

비동기적 방송을 사용하는 이동 컴퓨팅 환경에서 캐쉬 일관성 유지 정책

김 대 옹[†] · 박 성 배^{††} · 황 부 현^{†††}

요 약

이동 컴퓨팅 환경에서 이동 사용자는 무선 통신망의 협소한 대역폭을 효율적으로 사용하기 위하여 자주 사용하는 데이터를 자신의 캐쉬에 저장하여 사용한다. 그러나, 이동호스트의 이동과 서버와의 통신 단절로 인하여 서버에 있는 데이터와 일관성이 위배되어 캐쉬된 데이터의 유용성이 상실될 수 있다. 기존 연구에서는 이동호스트의 캐쉬 일관성을 유지하기 위한 정책에 중점을 두었으며, 모든 이동 기지국에서 무효화 메시지를 방송하는 시점이 동일하다고 가정하였다. 본 논문에서는 이동호스트의 이동으로 인하여 발생될 수 있는 캐쉬 일관성 위배의 문제를 해결하는 캐쉬 일관성 유지 정책을 제안한다. 제안하는 정책은 각 이동 기지국이 무효화 메시지의 방송하는 시점을 다르게 함으로서 기존 정책에서 방송 시점을 일치시키기 위한 오버헤드를 줄일 수 있고, 이동 기지국이 이동호스트에게로 전송하는 메시지의 양을 줄일 수 있기 때문에 통신비용이 절감되고 무선 통신의 협소한 대역폭이 보다 효율적으로 사용될 수 있다.

A Strategy to maintain Cache Consistency in Mobile Computing Environments using the Asynchronous Broadcasting

Dae-Ong Kim[†] · Seong-Bae Park^{††} · Bu-Hyun Hwang^{†††}

ABSTRACT

In mobile computing environments, to efficiently use the narrow bandwidth of wireless networks a mobile host caches the data that are frequently accessed. But, because the cached data can be inconsistent with the data in a server due to the host mobility and the disconnection from a server, the usefulness of cached data may be lost. The traditional methods emphasize on the cache consistency and assume that broadcasting times are synchronized at all mobile support stations. In this paper, we propose a strategy to maintain cache consistency, which resolves the problems that be caused by the migration of mobile host. The proposed strategy has asynchronous broadcasting time and reduces the communication overhead caused by the synchronization. Also, by preventing the unnecessary messages transmitting from the mobile support station to a mobile host, this strategy can reduce the communication cost and use the narrow bandwidth of wireless networks efficiently.

※ 이 논문은 1997학년도한국과학재단 핵심전문 연구비에 의하여 연구되었음.

† 정 회 원 : 서강정보대학 전자계산과 교수

†† 정 회 원 : 순천제일대학 전자계산과 교수

††† 정 회 원 : 전남대학교 전산학과 교수

논문접수 : 1998년 3월 12일, 심사완료 : 1998년 8월 11일

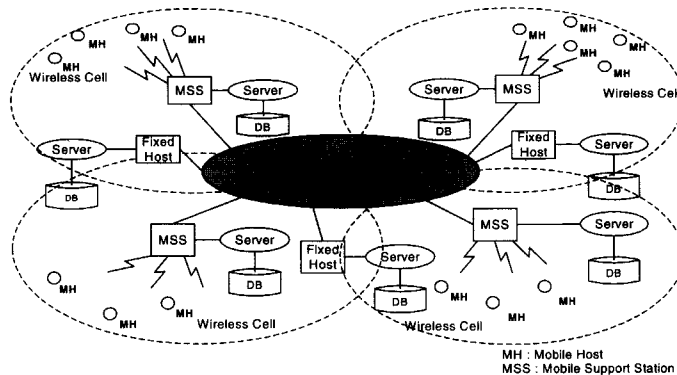
1. 서 론

오늘날 휴대용 컴퓨터의 대중화와 무선 통신망의 급속한 발전으로 인하여 기존의 분산 시스템과 다른 이동 컴퓨팅 환경이 출현하였다. 분산 시스템 환경에서 호스트의 연결과 위치는 변하지 않으나, 이동 컴퓨팅 환경은 분산 시스템 환경에 호스트의 이동성을 고려한 경우이다. 이동호스트는 이동을 하기 위해서 작고 가벼워야 하기 때문에, 이용할 수 있는 배터리의 양과 사용할 수 있는 자원은 한정된다. 이러한 제약으로 인하여 이동 컴퓨팅 환경은 기존의 분산 시스템 환경과 비교하여 다음과 같은 특징을 갖는다[3, 7, 8, 10, 14, 16, 17]. 첫째, 무선 통신망을 이용한다. 무선 통신은 유선 통신에 비하여 신뢰성이 낮고 협소한 대역폭을 갖는다. 협소한 대역폭으로 인하여 이동호스트가 이동 기지국과 교신할 메시지의 양이 많을 경우 무선 통신 채널에 대한 점유 경쟁이 발생될 수 있다. 둘째, 이동 사용자는 자유롭게 이동한다. 사용자의 이동성은 이동 컴퓨팅 환경의 가장 큰 특징이라 할 수 있는데, 이동 사용자들은 이동 중에도 이동 기지국과 연결을 유지하면서 지속적으로 작업을 수행하고자 한다. 셋째, 이동 사용자는 이동 기지국과 단절될 수 있다. 이동호스트는 지형적인 영향으로 이동 기지국과 단절될 수 있으며, 이동 사용자가 이동호스트의 배터리 수명을 연장하기 위하여 의도적으로 이동 기지국과 연결을 단절할 수 있다.

그림 1의 이동 컴퓨팅 모델에서 고정 호스트는 서버의 기능을 하고 이동호스트는 클라이언트의 역할을 한다. 한 지리적인 통신 영역 안에서 이동호스트(Mobile

Host : MH)와 통신을 하는 고정호스트를 이동 기지국(Mobile Support Station : MSS)이라 하며, MSS는 이동호스트들이 트랜잭션을 처리하는데 필요한 데이터를 제공하거나 이동호스트가 다른 호스트들과 통신할 수 있는 통로를 제공한다. 이 때, 각 MSS의 지리적인 통신 영역을 셀(Cell)이라 하며, 셀은 물리적으로 다른 셀과 중첩될 수 있으나 이동호스트는 단지 하나의 MSS와 통신이 가능하다. 그림 1에서 점선으로 된 타원은 셀을 나타내며, MSS를 포함한 고정호스트는 데이터베이스를 가지고 있으며, 고정호스트들은 고정 통신망을 이용하여 서로 메시지를 교환한다[2, 3, 8, 15].

기존의 캐쉬 일관성 유지 정책은 이동호스트의 이동과 단절 등으로 인하여 캐쉬 일관성이 위배되는 문제를 해결하는데 중점을 두고 연구가 진행되었으며, 이들 연구에서는 모든 MSS에서 무효화 메시지를 발송하는 시점이 동일한 경우로 가정하였다[4, 7, 9, 12, 16, 17]. 그러나, 일반적으로 각 MSS는 독립적으로 운영되는 서버이므로 다른 MSS의 간섭 없이 독자적으로 운용되어야 하므로 모든 MSS의 발송 시점을 일치시키는 데는 부가적인 과정이 요구되고, 이로 인하여 통신 대역폭에 대한 부담을 가중시키게 된다. 본 논문에서는 MSS들간의 발송 주기는 동일하나 발송 시점이 다를 때 이동호스트가 다른 셀로 이동하여 캐쉬 일관성이 위배되는 문제를 해결하기 위한 정책인 CCAB(Cache Consistency on the Asynchronous Broadcasting) 정책을 제안하고자 한다. 제안하는 정책은 이동 컴퓨팅 환경에서 2PC(Two Phase Commit) 프로토콜을 지원하는 중복 데이터베이스 시스템을 기반으로 한다. 제안하는 정책에서 이동호스트는 캐쉬 데이터 중



(그림 1) 이동 컴퓨팅 모델
(Fig. 1) The Mobile Computing Model

에서 갱신가능 데이터에 대한 일관된 정보를 얻기 위하여 데이터의 식별자만을 MSS에게 전송한다. 그리고, MSS는 이동호스트가 요청한 일관된 정보와 이동호스트의 이동으로 인하여 수신하지 못한 갱신 결과들을 이동호스트에게 전송하여 캐쉬 일관성을 유지하도록 한다. 이 정책은 조정자와 2PC 프로토콜을 고려하여 MSS에서 이동호스트로의 불필요한 전송을 예방하기 때문에 메시지의 양을 줄일 수 있어서 무선 통신 채널에 대한 부담을 줄일 수 있고, 각 MSS에서 비동기적으로 방송하여도 캐쉬 일관성을 유지할 수 있으므로 방송 시점을 일치시키기 위한 오버헤드를 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 캐쉬 일관성을 다룬 기존의 연구들을 살펴 보고, 3장에서는 MSS들간의 방송 주기는 동일하나 방송 시점이 비동기화되었을 때 캐쉬 일관성이 위배되는 문제점을 정의 하며, 4장에서는 이동 트랜잭션 처리 모델을 제시한다. 5장에서는 캐쉬 일관성 위배를 해결하는 기본 정책과 이를 개선한 CCAB 정책을 제안하고 성능을 분석하며, 마지막으로 6장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

이동 컴퓨팅 환경에서 이동호스트를 사용하는 사용자를 이동 사용자(Mobile User)라 하고 이동호스트에 의해 제출된 트랜잭션을 이동 트랜잭션(Mobile Transaction)이라 하며, 이동 사용자는 장소에 상관없이 이동 트랜잭션을 이용하여 데이터베이스를 검색한다.

이동 사용자가 캐쉬를 사용하면, 이동 트랜잭션들이 필요한 데이터를 MSS에게 요청하지 않고 캐쉬로부터 가져와 수행하므로, MSS로부터의 메시지 전송량을 줄일 수 있어서 무선 통신의 협소한 대역폭을 효율적으로 사용할 수 있고 이동 트랜잭션의 연산 속도를 향상시킬 수 있으며 단절 기간 동안에도 이동 트랜잭션의 수행을 계속할 수 있어서 시스템의 성능은 향상된다 [13]. 그러나, 이동 사용자의 이동이나 MSS와의 통신 단절로 인하여 캐쉬 데이터는 MSS에 있는 데이터와 일관되지 않을 수 있다. 캐쉬 일관성이 위배되면, 캐쉬 데이터를 이용하는 이동 트랜잭션의 정확성은 보장될 수 없으므로 캐쉬 일관성 유지 정책은 중요하다.

각 MSS는 캐쉬 일관성을 유지하기 위하여 무효화 메시지를 방송한다. 이동호스트가 무효화 메시지를 수

신하면 이동호스트는 무효화 메시지에 포함된 정보를 이용하여 캐쉬 데이터를 수정하거나 제거하여 캐쉬 일관성을 유지한다. 무효화 메시지를 방송하는 정책으로는 MSS에서 갱신이 발생하는 즉시 그 결과를 방송하는 즉각적인 캐쉬 무효화 정책과 일정한 시간동안에 갱신된 내용을 모아 주기적으로 방송하는 주기적인 캐쉬 무효화 정책이 있다. 무효화 메시지에 갱신된 데이터에 대한 메시지만을 포함하는 경우와 추가로 갱신가능 데이터에 대한 메시지를 포함하는 경우가 있다. 갱신가능 메시지는 2PC 프로토콜의 수행 과정에서 조정자가 vote 요청을 모든 참여자에게 보낸 후 참여자로부터 yes 또는 no 응답을 기다리는 동안에 방송 시점에 도달하였을 때 MSS가 방송하는 메시지이며, 참여자에서는 조정자에게 yes 응답을 보내고 commit 또는 abort 메시지를 기다리는 동안에 방송 시점에 도달하였을 때 MSS가 방송하는 메시지를 말한다[16].

이동 트랜잭션이 수행될 때 필요한 데이터가 캐쉬에 없으면 이동호스트는 해당 데이터를 MSS에게 요청하여 캐쉬한 후 연산을 수행한다. 이동호스트가 MSS에게 데이터를 캐쉬하기 위하여 요청할 때, MSS가 응답하는 방법에는 이동호스트가 요청한 데이터를 다음 방송 시점에 무효화 메시지와 함께 방송하는 방법[4]과 즉시 그 결과를 이동호스트에게 전송하는 방법[9, 15]으로 구분할 수 있다.

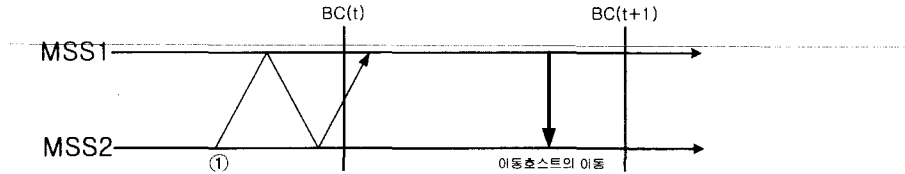
다음은 각 MSS에서 무효화 메시지의 방송 시점이 동일한 환경에서 캐쉬 일관성을 유지하기 위한 기존 정책들을 기술한다.

[4]에서는 무효화 메시지를 방송하였으나 이동호스트가 통신 단절로 인하여 방송된 무효화 메시지를 수신하지 못하였을 때, 캐쉬 일관성을 유지하는 방법으로 TS(broadcasting TimeStamps) 방법, AT(Amnestic Terminals) 방법을 제안하였다. TS 방법은 MSS가 일정한 크기 $w(=kL, L$ 은 방송주기)의 윈도우안에 갱신된 데이터의 정보를 기억하고 있다. 이동호스트가 MSS와 통신이 단절된 후 다시 연결을 요청하면, MSS는 현재의 타임스탬프 $T_i=iL$ 에서 윈도우에 다음과 같은 목록 U_i 를 유지하여 무효화 메시지를 방송한다.

$$U_i = \{[j, t_j] \mid j \in D\}$$

여기서, D 는 데이터베이스에 있는 데이터의 집합이고 t_j 는 $T_i - w \leq t_j \leq T_i$ 인 j 의 마지막 갱신의 타임스탬프이다.

AT 방법에서, 무효화 메시지는 마지막 방송주기 L 동안에 갱신된 데이터의 정보를 포함한다. 두 알고리



(그림 2) 무효화 메시지에 갱신만을 포함하는 기존 정책에서 발생하는 문제점
 (Fig. 2) The problems happened in the traditional strategies that include only the update in invalidation message.

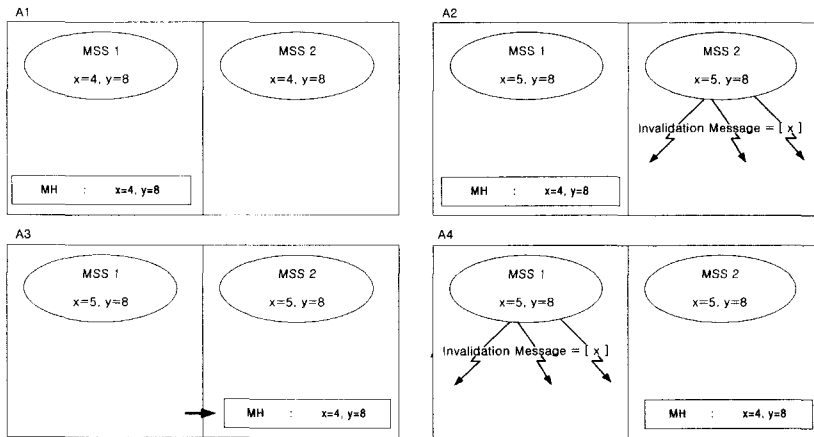
증에서 그들의 단절 기간이 특정 길이(TS 방법에서는 w , AT 방법에서는 L)를 넘어설 때 모든 캐쉬 데이터를 제거한다.

[16]에서는 무효화 메시지의 내용에 갱신 메시지뿐만 아니라 갱신가능 메시지를 포함하여 갱신 메시지를 포함하는 기존 정책에서 해결하기 어려운 문제들을 해결하였다. 그림 2에서 갱신 메시지를 포함하는 기존 정책에서 발생하는 문제점과 이를 해결하는 방법을 보였다. 그림 2의 ①과 같은 경우에 MSS1에서는 갱신 결과를 방송하지 않았지만 MSS2에서는 갱신 결과를 방송한 후에 이동호스트가 MSS2로 이동함으로써 이동호스트의 캐쉬 일관성은 유지될 수 없다. 이와 같은 문제점을 해결하기 위하여 갱신가능 메시지를 이용하였다. 갱신가능 메시지를 이용하면, ①의 경우에 이동호스트가 MSS1에서 갱신가능 메시지를 수신하고 MSS2로 이동하므로 이동호스트는 이동한 후 캐쉬 데이터 중에서 갱신가능 데이터에 대한 일관된 정보를 MSS2에게 요청하여 캐쉬 일관성을 유지하도록 하였다.

3. 문제 정의

일반적으로, 분산 시스템 환경에서 각 MSS는 다른 MSS에 종속되지 않고 독립적으로 운영된다. 그러나, 이동 컴퓨팅 환경에서 캐쉬 일관성 유지를 위한 연구는 MSS의 방송 주기와 시점을 일치시켰다. 모든 MSS들의 방송 시점을 일치시키기 위해서 어떤 MSSi는 다른 모든 MSSj에게 방송 시점을 일치시키기 위한 메시지를 전송하고 그들로부터 응답을 받은 시점을 중심으로 방송 시점을 조정한 후 결정된 방송 시점을 최종 통보하고, 약속된 시점에 무효화 메시지를 방송한다. 그러나 이와 같은 방법은 매우 복잡하고 어려운 일이며, 각 MSS들이 방송 시점을 일치시키기 위하여 한 MSSi에 방송 시점의 동기화 작업이 집중되므로 분산 시스템의 근본적인 목표인 MSS들의 독립성을 저해하는 요인이 된다.

이 장에서는 MSS들의 독립성을 보장하기 위하여 MSS들간의 방송 시점이 다른 2PC 프로토콜을 지원하



(그림 3) 방송 시점의 차이로 인한 캐쉬 일관성의 위배
 (Fig. 3) The violation of the cache consistency caused from the asynchronous broadcasting

는 중복 분산 데이터베이스의 이동 컴퓨팅 환경에서 이동호스트가 인접 셀로 이동할 때 발생하는 문제점들을 정의한다.

그림 3은 각 MSS들의 방송 시점이 다를 때, 이동호스트의 이동으로 인하여 캐쉬 일관성이 위배되는 경우를 보여 준다. 그림 3의 A1에서 MSS1과 MSS2, 그리고 이동호스트의 캐쉬는 모두 일관된 데이터를 유지하고 있다. A2에서는 임의의 MSS에서 발생한 x에 대한 갱신 결과가 MSS1과 MSS2를 비롯한 모든 MSS에 반영되었고, MSS2에서는 이 갱신 결과를 방송하였다. A3에서는 이동호스트가 MSS1에서 MSS2로 이동하였고, A4에서 MSS1이 x에 대한 갱신 결과를 방송하였다.

결과적으로, 그림 3에서는 임의의 MSS에서 발생한 x의 갱신에 대하여 MSS1과 MSS2를 비롯한 모든 MSS가 방송 시점에 그 갱신 결과를 방송하였으나, 이동호스트가 MSS1에서 MSS2로 이동함으로써 무효화 메시지의 수신 기회를 상실하게 되어 이동호스트의 캐쉬 데이터는 일관성을 유지하지 못한다. 이와 같이 이동호스트의 캐쉬가 일관성을 유지하지 못하는 이유는 이동호스트가 이동하기 전에 있던 MSS1에서는 이동호스트가 이동한 후에 갱신 결과를 방송하였고 이동한 MSS2에서는 이동하기 전에 이미 방송이 이루어졌기 때문이다.

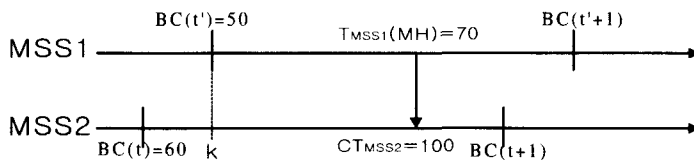
4. 이동 트랜잭션 처리 모델

다음은 본 논문에서 사용하는 이동 트랜잭션 처리 모델에 대한 환경을 기술한다. 첫째, 이동 트랜잭션은 짧은 시간을 요구하는 트랜잭션이다. 이동 사용자는 이동 중이거나 출장 중에 주식시장이나 전자잡지, 여행정보, 이동쇼핑, 전자메일 서비스 등과 같은 단순한 업무에 필요한 정보를 얻기 위하여 데이터베이스에 접근하기 때문에 보편적으로 짧은 수행 시간을 요구하는

트랜잭션을 이동호스트에서 수행시킨다. 둘째, 각 셀에 있는 이동호스트는 해당 MSS에 동기화된다. 한 셀안에 있는 이동호스트는 해당 MSS의 데이터베이스를 이용하여 캐쉬 일관성을 유지하고 MSS는 셀내의 모든 이동호스트를 지원하므로 이동 트랜잭션은 MSS에 동기되어 수행되어야 한다. 셋째, MSS는 이동호스트의 캐쉬 데이터 요청에 대하여 즉시 응답한다. 넷째, 고정 통신망은 중복분산 데이터베이스에서 ROWA (Read One Write All) 또는 QCP(Quorum Commit Protocol)를 이용하며, 트랜잭션 완료 프로토콜은 2PC 프로토콜이다. 이동 컴퓨팅 환경에서 이동호스트가 MSS로부터 데이터를 캐쉬하고자 할 때, 요구한 데이터가 자신의 MSS에 존재하지 않으면 데이터를 캐쉬하는데 시간이 소요되고 이 기간에 이동호스트의 위치가 변동될 수 있으므로 MSS는 데이터를 응답할 때마다 이동호스트의 위치를 확인하여야 한다. 그러므로, 이동 컴퓨팅 환경에서 데이터베이스의 완전 중복은 필수적이다. 또한, 2PC 프로토콜은 중복 분산 데이터베이스에서 가장 많이 적용되는 완료 프로토콜이다[5]. 다섯째, 각 MSS는 무효화 메시지를 주기적으로 방송하고 방송 주기는 동일하나 방송 시점은 다를 수 있다. 여섯째, MSS가 방송하는 무효화 메시지에는 갱신 메시지와 갱신가능 메시지를 포함한다.

5. CCAB 정책

이 장에서는 3장에서 제시한 문제들을 해결하기 위한 캐쉬 일관성 유지 정책을 제안한다. 제안하는 정책에서 우선적으로 고려하여야 할 점은 이동호스트가 MSS1에서 MSS2로 이동할 때 이동호스트가 MSS1에서 무효화 메시지를 마지막으로 수신한 방송 시점 $BC(t')$ 를 MSS2의 시각에 일치시키는 과정이다. 그림 4는 $BC(t')$ 를 MSS2의 시각으로 환산한 시점 k 를 구하는 과정을 보인다. 이동호스트가 MSS1로부터 수신한 방송 시점 $BC(t')$



(그림 4) 이동호스트가 이동한 후의 방송 시점의 조정
(Fig. 4) The adjustment of the broadcasting time after the moving of the mobile host

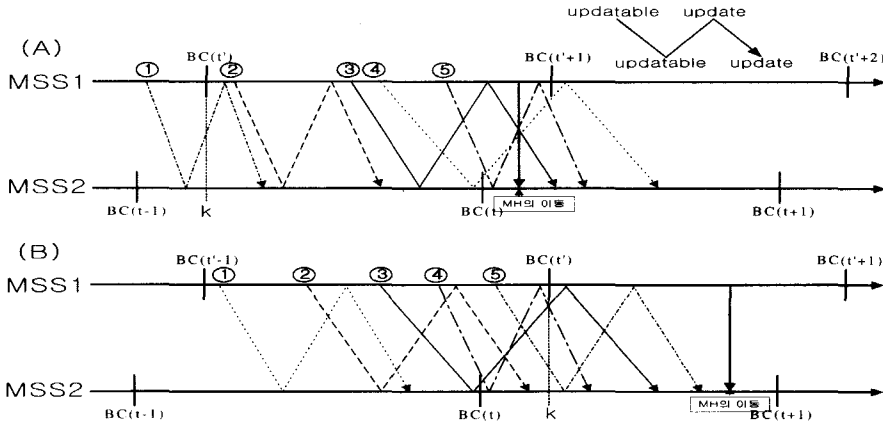
와 이동할 때의 MSS1의 시점 $T_{MSS1}(MH)$ 를 MSS2에 보내면, MSS2는 이동호스트로부터 수신 받은 시점 CT_{MSS2} 를 이용하여 k 의 값을 구한다.

$$k = CT_{MSS2} - (T_{MSS1}(MH) - BC(t)) \dots \dots \dots (식 1)$$

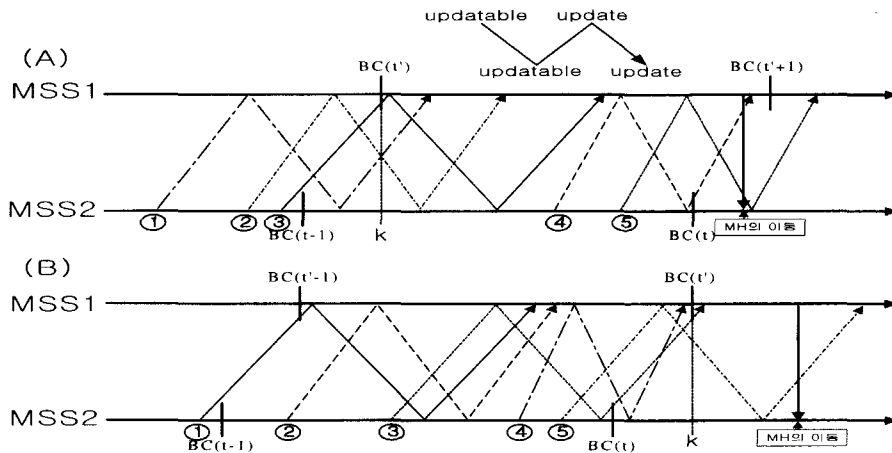
예를 들어, 그림 4에서 MSS1의 마지막 방송 시점 $BC(t)$ 을 50, 이동호스트가 MSS1에서 MSS2로 이동하는 시점 $T_{MSS1}(MH)$ 를 70이라 하고, MSS2가 이동호스트로부터 메시지를 수신하는 시점 CT_{MSS2} 를 100이라고 한다면, (식 1)에 의하여 $k = 50 + (100 - 70) = 80$ 이 된다.

5.1 기본 정책

이제 본 논문에서 제안하려는 캐쉬 일관성 유지 정책을 제안하기 위하여 이동호스트의 이동으로 인하여 캐쉬 일관성이 위배되는 다양한 경우들을 고려한다. 이동호스트는 MSS1에서 MSS2로 이동한다고 가정한다. 그림 5, 그림 6, 그리고 그림 7에서는 이동호스트가 이동하는 시점을 중심으로 2PC 프로토콜을 사용하는 트랜잭션이 완료되는 과정을 보여준다. 그림 5에서는 조정자가 MSS1인 경우, 그림 6에서는 조정자가 MSS2인 경우, 그리고 그림 7에서는 조정자가 MSS3인 경우로 구분하고, 각 경우에 대하여 $k < BC(t)$ 인 (A)의 경우와 $k > BC(t)$ 인 (B)의 경우로 구분하였다.



(그림 5) 조정자가 MSS1일 때, 이동호스트의 이동과 트랜잭션 완료 과정
 (Fig. 5) The course of mobile host's mobility and transaction's commit when the coordinator is MSS1



(그림 6) 조정자가 MSS2일 때, 이동호스트의 이동과 트랜잭션 완료 과정
 (Fig. 6) The course of mobile host's mobility and transaction's commit when the coordinator is MSS2

(1) 조정자가 MSS1인 경우

그림 5의 (A)에서 ③, ④, ⑤의 경우는 MSS2에서 마지막으로 방송한 무효화 메시지에 갱신 결과가 반영되지 않았으므로 이동호스트가 이동한 후에도 캐쉬 일관성이 유지되는 예이고, ①과 ②의 경우는 MSS1에서는 마지막 방송 후에 갱신이 발생되었고, MSS2에서는 갱신 결과를 방송한 후에 이동호스트가 이동하므로 갱신 결과를 수신할 기회를 상실하게 되어 캐쉬 일관성이 위배되는 예를 보여 주고 있다. ①과 ②의 경우를 구체적으로 살펴보면, ①의 경우는 이동호스트가 MSS1의 방송 시점 $BC(t')$ 에서 갱신가능 메시지를 수신하였지만, ②의 경우는 이동호스트가 MSS1에서 갱신에 대한 어떠한 정보도 얻지 못하였다. ①과 ②의 경우에 이동호스트는 캐쉬 데이터 중에서 갱신가능 데이터의 식별자를 MSS2에게 전송하고 MSS2는 갱신가능 데이터에 대한 일관된 정보(갱신 여부와 갱신 결과)와 k 부터 $BC(t)$ 사이에서 발생한 갱신을 이동호스트에게 전송하여 캐쉬 일관성을 유지하도록 한다. 그림 5의 (B)에서 ①, ②, 그리고 ④의 경우는 이동호스트가 갱신된 결과를 수신한 상태에서 이동하므로 캐쉬 일관성에 영향을 미치지 않으며, ③과 ⑤의 경우는 갱신가능 메시지를 수신한 상태에서 이동하므로 이동호스트는 이동한 후 MSS2에게 갱신가능한 캐쉬 데이터의 일관된

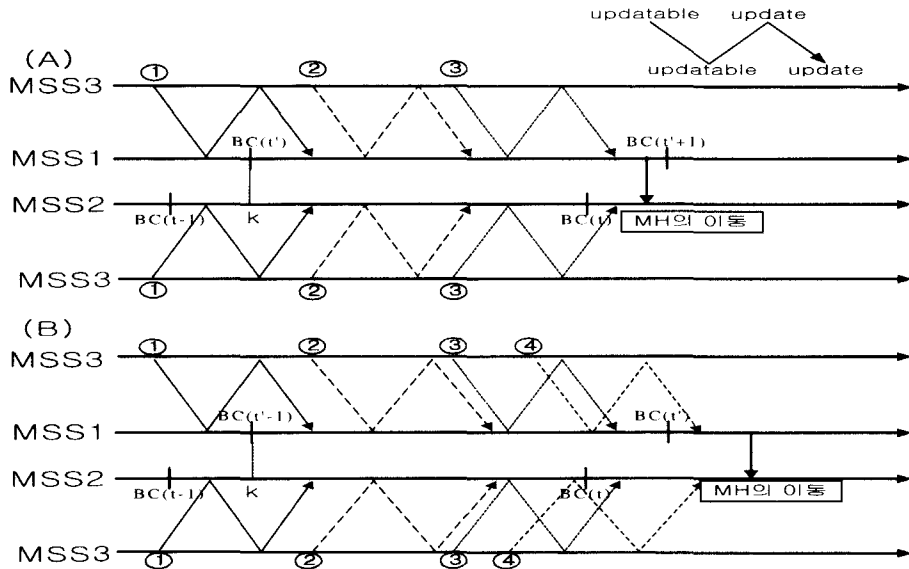
정보를 요청하여 캐쉬 일관성을 유지한다.

(2) 조정자가 MSS2인 경우

그림 6의 (A)에서 ①, ②, ③, ④의 경우는 그림 5의 (A)의 ①, ②처럼 이동호스트의 이동으로 인하여 캐쉬 일관성이 위배되는 예이고, ⑤의 경우는 그림 5의 (A)의 ③, ④, ⑤처럼 캐쉬 일관성을 유지하는 예이다. ①의 경우는 그림 5의 (A)의 ①에 해당되지만 MSS2에서의 갱신 시점이 그림 5의 (A)의 ①은 k 시점 이후이지만 그림 6의 (A)의 ①은 k 시점 이전이므로 k 부터 $BC(t)$ 사이의 갱신 정보로는 일관성을 유지할 수 없다. 따라서, 이동호스트는 캐쉬 데이터 중에서 갱신가능 데이터에 대한 일관된 정보를 MSS2에게 요청하여 캐쉬 일관성을 유지한다. 그림 6의 (B)에서 ①, ②, ④의 경우는 그림 5의 (B)의 ①, ②, ④와 동일하며, ③과 ⑤의 경우는 그림 5의 (B)의 ③과 ⑤와 동일하므로 캐쉬 일관성을 유지하는 방법도 동일하다.

(3) 조정자가 MSS3인 경우

그림 7의 (A)에서 ③의 경우는 그림 5의 (A)에서 ③부터 ⑤의 경우처럼 이동한 후에도 캐쉬 일관성을 유지하나, ①과 ②의 경우는 그림 5의 (A)에서 ①과 ②의 경우처럼 캐쉬 일관성이 위배된다. ①의 경우는



(그림 7) 조정자가 MSS3일 때, 이동호스트의 이동과 트랜잭션 완료 과정

(Fig. 7) The course of mobile host's mobility and transaction's commit when the coordinator is MSS3

그림 5의 (A)에서 ①의 경우와 동일한 경우이고, ②의 경우는 그림 5의 (A)에서 ②의 경우와 동일한 경우이므로 캐쉬 일관성을 유지하는 방법도 동일하다. 그림 7의 (B)에서 ①, ②, ③의 경우는 그림 5의 (B)에서 ①의 경우와 같으며, ④의 경우는 MSS1과 MSS2에서 갱신가능을 방송한 경우이므로 캐쉬 일관성을 유지한다.

위의 경우들을 종합하면, 각 그림에서 이동호스트의 이동으로 인하여 캐쉬 일관성이 위배되는 경우는 (A)의 경우에 존재한다. 캐쉬 일관성이 위배되는 경우들의 공통점은 이동호스트가 MSS1에서는 갱신 메시지를 수신하지 못하고 MSS2의 셀로 이동하며, MSS2는 갱신 결과를 이미 방송한 경우들이다. 이러한 경우들에 대하여 이동호스트가 다른 셀로 이동하면 이동호스트는 캐쉬 데이터 중에서 갱신가능한 데이터의 식별자를 그 셀의 MSS에게 전송하고, MSS2는 이동호스트로부터 수신된 갱신가능 데이터에 대한 일관된 정보와 이동호스트가 갱신 결과를 수신하지 못한 정보(그림 5, 그림 6, 그리고 그림 7에서 k부터 BC(t) 사이에 갱신되어 방송된 정보)를 전송하여 이동호스트의 캐쉬 일관성이 유지되도록 한다.

지금까지 MSS간에 방송 시점이 다른 경우에 캐쉬 일관성을 유지하는 기본 정책을 기술하였다. 이제는 제안한 정책을 기반으로 하여 알고리즘을 제시한다.

/* 기본 정책에서 캐쉬 일관성을 위하여 이동호스트가 MSS에게 보내는 메시지 */

```

Procedure Sending_to_MSS;
BEGIN
    send BC(t'), TMSS1(MH), the updatable
    cached data_id, and MSS_id
END
    
```

위 알고리즘에서, 이동호스트는 인접 셀로 이동하면 현재 자신이 가지고 있는 시점 $T_{MSS1}(MH)$ 과 마지막으로 무효화 메시지를 수신한 시점 $BC(t')$, 갱신가능한 캐쉬 데이터의 식별자, 그리고 이동하기 전의 MSS의 식별자를 이동한 후의 MSS에게 보낸다.

/* 이동호스트의 캐쉬 일관성을 유지하기 위하여 MSS가 이동호스트에게 응답할 정보를 구하는 알고리즘 */

```

Procedure Responsing_to_MH;
BEGIN
    
```

$$k = CT_{MSS2} - (T_{MSS1}(MH) - BC(t))$$

```

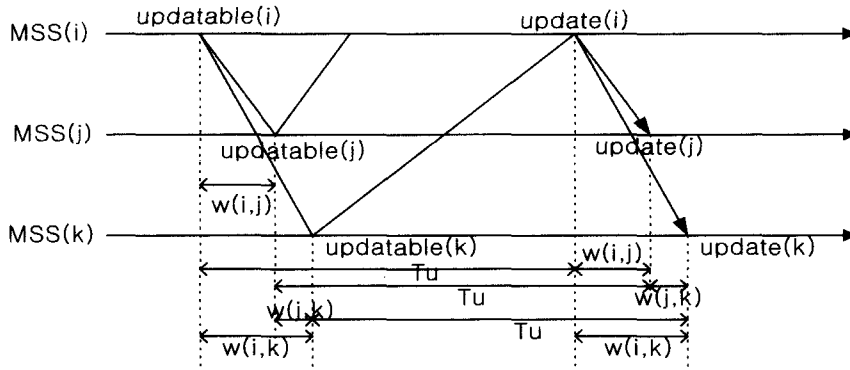
FOR the updatable cached data transmitted
from MH
    construct the Resp_Mess1
END_FOR
IF ( k < BC(t) )
    THEN construct the Resp_Mess2 with the
        updated result between k and BC(t)
    ENDIF
construct the Resp_Mess from the Resp_Mess1
and the Resp_Mess2
transmit the Resp_Mess
END
    
```

위 알고리즘에서 $BC(t')$ 는 이동호스트가 이동하기 전의 MSS(MSS1)에서 마지막으로 무효화 메시지를 수신한 방송 시점이며, $BC(t)$ 는 이동호스트가 이동한 후의 MSS(MSS2)에서 마지막으로 방송한 방송 시점이다. k는 MSS1에서의 방송 시점 $BC(t')$ 를 MSS2의 시간으로 환산한 시점이며, $T_{MSS1}(MH)$ 는 MSS1에서 이동호스트의 이동 시점이며, CT_{MSS2} 는 이동호스트가 MSS2로 이동해 온 시점이다. Resp_Mess1은 이동호스트가 MSS2로 이동하여 갱신가능한 캐쉬 데이터에 대하여 일관된 정보를 요청하였을 때 MSS2가 이동호스트에게 전달할 갱신 데이터이다. Resp_Mess2는 k부터 BC(t) 사이에 갱신된 데이터이다. Resp_Mess1의 데이터와 Resp_Mess2의 데이터가 중복될 때, 중복된 Resp_Mess1의 데이터를 제거한 후 Resp_Mess1의 데이터와 Resp_Mess2의 데이터를 합하여 Resp_Mess를 구성하고 이를 이동호스트에게 전달한다.

5.2 CCAB 정책

지금까지는 기본 정책에 대하여 설명하였고, 이제부터는 지금까지 설명된 기본 정책을 기반으로 교환되는 메시지의 양을 줄일 수 있는 CCAB(Cache Consistency on the Asynchronous Broadcasting) 정책을 제안한다. CCAB 정책에서 고려한 점은 MSS에서 수행되는 갱신 트랜잭션의 완료 과정에서 발생하는 지연 시간이다.

그림 8에서는 MSS간의 갱신가능 시점의 차이와 갱신 시점의 차이, 그리고 각 MSS에서의 갱신가능 시점부터 갱신 시점까지의 시간 등을 나타낸다. 임의의 조정자



(그림 8) MSS간의 갱신가능 시점 및 갱신 시점의 관계
 (Fig. 8) The relation of updatable_time and update_time among MSSs

MSS(i)에서 다른 임의의 한 참여자 MSS(j)로 메시지를 전송하는 시간은 동일하므로, MSS(i)와 MSS(j)에서 갱신가능 시점간의 차이 $|updatable(i) - updatable(j)|$ 와 갱신 시점간의 차이 $|update(i) - update(j)|$ 는 같으며, 이를 $w(i,j)$ 로 표현한다. 또한 MSS(i)와 MSS(k)간에도 같은 관계가 성립하여 $w(i,k)$ 가 성립한다. 그리고, 임의의 두 참여자 MSS(j)와 MSS(k)에서도 갱신가능 시점간의 차이와 갱신 시점간의 차이가 존재하고 이들의 값은 같으며, $w(j,k)$ 로 표현한다. 조정자 MSS(i)가 다른 모든 MSS에게 vote 요청을 하여 모든 참여자로부터 yes를 받아 갱신 할 때까지 소요되는 시간을 $T_u(i)$ 라 하면 $T_u(i) = update(i) - updatable(i)$ 이다. 참여자 MSS(j)에서는 yes를 응답한 후(갱신가능 시점) 조정자로부터 갱신 메시지를 수신하여 갱신 하는 시점까지 소요되는 시간은 다음 식에 의하여 $T_u(j)$ 와 같다.

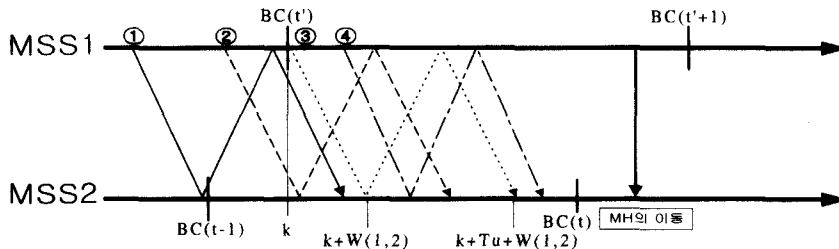
$$T_u(j) = update(j) - updatable(j) = \{update(i) - w(i,j)\}$$

$$\begin{aligned} &= \{updatable(i) - w(i,j)\} \\ &= update(i) - updatable(i) \\ &= T_u(i) \end{aligned}$$

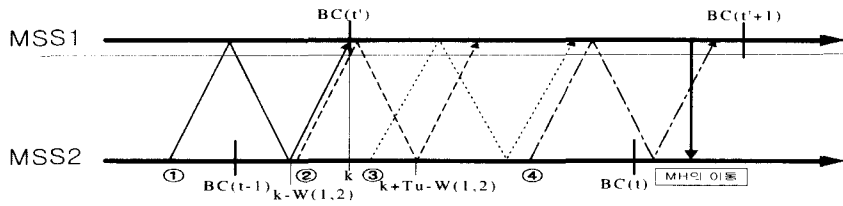
따라서, 모든 MSS에서 갱신가능 시점부터 갱신 시점까지의 시간은 T_u 로 동일하다. 다음은 캐쉬 일관성이 위배되는 그림 5, 6, 7의 (A)의 경우만을 고려한다.

(1) 조정자가 MSS1인 경우

그림 9에서 ①의 경우는 MSS1의 BC(t') 시점에서 갱신 결과를 이동호스트가 수신하고 이동하므로 캐쉬 일관성은 유지된다. 이 경우에, MSS2에서는 $k+w(1,2)$ 이전에 갱신이 발생하므로 k부터 $k+w(1,2)$ 사이의 갱신은 전송할 필요가 없다. MSS2에서 $k+w(1,2)$ 이후에 발생한 갱신들 중에서 $k+w(1,2)$ 부터 $k+w(1,2)+T_u$ 사이에서 발생한 갱신은 이동호스트가 BC(t) 시점에서 방송한 갱신가능 데이터에 대한 갱신이다. 이와 같은 ②



(그림 9) 조정자가 MSS1인 경우에 T_u 와 $w(1,2)$ 가 캐쉬 일관성에 미치는 영향
 (Fig. 9) The effect of T_u and $w(1,2)$ upon cache consistency when the coordinator is MSS1



(그림 10) 조정자가 MSS2인 경우에 T_u 와 $w(i,j)$ 가 캐쉬 일관성에 미치는 영향
 (Fig. 10) The effect of T_u and $w(i,j)$ upon cache consistency when the coordinator is MSS2

의 경우에 이동호스트는 MSS2로 이동한 후 MSS1에서 수신한 갱신가능한 캐쉬 데이터의 갱신 결과를 요청하고, MSS2는 이 요청에 대한 갱신이 있을 때 이 갱신 결과를 이동호스트에게 전송한다. ③과 ④의 경우처럼 $k+w(1,2)+T_u$ 이후부터 $BC(t)$ 사이에 발생한 갱신은 이동호스트가 두 MSS에서 갱신된 데이터에 대한 방송을 듣지 못하였기 때문에 MSS2는 이 범위에서 발생한 모든 갱신을 전송하여 이동호스트의 일관성을 유지한다. 이동호스트가 요청한 $k+w(1,2)$ 부터 $k+w(1,2)+T_u$ 사이에서 갱신된 데이터가 $k+w(1,2)+T_u$ 이후에서 갱신된 데이터에 있으면 $k+w(1,2)+T_u$ 이전의 갱신 결과를 삭제하고 이동호스트에게 전송하여 캐쉬 일관성을 유지하도록 한다.

(2) 조정자가 MSS2인 경우

그림 10의 ①의 경우는 그림 9의 ①의 경우처럼 이동호스트의 캐쉬 일관성에 영향을 미치지 않는다. MSS2에서 $k-w(1,2) \leq \text{update}(2) < k-w(1,2)+T_u$ 인 그림 10의 ②의 경우는 이동호스트가 MSS1의 $BC(t')$ 시점에서 갱신가능을 수신한 것으로 그림 9의 ②의 경우처럼 해결한다. MSS2에서 $k-w(1,2)+T_u \leq \text{update}(2) < BC(t)$ 인 그림 10의 ③의 경우는 그림 9의 ③과 ④의 경우처럼 해결한다.

(3) 조정자가 MSS3인 경우

조정자가 MSS1과 MSS2가 아닌 경우에 갱신 시점이 MSS1과 MSS2에서 다를 수 있으나 MSS1과 MSS2가 인접하므로 조정자로부터의 전송 시간은 큰 차이는 없을 것이다. 따라서, MSS1과 MSS2에서의 갱신 기점은 동일한 것으로 가정한다. 그림 11의 ①의 경우는 그림 9의 ②의 경우처럼 MSS1에서는 $BC(t')$ 시점의 방송에서 갱신가능을 방송하고 MSS2에서는 k 부터 $k+T_u$ 사이에서 갱신되어 캐쉬 일관성이 위배되는 경우로 이동호스트가 이동한 후 갱신가능한 캐쉬 데이터에 대하여 일관된 정보를 요청하면 MSS2에서는 이 데이터에 대한 k 부터 $k+T_u$ 사이에서의 갱신 정보를 전송한다. ③의 경우는 그림 9의 ④의 경우처럼 MSS2는 이 범위에서 발생한 모든 갱신을 이동호스트에게 전송한다.

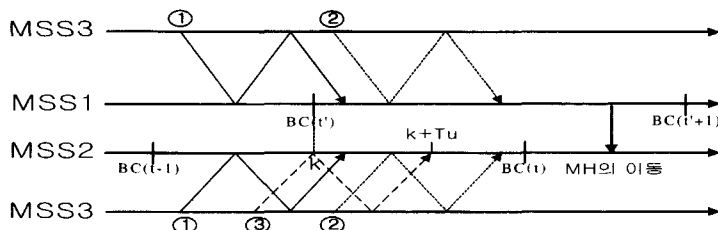
/* CCAB 정책에서 이동호스트의 캐쉬일관성을 위하여 MSS가 이동호스트에게 응답할 정보를 구하는 알고리즘 */

Procedure Responding_to_MH;

BEGIN

$$k = CT_{MSS2} - (T_{MSS1}(MH) - BC(t))$$

FOR the updatable cached data transmitted



(그림 11) 조정자가 MSS3인 경우에 T_u 가 캐쉬 일관성에 미치는 영향
 (Fig. 11) The effect of T_u upon cache consistency when the coordinator is MSS3

```

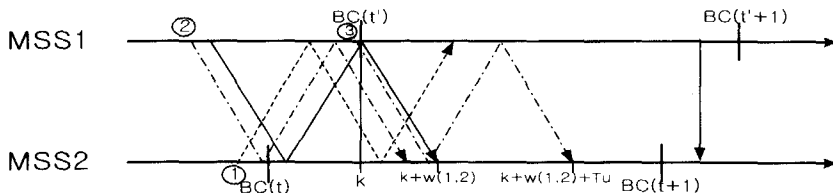
from MH
construct the Resp_Mess1
END_FOR
IF ( k < BC(t) ) THEN
    SWITCH( coord ) {
CASE MSS1 : construct the Resp_Mess2 with
              the updated result
              between k+w(1,2)+Tu and BC(t)
CASE MSS2 : construct the Resp_Mess2 with
              the updated result
              between k-w(1,2)+Tu and BC(t)
CASE MSS3 : construct the Resp_Mess2 with
              the updated result
              between k+Tu and BC(t)
    }
    ENDIF
    construct the Resp_Mess from the
    Resp_Mess1 and the Resp_Mess2
    transmit the Resp_Mess
END
    
```

위 알고리즘에서 $BC(t')$ 는 이동호스트가 이동하기 전의 MSS(MSS1)에서 마지막으로 무효화 메시지를 수신한 방송 시점이며, $BC(t)$ 는 이동호스트가 이동 후의 MSS(MSS2)에서 마지막으로 방송한 방송 시점이다. k 는 MSS1에서의 방송 시점 $BC(t')$ 를 MSS2의 시간으로 환산한 시점이며, $T_{MSS1}(MH)$ 는 MSS1에서의 이동호스트의 이동 시점이며, CT_{MSS2} 는 이동호스트가 이동해 온 MSS2의 시점이다. $Resp_Mess1$ 은 이동호스트가 MSS2로 이동하여 갱신가능한 캐쉬 데이터에 대한 일관된 정보를 요청하였을 때 MSS2가 이동호스트에게 전달할 갱신 데이터이다. $Resp_Mess2$ 는 이동호스트가 이동함으로 인하여 수신하지 못한 갱신 데이터이다. $Resp_Mess1$ 의 데이터와 $Resp_Mess2$ 의 데이터

가 중복될 때, 중복된 $Resp_Mess1$ 의 데이터를 제거한 후 $Resp_Mess1$ 의 데이터와 $Resp_Mess2$ 의 데이터를 합하여 $Resp_Mess$ 를 구성하고 이를 이동호스트에게 전달한다.

5.3 성능 분석

이동 컴퓨팅 환경에서 캐쉬 일관성을 유지하는 정책은 무선 통신 채널의 협소한 대역폭으로 인하여 많은 영향을 받는다. 이 절에서는 이동호스트가 인접 셀로 이동할 때 캐쉬 일관성을 유지하기 위하여 요구되는 메시지의 양에 대하여, 본 논문에서 제안한 CCAB 정책과 기본 정책과 [4]의 AT 방법을 비교 분석한다. 먼저, AT 방법을 비동기적인 방송에 적용하는 경우에 이동호스트가 MSS1에서 MSS2로 이동하면 MSS2는 이동호스트의 단절로 생각하여 마지막으로 방송한 내용을 이동호스트에게 전송한다. 기본 정책에서는 이동호스트가 자신이 이동하기 전에 MSS1로부터 마지막으로 수신한 방송 시점인 $BC(t')$ 를 MSS2의 시점으로 환산한 값을 k 라 하고, MSS2에서 마지막으로 방송한 시점을 $BC(t)$ 라 할 때, 이동호스트는 캐쉬 데이터 중에서 갱신가능한 데이터의 식별자를 MSS에게 보내고, MSS는 $k < BC(t)$ 인 경우에 k 부터 $BC(t)$ 사이에서 갱신된 내용과 이동호스트가 보내온 갱신가능한 데이터에 대한 일관된 정보를 전송하고 $k > BC(t)$ 인 경우에 이동호스트가 보내온 갱신가능 데이터에 대한 일관된 정보만을 전송한다. 기본 정책에서 이동호스트가 MSS에게 전송하는 메시지 양은 캐쉬 데이터 중에서 갱신가능한 데이터의 식별자들이며, MSS가 이동호스트에게 전송하는 메시지 양은 $k < BC(t)$ 인 경우와 $k > BC(t)$ 인 경우가 다르지만 갱신가능 데이터의 일관된 정보를 제외한다면 평균적으로 AT 방법의 1/4에 해당한다. 그리고, 한 방송 주기 L 동안에 갱신률을 μ 라 한다면, 이동호스트가 갱신가능 메시지를 수신하는 경우는 $BC(t') - Tu$ 부



(그림 12) 갱신가능에 대한 갱신 결과와 수신한 갱신 결과를 다시 전송하는 경우
 (Fig. 12) The case that the updated results of the updatable data and the received data are transmitted again

터 BC(t')에서 갱신가능이 발생한 경우이므로 갱신 가능성이 발생할 확률은 $\frac{Tu}{L}\mu$ 이며, 전체 데이터의 개수가 n 개이고 캐쉬 데이터의 개수가 m 개라면 캐쉬 데이터 중에서 갱신 가능성이 발생할 확률은 $\frac{m}{n}\frac{Tu}{L}\mu$ 이다. 따라서, 캐쉬 데이터 중에서 갱신가능 데이터는 극히 적다. 따라서, 기본 정책은 AT 방법보다 아주 적은 메시지 양으로 일관성을 유지한다.

기본 정책에서 k부터 BC(t)사이의 갱신 중에서 그림 12의 ①의 경우처럼 k부터 k+w(1,2)사이에서 갱신된 데이터는 이동호스트가 이동하기 전의 MSS인 MSS1에서 갱신가능을 수신하였으므로 k+w(1,2)-Tu부터 k+w(1,2)사이에서 갱신된 데이터 중에서 캐쉬 데이터에 속하는 데이터만을 전송하여도 되지만 기본 정책에서는 k부터 k+w(1,2)사이의 갱신 결과를 전부 전송하므로 불필요한 전송이 발생하게 된다. 그림 12의 ②의 경우처럼 k부터 k+w(1,2)사이에 갱신된 데이터는 이동호스트가 MSS1에서 이미 갱신 결과를 수신한 것이므로 갱신 결과를 전송할 필요가 없고, ③의 경우처럼 k+w(1,2)부터 k+w(1,2)+Tu 사이의 갱신 데이터 ①의 경우처럼 갱신가능을 수신하였으므로 캐쉬 데이터의 갱신가능 데이터에 대한 결과만을 전송하여도 되므로 불필요한 전송이 발생할 수 있다. 이러한 불필요한 전송을 제거한 정책이 CCAB 정책이다. CCAB 정책에서는 MSS2가 조정자의 MSS_id와 이동호스트가 이동하여 온 MSS1의 식별자를 알고 MSS1과 MSS2사이의 갱신 시점의 차이를 고려하여 이동호스트가 인식하지 못한 갱신 데이터들만을 전송하여 캐쉬 일관성을 유지하도록 하였다.

6. 결 론

이동 컴퓨팅 환경에서 캐쉬 일관성을 유지하는 것은 이동 트랜잭션의 관리를 위하여 꼭 필요한 과정이다. 지금까지 연구된 캐쉬 일관성 유지 정책에서는 모든 MSS가 무효화 메시지를 방송하는 시점들이 동일하였다. 그러나, 방송 시점을 일치시키는 것은 부가적인 복잡한 과정이 요구되고, 각 MSS의 독립성을 저해시키며, 이로 인하여 시스템의 성능이 떨어지는 요인으로 작용한다.

본 논문에서는 각 MSS가 무효화 메시지를 비동기적으로 방송할 때 발생할 수 있는 캐쉬 일관성 위배의

문제점을 조사하고, 이러한 문제점을 해결하기 위한 캐쉬 일관성 유지 정책을 제안하였다. 제안하는 기본 정책에서 이동호스트가 셀 A에서 방송 시점이 다른 셀 B로 이동할 때, 자신이 셀 A의 MSS1으로부터 마지막으로 수신한 방송 시점인 BC(t')를 셀 B의 MSS2의 시점으로 환산한 값을 k라 하고, MSS2에서 마지막으로 방송한 시점을 BC(t)라 할 때, 이동호스트는 캐쉬 데이터 중에서 갱신가능한 데이터의 식별자를 MSS1에게 보내고, MSS1은 k<BC(t)인 경우에 k부터 BC(t)사이에서 갱신된 내용과 이동호스트가 보내온 갱신가능 데이터에 대한 일관된 정보를 전송하고, k>BC(t)인 경우에 이동호스트가 보내온 갱신가능 데이터에 대한 일관된 정보만을 전송하여 캐쉬 일관성을 유지하도록 하였다.

그런데, 제안한 기본 정책에서는 불필요한 메시지가 발생할 수 있으므로 이를 제거하여 전송하는 메시지의 양을 줄인 CCAB 정책을 제안하였다. CCAB 정책에서는 MSS2가 조정자의 MSS_id와 이동호스트가 MSS1의 식별자를 알고 MSS2와 MSS1사이의 갱신 시점의 차이를 고려하여 이동호스트가 인식하지 못한 갱신 데이터들만을 전송하여 캐쉬 일관성을 유지하도록 하였다. CCAB 정책은 기존 정책과 비교할 때 중복된 메시지의 전송을 예방하고, MSS2가 전송할 메시지의 양을 최소화하였고, 방송 시점의 동기화를 위한 오버헤드를 제거하였다. 또한 본 연구는 지금까지 동기화된 환경에서 중점 연구되었던 캐쉬 일관성 유지 정책의 새로운 방향을 제시하였다. 앞으로 비동기적인 방송에서 이동 트랜잭션의 효율적인 관리에 관한 연구와 방송 주기의 불일치로 인한 문제점과 이에 대한 연구가 계속되어야 할 것이다.

참 고 문 헌

- [1] R. Alonso and H. F. Korth, "Database Systems Issues in Nomadic Computing," In Proc. of the 1993 SIGMOD Conference, pp.388-392, May 1993.
- [2] B. R. Badrinath and T. Imielinski, "Replication and Mobility," In second Workshop on the management of Replicated Data, pp.9-12, 1992.
- [3] B. R. Badrinath, A. Acharya, T. Imielinski, "Structuring Distributed Algorithms for Mobile Hosts," In Proc. of the 14th International Con-

ference on Distributed Computing Systems, pp. 21-28, Jun. 1994.

[4] D. Barbara et al., "Sleepers and Workaholics : Caching Strategies in Mobile Environments," Proc. ACM SIGMOD, June 1994.

[5] P. A. Bernstein, V. Hadzilacos and N. Goodman, "Concurrency Control and Recovery in database Systems," Addison Wesley, Reading, Massachusetts, 1987.

[6] S. Ceri, G. Pelagatti, "Distributed Databases Principles and Systems," McGraw-Hill, 1984.

[7] A. Elmagarmid, J. Jing, and T. Furukawa, "Wireless Client/Server Computing for Personal Information Service and Application," SIGMOD Record 1995.

[8] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computers, 27(6), pp.38-47, Apr. 1994.

[9] A. Fu et al., "Dynamic Policies in Selecting a Caching Set for a Distributed Mobile Computing Environment," Technical Report, May 13, 1995.

[10] Y. Huang, P. Sistla, and O. Wolfson, "Data Replication for Mobile Computers," In Proc. of the 1994 ACM SIGMOD International Conference on Management of Data, pp.13-24, 1994.

[11] T. Imielinski and B. R. Badrinath, "Quering in highly mobile-distributed environments," In Proc. of the 18th VLDB Conference Vancouver, British Columbia, Canada, pp.41-52, 1992.

[12] J. Jing, O. Bukhres, A. Elmagarmid, R. Alonso, "Bit-Sequences: A New Cache Invalidation Method in Mobile Environments," pp.1-25, May 1995.

[13] V. R. Narasayya, "Distributed Transactions in a Mobile Computing System," CSE552, pp.1-13, March 9, 1994.

[14] E. Pitoura et al., "Revising Transaction Concepts for Mobile Computing," Proc. IEEE Workshop on Mobile Systems and Applications, Dec. 1994.

[15] M. H. Wong and W. M. Leung, "A Caching Policy to Support Read-Only Transactions in a Mobile Computing Environment," 1994.

[16] 박성배, 황부현, "이동 컴퓨팅 환경에서 갱신가능

메시지를 이용한 캐쉬 일관성 유지 정책", 한국 정보처리학회 논문지 제4권 7호, pp.1694-1705, July 1997.

[17] 김대인, 황부현, "이동 컴퓨팅 환경에서 사이클 탐지를 이용한 트랜잭션 직렬성 유지 기법", 한국정보과학회, 춘계학술발표 논문집, pp.151-154, March 1997.



김대응

dokim@sunny.chonnam.ac.kr

1980년 전남대학교 사범대학 수학교육과(교육학사)

1984년 전남대학교 대학원 수학과(이학석사)

1990년 전남대학교 대학원 전산통계학과(이학석사)

1994년 전남대학교 대학원 전산통계학과(박사과정 수료)

1984년 3월~현재 서강정보대학 전자계산과 부교수

관심분야 : 분산시스템, 이동 컴퓨팅



박성배

psbsas@first.sunchun.ac.kr

1985년 전남대학교 계산통계학과(학사)

1987년 전남대학교 대학원 계산통계학과(이학석사)

1998년 전남대학교 대학원 전산통계학과(이학박사)

1988년~현재 순천제일대학 전자계산과 부교수

관심분야 : 분산시스템, 분산 멀티미디어, 이동 컴퓨팅



황부현

bhhwang@chonnam.chonnam.ac.kr

1978년 숭실대학교 전산학과(학사)

1980년 한국과학기술원 전산학과(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1980년~현재 전남대학교 전산학과 교수

관심분야 : 분산시스템, 분산 데이터베이스 보안, 객체지향 시스템