherche dans le domaine de l'informatique mobile connaît une activité très importante, en particulier sur les aspects liés à la gestion de données [BER 03]. Nous considérons qu'un système mobile comprend une application avec des utilisateurs mobiles ou fixes qui accèdent aux données localisées sur des sites également mobiles ou fixes. A cause des caractéristiques de l'environnement mobile [PIT 98], les techniques traditionnelles de duplication, de gestion de cache, de requêtes, de transactions, etc. deviennent pour la plupart inadéquates telles quelles.

La notion de transaction a été réétudiée [MAD 01, PIT 03, LEE 02, CUC 02] pour proposer des modèles et des protocoles permettant d'assurer certaines propriétés transactionnelles (atomicité, cohérence, isolation, durabilité) en environnement mobile. Les algorithmes et protocoles proposés tentent d'optimiser l'utilisation des ressources des unités mobiles et de surmonter les limitations du réseau mobile (déconnexions, faible bande passante, etc).

Dans cet article nous nous intéressons aux transactions mobiles. Un état de l'art complet sur ce thème est présenté dans [SER 04b]. Ici, nous nous intéressons plus spécifiquement aux protocoles de validation de transactions réparties sur plusieurs unités dont certaines mobiles. Nous présentons les résultats d'un travail qui évalue et compare des protocoles proposés pour la validation de transactions de ce type. L'objectif à long terme de ce travail est de classifier les différents protocoles existant dans le but de mieux cerner ceux qui sont les plus adaptés aux besoins des différentes classes d'applications et contextes réseaux. Le premier travail consiste a identifier les propriétés qualitatives et quantitatives de tels protocoles. Ces propriétés doivent ensuite être étudiées au regard des caractéristiques de l'environnement mobile. Notre contribution dans ce contexte porte sur deux points. D'une part, nous avons identifié des propriétés qualitatives et des indices de performance quantitatifs pouvant être utilisés pour des bancs de tests systématiques. D'autre part nous avons étudié ces propriétés pour quatre protocoles.

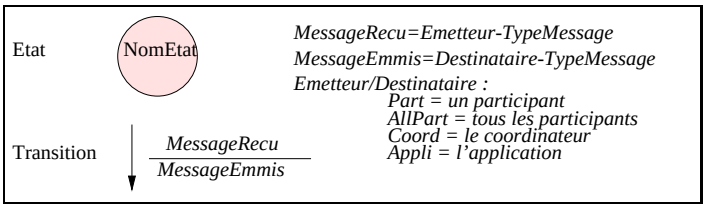

Figure 1. *Légende pour les diagrammes d'états*

Les protocoles analysés ici sont *Two Phases Commit* (2PC) [GRA 78], *Unilateral Commit protocol for Mobile and Disconnected Computing* (UCM) [BOB 02], *Combination of Optimistic approach and 2PC* (CO2PC) [SER 04a] et *Timeout-based Mobile Transaction Commitment Protocol* (TCOT) [KUM 02]. Les trois derniers sont des propositions spécifiques pour des environnements mobiles et sont représentatifs des propositions actuelles. 2PC et UCM assurent l'atomicité alors que CO2PC et TCOT garantissent l'atomicité sémantique.

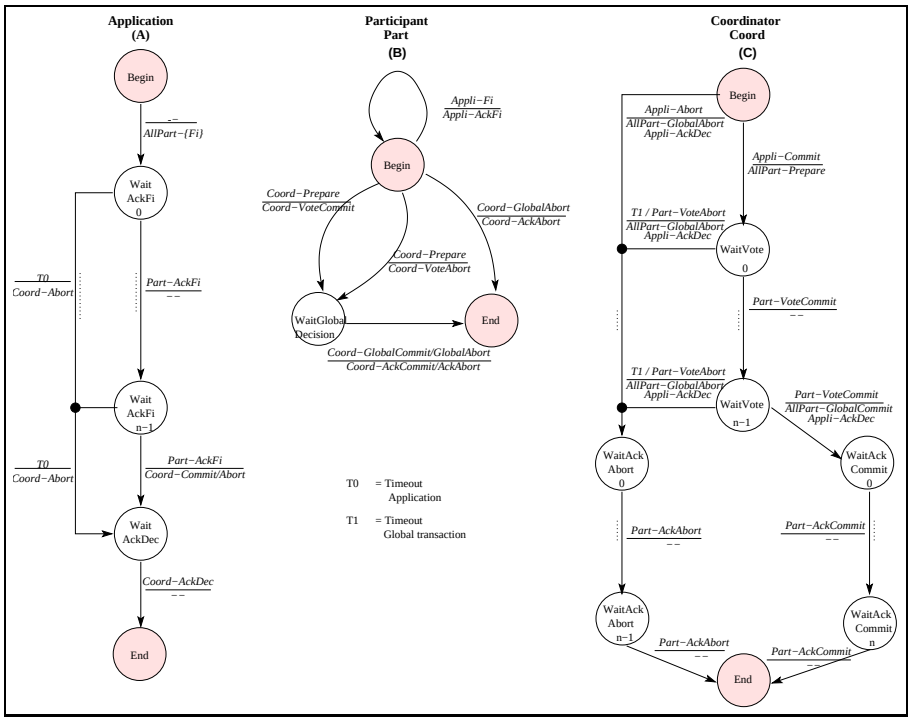

Figure 2. *Diagramme d'états de 2PC*

Dans la suite, nous introduisons brièvement les protocoles étudiés. Pour chacun nous présentons son diagramme d'états-transitions (nos conventions sont données par la figure 1) et ses caractéristiques essentielles. 2PC, UCM , CO2PC et TCOT font respectivement l'objet de la section 2, 3, 4 et 5. La section 6 présente une analyse qualitative des protocoles, alors que la section 7 présente une analyse quantitative des performances des protocoles. Cette analyse porte sur la phase de validation des transactions, ainsi que sur l'influence des différents protocoles sur les performances du déroulement de l'ensemble des transactions. La section 8 donne nos conclusions globales et introduit les travaux futurs.

2. Protocole 2PC

Description et propriétés

Le protocole 2PC n'a pas été conçu pour l'environnement mobile mais est utilisé par certaines propositions transactionnelles développées pour ces environnements. 2PC, dont le diagramme d'états est donné en fig. 2, permet la validation atomique de transactions distribuées sur plusieurs bases de données (SGBD sous-jacents). Chaque base exécute une transaction, dite locale, qui participe à la transaction globale. Les bases sont considérées comme les participants – fig. 2(B). La validation est initiée par l'application (fig. 2(A)) qui envoie un message au coordinateur du protocole – fig. 2(C). La décision de validation ou d'annulation globale de la transaction (phase de dé-

cision) est prise par le coordinateur après consultation des participants (phase de vote). Les participants qui votent pour une validation entrent dans un état de "préparation". Dans cet état, le participant n'a plus le droit de valider ou d'annuler unilatéralement la transaction locale. Ce protocole garantit la correction de la décision du coordinateur, dans le sens où la validation (resp. annulation) globale implique la validation (resp. annulation) de toutes les transactions locales qui la composent.

Prise de décision globale à tort

Dans certains cas le coordinateur décide l'annulation globale alors qu'une validation est possible. Cette situation survient lorsque le vote d'un participant ne parvient pas au coordinateur avant l'expiration de son temporisateur – état *Wait Vote i* sur la fig. 2(C).

Situations de blocage

Deux situations de blocage peuvent se produire dans 2PC. (1) Le coordinateur reste bloqué tant qu'il n'a pas reçu l'acquittement de la décision globale de la part de tous les participants – états *Wait Ack Abort i* et *Wait Ack Commit i* de la fig.2 (C). Ceci n'est pas très pénalisant car, à priori, aucune donnée ne reste bloquée au niveau du coordinateur. (2) Un participant reste bloqué s'il vote pour la validation et s'il ne reçoit pas la décision du coordinateur (état *Wait Global Decision* de la fig. 2(B). Cette situation pénalise le système car les ressources utilisées par la transaction locale sur le participant restent bloquées jusqu'à terminaison de la transaction globale. Le participant ne peut pas décider seul de la fin de cette transaction.

3. Protocole UCM

Description et propriétés

UCM a été spécifiquement créé pour les environnements mobiles. Son diagramme d'états est donné en fig. 3. 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 (ack envoyés à l'application après exécution). Dans le cas contraire, la transaction est immédiatement annulée. De ce fait, si la transaction termine, la décision est obligatoirement la validation et il n'y a pas de risque de décision à tort. Le protocole UCM se compose donc d'une seule phase (de décision), déclenchée 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 (message *Insert Record*).

Situation de blocage

Comme pour 2PC, le coordinateur reste bloqué si au moins un acquittement de la décision globale n'arrive pas (états *Wait Ack Abort i* et *Wait Ack Commit i* de la fig. 3(E)), mais cette situation n'est pas pénalisante pour le système.

4. Protocole CO2PC

Description et propriétés

CO2PC a également été spécialement conçu pour des environnements mobiles.

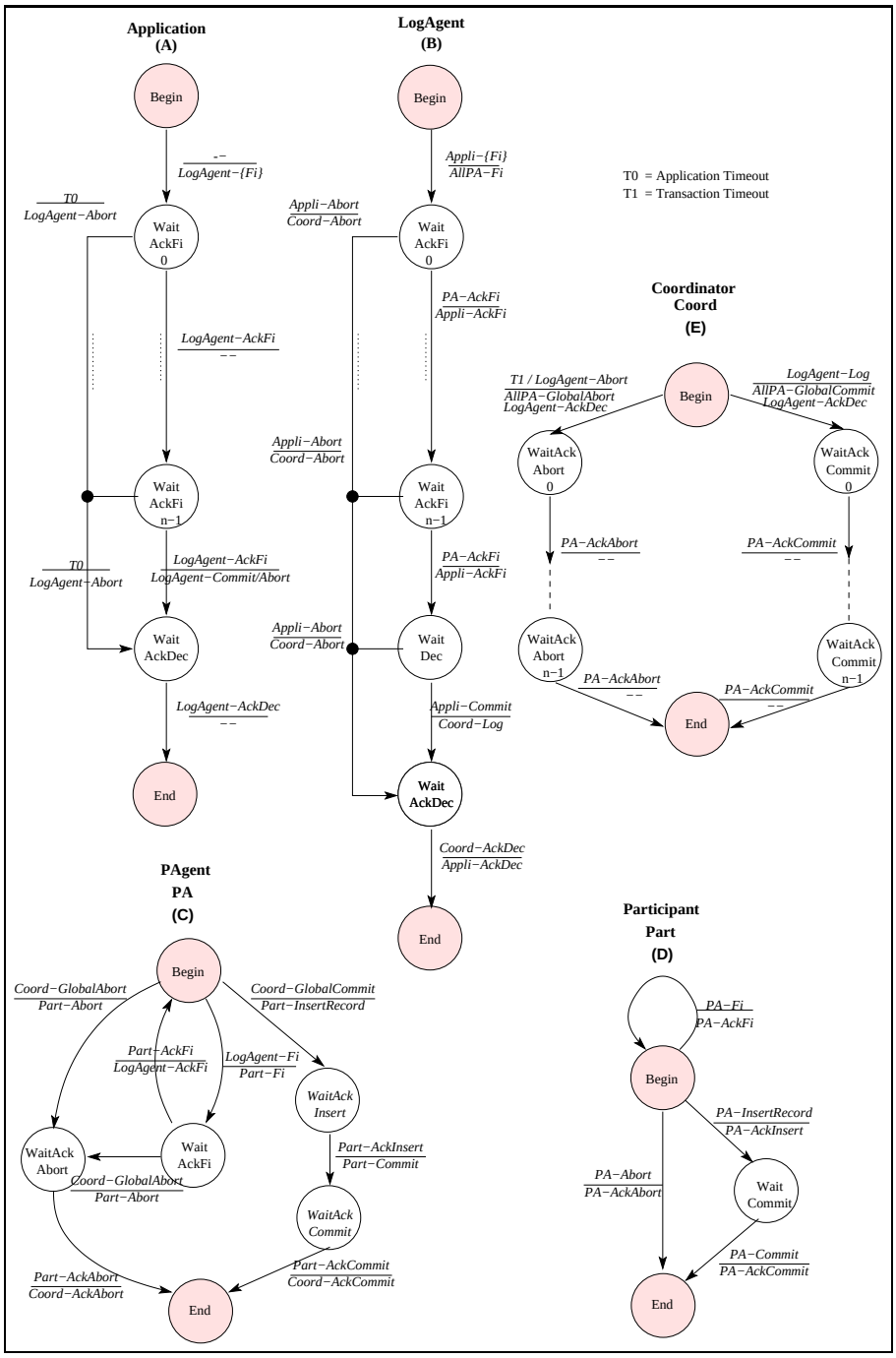

Figure 3. *Diagramme d'états de UCM*

Une transaction peut comprendre transactions composantes compensables (l'annulation d'une transaction se fait par une autre transaction, dite de compensation) et non compensables, de ce fait, CO2PC assure une atomicité sémantique des transactions globales[1]. Les transactions composantes compensables sont autorisées à valider unilatéralement de manière anticipée (validation optimiste) alors que la validation des transactions composantes non-compensables se fait de manière coordonnée, en accord avec la décision globale prise par le coordinateur. CO2PC augmente l'autonomie des participants sans imposer des interfaces particulières. Il fonctionne comme suit : chaque participant (représenté par un proxy) envoie son vote (*commit/abort*) au coordinateur qui prend la décision globale de validation à l'unanimité ou d'annulation sinon. Localement, chez les participants, l'envoi du vote se passe différemment selon que le participant exécute une transaction compensable (et fait une validation optimiste) ou non.

Un participant faisant une validation optimiste est représenté par un *OptProxy* – fig. 4(B). Celui-ci demande l'exécution de sa transaction composante au SGBD sous-jacent – appelé *OptPart* sur la fig. 4(E). Une fois l'exécution terminée, la transaction composante est validée ou annulée unilatéralement. L'*OptProxy* envoie ensuite le *vote* correspondant au *coordinateur* du CO2PC – fig. 4(D). Ultérieurement, si la décision globale est l'annulation, la transaction de compensation correspondante est exécutée par le SGBD sous-jacent, sinon, le participant a terminé.

Un participant exécutant une transaction composante non compensable ne peut pas valider unilatéralement. Dans ce cas, un *NonOptProxy* (fig. 4 (C)) coordonne un protocole 2PC localement avec le SGBD sous-jacent comme unique participant – *NonOptPart* sur la fig. 4(F). Le vote 2PC du SGBD est propagé par le *NonOptProxy* au coordinateur du CO2PC. Si le vote est l'annulation, le participant annule la transaction locale de façon unilatérale, sinon, il entre dans l'état de préparation, (*WaitGlobalDecision*). Dès que la décision globale du coordinateur CO2PC arrive au *NonOptProxy*, elle est transmise au SGBD. Cette décision sera considérée comme la décision pour le 2PC en cours localement.

Prise de décision globale à tort

Le coordinateur de CO2PC peut décider un abandon global à tort dans les mêmes circonstances que 2PC (vote arrivant après l'expiration du temporisateur).

Situations de blocage

Les participants et le coordinateur CO2PC ont des temporisateurs pour limiter les blocages. Malgré cela, (1) le coordinateur est bloqué, comme pour les protocoles 2PC et UCM, tant que tous les acquittements ne sont pas arrivés, et (2) suite à l'utilisation de 2PC pour des validations locales, un *NonOptPart* qui a voté commit reste bloqué jusqu'à réception de la décision globale. Les *OptPart* ne se bloquent pas (terminaison unilatérale).

1. 2PC et UCM considèrent exclusivement des transactions non compensables et assurent donc l'atomicité au sens stricte.

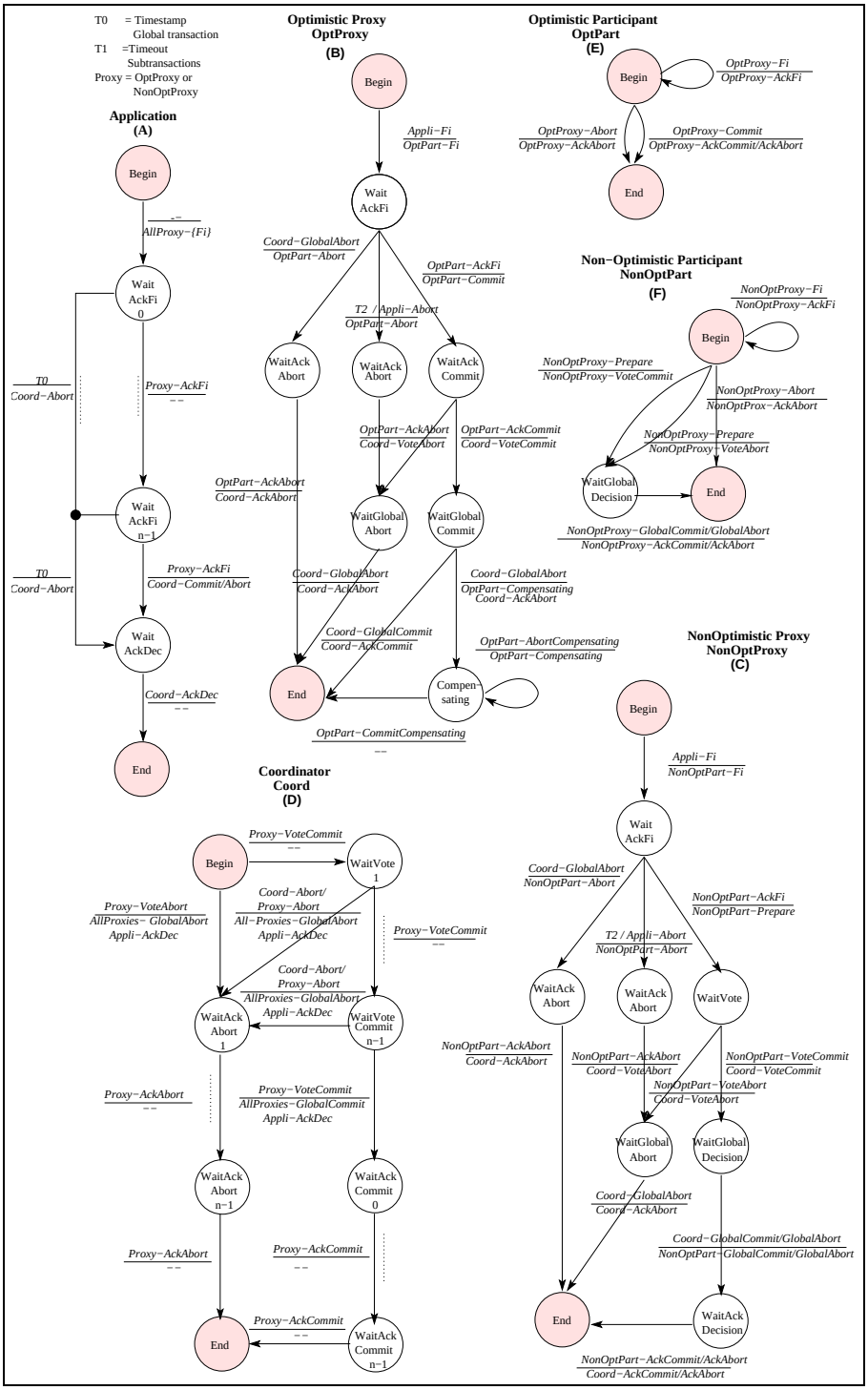

Figure 4. *Diagramme d'états de CO2PC*

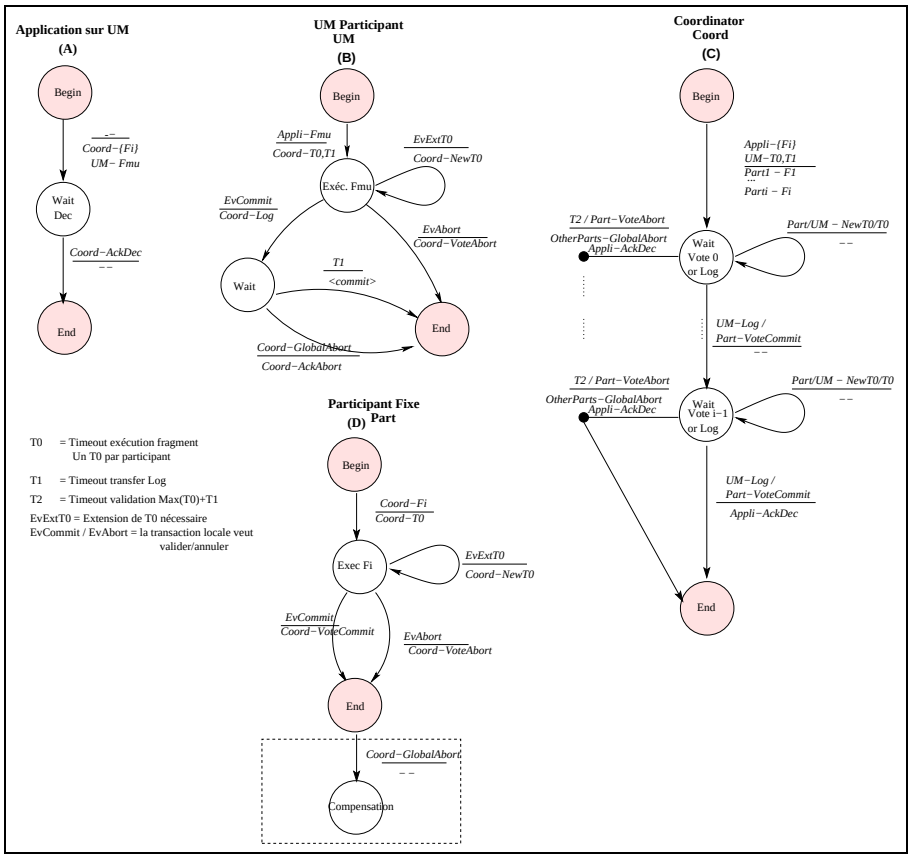

Figure 5. *Diagramme d'états de TCOT*

5. Protocole TCOT

Description et propriétés

TCOT a été conçu pour des transactions lancées par une unité mobile dont un fragment est exécuté sur l'unité mobile et d'autres sur des unités fixes. L'idée centrale de TCOT est l'utilisation de temporisateurs pour réduire les échanges de messages. Les participants fonctionnent de manière optimiste : ils valident/annulent la transaction locale indépendamment. Ceci impose que les fragments soient compensables. Pour que le protocole fonctionne correctement les auteurs de TCOT supposent une communication fiable (i.e. pas de perte de messages) [2]. Sous ces hypothèses, TCOT assure l'atomicité sémantique de la transaction.

TCOT distingue le participant mobile, un coordinateur et les participants fixes (voir fig. 5). Afin de faciliter notre travail de comparaison nous avons également dégagé

2. Hypothèse très forte pour un environnement mobile

l'application ((A) sur la figure 5) qui pour TCOT est localisée sur l'UM. Celle-ci envoie au coordinateur les fragments devant être exécutés sur les unités fixes et lance le fragment local à l'UM. Le coordinateur distribue les fragments.

Tous les participants (mobile et fixes) envoient au coordinateur la valeur de leur temporisateur (T0 sur la figure) qui correspond au temps estimé d'exécution de leur fragment. L'UM envoie de plus un deuxième temporisateur T1 car elle est tenue de transférer au coordinateur le journal des modifications faites par la transaction. T1 indique le temps (en plus de T0) pendant lequel le coordinateur attend ce journal. Si pendant l'exécution de leur fragment, un participant (mobile ou fixe) se rend compte qu'il ne terminera pas avant le T0 annoncé, il peut renvoyer au coordinateur une nouvelle valeur pour son T0. Cette extension de temps est décidée unilatéralement et n'est pas acquittée par le coordinateur.

Tous les participants peuvent annuler unilatéralement. Dans ce cas, ils envoient un message *VoteAbort* au coordinateur. Les participants fixes peuvent valider unilatéralement sans attendre et envoient un message *VoteCommit* au coordinateur. Le participant mobile valide différemment : à la fin de l'exécution de son fragment il envoie son journal au coordinateur et attend l'expiration de T1. Si pendant ce temps il ne reçoit pas de *GlobalAbort* de la part du coordinateur alors il valide localement. Si le *GlobalAbort* arrive avant l'expiration de T1 alors il annule localement.

De son côté le coordinateur envoie un *GlobalAbort* aux participants s'il reçoit un *VoteAbort* ou si au bout du temporisateur (Max(T0)+T1) il n'a pas reçu le journal de l'UM ou les *VoteCommit* des participants fixes ne sont pas tous arrivés.

Prise de décision globale à tort

Sous les hypothèses de communication fiable, sans délai de transfert de messages et sans panne, TCOT ne prend pas de décision globale à tort. Rappelons que TCOT n'inclut pas de message de décision de validation globale. Seul un message d'annulation globale est émis par le coordinateur. Si les participants ne reçoivent pas un tel message, ils valident.

Si, par contre, les hypothèses indiquées ne sont pas garanties, l'atomicité sémantique ne sera plus assurée.

Situations de blocage

Les hypothèses et les temporisateurs de TCOT évitent les situations de blocage. Le seul danger est lié à un temporisateur trop long décidé par un participant. Le coordinateur doit attendre l'expiration des temporisateurs.

6. Comparaison qualitative des protocoles

Les protocoles de validation analysés ici assurent différentes propriétés et imposent différents types de contraintes. Le compromis entre les propriétés et les contraintes donne lieu à différents types de protocoles. Cette section présente une discussion des protocoles analysés et plus particulièrement sur la propriété d'atomicité (ou atomicité sémantique), la tolérance aux pannes et le niveau d'autonomie imposé aux SGBD sous-jacents.

Atomicité.

2PC et UCM assurent l'atomicité. Pour cela, des contraintes très fortes sont imposées au système, par exemple, le blocage des données et le transfert d'un journal. CO2PC et TCOT ont été conçus pour limiter ce type de contraintes, en particulier, le temps de blocage. Ceci conduit à l'atomicité sémantique grâce à la validation optimiste et l'utilisation de transactions de compensation en cas de panne. CO2PC est suffisamment flexible pour offrir l'atomicité (et non juste l'atomicité sémantique) lorsque toutes les sous-transactions utilisées sont non-compensables. Dans ce cas, CO2PC se comporte comme une version modifiée de 2PC (avec un message en moins).

Tolérance aux pannes.

Afin d'éviter des annulations en cascade et pour assurer des histoires recouvrables, UCM et 2PC doivent être utilisés avec des protocoles de verrouillage (par exemple, 2PL [ESW 76]). De plus, UCM demande le transfert du journal de toutes les opérations de modification des transactions vers un support de stockage stable avant le début du processus de validation. Cela garantit la propriété de reprise des participants mobiles en cas de défaillance. De manière similaire, TCOT assure la tolérance aux pannes au niveau des sous-transactions. Pour cela, le journal des modifications doit être transféré vers un support stable avant le début d'une validation unilatérale. D'autre part, TCOT suppose que les participants fixes sont fiables mais pas UCM. En ayant comme objectif la limitation de l'utilisation du réseau sans fil, CO2PC ne demande aucun transfert de journal, en conséquence, il n'assure pas la récupération pendant l'exécution des sous-transactions. Cette propriété est déléguée aux SGBD sous-jacents. Ils sont libres d'utiliser la technique de récupération la plus appropriée, par exemple, faire le transfert du journal des modifications après la validation anticipée d'un certain nombre de sous-transactions ou lorsqu'une bonne communication sans fil est disponible. Au niveau global, CO2PC et TCOT assurent une récupération sémantique avec l'exécution de transactions de compensation.

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 le type d'information à partager avec le système global, l'indépendance d'exécution ainsi que la non imposition de changements au système [Ö 99]. 2PC est le protocole de validation le plus répandu. La quasi totalité des SGBD commerciaux implémente nativement son interface. Néanmoins ceci n'est pas le cas des SGBD commerciaux légers (par exemple, Oracle9i Lite [Ora 02], DB2 Everyplace [IBM 02]). Fournir l'état "préparation" peut être vu comme un relâchement de l'atomicité, cependant, comme il s'agit d'une interface largement implantée nous considérons que son utilisation n'affecte pas l'autonomie. UCM et TCOT n'imposent pas d'interface particulière. Néanmoins, TCOT affecte l'autonomie du fait qu'il demande un transfert de journal ce qui n'est pas le cas d'UCM puisque le transfert du journal est effectué par le *LogAgent*. CO2PC n'impose pas d'interface particulière

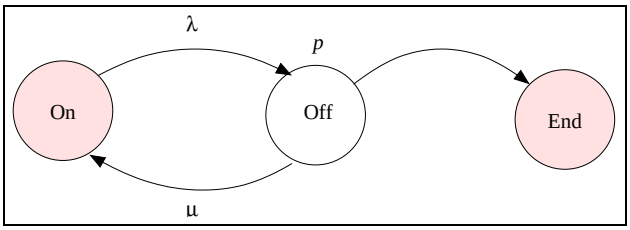

Figure 6. *Comportement des participants : automate à deux états On/Off.*

aux participants optimistes (ceux qui valident de manière anticipée), cependant, les participants non-optimistes doivent fournir l'interface de 2PC.

Type d'application.

Aussi bien 2PC que UCM sont utilisables pour un très large spectre d'applications. Ceci n'est pas le cas de TCOT, car il impose que toutes les transactions soient compensables. De son coté CO2PC propose plus de flexibilité en supportant des applications ayant des transactions compensables mais aussi non-compensables comme pour 2PC et UCM.

Par ailleurs, 2PC, UCM et CO2PC supportent un nombre théoriquement illimité de participants mobiles. Cependant, 2PC ne supporte pas de déconnexions, contrairement à UCM et CO2PC. De son coté TCOT n'est utilisable qu'avec un seul participant mobile.

7. Comparaison quantitative des protocoles

7.1. *Conditions de l'étude*

Notre étude compare certains aspects des performances des protocoles de validation.

Deux études différentes ont été menées. Dans la première (section 7.2) seule la phase de validation a été étudiée. Les résultats détaillés de cette étude, sans inclure TCOT, sont disponibles dans [BOB 04]. Dans ce cas, nous considérons le déroulement des protocoles à partir de l'envoi de la décision de l'application pour 2PC et UCM et à partir du départ du premier vote dans CO2PC. Pour CO2PC, cette démarche suppose que les participants sont, d'une certaine manière, synchronisés pour démarrer le protocole. Cette hypothèse simplificatrice est levée dans l'étude de la section 7.3 où le déroulement de la transaction dans son ensemble est pris en compte. L'objectif de ces deux études est d'identifier de façon précise les points forts et les points faibles des protocoles ainsi que leur provenance (phase d'exécution ou de validation).

Hypothèses réseaux

Nous considérons que les participants mobiles sont connectés au coordinateur par un canal de transmission plus ou moins fiable, et qu'ils peuvent aussi quitter définitivement le système sans prévenir le coordinateur. Le modèle choisi pour rendre compte

de cette situation est un automate à deux états On/Off (cf. fig.6). Le participant est connecté pendant un temps aléatoire distribué selon une loi exponentielle de paramètre λ. Le temps moyen de séjour dans l'état On est donc $\frac{1}{\lambda}$. Puis il reste déconnecté selon un temps aléatoire de distribution exponentielle de paramètre μ. Le temps moyen de séjour dans l'état Off est $\frac{1}{\mu}$. En sortant de l'état Off le participant peut soit quitter définitivement le système avec une probabilité p, soit retourner dans l'état On.

Lors des simulations, un délai constant est appliqué à toutes les transmissions pour refléter la latence du réseau. Un participant ne peut émettre de message que si la fenêtre de connexion (temps restant avant la prochaine déconnexion) est suffisamment grande pour le permettre.

Plate forme de simulation

Le comportement des différents protocoles a été simulé en utilisant la simulateur SimJava[SIM]. Simjava est un simulateur à événements discrets à base de processus pour java. Une simulation Simjava est un ensemble d'entités (nommées *Sim_entities*) s'exécutant chacune dans son propre processus. Ces entités sont connectées entre elles par des ports et peuvent communiquer les unes avec les autres en envoyant et en recevant des messages. Une classe *Sim_System* contrôle tous les processus, fait avancer l'horloge de la simulation et propage les évènements. L'évolution de la simulation est enregistrée dans un fichier grâce à des messages de trace envoyés par les entités.

Dans nos simulations un participant est représenté par deux entités. Une pour simuler le protocole de validation l'autre pour simuler le comportement du participant. Cette dernière utilise deux classes de générateurs aléatoires. *Sim_uniform_obj* est utilisée pour la génération de nombres aléatoires uniformes. Ceci pour décider quand le participant quitte définitivement le système. *Sim_negexp_obj* est utilisée pour générer des nombres distribués selon une loi exponentielle qui donnent les temps de séjour dans les états On et Off. L'entité qui simule le processus de validation du participant envoie les messages *vote*, *ack* et *leave* à l'entité utilisée pour simuler le coordinateur. Cette dernière entité collecte les évènements et calcule les statistiques sur les situations de blocage, le temps moyen de validation et les prises de décision à tort.

Indices de performance

L'étude est faite d'une part en fonction du nombre de participants mobiles, et d'autre part en fonction des caractéristiques de l'environnement.

Pour l'étude ne prenant en compte que la partie validation, deux indices de performances sont évalués :

– le temps moyen de validation (en unité de temps de SimJava),

– la probabilité de blocage du coordinateur.

Il faut noter que cette dernière probabilité dépend principalement de la valeur du temporisateur du coordinateur. Le choix de cette valeur est le résultat d'un arbitrage entre le temps moyen de validation et la probabilité de décision d'annulation à tort. En effet, un temporisateur court réduira le temps moyen de validation mais augmentera la pro-

babilité de décision d'annulation à tort. Pour toutes nos simulations les temporisateurs ont été fixés de manière relativement arbitraire en considérant un temps minimum pour l'envoi des messages requis plus une marge de sécurité de 50% de ce temps.

Pour l'étude prenant en compte tout le déroulement de la transaction les indices de performance évaluées sont :

– la durée moyenne de la transaction (notée T_{vu}) vue par l'application(unité SimJava),

– la durée moyenne de la transaction durée de validation comprise (avec les compensations si besoin) notée T_{tot},

– la probabilité Pb_{CoBLK} de blocage du coordinateur (2PC, UCM et CO2PC),

– la probabilité Pb_{PaBLK} de blocage d'un participant (2PC, CO2PC avec participants non-optimistes),

– la probabilité Pb_{ISW} de prise de décision d'annulation globale à tort (2PC et CO2PC avec participants non-optimistes),

– la probabilité Pb_{ab} d'annulation de la transaction.

Cette dernière probabilité a un sens car dans les simulations la transaction est toujours sensée valider. Les abandons proviennent uniquement des conditions réseaux et des dépassements de temporisateur. Dans ces simulations l'influence des messages du journal imposés par les protocoles UCM et TCOT est mesurable. Pour toutes les simulations, nous considérons que l'application est mobile et participe à la transaction (comme le présuppose le protocole TCOT).

7.2. *Mesures sur la phase de validation.*

Trois contextes ont été simulés de manière à représenter trois types d'environnements mobiles dégradés. Pour ces trois contextes la probabilité qu'un participant quitte définitivement le système est fixée à $p = 5\%$. Quand un participant n'a pas quitté définitivement le système, le premier contexte (noté *CTX1*) correspond à un environnement moyen[3] où le participant a 90% de chances d'être connecté (état On). Le deuxième (resp. troisième) contexte (noté *CTX2*, resp.*CTX3*) correspond à un environnement dégradé (resp. très dégradé) où la probabilité d'être dans l'état On est de 50% (resp. 10%).

Pour ces trois contextes, les trois indices de performances cités ont été mesurés en fonction du nombre de participants mobiles, de 0 à 10, sur 10 participants au total [4].

Probabilité de blocage

La fig. 7 montre la probabilité de blocage pour les trois environnements. Dans l'environnement moyen CTX1, les trois protocoles se comportent relativement bien. En effet, pour UCM et CO2PC ces probabilités sont inférieures à 0.2%. Par contre

3. Conditions déjà assez pessimistes
4. Nombre élevé pour une telle transaction

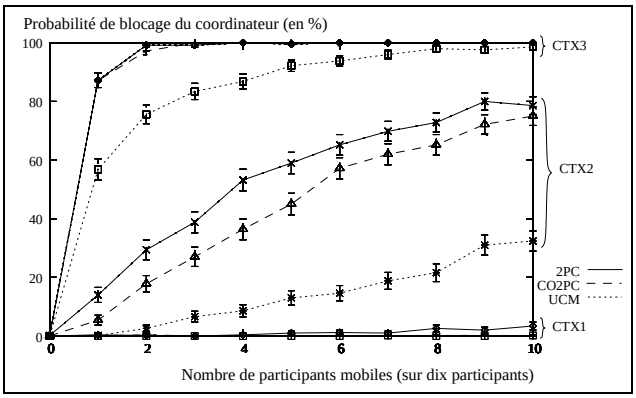

Figure 7. *Probabilité de blocage vs nombre de participants mobiles.*

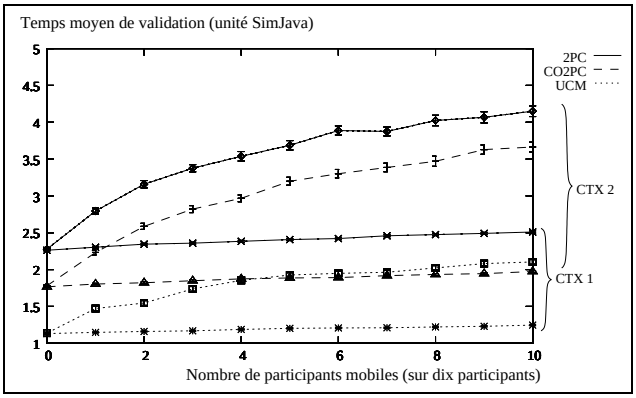

Figure 8. *Temps moyen de validation vs nombre de participants mobiles.*

pour 2PC elle est de l'ordre de 3% ce qui ne peut pas toujours être considéré comme négligeable. Dans les environnements très dégradés CTX2 et CTX3, les trois protocoles ne fonctionnent pas de manière correcte. Dans CXT2, seul UCM avec deux participants mobiles à une probabilité de blocage acceptable (autour de 3%). Dans CTX3 tous les protocoles ont des probabilités de blocage supérieur à 75% pour deux participants mobiles sur 10.

Temps moyen de validation

Dans CTX3 le nombre de validations qui restent bloquées est très important et font que l'estimation du temps moyen de validation a peu de sens. La fig. 8 présente le temps moyen de validation pour CTX1 et CTX2. Dans CTX1 le nombre de participants mobiles à peu d'influence sur le temps moyen de validation. Encore une fois on remarque donc que les trois protocoles supportent bien un environnement de qualité moyenne. On remarque aussi que le temps moyen de validation pour UCM est

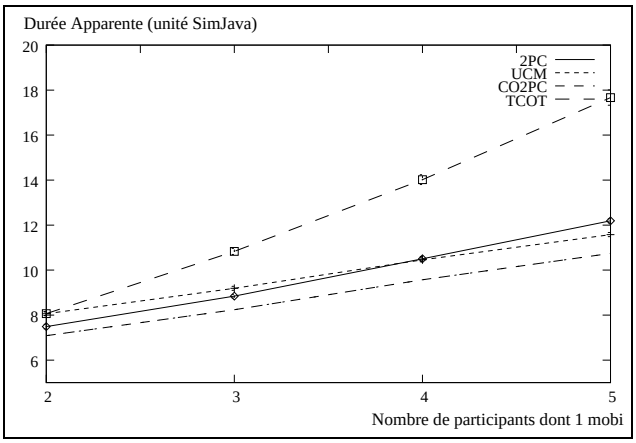

Figure 9. *T_{vu} vs nombre de participants dont 1 participant mobile.*

meilleur que celui de CO2PC qui est lui-même meilleur que celui de 2PC. Ceci est logique puisque le résultat des simulations ne dépend que du nombre de messages transitant sur le réseau mobile. Dans CTX2 les messages sont plus souvent en attente d'une fenêtre d'émission suffisante ce qui explique l'augmentation du temps moyen de validation.

7.3. *Mesures sur l'ensemble de la transaction*

Les simulations effectués sur CO2PC en faisant varier le nombre de participants optimistes et non-optimistes au sein d'une même transaction ont montré des performances très voisines. Ceci pour tous les indices de performances mesurés. C'est pourquoi dans la suite les courbes présentées ne différencient pas ces différents cas.

Variation du nombre de participants à la transaction

Dans un premier temps le cas d'un unique participant mobile (colocalisé avec l'application) est considéré.[5] Ce participant a accès au réseau 75% du temps et sa probabilité de quitter définitivement le système est de $2,5\%$.

Les fig. 9 et 10 montrent respectivement le temps moyen d'exécution T_{vu} vu par l'application et le temps total d'exécution T_{tot} en fonction du nombre de participants.

On remarque que pour TCOT ces deux temps peuvent être très différents surtout quand le nombre de participants est faible. Cette différence s'explique par le fait que le participant mobile doit attendre la fin de son temporisateur pour que la transaction et la validation soient terminées. Par contre pour les autres protocoles la différence entre les deux mesures est peu importante. Les trois protocoles UCM, 2PC et CO2PC ont des performances similaires avec un légère avantage à CO2PC.

5. Cas choisit car TCOT n'autorise qu'un seul participant mobile.

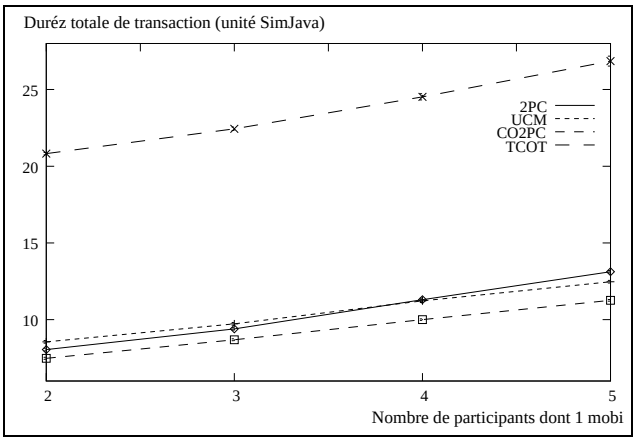

Figure 10. *T_{tot} vs nombre de participants dont 1 participant mobile.*

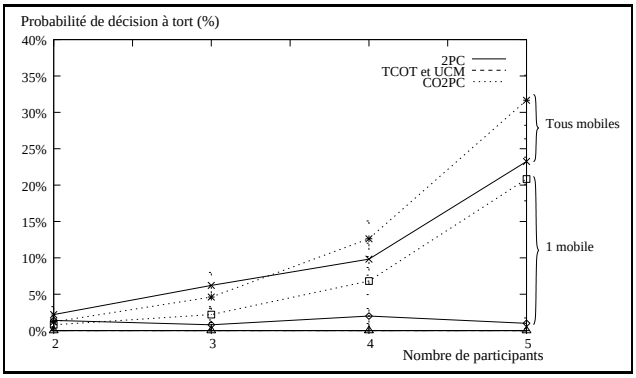

Figure 11. *Pb_{ISW} vs nombre de participants, avec un participant mobile et tous les participants mobiles.*

Les fig. 11 et 12 présentent respectivement les probabilités de décision d'annulation à tort Pb_{ISW} et d'abandon Pb_{ab}. Nous considérons un participant mobile de manière à inclure TCOT puis avec tous les participants mobiles. Dans ce dernier cas seul UCM, 2PC et CO2PC sont concernés. Pour UCM et TCOT Pb_{ISW} est de 0% puisque ces protocoles ne présentent pas de cas de décision d'annulation à tort. Pour les autres on peut remarquer que ces probabilités sont assez élevées variant de quelques pour cents (ce qui est déjà beaucoup) à plus de 30% pour les pires cas.

Variation du temporisateur du coordinateur

Cette série de mesures présente les indices de performances en fonction de la valeur choisie pour le temporisateur du coordinateur. Ces mesures ont été réalisées pour trois participants dont un mobile de manière à inclure TCOT à l'expérience. Les par-

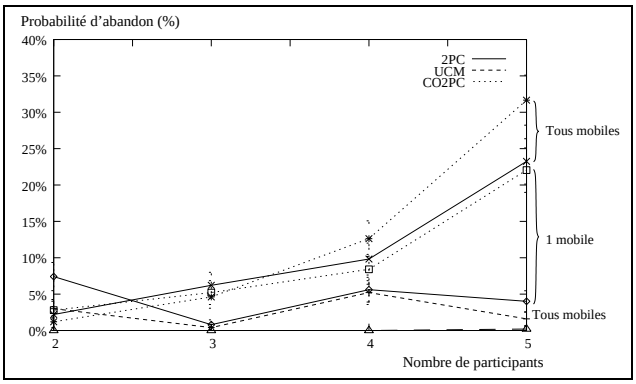

Figure 12. *Pb_{ab} vs nombre de participants, avec un participant mobile et tous les participants mobiles.*

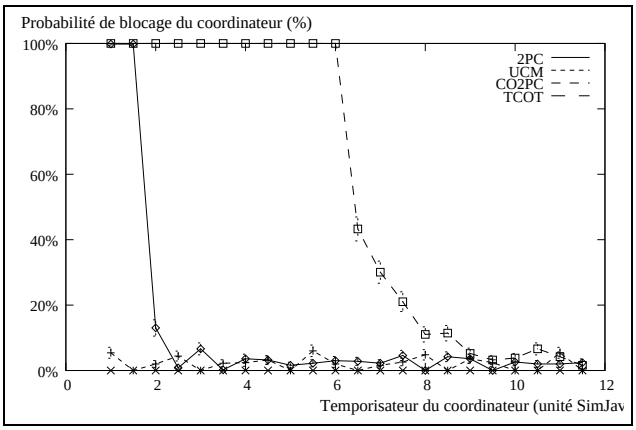

Figure 13. *Pb_{CoBLK} vs temporisateur du coordinateur.*

ticipants mobiles ont une probabilité de 75% d'être connectés avec une probabilité de 2.5% de quitter définitivement le système.

Les courbes de la probabilité de blocage d'un participants Pb_{PaBLK} ne sont pas présentées, en effet le nombre de blocage observés pour tous les protocoles n'est pas significatif. On peut en déduire que, pour le contexte réseau choisi les quatre protocoles étudiés ont de bonnes performances.

Les courbes sur les temps d'exécution ne sont pas présentées non plus. En effet, le choix de la valeur pour ce temporisateur ne joue pas sur le temps moyen d'exécution vu par l'application (T_{vu}). Il ne change pas non plus le temps moyen d'exécution total (T_{tot}) dans lequel nous n'avons pas inclus les cas d'annulation à tort. Si ces cas avaient été comptabilisés un grand temporisateur aurait rallongé le temps moyen de la validation et donc le temps total.

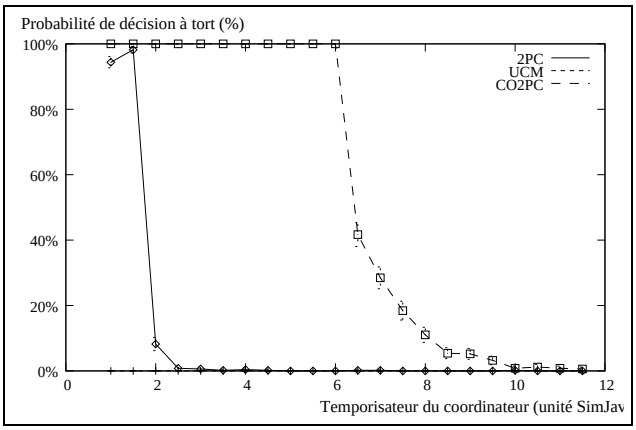

Figure 14. *Pb_{ISW} vs temporisateur du coordinateur.*

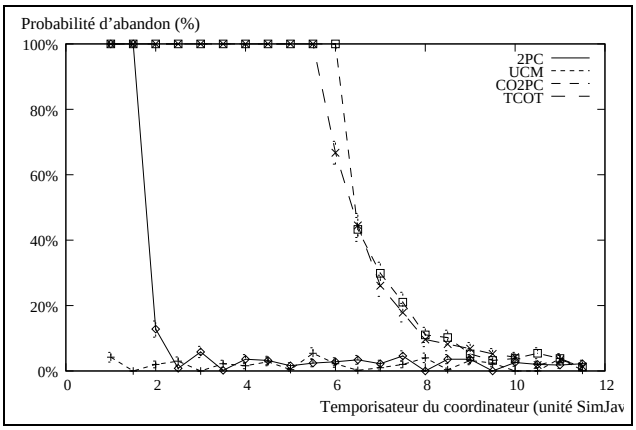

Figure 15. *Pb_{ab} vs temporisateur du coordinateur.*

La courbe 13 présente la probabilité de blocage du coordinateur (Pb_{CoBLK}). La courbe 14 présente la probabilité que le coordinateur des protocoles prenne une mauvaise décision (Pb_{ISW}). Et enfin la courbe 15 présente la probabilité d'abandon (Pb_{ab}). Ces courbes permettent pour chaque protocole de calibrer le temporisateur du coordinateur de manière à garantir une qualité de service souhaitée par une application. De manière générale, on remarque que ce temporisateur doit être beaucoup plus grand pour CO2PC que pour les autres protocoles. Ceci s'explique par le fait que dans CO2PC ce temporisateur débute dés le début de la transaction alors que pour les autres protocoles, il ne débute qu'au début de la phase de validation proprement dite.

8. Conclusion et perspectives

Nous avons présenté une étude des performances de certains aspects des protocoles de validation 2PC, UCM, CO2PC et TCOT. Cette étude a porté sur le processus de coordination de la validation de transactions réparties sur plusieurs unités mobiles et/ou fixes. 2PC et UCM assurent l'atomicité alors que CO2PC et TCOT assurent une atomicité sémantique. La comparaison des protocoles n'a pas pour objectif de faire ressortir un protocole comme étant *le meilleur*. En effet, 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. Cependant, nous nous sommes efforcés de dégager des indices de performance et des conditions d'exécution permettant certaines comparaisons sur le plan quantitatif : probabilité de blocage du coordinateur et des participants, durée moyenne de la transaction (perçue et réelle), probabilité de décision à tort et probabilité d'annulation induite par un contexte réseau peu favorable. La transaction a été considérée dans son ensemble, car certains protocoles de validation imposent des actions pendant l'exécution des transactions.

Nous avons considéré l'évaluation de la probabilité de blocage des participants, car cette situation peut entraîner le blocage des ressources. Théoriquement, ceci peut arriver pour certains protocoles étudiés, mais dans le contexte réseau simulé ces cas étaient trop rares pour être mesurés de façon fiable.

Nous constatons que malgré le fait que 2PC n'a pas été prévu pour des transactions impliquant des mobiles, ses performances ne sont pas mauvaises comparées à celles des autres protocoles étudiés. Néanmoins, il est le protocole qui nécessite le plus de messages et qui laisse le moins d'autonomie aux participants, ce qui pour les unités mobiles est pénalisant. A l'opposé, TCOT a une approche optimiste et laisse toute autonomie de validation aux participants. L'approche est basée sur des temporisateurs. Le choix des valeurs des temporisateurs a un fort impact sur les performances de ce protocole et peut engendrer des temps d'exécution importants s'ils sont choisis trop longs.

Le temps d'exécution global des transactions est très similaire pour 2PC et UCM (TCOT dépend beaucoup du choix des temporisateurs). Le temps d'UCM est pénalisé par le transfert du journal vers le coordinateur (unité fixe), alors que CO2PC est avantagé par un transfert non immédiat du journal. Néanmoins, ceci a un impact sur la tolérance aux fautes.

Plusieurs aspects doivent encore être approfondis, par exemple, le choix des valeurs des temporisateurs et leur éventuelle extension, l'impact de la taille des fragments de transaction et le coût de la reprise après panne.

Remerciements : nous tenons à remercier Stéphane Drapeau pour ses précieuses relectures.

9. Bibliographie

[BER 03] BERNARD G., BEN-OTHMAN J., BOUGANIM L., CANALS G., DEFUDE B., FERRIÉ J., GANÇARSKI S., GUERRAOUI R., MOLLI P., PUCHERAL P., RONCANCIO C.,

SERRANO-ALVARADO P., VALDURIEZ P., « Mobilité et Bases de Données : Etat de l'Art et Perspectives (Partie I et II) », *Chronique Technique et science informatiques (TSI)*, vol. 22, n° 3 and 4, 2003.

[BOB 02] BOBINEAU C., « Gestion de Transactions en Environnement Mobile », PhD thesis, Université de Versailles, France, December 2002.

[BOB 04] BOBINEAU C., LABBÉ C., RONCANCIO C., SERRANO-ALVARADO P., « Protocoles de validation pour transactions en environnements mobiles : première étude », *Mobilité et Ubiquité 2004*, France, Juin 2004.

[CUC 02] CUCE S., ZASLAVSKY A. B., HU B., RAMBHIA J., « Maintaining Consistency of Twin Transaction Model Using Mobility-Enabled Distributed File System Environment », *MDDS 02*, France, September 2002.

[ESW 76] ESWARN K. P., GRAY J., LORIE R. A., TRIGER I. L., « The Notions of Consistency and Predicate Locks in a Database System », *Communications of the ACM (CACM)*, vol. 19, n° 11, 1976.

[GRA 78] GRAY J., « Notes on Database Operating Systems , Operating Systems, an Advanced Cours », *LNCS 60, Springer-Verlag*, , 1978.

[IBM 02] IBM SOFTWARE PRODUCTS, « DB2 Everyplace http ://www-3.ibm.com/software/data/db2/everyplace/ », 2002.

[KUM 02] KUMAR V., PRABHU N., DUNHAM M. H., SEYDIM A. Y., « TCOT- A Timeout-Based Mobile Transaction Commitment Protocol », *IEEE Transactions on Computers*, vol. 51, n° 10, 2002.

[LEE 02] LEE M., HELAL A., « HiCoMo : High Commit Mobile Transactions », *Parallel and Distributed Database Systems*, vol. 11, n° 1, 2002.

[MAD 01] MADRIA S. K., BHARGAVA B., « A Transaction Model for Improving Data Availability in Mobile Computing », *Kluwer Academic Publishers Distributed and Parallel Databases (DAPD)*, vol. 10, n° 2, 2001.

[Ö 99] ÖZSU T., VALDURIEZ P., *Principles of Distributed Database Systems*, Prentice Hall, 2nd édition, 1999.

[Ora 02] ORACLE CORPORATION, « Oracle9i Lite : The Internet Platform For Mobile Computing http ://otn.oracle.com/products/lite/ », 2002.

[PIT 98] PITOURA E., SAMARAS G., *Data Management for Mobile Computing*, Kluwer Academic Publishers, 1998.

[PIT 03] PITOURA E., CHRISANTHIS P. K., RAMAMRITHAM K., « Characterizing the Temporal and Semantic Coherency of Broadcast-Based Data Dissemination », *Int. Conf. on Database Theory (ICDT)*, vol. 2572 de *LNCS*, Siena, Italy, January 2003.

[SER 04a] SERRANO-ALVARADO P., « Transactions Adaptables pour les Environnements Mobiles », PhD thesis, Université J. Fourier, France, February 2004.

[SER 04b] SERRANO-ALVARADO P., RONCANCIO C. L., ADIBA M., « A Survey of Mobile Transactions », *Int'l Journal on Distributed and Parallel Databases (DAPD)*, , à paraître 2004, Kluwer Academic Publishers.

[SIM] SIMJAVA, « www.dcs.ed.ac.uk/home/hase/simjava/ ».