

Aleksandra BIEŃKOWSKA
Uniwersytet Jagielloński, Instytut Informatyki

BADANIE SYMULACYJNE PROTOKOŁU ZATWIERDZANIA TRANSAKCJI MOBILNYCH

Streszczenie. Niniejszy artykuł jest poświęcony badaniom symulacyjnym protokołu zatwierdzania mobilnych transakcji, znanym jako *Transaction Commit on Timeout* (TCOT). Zostały w nim przedstawione wyniki symulacji poprawności działania protokołu TCOT, które przeprowadzono przy użyciu autorskiego programu, umożliwiającego symulację oraz wizualizację wykonania transakcji w środowisku mobilnym. Uzyskane wyniki badań symulacyjnych pozwoliły na wprowadzenie modyfikacji i usprawnienia protokołu TCOT.

Słowa kluczowe: protokół zatwierdzania transakcji mobilnych, transakcje mobilne, systemy mobilne, symulacja

SIMULATION RESEARCHES OF PROTOCOL FOR COMMITTING MOBILE TRANSACTIONS

Summary. The article treats with problems related to the *Transaction Commit on Timeout* protocol. The author created an application that enables simulation and visualisation of a protocol execution in the mobile system. The program was used to carry out simulations of TCOT execution for a number of events. On the basis of the simulations results several modifications of the protocol are recommended.

Keywords: mobile systems, transaction commit protocols, mobile transactions, simulation

1. Wprowadzenie

Systemy mobilne, opisane m.in. w monografii dotyczącej radiokomunikacji ruchomej [11], to jeden z najnowszych i najbardziej dynamicznie rozwijających się obszarów informatyki. W ostatnich latach nastąpił znaczny wzrost rynku zastosowań mobilnych oraz obszaru

badania naukowych i rozwojowych z nim związanych. Z tego względu coraz więcej uwagi poświęca się rozwiązaniom przeznaczonym dla środowisk mobilnych. Do istotnych zagadnień należą między innymi kwestie związane z działaniem mobilnych baz danych.

Systemy baz danych przeznaczone dla środowisk mobilnych, opisane m.in. w monografiach [5, 7, 10], wymagają uwzględnienia problemów związanych z mobilnością użytkowników oraz specyfiką środowiska mobilnego, takich jak: ograniczona jakość komunikacji pomiędzy jednostką mobilną a stacją bazową wynikająca z niskiej jakości łączy bezprzewodowych, ograniczone zasoby energii i pamięci w ruchomej jednostce (ang. *mobile unit*, MU), możliwość przejścia MU w stan tzw. „drzemki” lub rozłączenia MU z systemem w celu prowadzenia pracy w trybie bezpołączeniowym. W wyniku pojawienia się wymienionych problemów mogą wystąpić nieoczekiwane sytuacje, takie jak: przerwanie komunikacji ze stacją bazową przez MU, na przykład z powodu wyczerpania się baterii lub uszkodzenia MU, wyczerpanie się przestrzeni dyskowej w MU lub zakłócenia w połączeniu pomiędzy MU a stacją bazową. Z mobilnością uczestników wiąże się również niejednokrotnie konieczność rekonfiguracji początkowych ustawień systemu w czasie wykonywania transakcji, a także możliwość przejścia stacji ruchomej do sąsiedniej komórki (*handoff*).

Ze względu na powyższe czynniki, specjalnego podejścia wymagają kwestie związane z przetwarzaniem zapytań w transakcjach, w szczególności protokoły zatwierdzania transakcji. Transakcje takie, określane mianem transakcji mobilnych, są realizowane w odmienny sposób niż w statycznych środowiskach rozproszonych. W ostatnich latach zaproponowano wiele protokołów zatwierdzania transakcji dla systemów mobilnych, między innymi zmodyfikowany protokół *Two Phase Commit* (2PC) [1] i *Transaction Commit on Timeout* (TCOT) [7]. Opisy działania wielu innych protokołów zatwierdzania transakcji można znaleźć w pracach [3, 9]. Formalne podejście, dotyczące zatwierdzania transakcji, podano m.in. w publikacji [6].

Celem niniejszej pracy jest symulacyjne badanie podstawowych własności protokołu TCOT. Jedną z najważniejszych jest wykorzystanie parametru *timeout* w celu weryfikacji poprawności wykonania poszczególnych podtransakcji mobilnych. Przeprowadzone badania symulacyjne poświęcono sprawdzeniu poprawności zatwierdzania transakcji mobilnych w protokole TCOT, w tym przede wszystkim zachowania atomowości transakcji globalnej oraz zachowania spójności danych w systemie mobilnym.

W kolejnych rozdziałach podano podstawowe pojęcia przedstawione w pracy. W trzecim rozdziale opisano protokół TCOT dla zatwierdzania transakcji. Czwarty rozdział prezentuje uzyskane wyniki przeprowadzonych badań symulacyjnych. Wnioski dotyczące usprawnienia protokołu TCOT przedstawiono w rozdziale 5. Pracę podsumowano w rozdziale szóstym.

2. Charakterystyka protokołów w systemach mobilnych

Mobilny System Baz Danych (ang. *Mobile Database System*, MDS) to rozproszony system złożony z węzłów, na których odbywa się przetwarzanie danych, oparty na systemie telefonii komórkowej [1, 11]. System mobilny od tradycyjnego systemu rozproszonego różni się tym, że użytkownik zmienia swoje położenie, co powoduje ciągłą zmianę topologii sieci. Elementami mobilnego systemu baz danych są węzły stacjonarne oraz węzły mobilne. Do węzłów stacjonarnych zalicza się stacje bazowe (ang. *base transceiver station*, BTS), stacjonarne serwery baz danych (DBS) oraz klientów stacjonarnych (ang. *fixed host*, FH). Element mobilny to jednostka ruchoma (MU). Węzły stacjonarne są połączone między sobą szybkimi łączami stacjonarnymi (np. przewodowymi, satelitarnymi). Węzły mobilne łączą się ze stacjonarnymi elementami sieci poprzez łącza bezprzewodowe za pośrednictwem węzłów bazowych.

Podobnie jak w klasycznym systemie rozproszonym, również tutaj występuje całkowity brak wspólnej pamięci, każdy węzeł dysponuje własną pamięcią lokalną. Przetwarzanie rozproszone, jakie ma miejsce w tym systemie, charakteryzuje również całkowity asynchronizm. Komunikacja między węzłami i przekazywanie informacji odbywa się tylko na zasadzie wymiany wiadomości. W systemie mobilnym nie istnieje globalny zegar, podejmuje się jedynie próby synchronizacji zegarów kwarcowych, zainstalowanych w komputerach mobilnych.

Mobilna transakcja rozproszona (*mobilna transakcja globalna*) to mobilna transakcja, której polecenia DML odwołują się do tabel, znajdujących się w co najmniej dwóch węzłach rozproszonej bazy danych. Mobilna transakcja rozproszona składa się ze zbioru *transakcji lokalnych*. W każdej z baz danych, do której odwołuje się mobilna transakcja rozproszona, jest tworzona jedna transakcja lokalna. Zarówno każda z transakcji lokalnych, jak również transakcja rozproszona powinny posiadać cechy trwałości, spójności, izolacji i atomowości. W transakcji realizowanej w systemie mobilnym biorą udział uczestnicy systemu mobilnego, tacy jak DBS, MU, BTS.

Uwzględniając ograniczenia wynikające ze specyfiki systemów mobilnych, wyróżniono zasadnicze cechy, które powinien posiadać protokół zatwierdzania transakcji, a mianowicie:

- generować wymianę jak najmniejszej liczby komunikatów w sieci bezprzewodowej, przez co zmniejsza się obciążenie systemu komunikatami,
- ryzyko utraty komunikatu w zawodnej sieci bezprzewodowej,
- podobnie jak w każdym systemie rozproszonym, również w systemie mobilnym protokół taki musi zapewniać atomowość zatwierdzania transakcji oraz spójność danych w systemie po jej zakończeniu,
- być nieblokujący, czyli umożliwiać każdemu uczestnikowi niezależne zakończenia transakcji bez konieczności oczekiwania na wynik transakcji w innych węzłach, w szczegó-

ności konieczności oczekiwania na odzyskanie (naprawę) węzła uszkodzonego, co zapobiega zakleszczeniom w systemie (nieskończonemu oczekiwaniu na odpowiedź od innego uczestnika transakcji) oraz ogranicza liczbę wysyłanych komunikatów.

Zastosowanie protokołów używanych w systemach rozproszonych, takich jak 2PC lub 3PC jest niewystarczającym rozwiązaniem w systemie mobilnym, ze względu na występujące w nim ograniczenia w komunikacji. Dla przykładu, 2PC wymaga trzech faz wymiany komunikatów pomiędzy uczestnikami w przypadku poprawnego wykonania transakcji i pięciu faz komunikacji w przypadku zaistnienia awarii. W systemie mobilnym taka ilość przesyłanych komunikatów może okazać się zbyt duża. Zakłócenia w komunikacji mogą powodować utratę dużej ilości komunikatów i wielokrotne, niepotrzebne wycofywanie transakcji (ang. *transaction abort*). Ponadto, protokoły te mają właściwość blokowania zasobów, co wyklucza ich zastosowanie w środowisku mobilnym.

3. Protokół TCOT dla zatwierdzania transakcji mobilnych

W zaprezentowanym w pracy [7] protokole TCOT zaproponowano rozwiązania, uwzględniające powyższe wymagania dla protokołów w systemach mobilnych.

W protokole zatwierdzania transakcji globalnej w systemie mobilnym TCOT bierze udział kilka rodzajów uczestników mobilnego systemu baz danych (MDS). Uczestnik każdego typu pełni ściśle określoną funkcję. Wyróżnia się następujące rodzaje funkcji:

Koordinator (CO) – jego funkcję pełni stacja bazowa (BTS). Zadaniem koordynatora jest zarządzanie wykonaniem transakcji globalnej. Odpowiada on za jej atomowe zatwierdzenie lub wycofanie.

Uczestnik statyczny – serwer baz danych (DBS), do którego są kierowane przez koordynatora żądania wykonania transakcji lokalnej.

Uczestnik mobilny – jednostka mobilna (MU), w czasie wykonywania transakcji może zmieniać swoje położenie. Inicjuje transakcję globalną, komunikując się z koordynatorem. W MU jest przechowywana lokalna kopia (*cache*) bazy danych znajdującej się na serwerze baz danych (DBS). Po wykonaniu transakcji lokalnej log zmian w *cache*'u jest przesyłany do koordynatora.

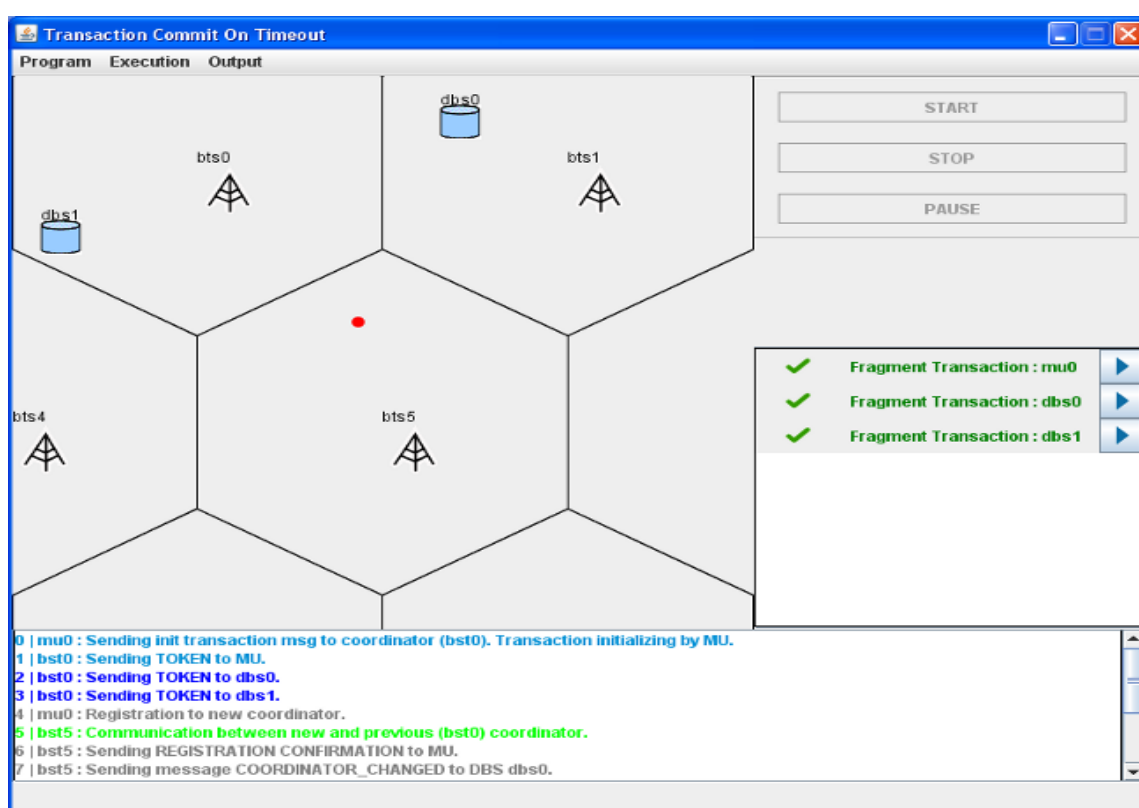
Dokładny opis wykonania protokołu TCOT został przedstawiony w pracy [7].

Dzięki wykorzystaniu w protokole parametru *timeout*, zarówno w przypadku uczestników statycznych, jak również mobilnych, decyzja o zatwierdzeniu bądź wycofaniu transakcji lokalnej jest podejmowana niezależnie, co pozwala na uniknięcie blokowania w systemie.

Podczas wykonywania protokołu przebieg transakcji realizowanych przez koordynatora i uczestników zbioru zatwierdzania jest zapisywany w dziennikach transakcji (logach). Koordy-

nator przechowuje log wykonania transakcji globalnej. Log przechowywany w MU po zakończeniu transakcji jest przesyłany do koordynatora, który następnie kieruje go do odpowiednich DBS w celu aktualizacji danych w bazach na serwerach. Pozwala to na zachowanie globalnej spójności w systemie. Dodatkowo, umieszczenie logu transakcji w bardziej bezpiecznej, stacjonarnej części systemu umożliwia odtworzenie danych zawartych w *cache*'u MU w razie awarii.

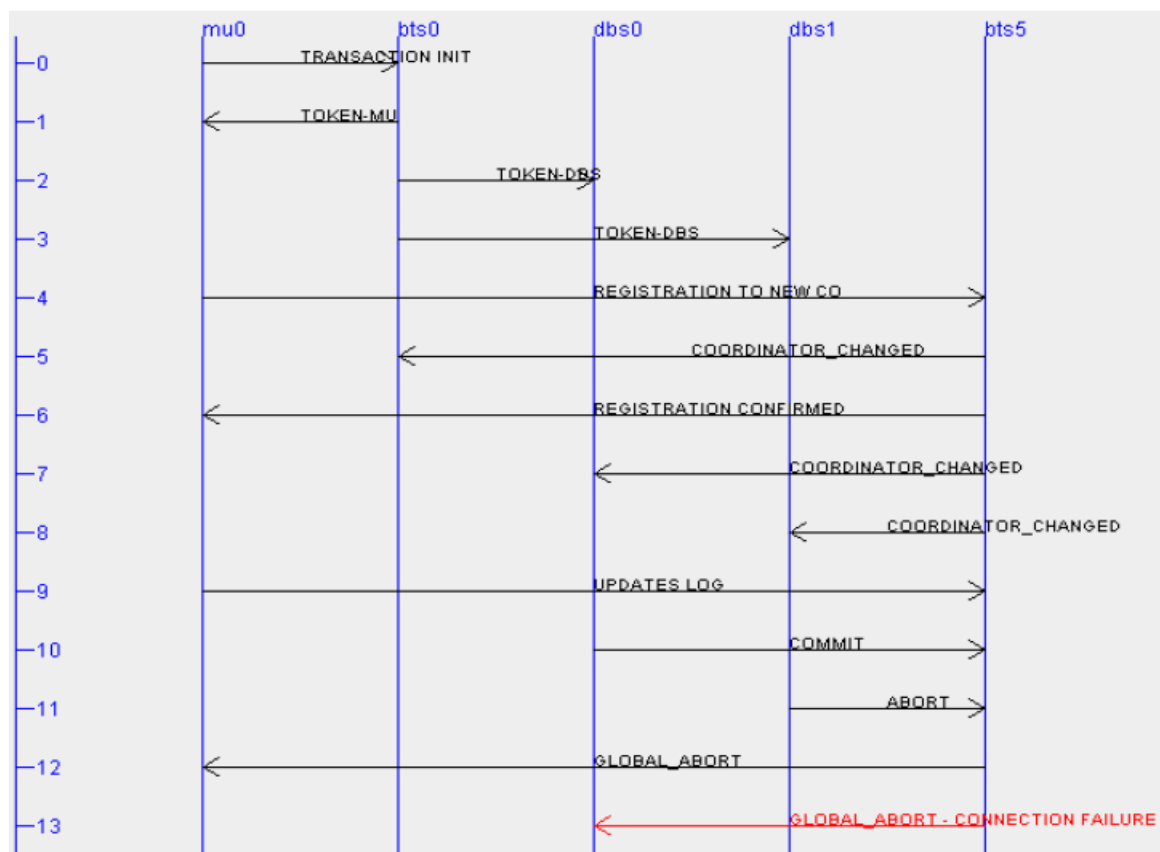
W kontekście omawianego protokołu na uwagę zasługuje również zjawisko przenoszenia połączenia. Ponieważ duża odległość pomiędzy koordynatorem a MU może obniżać jakość połączenia, przyjęto tzw. migrujący model koordynacji; w przypadku wystąpienia *handoff*'u przeprowadza się zmianę koordynatora transakcji.



Rys. 1. Graficzny interfejs użytkownika
Fig. 1. Graphical user interface

4. Badanie symulacyjne protokołu TCOT

W celu przeprowadzenia symulacji wykonania tego protokołu utworzono aplikację, umożliwiającą analizę przebiegu jego kolejnych kroków. W stworzonej aplikacji zaimplementowano protokół TCOT oraz wprowadzono funkcjonalność, umożliwiającą tekstową i graficzną wizualizację jego wykonania. Graficzny interfejs użytkownika przedstawia rys. 1.



Rys. 2. Przykład linii czasowej pojedynczego przebiegu symulacji
 Fig. 2. An example of run time of single simulation

Napisany program umożliwia m.in. podgląd zawartości lokalnych baz danych po wykonaniu symulacji, a także wyświetlenie przepływu komunikatów pomiędzy uczestnikami systemu w czasie wykonywania protokołu. Przykład linii czasowej uzyskanej w wyniku pojedynczego przebiegu symulacji przedstawia rys. 2, natomiast przykładowe kroki protokołu TCOT pokazano na rys. 3.

Zadaniem stworzonego programu jest umożliwienie określenia warunków działania protokołu w postaci parametrów oraz symulacja działania protokołu przy zadanych parametrach umożliwiającą analizę przebiegu jego wykonania w różnych sytuacjach działania systemu.

W aplikacji przyjęto dodatkowe założenia:

- w symulowanym systemie występuje tylko jedna jednostka mobilna (MU), pewna liczba BTS, obliczana podczas działania programu oraz pewna zdefiniowana przez użytkownika liczba serwerów baz danych DBS n ($n \geq 1$),
- w czasie działania symulacji jest wykonywana jedna transakcja globalna (nie są rozpatrywane zagadnienia związane z izolacją transakcji i współbieżnością podczas działania transakcji współbieżnych).

Protocol steps	
0	mu0 : Sending init transaction msg to coordinator (bts0). Transaction initializing by MU.
1	bts0 : Failure while sending TOKEN to MU.
2	bts0 : Sending TOKEN to dbs0.
3	bts0 : Sending TOKEN to dbs1.
4	dbs0 : Fragment transaction succeded in dbs0.
5	dbs1 : Fragment transaction succeded in dbs1.
6	dbs0 : Sending COMMIT message to coordinator bts0.
7	dbs1 : Sending COMMIT message to coordinator bts0.
8	mu0 : Sending ABORT message to coordinator bts0.
9	bts0 : Sending GLOBAL_ABORT message to dbs0.
10	bts0 : Sending GLOBAL_ABORT message to dbs1.
11	dbs0 : OPERATIONS COMPENSATION - TRANSACTION DISCARDED.
12	dbs1 : OPERATIONS COMPENSATION - TRANSACTION DISCARDED.

Rys. 3. Przykładowe kroki protokołu TCOT
 Fig. 3. An example of TCOT protocol steps

Przy użyciu utworzonego programu przeprowadzono symulację wykonania protokołu w przypadku wystąpienia różnych zdarzeń w systemie, specyficznych dla środowiska mobilnego. Wyniki symulacji pozwoliły na dokładne zaobserwowanie zachowania omawianego protokołu w różnych sytuacjach oraz ukazały ważne jego cechy.

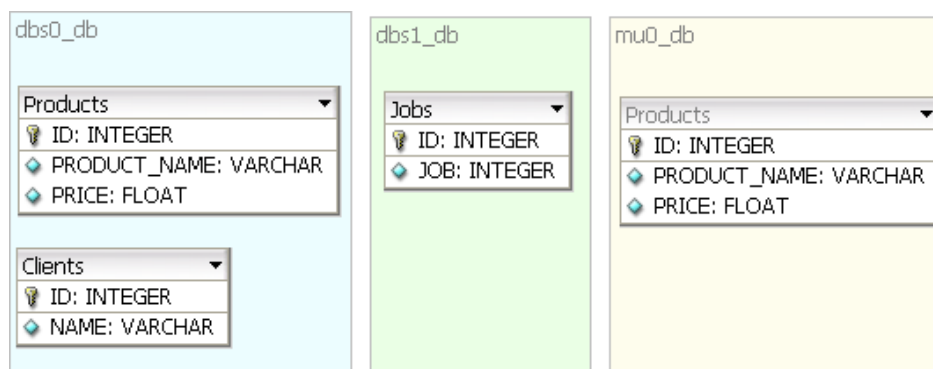
Dla celów symulacji zaprojektowano system złożony z jednej jednostki mobilnej o nazwie *mu0*, dwóch węzłów DBS (*dbs0* i *dbs1*) oraz wielu stacji BTS (*bts0*, *bts1*...*bts_n*).

Każdy z serwerów *dbs0* i *dbs1* zawiera jedną bazę danych (odpowiednio *dbs0_db* i *dbs1_db*).

W bazie *dbs0_db* na serwerze *dbs0* znajdują się dwie tabele o nazwach: *Products* oraz *Clients*, a w bazie *dbs1_db* na serwerze *dbs1* znajduje się jedna tabela o nazwie *Jobs*.

Jednostka mobilna zawiera *cache* o nazwie *mu0_db* fragmentu bazy danych znajdującej się na serwerze *dbs0*. *Cache* zawiera kopię tabeli *Products* z bazy *dbs0_db*.

Strukturę mobilnej bazy danych przedstawiono na rys. 4.



Rys. 4. Struktura mobilnej bazy danych
 Fig. 4. Structure of mobile database

Transakcja globalna zdefiniowana w parametrze *transaction* dzieli się na trzy transakcje lokalne przeznaczone do wykonania przez uczestników symulacji:

- Fragment transakcji wykonywany przez mu0: wstawienie do tabeli *Products* wiersza (1, computer, 5000).
- Fragment transakcji wykonywany przez dbs0: wstawienie do tabeli *Clients* wiersza (1, Kowalski).
- Fragment transakcji wykonywany przez dbs1: wstawienie do tabeli *Jobs* dwóch wierszy: (1, assistant) oraz (2, manager).

Poniższy scenariusz przedstawia przykład przebiegu wykonania protokołu TCOT w sytuacji, gdy w systemie nie wystąpiła awaria podczas działania protokołu oraz wszystkie podtransakcje zostały wykonane poprawnie:

- MU wysyła do koordynatora żądanie rozpoczęcia transakcji (TRANSACTION INIT),
- koordynator wysyła tokeny do uczestników transakcji,
- po otrzymaniu tokenu każdy z uczestników rozpoczyna wykonywanie transakcji lokalnej,
- po wykonaniu transakcji serwery DBS wysyłają komunikat COMMIT do CO, natomiast MU wysyła log dokonanych zmian w komunikacie UPDATES LOG,
- po upływie czasu oczekiwania na ewentualny GLOBAL ABORT, uczestnicy transakcji uznają transakcję za zatwierdzoną,
- aktualizacja *cache*'u MU zostaje przesłana do serwerów DBS,
- w przypadku poprawnego wykonania protokołu po wykonaniu transakcji globalnej w lokalnych bazach danych u wszystkich uczestników systemu znajdują się poprawne i spójne dane. Atomowość i spójność transakcji zostają zachowane.

Jeżeli transakcja lokalna u jednego z uczestników się nie powiodła, wysyła on do koordynatora komunikat ABORT i anuluje lokalną transakcję. Po otrzymaniu komunikatu ABORT koordynator anuluje transakcję globalną i wysyła komunikat GLOBAL ABORT do tych uczestników, którzy nadesłali COMMIT. Po otrzymaniu komunikatu GLOBAL ABORT uczestnicy wycofują transakcje lokalne.

W tym przypadku po zakończeniu transakcji globalnej tabele w bazach danych u uczestników są puste. Transakcja globalna nie zostaje zatwierdzona, wszystkie podtransakcje zostają wycofane, zostaje zachowana spójność danych i atomowość transakcji.

W tabeli 1 zostały przedstawione wyniki symulacji przeprowadzonych dla wybranych przypadków awarii systemu mobilnego.

Tabela 1

Wyniki symulacji dla wybranych przypadków awarii system mobilnego

Przypadek awarii	Zdarzenia wyjątkowe w przebiegu wykonania protokołu	Wyniki wykonania protokołu
A. Awaria na łączach podczas próby wysłania tokenu przez koordynatora do jednostki mobilnej	Podczas wysyłania tokenu do MU następuje awaria łącza, token nie dociera do MU. Kiedy w jednostce mobilnej czas oczekiwania na token zostanie przekroczony, MU wysyła komunikat ABORT do koordynatora (MU nie ponawia żądania wysłania tokenu). Po otrzymaniu komunikatu ABORT od MU koordynator wysyła komunikat GLOBAL ABORT do wszystkich uczestników transakcji, którzy nadesłali COMMIT (dbs0 i dbs1). Po otrzymaniu komunikatu GLOBAL ABORT serwery dbs0 oraz dbs1 wycofują transakcje lokalne oraz dokonują kompensacji wykonanych przez nie zmian w lokalnych bazach danych.	W wyniku wykonania powyższych operacji tabele w bazach danych u wszystkich uczestników pozostają puste. Transakcja globalna nie zostaje zatwierdzona, ale atomowość transakcji i spójność danych we wszystkich węzłach zostaje zachowana.
B. Awaria koordynatora po wysłaniu tokenów do uczestników transakcji	Po wysłaniu tokenów następuje awaria koordynatora. Komunikaty UPDATES LOG oraz COMMIT nie zostają obsłużone, globalna transakcja nie zostaje zatwierdzona. Uczestnicy transakcji nie otrzymują w określonym czasie komunikatu GLOBAL ABORT i uznają transakcję za zatwierdzoną.	Atomowość transakcji nie zostanie naruszona, ponieważ wszystkie lokalne transakcje zostały wykonane. Może jednak powstać w systemie niespójność danych, ponieważ aktualizacje <i>cache</i> 'u MU nie zostają przekazane przez CO do serwera dbs0.
C. Przypadek awarii łącza podczas wysyłania przez jednostkę mobilną (MU) komunikatu UPDATES LOG do koordynatora	Podczas wysyłania komunikatu UPDATES LOG przez MU następuje awaria łącza i komunikat nie dociera do koordynatora. Ponieważ CO nie otrzymuje żadnego komunikatu od MU w określonym czasie (następuje <i>timeout</i>), to nie zatwierdza transakcji globalnej i wysyła komunikat GLOBAL ABORT do wszystkich uczestników transakcji, którzy nadesłali COMMIT (w tym przypadku do serwerów dbs0 i dbs1). Po odebraniu komunikatu GLOBAL ABORT serwery dbs0 i dbs1 wycofują transakcje lokalne.	W systemie występuje niespójność danych. Tabela <i>Products</i> w <i>cache</i> 'u MU zawiera jeden wpis. Tabele na serwerach dbs0 i dbs1 są puste. Naruszona zostaje również atomowość transakcji globalnej, ponieważ w MU nie została wycofana transakcja lokalna.

cd. tabeli 1

<p>D. Przypadek awarii łącza podczas wysyłania komunikatu GLOBAL ABORT przez koordynatora do jednostki mobilnej</p>	<p>Serwer dbs1, w którym jedna z operacji się nie powiodła, wysyła do koordynatora komunikat ABORT i anuluje lokalną transakcję. Po otrzymaniu komunikatu ABORT koordynator anuluje transakcję globalną i wysyła do tych uczestników, którzy nadesłali COMMIT (dbs0 i mu0) komunikat GLOBAL ABORT. Podczas wysyłania komunikatu GLOBAL ABORT do MU następuje awaria łącza i komunikat nie dociera do jednostki mobilnej. Po upływie czasu oczekiwania na ewentualny komunikat GLOBAL ABORT w MU transakcja zostaje uznana za zatwierdzoną.</p>	<p>Po zakończeniu transakcji globalnej tabela <i>Products</i> w <i>cache</i>'u MU zawiera jeden wpis. Tabele na serwerach dbs0 i dbs1 są puste. W systemie występuje niespójność danych. Naruszona zostaje również atomowość transakcji globalnej, ponieważ nie u wszystkich uczestników została wycofana transakcja lokalna.</p>
<p>E. Przypadek awarii łącza podczas wysyłania do koordynatora komunikatu ABORT przez serwer baz danych (DBS)</p>	<p>W węźle dbs1 transakcja lokalna nie powiodła się, zostaje wycofana, a do koordynatora zostaje wysłany komunikat ABORT. Podczas wysyłania komunikatu ABORT przez serwer dbs1 następuje awaria łącza i komunikat nie dociera do CO. Ponieważ CO nie otrzymuje żadnego komunikatu od dbs1 w określonym czasie (<i>timeout</i>), nie zatwierdza transakcji globalnej, wysyła komunikat GLOBAL ABORT do wszystkich uczestników transakcji, którzy nadesłali COMMIT lub UPDATES LOG (w tym przypadku do mu0 i dbs0) Po odebraniu komunikatu GLOBAL ABORT uczestnicy mu0 i dbs0 wycofują transakcje lokalne.</p>	<p>Po zakończeniu transakcji globalnej tabele w bazach danych u uczestników są puste. Spójność danych w węzłach nie została naruszona. Transakcja globalna nie została zatwierdzona, wszystkie podtransakcje zostały wycofane, została zachowana atomowość transakcji.</p>

Na podstawie dokonanych obserwacji zostały zaproponowane modyfikacje i usprawnienia protokołu TCOT.

5. Wnioski i propozycje usprawnienia protokołu TCOT

Jakkolwiek z teoretycznego punktu widzenia globalna decyzja o zatwierdzeniu lub odrzuceniu transakcji jest podejmowana w poprawny sposób, to w wyniku wystąpienia pewnych zdarzeń, po zakończeniu wykonania protokołu dane w systemie mogą znajdować się w niespójnym stanie.

Poniżej przedstawiono analizę wyników symulacji opisanych w poprzednim rozdziale, ukazujących sytuacje, w których protokół działa niepoprawnie, oraz adekwatne do omawianych przypadków propozycje usprawnień protokołu.

5.1. Analiza przypadku B (awaria koordynatora po wysłaniu tokenów do uczestników transakcji)

W przypadku awarii koordynatora po wysłaniu tokenów do (jednego lub wielu) uczestników transakcji, niespójność danych w węzłach powstaje na skutek niedostarczenia przez koordynatora aktualizacji *cache*'u MU do serwera db0.

Propozycją rozwiązania tego problemu może być nawiązanie komunikacji z koordynatorem przez DBS w celu wysłania dodatkowego komunikatu informującego o nieotrzymaniu komunikatu UPDATES LOG. Po usunięciu awarii koordynator mógłby ponownie nawiązać komunikację z MU i przekazać do serwera DBS logi aktualizacji.

5.2. Analiza przypadku C (awaria łącza podczas wysyłania przez jednostkę mobilną do koordynatora komunikatu UPDATES LOG)

Ponieważ komunikat UPDATES LOG od jednostki mobilnej nie dociera do koordynatora, CO uznaje brak wiadomości za oznakę awarii jednostki mobilnej i nie wysyła do niej komunikatu GLOBAL ABORT. Komunikat GLOBAL ABORT jest wysyłany jedynie do tych uczestników, którzy wysłali COMMIT (lub UPDATES LOG). Tymczasem jednostka mobilna zatwierdza transakcję lokalną, ponieważ nie otrzymuje komunikatu GLOBAL ABORT. Gdyby komunikat ten dotarł do MU, nie wystąpiłaby niespójność danych w systemie.

Usprawnieniem protokołu, które redukowałoby wystąpienie sytuacji tego rodzaju, mogłoby być wysyłanie przez koordynatora (w przypadku podjęcia decyzji o zakończeniu transakcji globalnej) komunikatu GLOBAL ABORT do wszystkich uczestników, bez względu na rodzaj otrzymanych od nich komunikatów.

5.3. Analiza przypadku D (awaria łącza podczas wysyłania komunikatu GLOBAL ABORT przez CO do jednostki mobilnej)

W tym przypadku niespójność danych w systemie po zakończeniu działania protokołu jest konsekwencją przyjętego założenia, że w jednostce mobilnej nieotrzymanie komunikatu GLOBAL ABORT od koordynatora po upływie określonego czasu od wysłania UPDATES LOG jest równoznaczne z zatwierdzeniem transakcji.

Propozycją modyfikacji protokołu mogłoby być wysyłanie przez CO do uczestników transakcji dodatkowego komunikatu GLOBAL COMMIT w czasie E_t potwierdzającego zatwierdzenie transakcji globalnej. Wówczas nieotrzymanie jakiegokolwiek komunikatu od CO byłoby jednoznacznie interpretowane przez uczestnika jako brak zatwierdzenia transakcji.

6. Podsumowanie

W niniejszym artykule przedstawiono sposób wykonania protokołu TCOT. Zostały zaprezentowane wyniki symulacji przeprowadzonych przy użyciu utworzonej w tym celu aplikacji.

Analiza wyników symulacji działania protokołu TCOT w różnych przypadkach pozwala zauważyć, że w sytuacji wystąpienia niektórych rodzajów awarii działanie protokołu prowadzi do niespójności danych w systemie oraz naruszenia atomowości transakcji globalnej. Wystąpienie niespójności po zakończeniu działania protokołu w wielu przypadkach pojawia się w wyniku nieodebrania przez uczestnika informacji o niepowodzeniu transakcji globalnej w wyniku błędnej interpretacji braku komunikatu od koordynatora. Przyjęcie takiego założenia może być korzystne, jeżeli w systemie jest wycofywany niewielki odsetek transakcji globalnych (komunikaty GLOBAL ABORT są wysyłane rzadko), natomiast dane w jednostce mobilnej mogą być łatwo uaktualnione. Dzięki temu jest możliwe wcześniejsze zakończenie transakcji przez jednostkę mobilną oraz szybkie zwolnienie zasobów wykorzystywanych przez transakcję, co może mieć znaczenie w przypadku wykonywania przez MU dużej liczby transakcji.

Przeprowadzone symulacje ilustrują również własność nieblokowania protokołu TCOT. Dzięki zastosowaniu w protokole *timeout*'u w żadnym z powyższych przypadków nie doszło do zablokowania uczestnika w nieskończonej pętli oczekiwania. Jest to szczególnie ważna cecha protokołu w przypadku komunikacji w zawodnej sieci bezprzewodowej, gdzie komunikaty są często gubione i częste blokowanie utrudniałoby przeprowadzanie transakcji. Jak wykazuje analiza wyników symulacji, niejednokrotnie odbywa się ona kosztem braku zachowania atomowości transakcji globalnej i spójności danych w systemie. Ta własność protokołu może być akceptowana, pod warunkiem że występowanie takich sytuacji nie będzie powodować poważnych zakłóceń w działaniu systemu. W przeciwnym przypadku jest konieczne wprowadzenie odpowiednich mechanizmów kompensacji oraz strategii ich wykorzystania.

BIBLIOGRAFIA

1. Bernstein Ph. A., Hadzilacos V., Goodman N.: Concurrency Control and Recovery in Database Systems. Addison Wesley, Reading MA 1987.
2. Kumar Y., Dash K., Dunham M. H., Seydin A. Y.: A Timeout-Based Mobile Transaction Commitment Protocol. [in:] Proc. of ADBIS-DASEAA 2000, Advances in Database Systems and Information Systems, in cooperation with ACM SIGMOD, Prague, Czech Republic 2000.

3. Bobineau C., Pucheral P., Abdallah M.: A Unilateral Commit Protocol for Mobile and Disconnected Computing. [in:] Proceedings of the International Conference on Parallel and Distributed Computing Systems (PDCS), USA, August 2000.
4. Lin Y. W., Wu H. U.: Commit Protocol for Low-Powered Mobile Clients. IEICE Trans. on Information and System, Vol. E82-D, No. 8, 1999, s. 1167÷1179.
5. Imieliński T., Korth H. F.: Mobile Computing. Kluwer Academic Publishers, 1996.
6. Korth H. F., Levy E., Silberschatz A.: A formal approach to recovery by Compensating Transactions. [in:] Proceedings of the 16th VLDB Conference, Brisbane, Australia 1990.
7. Kumar V.: Mobile Database Systems. John Wiley and Sons, Hoboken 2006.
8. Nouali N., Doucet A., Drias H.: A Two-Phase Commit Protocol for Mobile Wireless Environment. [in:] Proceedings of the 16th Australasian database conference, Newcastle, Australia 2005.
9. Nouali N., Drias H., Doucet A.: Revisiting Distributed Protocols for Mobility at the Application Layer. [in:] The Third World Enformatika Conference, WEC'05, April 27-29, Istanbul, Turkey 2005.
10. Tanenbaum S., van Steen M.: Systemy rozproszone. Zasady i paradygmaty. WNT, Warszawa 2006.
11. Wesołowski K.: Systemy radiokomunikacji ruchomej. WKŁ, Warszawa 2003.
12. Wrembel R., Bębel B.: Oracle. Projektowanie rozproszonych baz danych. Helion, Gliwice 2003.

Recenzent: Dr inż. Andrzej Sikorski

Wpłynęło do Redakcji 3 marca 2011 r.

Abstract

The design of transaction commit protocols intended for mobile systems requires the consideration of specific issues related to the characteristics of a mobile environment and a users' mobility. The article discusses in detail the *Transaction Commit on Timeout* protocol, which meets the requirements of a mobile system. Its main idea is to use the *timeout* parameter to identify the successful completion of a transaction in the order to realize a characteristic of non-blocking.

The author created an application that enables simulation and visualisation of a protocol execution in the mobile system. The program was used to carry out simulations that illustrate

different ways of TCOT execution for a number of events. A structure of a mobile database system designed for the purpose of the simulations is depicted on the figure Fig. 4. Results of the simulations are presented in table Table 1. The results are discussed in detail. In the context of TCOT protocol execution issues connected with transaction atomicity and consistency of data in the mobile system are considered. On the basis of the results, several modifications of the protocol are recommended.

Adres

Aleksandra BIEŃKOWSKA: Uniwersytet Jagielloński, Instytut Informatyki,
ul. Prof. S. Łojasiewicza 6, 30-348 Kraków, Polska, ab.bienkowska@gmail.com.