

# 이동 컴퓨팅 환경에서 비동기적 데이터방송을 사용한 동시성 제어 기법

고 승 민\* · 김 대 인\*\* · 임 선 모\*\*\* · 황 부 현\*\*\*\*

## 요 약

이동 컴퓨팅 환경에서 이동 호스트는 대역폭의 사용을 줄이고 이동 트랜잭션의 응답 시간을 향상시키기 위하여 캐쉬를 사용한다. 그리고 서버는 이동 호스트의 캐쉬 일관성을 유지하기 위해서 주기적으로 갱신된 데이터를 방송한다. 그러나 무효화 보고서를 사용한 캐쉬 일관성 유지 기법은 이동 트랜잭션의 완료 결정을 지연한다는 문제점을 가지고 있다. 본 논문에서는 캐싱된 데이터를 사용하여 이동 트랜잭션을 수행하는 경우 이동 트랜잭션의 직렬 가능한 수행을 보장하는 CCM-AD 방법을 제안한다. 제안하는 CCM-AD 방법은 서버에서 이동 호스트에게 방송한 데이터와 갱신된 데이터와의 교집합을 구하여 공통되는 데이터를 요청 데이터와 함께 방송함으로써 이동 트랜잭션의 직렬 가능한 수행을 보장한다. 그리고 수행 후 완료 결정을 위한 이동 트랜잭션의 지연도 줄일 수 있다. 또한 주기적으로 방송하는 무효화 보고서의 메시지 크기도 줄일 수 있다.

## Concurrency Control Method using the Asynchronous Data Broadcast in Mobile Computing Environments

Seung-Min Ko<sup>\*</sup> · Dae-In Kim<sup>\*\*</sup> · Sun-Mo Im<sup>\*\*\*</sup> · Bu-Hyun Hwang<sup>\*\*\*\*</sup>

## ABSTRACT

In mobile computing environments, a mobile host caches the data to use the narrow bandwidth efficiently and improve the response time. A server periodically broadcasts the update data for maintaining the cache consistency of a mobile host. However, a method for maintaining cache consistency using periodic invalidation report has a problem that defers the commit decision of a mobile transaction to guarantee its serializable execution. In this paper, we propose CCM-AD method that guarantees the serializable execution of a mobile transaction even in the case that it is executed using cached data. As CCM-AD method guarantees the serializable execution of mobile transactions by using the intersection between the data broadcast to mobile host and the data updated at server. Then the CCM-AD method can reduce the delay of commitment decision of a mobile transaction. Also our method can reduce the size of invalidation report.

**키워드 :** 이동 컴퓨팅(Mobile Computing), 동시성 제어(Concurrency Control), 캐쉬 일관성(Cache Consistency), 비동기 데이터방송(Asynchronous data broadcast), 무효화 보고서(Invalidation Report)

## 1. 서 론

이동 컴퓨팅 환경은 기존의 분산 환경에 비하여 이동 호스트의 이동성, 잦은 단절(disconnection), 제한된 대역폭(bandwidth), 그리고 배터리 용량 등의 제약을 받으며 이동 컴퓨팅 환경의 특성을 고려한 효율적인 이동 트랜잭션의 처리 방법으로 서버에 있는 데이터베이스의 일부분을 이동 호스트로 가져오는 캐싱 방법이 제안되었다[2, 10, 13].

그러나 캐싱 방법은 서버에서 데이터가 갱신되는 경우 이동 호스트는 지구국에서의 갱신 발생 여부를 알지 못하므로 이동 호스트의 데이터의 일관성 검증과정에서 서버의 처리 능력은 감소하고 제한된 대역폭의 빈번한 사용으로 인한 병목현상(bottleneck)이 발생한다는 문제점이 있다. 이러한 문제점을 해결하기 위하여 무효화 보고서(invalidation report)를 이동 호스트로 방송해 주는 방법이 제안되었다[2, 10]. 그러나 주기적인 무효화 보고서 방송을 사용한 캐쉬 일관성 유지 방법은 배터리 절약을 위한 고의적인 단절 및 이동 호스트의 이동으로 인하여 무효화 보고서를 수신하지 못하는 경우 정확한 데이터를 무효화하는 거짓 무효화(false invalidation)가 발생할 수 있다[2].

본 연구에서는 이동 컴퓨팅 환경에서 이동 트랜잭션의

\* 본 연구는 2004년도 한국과학기술원 지역우수과학자 육성지원연구(R05-2003-000-10532-0)의 지원으로 수행되었음.

† 준 회 원 : 전남대학교 전산학과

\*\* 준 회 원 : 전남대학교 전산학과 시간강사

\*\*\* 정 회 원 : 동강대학 컴퓨터정보과 조교수

\*\*\*\* 중신회원 : 전남대학교 컴퓨터정보학과

논문접수 : 2004년 9월 20일, 심사완료 : 2004년 12월 15일

직렬 가능한 수행을 보장하고 동시에 이동 호스트의 캐쉬 일관성을 유지하기 위하여 방송하는 무효화 보고서를 최적화할 수 있는 방법을 제안한다. 제안하는 방법은 이동 트랜잭션이 접근한 데이터를 무효화 보고서 수신 시점을 기준으로 분류하여 무효화 보고서를 수신하기 이전에 완료 결정을 내림으로써 이동 트랜잭션의 빠른 응답 시간을 제공하고, 지구국에서의 데이터 갱신 여부를 이동 호스트가 요청한 데이터 방송시 함께 방송함으로써 무효화 보고서의 크기를 줄일 수 있다.

본 연구의 내용은 다음과 같다. 제2장에서는 기본적인 이동 컴퓨팅 환경 및 캐쉬 일관성방법에 대한 관련 연구를 기술하고, 제3장에서는 OCC-UTS<sup>2</sup> 방법의 문제점을 기술한다. 제4장에서는 갱신된 데이터 정보를 방송하여 이동 트랜잭션의 직렬 가능한 수행을 보장하고 무효화 보고서를 최적화 할 수 있는 방법을 제안하고, 제5장에서는 해석적 모델을 이용하여 제안하는 방법의 성능을 분석한다. 마지막으로 제6장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

## 2. 관련연구

### 2.1 이동 컴퓨팅 환경

이동 컴퓨팅 환경에서 하나의 서버가 서비스를 제공해 주는 영역을 셀(cell)이라 하며 하나의 셀 안에 있는 모든 이동 호스트는 같은 서버에서 서비스를 제공받으며 관리된다[2].

이동 컴퓨팅 환경에서 이동 호스트가 정보 검색과 같은 작업을 수행하는 트랜잭션을 이동 트랜잭션이라 하며 이동 트랜잭션은 기존의 분산 환경에서 수행되는 트랜잭션에 비하여 짧은 트랜잭션이다[1].

이러한 이동 호스트의 요구에 서버는 즉각적인 응답을 해야 하지만 이동 컴퓨팅 환경은 유선 환경에 비하여 낮은 대역폭의 특성을 갖고 있다. 이러한 제약환경을 고려한 데이터 전송방법으로 풀 기반(pull based) 데이터 방송과 푸시 기반(push based) 데이터 방송, 그리고 이 둘을 혼합한 혼합(hybrid) 데이터 방송 방법이 제안되었다[3, 4].

풀 기반 데이터 방송 방법은 이동 호스트로부터 데이터 요구가 있는 경우 서버에서 데이터를 방송하는 방법으로, 이동 호스트에서 서버로의 상향 채널(uplink channel)의 사용이 많아지지만 이동 호스트가 요구하는 데이터를 즉시 얻을 수 있는 장점이 있다. 그러나 푸시 기반 데이터 방송 방법은 이동 호스트의 요구가 없어도 서버에서 구성된 방송 스케줄에 따라 데이터를 방송하는 방법으로, 대역폭의 사용은 줄일 수 있지만 이동 호스트가 필요로 하는 데이터를 수신할 때까지 이동 트랜잭션의 수행이 지연됨으로써 이동 트랜잭션의 응답 시간이 길어지는 단점이 있다[12]. 그리고 혼합 데이터 방송 방법은 서버에서의 방송 스케줄 구성시 상향 채널을 통하여 들어오는 이동 호스트의 데이터 요청을 반영함으로써 가능한 한 대역폭의 사용을 줄이고 이동 호스트의 데이터 수신 대기 시간을 줄일 수 있다는 장점이 있다[3, 4].

### 2.2 캐쉬 일관성

이동 컴퓨팅 환경에서 캐쉬 사용은 이동 컴퓨팅 환경의 무선 통신망이 갖는 제한된 대역폭의 한계를 극복하고 이동 트랜잭션의 효율적인 수행을 위하여 반드시 필수적이다.

[2]에서는 이동 컴퓨팅 환경에서 지구국에서 주기적으로 무효화 보고서를 방송하여 이동 호스트의 캐쉬 일관성을 유지하는 방법으로 TS(broadcasting TimeStamp), AT(Amnesiac Terminals), SIG(SIGnatures) 방법을 제안하였다. 제안한 방법에서 이동 호스트는 현재 무효화 보고서 방송주기 동안에 제출되는 질의를 리스트 형태로 유지한다. 그리고 이동 호스트는 수신한 무효화 보고서를 사용하여 캐쉬된 데이터의 유효성을 검증하고 질의를 처리한다.

[10]에서는 비교적 단절 시간과 갱신 패턴에 영향을 적게 받는 이동 호스트의 캐쉬 일관성 유지 방법으로 BS(Bit Sequence) 방법을 제안하였다. BS 방법에서 지구국은 이동 호스트의 단절에 대비하여 하나 이상의 무효화 보고서 방송 구간 동안에 갱신된 데이터 항목에 대한 정보를 비트 열로 구성하여 방송한다. 그리고 이동 호스트는 무효화 보고서에 포함된 타임스탬프와 비트 열을 사용하여 이동 호스트의 캐쉬 일관성을 유지한다. 그러나 [2, 10]에서 제안한 주기적인 무효화 보고서 방송을 사용한 이동 호스트의 캐쉬 일관성 유지 방법은 이동 트랜잭션의 직렬 가능한 수행을 고려하지 않는다는 문제점이 있다. 또한 이동 호스트의 단절 등으로 인하여 실제로는 정확한 데이터를 무효화하는 거짓 무효화(false invalidation)가 발생할 수 있다.

## 3. 문제점 제시

이동 컴퓨팅 환경에서 이동 트랜잭션에 관한 연구는 제한된 대역폭, 전력의 한계, 잦은 접속 단절, 그리고 사용자의 이동성과 같은 이동 컴퓨팅 환경의 특성을 고려하여야 한다.

[9]에서는 이동 트랜잭션의 직렬 가능한 수행을 보장하는 방법으로 OCC-UTS(Optimistic Concurrency Control with Update TimeStamp) 방법과 OCC-UTS<sup>2</sup>(Optimistic Concurrency Control with Update TimeStamp Span) 방법을 제안하였다. OCC-UTS 방법에서 이동 호스트는 이동 트랜잭션 수행 후 완료 시점에 수행의 정확성 검증을 위하여 이동 트랜잭션이 접근한 데이터 집합을 서버로 보내며, 서버는 현재 무효화 보고서 방송주기 동안에 발생한 이동 트랜잭션 수행의 정확성 검증 결과를 완료 결정 리스트와 철회 결정 리스트로 구성하여 무효화 보고서와 함께 방송한다. 그리고 이동 호스트는 수신한 무효화 보고서에 포함된 결과에 따라 이동 트랜잭션을 완료 또는 철회한다. 그러나 OCC-UTS 방법은 이동 트랜잭션 완료시 정확성 검증 요구로 인한 이동 호스트에서 서버로의 상향 채널 사용이 빈번해지고 이동 트랜잭션의 정확성을 서버에서 검증함으로써 서버에 포화(saturation) 현상이 발생한다는 문제점을 갖고 있다. 그러므로 [9]에서는 OCC-UTS 방법의 문제점을 해결하기 위하여 이동 호스트에서 이동 트랜잭션의 정확성을 검증하는 OCC-UTS<sup>2</sup> 방법을

제안하였다.

OCC-UTS<sup>2</sup> 방법은 이동 트랜잭션이 접근한 데이터의 타임스탬프가 모두 같거나 가장 최근에 수신한 무효화 보고서의 방송 시점보다 작은 경우에 이동 트랜잭션을 즉시 완료함으로써 이동 트랜잭션의 응답 시간을 향상시키고 이동 호스트에서 이동 트랜잭션의 완료 결정을 함으로써 정확성 검증으로 인한 서버의 포화 현상을 제거하였다. 그러나 OCC-UTS<sup>2</sup> 방법 역시 대부분의 이동 트랜잭션의 완료 결정을 무효화 보고서 수신 시점에 함으로써 이동 트랜잭션의 응답 시간이 길어진다는 문제점이 있다. 따라서 가능한 한 이동 트랜잭션 수행의 빠른 응답 시간을 제공할 수 있으며, 정확한 이동 트랜잭션 수행의 정확성을 검증할 수 있는 방법에 대한 연구가 필요하다.

본 연구에서는 이러한 문제점을 해결하고 이동 트랜잭션의 직렬 가능한 수행을 보장하는 방법으로 CCM-AD(Concurrency Control Method using the Asynchronous Data Broadcast) 방법을 제안한다. 제안하는 CCM-AD 방법은 무효화 보고서를 수신하기 이전에 이동 트랜잭션의 완료 및 지연 여부를 결정함으로써 가능한 한 이동 트랜잭션의 지연을 최소화하고 응답 시간을 향상시킬 수 있다. 또한 지구국에서의 데이터 갱신 여부를 이동 호스트가 요청한 데이터 방송시 함께 방송함으로써 무효화 보고서 크기를 줄일 수 있다.

#### 4. CCM-AD 방법

본 연구에서 제안하는 CCM-AD 방법에서 이동 트랜잭션은 읽기 연산만을 수행하며 모든 데이터의 갱신 연산은 서버에서 수행된다고 가정한다. CCM-AD 방법에서 이동 트랜잭션은 이동 호스트에 캐칭된 데이터를 사용하여 수행되며 이동 트랜잭션이 캐쉬에 없는 데이터를 요구하는 경우 서버로 데이터를 요청한다. 그리고 서버는 이동 트랜잭션의 빠른 수행을 위하여 이동 호스트가 요청한 데이터를 가능한 한 즉시 방송한다고 가정한다.

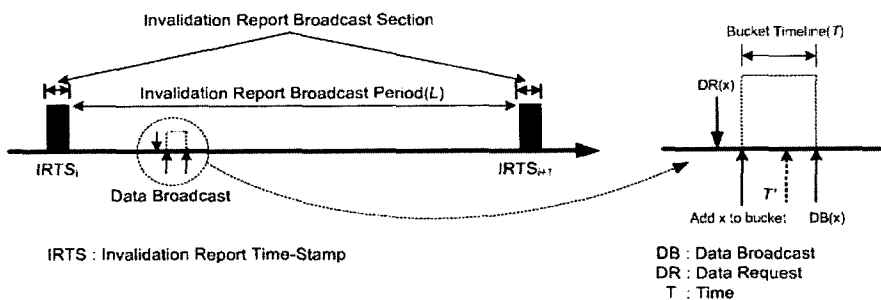
이동 트랜잭션 수행을 위한 데이터 캐칭 방법은 하나의 이동 트랜잭션이 접근하는 모든 데이터를 이동 트랜잭션의 수행 시작 전에 모두 얻는 Cache-ahead 방법과 이동 트랜잭션의 수행 도중에 데이터를 얻는 Cache-on-demand 방법으로 나눌 수 있다.

Cache-ahead 방법을 사용하는 경우 이동 호스트는 선처리기(preprocessor)를 사용하여 이동 트랜잭션을 수행하기 이전에 미리 접근하는 데이터를 파악할 수 있으며, 필요로 하는 여러 개의 데이터를 한 번의 요청 메시지를 사용하여 지구국에 전달함으로써 제한된 상향 채널의 사용을 줄이고 빠른 데이터 확보가 가능하다. 따라서 CCM-AD 방법은 데이터 캐칭 방법으로 Cache-ahead 방법을 사용한다고 가정한다.

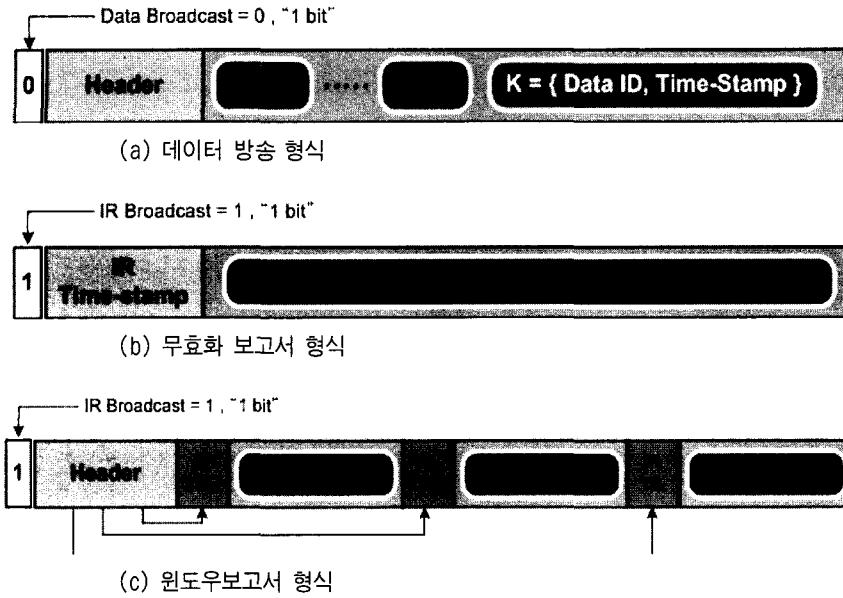
##### 4.1 서버 알고리즘

본 절에서는 제안하는 CCM-AD 방법에서 수행되는 서버 알고리즘에 대하여 기술한다. CCM-AD 방법에서 서버에서 유지하는 자료구조는 다음과 같다. 서버는 무효화 보고서 방송주기 동안에 서버에서 갱신된 데이터의 집합인 U\_DS(Updated Data Set)와, 무효화 보고서 방송주기 동안에 이동 호스트의 요구로 인하여 방송한 데이터 집합인 B\_DS(Broadcast Data Set)를 유지한다. 또한 서버는 집합 B\_DS와 집합 U\_DS에 공통적으로 포함되는 교집합 K를 유지한다.

제안하는 CCM-AD 방법에서 교집합 K는 현재 무효화 보고서 방송주기 동안에 이동 호스트에게 방송한 데이터가 갱신되었음을 이동 호스트에게 알려주기 위한 정보이며, 교집합 K는 서버에서 이동 호스트의 데이터 요구시에 데이터와 함께 방송한다. 즉 이동 호스트에서 데이터 x를 요구하는 경우 서버는 이동 호스트가 요구한 데이터 x를 방송할 때 교집합 K를 함께 방송한다. 그리고 CCM-AD 방법에서 서버는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 갱신된 데이터의 식별자(identifier)와 타임스탬프로 구성되는 무효화 보고서를 주기적으로 방송한다. 일반적으로 무효화 보고서는 현재 방송주기 동안에 발생한 모든 갱신 정보로 구성된다. 그러나 제안하는 CCM-AD 방법에서는 현재 무효화 보고서 방송주기 동안에 이동 호스트에게 방송한 데이터가 갱신되는 경우 교집합 K를 이용하여 방송된 데이터의 갱신 여부를 알려주므로 교집합 K를 제외한 갱신 정보로 무효화 보고서를 구성한다. 또한 CCM-AD 방법에서 서버는 이동 호스트가 단절로 인하여 무효화 보고서를 수신하지 못할 경우에 대비하여 하나 이상의 무효화 보고서 정보로 구성되는 윈도우 보고서를 유지한다. 그리고 이동 호스트는 단절 후 재연결시 캐칭된 데이터의 유효성을 검증하기 위하여 서버로 윈도우 보고서를 요청한다.



( 그림 1 ) 데이터 방송 및 무효화 보고서 방송



(그림 2) 서버에서 방송하는 메시지 구조

CCM-AD 방법에서 서버는 (그림 1)과 같이 DR(x) 시점에 이동 호스트로부터 데이터 요청을 받는 경우 요청받은 데이터를 방송 단위인 버킷에 추가한다. 버킷을 사용하는 이유는 이동 호스트로부터 데이터 요청을 받을 때마다 방송하는 것은 잦은 대역폭의 사용을 유발하므로 일정 기간 동안 이동 호스트의 데이터 요청 정보를 수집하여 한번에 버킷 단위로 방송하여 제한된 대역폭의 사용을 줄이기 위함이다. 서버는 기본적으로 이동 호스트로부터 데이터 요구를 수집하여 버킷이 채워지는 즉시 버킷 단위로 데이터를 방송한다. 그리고 버킷이 차지 않더라도 마감시간(deadline)  $T$ 를 설정하여  $T$ 초가 경과한 버킷은 즉시 이동 호스트에게 방송한다. 즉 무효화 보고서 방송주기가  $L$ 이고 만약  $T'(L > T > T')$  초에 버킷이 채워지는 경우 시점  $T'$ 초에 데이터를 방송하고 그렇지 않으면  $T$ 초에 버킷을 방송한다.

제안하는 CCM-AD 방법에서 서버는 (그림 2)와 같이 크게 세 가지의 메시지를 이동 호스트에게 방송한다. (그림 2)의 각각의 메시지에서 첫 1비트는 데이터 방송과 무효화 보고서 방송을 구분하기 위한 정보이다. 첫 비트가 0인 경우 헤더에 포함된 데이터 정보와  $K$ 의 포인터를 사용하여 필요로 하는 데이터만을 선택하여 사용한다. 예를 들어 AT&T의 Hobbit-Chip을 기준으로 할 때 데이터를 처리하는데 소모되는 전력은 대기 모드일 때보다 더 많은 전력이 소모된다. 그러므로 CCM-AD 방법에서는 필요한 데이터에만 접근한다.

무효화 보고서와 윈도우보고서는 (그림 2)의 (b)와 (c)에서와 같이 서버에서 갱신된 데이터 식별자와 타임스탬프의 쌍으로 구성된다. 윈도우보고서는 무효화 보고서를 수신하지 못한 이동 호스트를 위하여 방송한다. 그리고 윈도우보고서는 (그림 2)의 (c)와 같이 가능한 한 이동 호스트의 불필요한 데이터 처리를 막기 위하여 각 무효화 보고서 주기의 인

덱스 값을 가지고 있다. 예를 들어 이동 호스트는 단절 기간에 따라  $IRTS_{i-2}$  주기의 무효화 보고서를 수신하지 못하거나 또는  $IRTS_{i-1}$  주기의 무효화 보고서를 수신하지 못할 수 있다. 전자에 해당하는 이동 호스트의 경우는 무효화 보고서의 전체 내용을 수신하여야 하지만 후자의 경우는  $IRTS_{i-2}$  주기의 내용은 수신할 필요가 없다. 그러므로 CCM-AD 방법에서는 이동 호스트가 윈도우보고서를 수신하는 경우 단절 시점의 시간을 고려하여 가장 적합한 시점의 갱신 정보를 선택하여 데이터를 무효화한다. 이러한 서버의 처리 과정은 (알고리즘 1)과 같이 기술할 수 있다.

1. Data Broadcast {

```

If(received data request from client) {
    while(bucket is not full) {
        add request data(j) to bucket;
        If(data broadcast time interval  $T \leq$  deferred Time) {
            exit; }
        //지연된 시간이 마감시간  $T$ 보다 크면 빠져나감
    }
}

```

$K = U_{DS} \cap B_{DS}$

```

broadcast data bucket and set  $K$  to Client;
remove set  $K$  both  $U_{DS}$  and  $B_{DS}$ ; // 방송한  $K$ 에
속한 데이터아이템 제거
add  $K$  to  $K_w$ ; // 접속 단절된 이동 호스트를 위해  $K$ 의
사본을  $K_w$ 만큼 유지
add requested data to  $B_{DS}$ 
}

```

2. Invalidation Report Broadcast {

```

If(received re-connection request from the Client) {

```

```

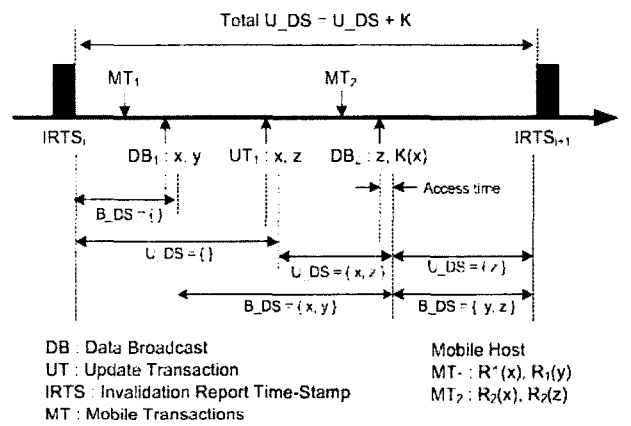
broadcast U_DSw; U_Kw;
}
else broadcast U_DS; // U_DS는 K만큼의 Data list가 제
거된 무효화 리스트
}
3. For each Update Transaction U_T {
If( U_T has been committed ) {
add its updated data items to U_DS;
}
}
    
```

(알고리즘 1) CCM-AD 방법에서의 서버 알고리즘

(알고리즘 1)의 데이터 방송 모듈은 서버가 이동 호스트로부터 데이터 요청을 받았을 때 요청받은 데이터를 방송하는 모듈이다. 방송 모듈에서 서버는 요청 데이터를 방송하기 전에 교집합  $K$ 를 구한 후 이동 호스트가 요청한 데이터와 함께 방송한다. 그리고 방송 후 집합  $U\_DS$ 와 집합  $B\_DS$ 에 공통으로 포함된 교집합  $K$ 에 해당하는 데이터를 삭제한다. 교집합  $K$ 는 이동 호스트로 보내는 무효화 정보이면서 동시에 현재 방송주기 동안에 이동 호스트로 방송된 데이터가 갱신되었음을 알려주는 정보이다. 그리고 이동 호스트는 데이터와 함께 방송된 교집합  $K$ 를 사용하여  $K$ 에 포함된 데이터를 제거함으로써 교집합  $K$ 는 이동 호스트의 캐쉬 일관성 유지에도 사용할 수 있으며 서버에서 주기적으로 방송하는 무효화 보고서의 크기를 줄일 수 있다. 기존의 [2,9,10]에서는 이동 호스트의 캐쉬 일관성을 유지하기 위하여 무효화 보고서를 방송할 때 (그림 3)의  $IRTS_i$ 와  $IRTS_{i+1}$  사이에 갱신된 모든 데이터 정보를 방송한다. 그러나 제안하는 CCM-AD 방법은 갱신된 데이터 식별자에서 교집합  $K$ 를 제거한 무효화 보고서를 방송함으로써 무효화 보고서의 크기를 줄일 수 있다. 또한 서버는 이동 호스트의 단절에 대비하여 하나 이상의 무효화 보고서 방송 주기 동안에 발생한 교집합  $K$ 의 집합인  $K_w$ 를 유지한다. 그리고 서버에서 유지하는  $K_w$ 는 기존의 방법에서 유지하는 윈도우 정보와 비슷하므로 별도의 추가적인 비용은 없다고 볼 수 있다. 즉 제안하는 CCM-AD 방법에서 집합  $U\_DS$ 와 교집합  $K$ 의 합은 (그림 3)의  $IRTS_i$ 와  $IRTS_{i+1}$  사이에 서버에서 발생하는 총 갱신 집합이 되며, 서버에서 데이터와 함께 방송한 교집합  $K$ 를 집합  $U\_DS$ 와 집합  $B\_DS$ 에서 제거하므로 무효화 보고서 방송주기 동안 서버에서 갱신된 집합  $U\_DS$ 에는  $K$ 에 속한 데이터 아이템은 제외된다. CCM-AD 방법에서 이동 호스트의 단절 후 서버에서 방송하는 윈도우보고서는 최종적으로 윈도우 방송주기  $w$  동안에 발생한 갱신 정보  $U\_DS_w$ 와  $K_w$ 를 (그림 2)의 (c)와 같은 형태로 전송한다. 즉 이동 호스트가 단절 후 재연결시 윈도우보고서를 요구하는 경우 서버에서는 불필요한 메시지 방송을 피하기 위하여 무효화 보고서 대신 윈도우보고서를 이동 호스트에게 방송한다.

(그림 3)은 서버 알고리즘의 예이다. 하나의 이동 호스트에 이동 트랜잭션  $MT_1$ 과  $MT_2$ 가 제출되었고 이동 트랜잭션

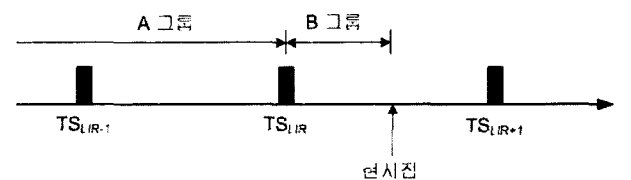
$MT_1$ 의 수행을 위하여 이동 호스트는 데이터  $x, y$ 를 서버로 요청하고 이동 트랜잭션  $MT_1$ 을 수행한다. 그리고 제출된 이동 트랜잭션  $MT_2$ 의 수행을 위하여 캐쉬에 없는 데이터  $z$ 를 서버로 요청하고 요청 데이터를 서버로부터 수신한 후 이동 트랜잭션  $MT_2$ 를 수행한다. 이동 트랜잭션  $MT_2$ 가 데이터  $z$ 를 요청하는 시점에서 집합  $U\_DS$ 와 집합  $B\_DS$ 는 각각  $\{x, z\}$ 와  $\{x, y\}$ 이며 교집합  $K$ 는  $\{x\}$ 가 된다. 그러나 이동 트랜잭션  $MT_2$ 의 수행을 위하여 이동 호스트가 요청한 데이터  $z$ 와  $K$ 를 방송한 후 서버에서 유지하는 집합  $U\_DS$ 와  $B\_DS$ 는 각각  $\{z\}$ 와  $\{y, z\}$ 가 된다. 즉 서버에서 방송 시점  $DB_2$ 에 이동 호스트에게 데이터  $z$ 와 교집합  $K$ 를 방송한 후  $K$ 에 속하는 데이터 아이템  $x$ 를  $U\_DS$ 에서 제거하기 때문이다. 제안한 CCM-AD 방법은 서버 알고리즘을 통하여 무효화 보고서 방송주기 동안에 서버로부터 얻은 데이터의 갱신 여부를 교집합  $K$ 를 이동 호스트가 수신함으로써 데이터의 갱신 여부를 알 수 있으며 현재 무효화 보고서 방송주기 동안에 캐싱된 데이터의 정확성을 보장할 수 있다.



(그림 3) 서버 알고리즘의 예

#### 4.2 이동 호스트 알고리즘

본 절에서는 제안하는 CCM-AD 방법에서 수행되는 이동 호스트 알고리즘에 대하여 기술한다. CCM-AD 방법에서 이동 호스트가 유지하는 정보와 이동 트랜잭션의 처리 과정은 다음과 같다.



(그림 4) 이동 호스트의 데이터 그룹

이동 호스트는 이동 트랜잭션을 수행하기 위하여 Cache-ahead 캐싱 방법을 사용하며, 이동 호스트의 캐쉬 일관성 유지를 위한 정보로 가장 최근에 수신한 무효화 보고서의 타임스탬프  $TS_{LIR}$ 를 유지한다. 그리고 이동 호스트는 캐싱된

데이터를 논리적으로 (그림 4)와 같이 두 개의 그룹 A, B로 분류하여 관리한다. 그룹 A의 데이터는 가장 최근에 수신한 무효화 보고서의 타임스탬프  $TS_{LIR}$ 을 기준으로 이전에 서버로부터 캐싱한 데이터 집합이며, 그룹 B의 데이터는  $TS_{LIR}$ 이후에 서버로부터 캐싱한 데이터 집합이다. 두 그룹의 분류는 이동 호스트가 요청 데이터를 수신한 시점을 기준으로 한다. 즉  $TS_{LIR}$ 과  $TS_{LIR+1}$ 구간에 수신한 데이터의 타임스탬프를  $TS_j$ 라고 하는 경우 이동 호스트에 캐싱된 데이터의 타임스탬프가  $TS_j < TS_{LIR}$ 이면 이동 호스트가 수신한 데이터가  $TS_{LIR}$  이후에 갱신이 없었다는 것을 의미하며, 이 데이터는 그룹 B의 데이터가 된다. 그리고 CCM-AD 방법은 이동 호스트가 스스로 이동 트랜잭션의 완료나 철회 결정을 내릴 수 없는 경우 서버에서 방송하는 무효화 보고서를 사용하여 이동 트랜잭션의 완료 결정을 내리며 수행 후 완료 결정을 기다리는 이동 트랜잭션을 일종의 큐 형태인 TransactionQue에 유지한다.

CCM-AD 방법은 이동 호스트로 제출된 이동 트랜잭션이 접근한 모든 데이터가 그룹 A에 해당되는 경우 서버와 별도의 통신 없이 이동 트랜잭션을 즉시 완료한다. 즉 이러한 경우는 이동 트랜잭션이 접근한 데이터가 가장 최근에 수신한 무효화 보고서 수신 시점을 기준으로 일관성 있는 데이터이기 때문이다. 그리고 제출된 이동 트랜잭션이 접근한 모든 데이터가 그룹 B에 속하는 경우에도 이동 트랜잭션은 즉시 완료 가능하다. 즉 현재 무효화 보고서 방송주기 동안에 캐싱된 그룹 B의 데이터들은 교집합  $K$ 를 사용하여 서버에서의 갱신 발생 여부를 알 수 있으며, 서버에서 현재 무효화 보고서 방송주기 동안에 이동 호스트가 캐싱한 데이터가 갱신되는 경우 갱신된 데이터를 캐싱하고 있는 이동 호스트는 해당 데이터를 무효화함으로써 그룹 B에 있는 데이터는 교집합  $K$ 를 수신한 시점을 기준으로 일관성 있는 데이터이기 때문이다. 이러한 이동 호스트의 이동 트랜잭션 처리 및 완료 결정 과정은 (알고리즘 2)와 같이 기술할 수 있다.

**1. Beginning of Mobile Transaction MT**

All required data items for MT are computed by preprocessor:

```
while(acquire every required data item by MT) {
  for each required data item {
    If ( j is in the cache ) {
      using the data in its cache;
    }
    else { request to the server; }
  }
  If(acquire every required data item) {
    Starting MT;
  }
}
```

**2. Decision of commit or defer**

```
If ( every data item in ReadSet is in Group A )
  { immediately commit; }
```

```
else If( every data item in ReadSet is in Group B )
  { immediately commit; }
else If( every time-stamp of data item in ReadSet
  <  $TS_{LIR}$  )
  { immediately commit; } //  $TS_{LIR}$ : 가장 최근에 수신한 무효화 보고서의 타임스탬프
else { put into the TransactionQue;
  defer until next the invalidation report is arrived;
}
```

**3. When receiving the Data Broadcast**

```
If ( request = 1 ) { // 데이터를 요청한 경우
  Acquire requested data from the Data Broadcast ;
}
```

```
for every data item j in Group B {
  If there is a pair ( j,  $ts_j$  ) in K {
    If  $ts_j^{GB} < ts_j$  { throw j out of the Group B }
    //  $ts_j^{GB}$ : 그룹B에 속하는 데이터 j의 타임스탬프
    //  $ts_j$ : 데이터 j의 타임스탬프
  }
}
```

**4. When receiving the Invalidation Report Broadcast(  $TS_i$  )**

```
If (  $TS_i - TS_{LIR} > W$  ) { drop the entire cache }
else { for every data item j in Cache {
  If there is a pair ( j,  $ts_j$  ) in IR(  $TS_i$  ) {
    If  $ts_j^C < ts_j$  { throw j out of the Cache; }
    //  $ts_j^C$ : 캐싱된 데이터 j의 타임스탬프
  }
}
```

```
while ( TransactionQue is not empty ) {
  for every  $T \in$  TransactionQue
    If ( ReadSet( T )  $\cap$  IR =  $\emptyset$  )
      { commit; }
    else { abort; }
}
```

Group A = Group A  $\cup$  Group B ;

(알고리즘 2) 이동 호스트의 완료 결정 알고리즘

(알고리즘 2)에서 이동 트랜잭션의 시작 부분은 이동 호스트가 이동 트랜잭션을 수행하기 이전에 선처리기를 사용하여 이동 트랜잭션이 접근하는 데이터를 서버로 요청하여 미리 확보하는 과정이다. 제안하는 CCM-AD 방법은 이동 트랜잭션이 접근한 데이터가 다음의 세 가지 조건 중 하나를 만족할 때 즉시 완료된다. 첫 번째 조건은 이동 트랜잭션이 접근한 모든 데이터가 그룹 A에 속하는 경우이고 두 번째 조건은 이동 트랜잭션이 접근한 모든 데이터가 그룹 B에 속하는 경우이다. 그리고 세 번째 조건은 이동 트랜잭션이 접근한 데이터가 그룹 A 또는 그룹 B에 속하더라도 데이터의 타임스탬프가  $TS_{LIR}$ 보다 작은 경우에는 즉시 완료할 수 있다.

그러나 이러한 조건을 만족하지 못하는 이동 트랜잭션은 다음 무효화 보고서 방송시점까지 TransactionQue에 대기한 후 데이터의 갱신 여부에 따라 완료 또는 철회된다. 또한 제안하는 CCM-AD 방법은 서버에서 주기적으로 방송하는 무효화 보고서를 사용하여 이동 호스트의 캐쉬 일관성과 이동 트랜잭션의 직렬 가능한 수행을 보장한다.

## 5. 성능평가

이 장에서는 본 연구에서 제안한 CCM-AD 방법을 해석적 모델을 이용하여 이동 트랜잭션의 완료율 및 서버에서의

$$P_Q = \text{Prob} [\text{no queries in an interval} \mid \text{unit is awake during the interval}] = e^{-\lambda L} \quad (1)$$

$$P_Q' = \text{Prob} [\text{one or more queries in an interval}] = (1 - e^{-\lambda L}) \quad (2)$$

식(1)은 데이터 질의 발생률이 지수분포를 따르는 경우 하나의 방송주기  $L$  동안에 이동 호스트가 연결 상태이면서 질의가 제출되지 않을 확률이다[2,9]. 그러므로 방송주기  $L$  동안에 이동 호스트에 질의가 발생할 확률은 질의가 발생하지

않을 확률에 따른 무효화 보고서 전송량의 성능을 분석한다. 그리고 CCM-AD 방법의 성능을 분석하기 위하여 [9]에서 제안한 OCC-UTS<sup>2</sup> 방법을 비교 대상으로 한다.

OCC-UTS<sup>2</sup> 방법은 제안한 CCM-AD 방법과 같은 풀기반 방송 환경이며 이동 트랜잭션의 직렬 가능한 수행을 보장하기 위하여 무효화 보고서를 이용하여 이동 트랜잭션의 완료 결정을 하므로 제안한 CCM-AD 방법의 비교 대상으로 적합하다. 본 연구에서의 해석적 모델은 비교 대상인 OCC-UTS<sup>2</sup>의 성능 분석 방법에 기초하여 [9]의 해석적 모델과 같다고 정의한다. 그리고 하나의 무효화 보고서 방송주기  $L$  동안에 질의가 발생할 확률  $P_Q'$ 는 다음과 같이 정의할 수 있다.

않을 확률을 제외한 확률로 식(2)와 같이 정의된다. 또한, 서버에서 데이터가 갱신될 확률  $P_U'$ 는 다음과 같이 정의할 수 있다.

$$P_U = \text{Prob} [\text{no updates during an interval}] = e^{-\mu L} \quad (3)$$

$$P_U' = \text{Prob} [\text{one or more updates during an interval}] = (1 - e^{-\mu L}) \quad (4)$$

식(3)은 데이터 갱신 발생률이 지수분포를 따르는 경우 방송주기  $L$  동안에 서버에서 데이터가 갱신되지 않을 확률이다[2,9]. 그러므로 방송주기  $L$  동안에 서버에서 데이터가 갱신될 확률은 데이터가 갱신되지 않을 확률을 제외한 확률로 식(4)와 같이 정의된다.

### 5.1 이동 트랜잭션의 완료율

제안한 CCM-AD 방법과 OCC-UTS<sup>2</sup> 방법에서 이동 호스트는 이동 트랜잭션을 수행하기 위하여 자신에 캐쉬에 있는 데이터를 사용하며 캐쉬에 존재하지 않는 데이터는 서버로 요청하여 이동 트랜잭션을 수행한다. 그러나 서버에서 수행 중인 갱신 트랜잭션과 이동 호스트에서 수행되는 이동 트랜잭션간의 충돌로 인한 직렬화 그래프에 사이클이 발생할 수 있으며, 이로 인하여 이동 트랜잭션의 완료 결정을 무효화 보고서를 수신할 때까지 지연된다. 그러므로 OCC-UTS<sup>2</sup> 방법에서 이동 트랜잭션은 수행 후 즉시 완료하는 경우와 무효화 보고서를 수신할 때까지 지연하는 경우로 나누어지며, 이를 적용한 완료율  $P_{OCC}$ 는 다음과 같이 정의할 수 있다.

$$P_{OCC} = OCC_{immediate} + OCC_{def\ errred} \quad (5)$$

식(5)에서  $OCC_{immediate}$ 는 이동 트랜잭션 수행 후 바로 완

료할 수 있는 경우이며  $OCC_{def\ errred}$ 는 이동 트랜잭션의 정 확성 검증을 위하여 서버로부터 무효화 보고서 수신시점까지 지연되는 경우로 이동 트랜잭션이 즉시 완료되는 경우를 제외한 경우에 해당된다. OCC-UTS<sup>2</sup> 방법에서 이동 트랜잭션이 즉시 완료하는 경우는 이전 방송 주기 동안에 캐싱된 데이터에 접근하거나 현재 무효화 보고서 방송주기 동안에 갱신되지 않은 데이터에 접근하는 경우이다. 그러나 이동 트랜잭션이 이전 방송 주기 동안에 캐싱된 데이터에 접근하는 경우는 OCC-UTS<sup>2</sup> 방법과 제안 방법과 모두 공통적인 부분이므로 해석적 모델에서는 현재 무효화 보고서 방송주기 동안에 캐싱된 데이터만 고려한다. 따라서 캐쉬 적중률이  $h$ 인 경우에  $OCC_{immediate}$ 는 제출된 이동 트랜잭션이 이동 호스트의 캐쉬에 없는 데이터를 요구하며, 이 때 데이터가 갱신되지 않는 경우이며 다음과 같이 정의할 수 있다.

$$OCC_{immediate} = (1 - h) \times (1 - e^{-\lambda L}) \times e^{-\mu L} \quad (6)$$

그리고  $OCC_{def\ errred}$ 는  $OCC_{immediate}$ 의 경우를 제외한 모든 경우이며 이동 트랜잭션은 현재 무효화 보고서에 접근한 데이터가 존재하지 않아야 완료할 수 있다. 즉 이동 트랜잭션이 접근한 데이터는 현재 무효화 보고서 방송주기 동안에 갱신되지 않은 경우로 다음과 같이 정의할 수 있다.

$$OCC_{def\ erred} = (1 - OCC_{immediate}) \times e^{-\mu L} \quad (7)$$

제안한 CCM-AD 방법에서 이동 트랜잭션은 OCC-UTS<sup>2</sup> 방법과 유사하게 수행 후 바로 완료할 수 있는 경우와 무효화 보고서 수신 시점까지 지연되는 두 가지 경우로 나누어진 다. 그러므로 CCM-AD 방법의 이동 트랜잭션 완료율  $P_{CCM}$  은 이동 트랜잭션이 즉시 완료하는 경우와 지연하는 경우로 나누어지며 다음과 같이 정의할 수 있다.

$$P_{CCM} = CCM_{immediate} + CCM_{def\ erred} \quad (8)$$

제안한 CCM-AD 방법에서 서버는 현재 무효화 보고서 방 송주기  $L$  동안에 이동 호스트에게 방송한 데이터가 갱신되 는 경우 교집합  $K$ 를 계산하여 재방송함으로써 항상 일관된 상태를 유지할 수 있으므로 즉시 완료될 수 있다. 따라서 CCM-AD 방법에서 이동 트랜잭션이 즉시 완료하는 경우인  $CCM_{immediate}$  는 이동 트랜잭션이 현재 방송주기 동안에 수신 한 데이터에만 접근한 경우이며 다음과 같이 정의할 수 있다.

$$CCM_{immediate} = (1 - h)(1 - e^{-\lambda L}) \quad (9)$$

그리고  $CCM_{def\ erred}$  는  $CCM_{immediate}$  의 경우를 제외한 모 든 경우이며 이동 트랜잭션은 현재 무효화 보고서에 접근 한 데이터가 존재하지 않아야 완료할 수 있다. 즉 이동 트랜잭션이 접근한 데이터는 현재 무효화 보고서 방송주 기 동안에 갱신되지 않은 경우로 다음과 같이 정의할 수 있다.

$$CCM_{def\ erred} = (1 - CCM_{immediate}) \times (e^{-\mu L}) \quad (10)$$

그러므로 CCM-AD 방법의 이동 트랜잭션 완료율  $P_{CCM}$  은 다음과 같이 정의할 수 있다.

$$P_{CCM} = (1 - h)(1 - e^{-\lambda L}) + (1 - CCM_{immediate}) \times (e^{-\mu L}) \quad (11)$$

### 5.2 무효화 보고서 전송량

제안한 CCM-AD 방법과 OCC-UTS<sup>2</sup> 방법은 서버에서 방 송하는 무효화 보고서를 수신하여 지연된 이동 트랜잭션의 일관성을 검증한다. OCC-UTS<sup>2</sup> 방법에서 서버는 이동 호스 트의 캐쉬 일관성을 유지하기 위하여 현재 무효화 보고서 방 송주기 동안에 갱신된 정보를 주기적으로 방송한다. 현재 무 효화 보고서 방송주기 동안에 서버에서 갱신된 데이터 수  $n_L$  은 서버의 데이터베이스의 아이템 개수를  $n$ 이라 할 때 다음과 같이 정의할 수 있다[2,9].

$$n_L = n(1 - e^{-\mu L}) \quad (12)$$

그리고 OCC-UTS<sup>2</sup> 방법의 무효화 보고서 크기  $IR_{OCC}$  는 [9]의 해석적 모델에서와 같이 데이터 갱신율이 지수분포 형 태를 따르는 경우 다음과 같이 정의할 수 있다[2].

$$IR_{OCC} = n_L \times \ln(n) \quad (13)$$

CCM-AD 방법은 현재 무효화 보고서 방송주기 동안에 이동 호스트에게 방송한 데이터가 갱신되는 경우 최신의 데 이터를 재방송함으로써 현재 무효화 보고서 방송주기 동안에 방송한 데이터의 캐쉬 일관성을 보장할 수 있다. 따라서 서 버는 데이터가 갱신되어도 현재 무효화 보고서 방송주기 동안에 이동 호스트에게 방송된 데이터는 무효화 보고서에 포함하지 않으므로 CCM-AD 방법의 무효화 보고서 크기  $IR_{CCM}$  은 다음과 같이 정의할 수 있다.

$$IR_{CCM} = n_L \times \ln(n) \times (1 - K) \quad (14)$$

집합  $K$ 는 현재 무효화 보고서 방송주기  $L$  동안에 이동 호스트에게 방송한 데이터 집합과 갱신된 데이터 집합과의 교집합이다. 그러므로 하나의 데이터가 집합  $K$ 에 포함될 확 률은 질의가 있을 확률과 데이터가 갱신될 확률을 이용하여 다음과 같이 정의할 수 있다.

$$K = (1 - h) \times (1 - e^{-\lambda L}) \times (1 - e^{-\mu L}) \quad (15)$$

그러므로 최종적인 CCM-AD 방법의 무효화 보고서 크기는 다음과 같이 정의할 수 있다.

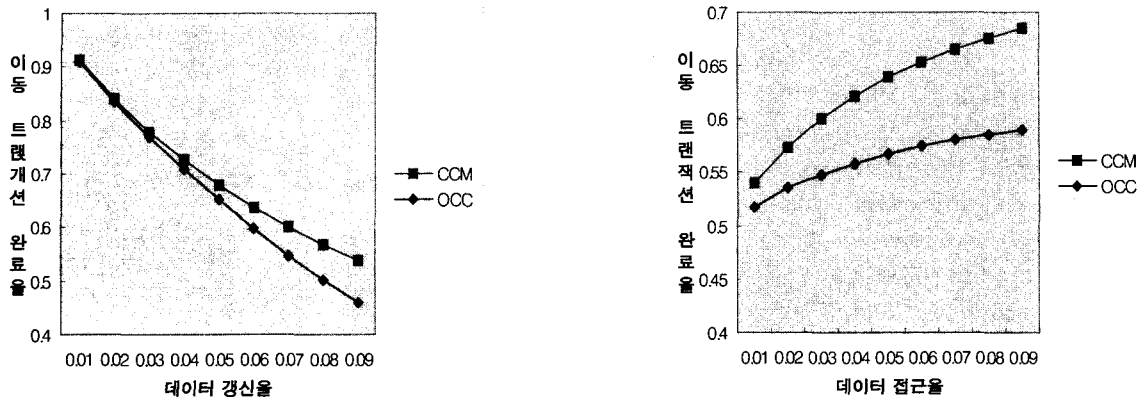
$$IR_{CCM} = n_L \times \ln(n) \times (1 - (1 - h) \times (1 - e^{-\lambda L}) \times (1 - e^{-\mu L})) \quad (16)$$

본 연구의 해석적 모델에 성능 분석을 위하여 적용하는 매개변수와 값들은 [2]에서 OCC-UTS<sup>2</sup> 방법의 분석에 이용 한 매개변수와 값을 적용한다. 각각의 매개변수와 값은 다음 <표 1>과 같다. 매개변수  $\lambda, \mu$  각각을 기준으로 하여 성능 평가를 실시하기 위해서 이들 각각에 기준이 되는 값과 동적 인 값을 정하여 실험의 객관성을 높였다. 그리고 [2]의 캐쉬 적중률을 고려하여 성능평가에 반영하였다.

<표 1> 적용하는 매개변수와 값

변수명	설	명	변수값
$L$	무효화 보고서 방송주기		10sec
$\lambda$	이동 호스트의 특정 데이터 접근율		0.03/동적
$\mu$	서버에서의 데이터 갱신율(hot spot)		0.07/동적
$h$	캐쉬 적중률		동적
$n$	데이터베이스의 데이터 아이템 갯수		500

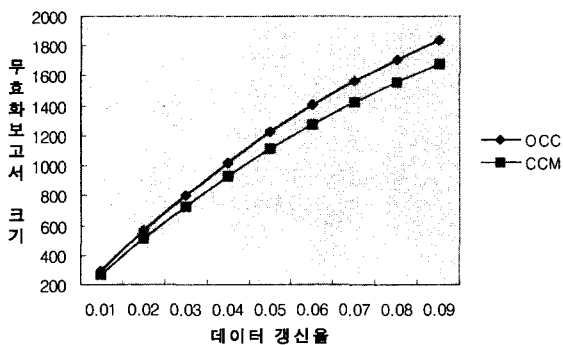




(그림 5) 이동 트랜잭션의 완료율

<표 1>의 매개변수를 식(5)와 식(8)에 적용하면 각 방법에서 이동 트랜잭션의 완료율을 구할 수 있다. 서버에서의 데이터 갱신율  $\mu$ 와 데이터 접근율  $\lambda$ 를 각각 기준으로 한 이동 트랜잭션의 완료율은 (그림 5)와 같다.

OCC-UTS<sup>2</sup> 방법은 데이터 갱신율이 높아지는 경우 즉시 완료 조건을 위반하여 무효화 보고서 수신 시점까지 지연되는 이동 트랜잭션이 늘어난다. 그리고 무효화 보고서에 포함된 데이터에 이동 트랜잭션이 접근하는 경우 이동 트랜잭션은 철회되므로 갱신율이 높을수록 많은 이동 트랜잭션이 철회된다. 그러나 제안한 CCM-AD 방법은 데이터의 갱신율이 높더라도 현재 무효화 보고서 방송주기 동안에 이동 호스트에게 방송한 데이터는 갱신 여부를 즉시 알려줌으로써 이동 트랜잭션 결정이 지연되지 않고 완료할 수 있다. 마지막으로 <표 1>의 매개변수를 식 (13), 식 (14)에 적용한 CCM-AD 방법과 OCC-UTS<sup>2</sup> 방법의 무효화 보고서 메시지 크기는 (그림 6)과 같다.



(그림 6) 무효화 보고서의 메시지 크기

(그림 6)과 같이 제안한 CCM-AD 방법은 서버에서의 데이터 갱신율이 높아지더라도 OCC-UTS<sup>2</sup> 방법에 비하여 방송되는 무효화 보고서의 메시지 크기가 작음을 알 수 있다.

### 6. 결론 및 향후 연구 방향

본 연구에서는 이동 트랜잭션의 직렬 가능한 수행을 보장하면서 이동 호스트가 스스로 이동 트랜잭션의 수행 및 지연 여부를 결정할 수 있는 CCM-AD 방법을 제안하였다. 제안한 CCM-AD 방법은 교집합  $K$ 를 이동 호스트가 요청한 데이터 방송시 함께 방송함으로써 현재 무효화 보고서 방송주기 동안에 캐칭된 데이터의 정확성을 보장하며 이동 트랜잭션의 즉시 완료 결정을 가능하게 해준다. 또한 CCM-AD 방법은 현재 무효화 보고서 방송주기 동안에 캐칭된 데이터가 서버에서 갱신되는 경우 교집합  $K$ 를 통하여 이동 호스트에게 알려줌으로써 이동 호스트의 캐쉬 일관성을 유지하기 위하여 서버에서 주기적으로 방송하는 무효화 보고서의 크기를 줄일 수 있다.

향후 연구 방향은 이동 호스트가 요구한 데이터에 대하여 보다 효율적인 방송을 통하여 가능한 한 대역폭의 사용을 줄이고 이동 호스트의 완료율을 높일 수 있는 방법에 대하여 연구하고자 한다.

### 참고 문헌

- [1] Kam-Yiu Lam, Mei-Wai Au and Edward Chan, "Broadcasting Consistent Data to Read-Only Transactions from Mobile Clients," The Computer Journal, Vol.45 No.2, pp.129-146, 2002.
- [2] D. Barbara and T. Imielinski, "Sleepers and Workaholics : Caching in Mobile Distributed Environments," Proc. 1994 ACM-SIGMOD Int'l Conf. Management of Data, pp.1-12, June, 1994.
- [3] S. Acharya, M. Franklin, and S. Zdonik, "Balancing push and pull for data broadcast," In Proceedings of ACM SIGMOD Conference on Management of Data, Tucson, AZ, USA, pp.183-194, May, 1997.

[4] T. Imielinski and S. Viswanathan, "Adaptive wireless information systems," In Proceedings of the Special Interest Group in DataBase Systems(SIGDBS) Confrence, Tokyo, Japan, pp.19-41, October, 1994.

[5] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. "Broadcast disks : Data management for sysmmertic communications environments," In Proceedings of ACM-SIGMOD Conference on Management of Data, San Jose, CA, USA, pp.199-210, May, 1995.

[6] S. Hameed and N. H. Vaidya, "Efficient Algorithms for scheduling data broadcast," ACM/Baltzer Journal of Wireless Networks(WINET), 5(3):183-193, 1999.

[7] Potoura, E.(1998) Supporting Read-Only Transactions in Wireless Broadcasting, Proceedings of the DEXA'98 Workshop on Mobility in Databases and Distributed Systems, 26-28 August, IEEE Online Publications, pp.428-433.

[8] Pitoura, E. & Chrysanthis, P. K.(1999) Scalable Proceeding of Read-Only Transaction in Broadcast Push. Proceeding of the 19th IEEE International Conference on Distributed Computing System, 31 May-4 Jun, IEEE Online Publications, pp.432-439.

[9] SangKeun LEE, "Caching and Concurrency Control in a Wireless Mobile Computing Environment," IEICE TRANS. INF. & SYST., Vol.E85-D, No.8, August, 2002.

[10] J. Jing, O. Bukhres, A. Elmagarmid and R. Alonso, "Bit-Sequences : A New Cache Invalidation Method in Mobile Environments," pp.1-25, 1993.

[11] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Data Broadcast," in Proceedings of ACM SIGMOD, Tucson, Arizona, May, 1997.

[12] W.-C. Lee, Q. L. Hu, and D. L. Lee, "Channel Allocation Methods for Data Dissemination in Mobile Computing Environments," 6th International Symposium in High Performance Distributed Computing(HPDC '97) Portland, USA, August, 1997.

[13] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing : Solutions and Challenges in Data Management," Communication of ACM. Vol.37. No.10. 1994.



**고 승 민**

E-mail : smgo@sunny.chonnam.ac.kr  
 2003년 호원대학교 전기전자통신공학부(학사)  
 2003년~현재 전남대학교 전산학과(석사)  
 관심분야 : 이동 컴퓨팅, XML, 임베디드 시스템



**김 대 인**

E-mail : dikim@chonnam.ac.kr  
 1996년 동신대학교 전자계산학(학사)  
 1998년 전남대학교 전산학과(석사)  
 2000년 전남대학교 전산학과(박사수료)  
 2004년~현재 전남대학교 전산학과 시간강사  
 관심분야 : 분산 처리 시스템, XML, 이동 컴퓨팅



**임 선 모**

e-mail : smim@dongkang.ac.kr  
 1987년 조선대학교 공과대학 컴퓨터공학과(공학사)  
 1989년 조선대학교 대학원 컴퓨터공학과(공학석사)  
 2004년 전남대학교 대학원 전산통계학과(박사과정)

1987년~1997년 광주은행 전산부 근무  
 1997년~2000년 동강대학 컴퓨터정보과 전임강사  
 2000년~현재 동강대학 컴퓨터정보과 조교수  
 관심분야 : 이동컴퓨팅, 트랜잭션 처리



**황 부 현**

E-mail : bhhwang@chonnam.ac.kr  
 1978년 승실대학교 전산학과(이학사)  
 1980년 한국과학기술원 전산학과(이학석사)  
 1994년 한국과학기술원 전산학과(공학박사)  
 1980년~현재 전남대학교 컴퓨터정보학부  
 관심분야 : 분산시스템, 분산데이터베이스, 객체지향시스템, 전자상거래