

모바일 애드혹 네트워크에서 캐쉬 접근 시간 향상을 위한 시간보증 기반의 캐쉬무효화 기법

최 재 호[†] · 오 재 오^{††} · 이 명 수[†] · 이 상 근^{†††}

요 약

모바일 장치의 증가와 무선 네트워크 기술의 발전으로 인해 새롭게 모바일 애드혹 네트워크에 대한 관심이 증가하고 있다. 최근까지의 연구는 데이터 접근성(Accessibility) 향상을 위해 데이터 복사본 활용이 효율적임을 증명하였다. 그러나, 데이터 복사본이 모바일 애드혹 네트워크에서 활용되기 위해서는 원본과 복사본 사이에 일관성(Consistency)이 확보되어야만 한다. 기존의 연구에서 사용되고 있는 일관성 확보 기법은 원본을 가진 노드에게 복사본 사용 노드들이 매번 일관성을 가지는지 확인하는 기법이 활용되었다. 그러나, 시간제약을 가지는 응용의 경우 원본을 가진 노드에게 확인하는 기법은 효율적이지 못하거나 때로는 심각한 문제를 일으킬 수도 있다. 본 논문에서는 시간제약응용을 위한 캐쉬 무효화 기법을 제안하고 있다. 제안하는 기법을 통해 기존의 일관성 확인 기법에서 일어나는 '롤백(Rollback)' 문제를 해결할 수 있었으며, 접근 시간(Access time) 측면에서 성능향상을 얻을 수 있었다. 또한 본 논문에서는 임계값 기반의 보완 기법을 제안한다. 우리는 실험 결과를 통해 제안하는 기법이 약간의 현재성(Currency) 손해를 있지만, 접근시간 측면에서 효율적임을 증명하여 준다.

키워드 : 모바일 애드혹 네트워크, 데이터 일관성, 접근시간, 시간제약응용

Cache Invalidation Schemes based on Time Guarantee for Improving Access Time in Mobile Ad hoc Networks

Jae-ho Choi[†] · Oh, Jae-Oh^{††} · Myong-Soo Lee[†] · SangKeun Lee^{†††}

ABSTRACT

Due to the popularity of mobile devices and advances in wireless communication technologies, a mobile ad hoc network has received a lot of attention. In the existing data replication management research, the use of a replica has been shown to be an efficient technique for improving data accessibility. However, to use a replica in ad hoc networks, the data consistency between the original data and the replica should be guaranteed. In the traditional research, a mobile node should check an original data whether the data is updated or not. However, It may be costly or sometimes impossible to check the original data. In the case of the time constraint applications, the checking cost can cause more serious problem. In this paper, we propose the time-guarantee based cache invalidation schemes for time constraint applications and the threshold based compensation method to enhance the time-guarantee based scheme. The proposed schemes can remove the "rollback" problem. Simulation results show that our schemes outperform the previous ones in terms of access time with little loss of data currency.

Keywords : Mobile Ad hoc Network, Data Consistency, Access Time, Time Constraint Applications

1. 서 론

모바일 장치들의 발전과 무선 네트워크 기술의 발전은 모바일 애드혹 네트워크 통신에 대한 관심을 증가시키고 있다[1][2]. 모바일 애드혹 네트워크는 전쟁상황이나

재난상황과 같은 인프라를 활용할 수 없는 상황에서 네트워크 구성이 가능하도록 해준다. 각각의 노드들은 데이터 공급자, 데이터 소비자의 역할과 함께 네트워크 구성을 위한 라우팅 기능을 가진다. 앞서 언급된 모바일 애드혹 네트워크의 주요 응용 분야인 전쟁이나 재난 상황응용(Application)의 경우, 응답시간이 중요한 시간제약을 가지는 응용이라는 특징이 있다. 예를 들면, 전쟁상황응용의 경우 위협에 처한 아군에 구조 신호에 대한 처리가 늦으면 아군이 생명을 잃을 수도 있으며, 응급 구조를 위한 응용의 경우 응답시간이 늦어지면 구조 가능한 조

※ 이 연구에 참여한 연구자의 일부는 '2단계 BK21사업'의 지원비를 받았다.
† 준 회 원 : 고려대학교 컴퓨터학과 박사과정
†† 준 회 원 : 고려대학교 컴퓨터학과 석사과정
††† 정 회 원 : 고려대학교 컴퓨터학과 부교수
논문접수 : 2008년 11월 7일
수 정 일 : 1차 2009년 1월 5일, 2차 2009년 1월 16일
심사완료 : 2009년 1월 16일

난자가 생명을 잃을 수도 있다.

모바일 애드혹 네트워크 분야에서 지금까지 가장 주목을 받아온 연구 분야는 네트워크 관련 연구 분야이다. 지금까지 다양한 모바일 애드혹 네트워크 라우팅 프로토콜이 제안되었으며, 이러한 라우팅 프로토콜은 매우 중요한 연구 영역으로 고려되어 왔다. 비록 라우팅 프로토콜이 매우 중요한 연구 분야이긴 하지만 네트워크 구성의 최종목적이 응용을 사용하기 위해서라는 점을 생각하면 캐쉬 관리, 데이터 복제관리, 업데이트 관리 등의 데이터 관리 연구 또한 매우 중요한 연구 분야이다. 이와 같은 이유로 최근 모바일 애드혹 네트워크에서 데이터 관리에 대한 연구결과 또한 많이 발표되고 있다[3-7].

지금까지 연구된 바에 따르면 캐쉬의 효율적인 활용은 데이터의 접근성(Accessibility)을 높이고, 데이터의 접근 시간(Access time)을 줄이며, 좁은 대역폭의 효율적인 활용을 위해 매우 유용한 방법이다. 업데이트가 없는 경우라면, 캐쉬에 저장된 데이터와 원본 데이터가 항상 일관성(Consistency)을 가지기 때문에 문제가 발생하지 않을 수도 있지만, 실제 응용에서 대부분의 데이터는 업데이트가 일어난다. 업데이트가 일어날 때, 원본 데이터와 캐쉬에 저장된 복제본간의 일관성이 유지 되지 않으면, 심각한 문제가 발생할 수도 있다. 우리는 최근 재난 상황에서 캐쉬를 활용하는 실제 응용을 발표하였다[8]. 논문 [8]의 시스템 구조는 (그림 1)과 같다. (그림 1)에서 모니터링 시스템은 (그림 2)와 같이 각각의 응급 구조사의 헬멧에 장착된 센서를 통해 위험 지역의 정보를 전달 받는다. 이 정보는 다시 (그림 2)의 단말 장치로 전송되어 응급 구조사들에 의해 활용된다. 응급 구조사는 이러한 위험 정보를 개인별 단말 장치에 저장하여 응급 구조 활동을 펼치게 된다. 이 시스템에서 응급 구조사 및 모니터링 시스템은 모바일 애드혹 네트워크로 연결되어 있다.

위와 같은 응용에서 각각의 응급 구조사들을 노드라 가정하고, 각각의 노드들이 화재상황이 심각한 지역의 위치 정보를 각각의 노드들이 캐쉬하고 있다고 가정해보자. 이러한 경우 서버에서 파악된 위험 지역 정보와 응급 구조사 단말기에 저장된 위험 지역 정보가 다르다면, 오히려 응급 구조사를 위험에 빠뜨릴 수도 있다. 이러한 이유로 복제본



(그림 1) 재난 상황을 위한 애드혹 네트워크 구조

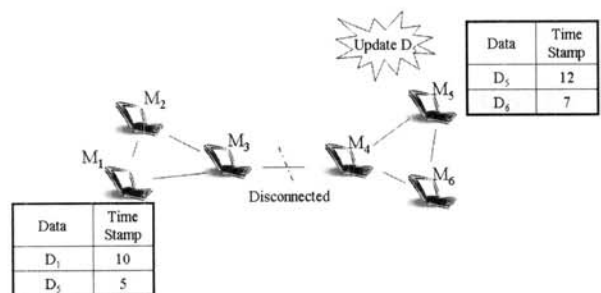


(그림 2) 헬멧 장착 센서 및 응급 구조사용 단말장치

을 활용한 캐쉬가 사용되기 위해서는 반드시 서버의 원본과 단말기의 복제본 간의 일관성(Consistency)이 보장되어야만 한다.

최근 연구에서 사용된 일관성 해결방법은 다음과 같다. 어떠한 노드가 복제본을 활용하기 위해서는 반드시 원본을 가지고 있는 노드에게 데이터의 업데이트 유무를 확인해야한다[4][6][7]. 지금까지의 기법은 가장 일관성 높은 데이터를 사용할 수 있다는 장점이 있으나, 캐쉬 일관성을 유지하기 위해 데이터 사용 시 매번 통신비용을 감수해야 하는 문제가 있으며, 비용을 감수하더라도 모바일 노드의 자유로운 이동성 때문에 때로는 일관성을 유지하지 못할 수도 있다. 이러한 문제로 인해 TTL(Time-to-live) 기반의 약한 일관성 기법이 제안되었다[1]. 그러나 TTL 기반의 기법은 특정한 인터벌에 제약을 받지 않고 업데이트가 일어나는 모바일 애드혹 네트워크 환경에 적합하지 않다. 기존 연구 중 우리의 최근 연구[9]를 제외하고는 업데이트가 일어나는 모바일 애드혹 네트워크 환경에서 시간제한 응용에서 접근시간 측면을 고려한 연구는 없었다.

(그림 3)은 임시 데이터 활용의 예를 보여준다. 본 논문에서는 임시 데이터 활용으로 인해 발생하는 '롤백(Rollback)' 문제를 해결하고자 한다. M_1 노드는 시간 5에 데이터 D_5 를 M_5 노드로부터 가져와서 활용한 뒤 캐쉬에 저장하였다. 저장 후 시간 7에 M_5 노드와 M_4 노드의 연결이 끊어졌으며, D_5 는 시간 12에 다시 업데이트



(그림 3) '임시 데이터 활용'의 예

되었다고 가정하자. 이러한 상황에서 시간 13에 M_1 노드가 D_5 를 사용하고자 할 때 기존의 기법에서는 M_1 노드가 '임시 데이터 활용(Tentative Access)'을 한다[4][6][7]. 임시로 활용된 데이터는 이후 M_1 노드가 M_5 노드와 연결될 때 M_1 노드의 요청을 통해 원본과 대조하게 된다. 이때 만약 원본 데이터와 시간 13에 활용된 데이터가 다른 데이터라면 시간 13이후의 모든 프로세싱은 '롤백'하게 되어 모두 취소된다. (그림 1)에서 설명된 시간제약을 가지는 응용의 경우 특정 기간 동안의 프로세싱 취소로 정의되는 '롤백' 문제를 지니는 무효화 기법은 활용이 불가능하다. 순간순간 생명과 연계된 판단을 해야 하는 (그림 1)과 같은 응용에서 특정기간의 프로세싱 다시 말하면, 특정 기간 동안의 획득한 위험 지역 정보를 모두 무효화 버리면, 복사본 전체를 응급 구조사가 신뢰하지 못하는 상황이 올 수도 있다. 더욱 큰 문제는 이러한 '롤백'이 이동이 빈번한 애드혹 네트워크 상황에서 자주 발생할 수 있다는 점이다. 본 논문에서는 약간의 데이터 현재성(Currency) 감소를 통해 '롤백'문제를 제거할 수 있는 캐쉬 일관성 관리 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 간단하게 관련 연구를 설명한다. 3장에서는 제안하는 기법을 설명하며, 4장에서는 시뮬레이션 결과를 통해 제안하는 기법의 효율성을 증명한다. 5장에서는 본 논문의 결론이 소개된다.

2. 관련 연구

기존의 무선 환경에서 업데이트 관리를 위해 가장 널리 사용되는 기법은 무효화 리포트(Invalidation Report) 기반의 업데이트 관리 기법이다[10]. 기존의 사용되는 무효화 리포트는 다음과 같이 정의될 수 있다.

$$IR_i = \{ \langle d_x, t_x \rangle \mid (d_x \in D) \wedge (T_i - \omega * L < t_x \leq T_i) \} \quad (1)$$

수식 (1)에서 나타난 바와 같이 기본적인 무효화 리포트 기반 기법에서는 원본을 가지고 있는 서버는 L 주기마다 무효화 리포트를 방송한다. 무효화 리포트는 업데이트된 데이터의 아이디 d_x 와 d_x 가 업데이트된 시간 t_x 정보를 포함한다. d_x 는 전체 데이터 D 에 포함되어 있는 데이터 아이디 중 현재 시간 T_i 로부터 특정 시간 $T_i - \omega * L$ 사이의 업데이트된 데이터 아이디를 나타낸다. ω 는 윈도우 사이즈로써 시스템 파라미터로 결정된다. 복제본을 가지는 노드 혹은 클라이언트들은 방송되는 무효화 리포트를 기반으로 자신의 캐쉬에 저장된 복사본의 일관성을 유지 한다. 기존의 모바일 애드혹 네트워크에서 데이터 관리를 하는 많은 기법들이 무효화 리포트 기법 기반의 업데이트 관리를 하고 있다[4-7]. 그러나, 이러한 무선 환경에서의 일관성 유지 기법은 애드혹 네트워크에서는 적

합하지 않다. 왜냐하면, 원본을 가진 노드의 연결이 무선 환경에서보다 애드혹 네트워크에서는 더욱 자주 끊어지기 때문에 무효화 리포트를 받지 못한 노드들이 많아지고 이러한 상황은 캐쉬를 거의 활용하지 못하는 결과를 초래하기 때문이다.

기존의 모바일 애드혹 네트워크에서 업데이트를 고려한 논문들은 최근값(Latest Value) 기반의 일관성 모델을 사용하고 있다[4][7]. 그러나 이러한 일관성 모델은 시간 제약적인 응용에는 적합하지 않다. 최근값 기반의 일관성 모델을 사용하기 위해서는 기본적으로 데이터를 사용하기 전에 원본을 가진 노드들에게 확인하는 과정을 거쳐야 하기 때문이다. 캐쉬 일관성 모델은 응용에 따라 다양하게 사용될 수 있기 때문에[11-13], 본 논문에서는 기존의 모바일 애드혹 네트워크 데이터관리와는 다르게 보다 빠른 접근시간 확보와 '롤백' 문제 해결을 위해 기존에 사용되던 일관성 모델 보다는 약한 일관성 모델을 사용하고자 한다. 약한 일관성 모델은 데이터를 사용할 때 가장 최신의 데이터가 아닐지라도 사용할 수 있도록 하는 데이터 모델을 말한다. 가장 약한 일관성 모델의 예로 대중 가용의 경우 다양한 버전의 음악들이 동일한 음악으로 고려되는 경우가 있는데 이러한 경우가 약한 일관성 모델을 사용하는 경우로 볼 수 있다. 이러한 약한 일관성 모델의 경우 데이터 현재성(Currency) 손해가 생기는 하지만, (그림 1)에서 설명한 시간제약을 가지는 응용의 경우, 앞서 설명한 바와 같이 강한 일관성 모델 사용을 통해 시스템 전체를 사용하지 못하는 경우가 생길 수도 있다. 물론, 애드혹 네트워크를 통해 돈이나 비밀 정보 같은 데이터를 다루는 응용의 경우, 강한 일관성 모델이 반드시 필요하기도 하지만, 본 논문에서는 앞서 설명한 응용에 보다 적합하도록 약간의 현재성 손해가 있지만, 약한 일관성 모델을 사용하고자 한다.

논문 [14]에서는 자기중심적 무효화 기법인 Invalidation by Absolute Validity Interval (IAVI)을 제안하였다. IAVI에서 클라이언트들은 데이터에 표시된 일종의 시간 표시(Time stamp)인 AVI(Absolute Validity Interval)를 통해 캐쉬를 무효화한다. IAVI 기법에서는 AVI를 활용함으로써, 무효화 리포트의 크기를 줄이는데 그 목적이 있었다. 일종의 예측된 무효화 기간인 AVI를 데이터와 함께 방송함으로써, 그 기간과 동일한 기간에 데이터 업데이트가 일어나는 데이터들은 무효화 리포트에 포함시키지 않는 방법을 통해 IAVI 기법에서는 무효화 리포트의 크기를 줄일 수 있었다. 그러나, IAVI 기법에서도 원본을 가진 서버는 기존 무효화 기법과 동일하게 주기적으로 무효화 리포트를 방송해야만 한다. 애드혹 네트워크의 경우, 원본을 가진 노드는 애드혹 네트워크에 참여할 모든 노드가 될 수 있기 때문에 IAVI 기법과 동일하게 원본을 가진 노드들이 모두 무효화 리포트를 방송할 경우 통신 트래픽이 급격하게 증가하는 문제를 야기시킬 수도 있다. 또한 IAVI 기법에서는 단지 데이터가 업데이

트 되는 주기만을 고려했을 뿐, 업데이트가 일어나는 패턴은 고려하지 않았다. 예를 들면, IAVI에서는 5초마다 주기적으로 업데이트가 꾸준히 일어나는 데이터나 평균적으로 5초마다 불규칙하게 업데이트가 일어나는 데이터 모두가 동일한 업데이트 주기로 고려되었다. 본 논문에서는 업데이트 패턴을 고려하며, 무효화 리포트를 사용하지 않는 무효화 기법을 제안한다.

3. 제안하는 기법

3.1 시간보증(Time-guarantee)

이번 장에서는 모바일 애드혹 네트워크에서 접근시간을 줄이기 위한 캐쉬 무효화 기법을 설명한다. 기본적으로 본 논문에서 제안하는 기법은 시간보증에 기반한다. 원본 데이터를 가지고 있는 노드는 복제본을 모바일 애드혹 네트워크 상의 다른 노드에게 전달해 줄 때 데이터와 함께 그 데이터에 대한 시간보증을 함께 전달한다. 시간보증은 원본 데이터를 가지고 있는 노드에 의해서 생성된다. 시간보증이 가지는 근본적인 의미는 ‘그 시간동안 복제본 데이터가 원본과 일관성을 가진다’는 의미이다. 생성된 시간보증은 실제 업데이트 시간과 일치하지 않을 수도 있다. 만약 M_i 노드가 D_i 에 대해 시간보증을 7이란 기간 동안 했다고 해서 반드시 7이란 시간이 흐른 후에 D_i 가 업데이트 되는 것은 아니다. 따라서, 다음 업데이트가 일어날 것으로 예상되는 시점까지 시간보증을 부여하는 것이 중요하다. 비록 시간보증 기반의 일관성 모델이 원본과 복사본 사이의 완벽한 현재성(Currency)을 보장해 주지는 않지만, 보다 빠른 응답 시간을 보장해 준다. 본 논문에서 가정하고 있는 시간제약이 있는 애플리케이션에서는 약간의 일관성 손해보다 빠른 응답시간이 더 중요하다. 예를 들면, 응급 구조 상황에서 환자의 위치관련 데이터를 처리하는 애플리케이션을 생각해 볼 수 있다. 그러한 애플리케이션에서 환자의 정확한 위치정보 만큼 중요한 요소는 위치정보가 주변 노드에게 전달되는 시간이다. 응급 구조사가 현장에 도착해서 환자에게 도착했을 때 다른 구조사 등에 그 환자가 구조되어 없는 상황은 받아들여 질 수 있지만, 응급 구조사의 출동 시간이 늦어지는 것은 용납될 수 없는 상황이다. 또한 올바르게 적은 시간보증이 적용된 데이터를 사용할 확률은 다음과 같은 조건을 만족시켜야만 하므로 낮다. 첫 번째, 원본 데이터의 업데이트가 시간보증 이전에 일어나야만 한다. 두 번째, 데이터를 사용하는 질의 또한 시간보증 전에 나타나야만 한다. 세 번째 요청하는 데이터가 캐쉬에 저장되어 있어야만 한다. 위의 세 가지 조건이 모두 만족되는 상황에서만 올바르게 적은 시간보증이 적용된 데이터 사용이 일어난다.

3.2 시간보증 계산방법

시간보증 기반의 캐쉬 일관성 모델을 사용하기 위해

서 가장 중요한 점은 업데이트 주기와 가장 유사한 또는 가장 응용분야에 적합한 시간보증 계산방법을 선택하는 것이다. 이번 절에서는 이러한 시간보증 계산방법들을 제안한다. 시간보증을 계산하기 위해서 가장 기본적인 방법은 원본 데이터를 가지고 있는 노드가 자신의 원본의 업데이트 주기 평균을 계산해서 가지고 있다가, 복사본을 생성하는 시점에 업데이트 주기의 평균을 주는 방법이다. 본 논문에서는 첫 번째 방법을 평균 업데이트 시간(Average Update Time)이라 하며, 줄여서 AUT라 한다. AUT는 다음과 같이 식(1)을 통해 계산된다. 식(1)에서 TG는 제안된 시간보증이며, P_i 와 U_i 는 각각 실제 업데이트 주기와 업데이트 횟수를 나타낸다.

$$TG = \frac{\sum_{i=1}^n P_i}{U_i} \quad (1)$$

AUT 기법에서 시간보증은 원본 데이터 업데이트 주기의 평균으로 계산된다. AUT계산을 위해 모바일 애드혹 네트워크의 노드들은 원본 데이터의 업데이트 주기 정보를 보존해야만 한다. 이러한 정보가 시간에 따라 증가할 경우 리소스가 부족한 모바일 노드들에 부담이 될 수도 있다. 따라서, 식 (2)와 같이 모바일 노드의 리소스 사용을 줄이는 k-UT(k-Update Time)기법도 활용될 수 있다.

$$TG_k = \frac{\sum_{i=n-k+1}^n P_i}{k} \quad (2)$$

시간보증을 계산하는 방법으로 응용에 따라서는 평균과는 다른 계산법이 필요할 경우도 있다. 예를 들면, 보다 일관성이 덜 중요한 응용의 경우 보다 업데이트 주기의 평균이 아닌 업데이트 주기의 최대값을 사용할 수도 있다. 최대값을 사용하는 것의 의미는 지금까지 업데이트가 일어난 가장 큰 기간을 시간보증으로 사용함으로써 데이터의 접근성(Accessibility)을 최대한 보장하기 위함이다. 이러한 경우 복제본은 천천히 만료(Expire)된다. 따라서, 각각의 모바일 노드내의 복제본의 개수는 늘어나게 되지만, 데이터의 일관성을 떨어뜨리는 결과를 낳게 된다. 예에서 설명한 것과 정반대의 경우, 즉, 일관성이 보다 중요한 경우에는 최소값을 사용할 수도 있다. 위에 설명한 최대값, 최소값을 사용한 시간보증은 식(3)과 식(4)를 통해서 계산된다.

$$TG_{\max} = \{P_{\max} | \max \text{ value of } P_i, (n-k < i \leq n)\} \quad (3)$$

$$TG_{\min} = \{P_{\min} | \min \text{ value of } P_i, (n-k < i \leq n)\} \quad (4)$$

시간보증을 계산하는 마지막 방법은 최근 업데이트 주기를 보다 중요시하는 기법이다. 본 논문에서는 마지막 기법을 HISTORY라 하였다. 시간보증은 최근 시간보증 값과 최근 업데이트 주기를 통해 계산된다. HISTORY기

법의 장점은 업데이트 주기가 일정한 패턴을 가지고 변화하는 경우 시간보증의 일관성 보장이 보다 정확해 진다는데 있다. HISTORY기법에서 시간보증은 식(5)에 의해서 계산된다. 식 (5)에서 α 는 최근 시간보증의 중요도를 결정하기 위한 시스템 파라미터로 사용된다. 예를 들면, α 가 0에 가까울수록 다음 시간보증 값은 마지막 업데이트 주기와 유사한 시간보증 값을 가지게 된다. 따라서 예전 업데이트 주기 보다는 마지막 업데이트를 가장 중요하게 고려하기 위해서는 α 를 0에 가깝도록 설정하면 된다.

$$TG_{history} = \alpha(\text{the latest } TG) + (1-\alpha)(\text{the latest } P_i), (0 < \alpha < 1) \quad (5)$$

3.3 시간보증 보완기법

앞서 제안된 시간보증 기법은 다음과 같은 보완기법에 의해 보다 높은 정확성을 가질 수 있다. <표 1>에서 볼 수 있듯이 평균 값 기반의 시간보증은 문제를 가질 수도 있다. <표 1>에는 두 가지 데이터의 업데이트를 보여준다. D_1 과 D_4 는 동일한 평균 업데이트 시간을 가지지만, 업데이트 패턴에 있어서는 매우 큰 차이를 보인다. D_4 는 D_1 에 비해 매우 불규칙한 업데이트가 이루어지고 있음을 볼 수 있다. 이러한 경우 두 데이터에 같은 시간보증을 부여하는 것은 큰 데이터 현재성 손해를 야기 시킬 수 있다. 데이터 업데이트 패턴이 불규칙한 경우, 이를 보완하기 위해 시간보증을 감소시키는 방법이 사용될 수 있다. 식(6)은 이러한 보완 기법이 적용되어 결정되는 시간보증을 구하는 식이다. Δ 값은 시간 보증의 크기의 변화를 업데이트 주기의 분산 계산하여, 0과 1사이에서 정규화한 값이다. 예를 들면, Δ 값이 1에 가까울수록 시간보증 값은 작아지게 된다. 이렇게 Δ 값이 크게 결정되는 경우 잦은 캐쉬내 복제본 삭제가 일어나기 때문에 캐쉬의 활용성이 떨어질 수 있을 것이고, 반대의 경우에는 데이터의 현재성 손실이 보다 빈번하게 발생할 수 있다.

$$TG_c = TG_{origin} * (1 - \Delta), (0 < \Delta < 1) \quad (6)$$

<표 1> 데이터 업데이트의 예

데이터 ID	업데이트 시간	데이터 ID	업데이트 시간
D_1	5	D_4	5
D_1	15	D_4	25
D_1	24	D_4	30
D_1	35	D_4	35
D_1	45	D_4	45

3.4 알고리즘

이번 절에서는 시간보증을 사용하기 위해 활용되는 알고리즘을 설명한다. 시간보증 활용을 위해서는 두 종류의

알고리즘이 활용된다. 첫 번째는 시간보증을 생성하기 위한 데이터 공급자의 알고리즘이다. 데이터 공급자는 원본 데이터를 가지고 있는 노드들이다. 각각의 데이터 공급자는 원본 데이터가 업데이트될 때마다 시간보증을 생성하고 업데이트 한다. 또한 원본데이터에 대한 요청이 들어올 때는 원본데이터와 시간보증을 요청한 노드에게 전송한다. 데이터 공급자의 알고리즘은 알고리즘 1에서 설명된다. 알고리즘에서 t_c 는 현재의 타임스탬프를 나타내며, TG_i 는 데이터 d_i 의 시간보증을 나타낸다.

```

알고리즘 1. 데이터 공급자의 알고리즘
(A)When data  $d_i$  is updated
    CalculationTG (<  $d_i, t_c$  >);
(B)When receives a request of data  $d_i$ 
    Send data < $d_i, TG_i$ >; /* TG for  $d_i$  is not changed
until  $TG_i$  */
(C)CalculationTG(){
    /*implements using proposed calculation methods*/
    /*< $d_i, TG_i$ > and equation (1)~(5) are used in this
function*/
    
```

데이터를 요청하는 노드는 데이터 사용자로써 동작한다. 데이터 사용자 알고리즘은 알고리즘2에 나타나있다.

```

알고리즘 2. 데이터 사용자의 알고리즘
(A)When data  $d_i$  is requested
    if  $d_i$  is placed in its local cache
        use  $d_i$  immediately;
    else send request  $d_i$  to neighbors;
(B)When receives response
    store < $d_i, TG_i$ > in local cache;
(C)if ( $t_c = TG_i$ )
    remove < $d_i, TG_i$ > from local cache;
    
```

4. 실험 결과

4.1 시뮬레이션 모델

제안한 기법의 성능 평가를 위해 본 논문에서 고려된 시뮬레이션 환경은 기본적으로 [3-5]논문에서 고려된 환경과 동일하다. 본 논문의 시뮬레이션에서는 전체 모바일 애드혹 네트워크에 40개의 모바일 노드가 있다고 가정하였다. 각각의 노드($M_1, M_2, M_3, \dots, M_n$)들은 로컬 캐쉬 저장소를 가지고 있으며, 최대 속도 V 로 $100*100$ 크기의 영역

을 랜덤하게 이동한다. 모바일 노드의 이동 모델링을 위해서는 목적지를 랜덤하게 선택하고 일정속도로 이동한 후 다시 랜덤한 목적지를 선택하는 모델이 사용되었다. 각각의 모바일 노드들과 같은 수의 원본 데이터($D_1, D_2, D_3, \dots, D_n$)가 네트워크 상에 존재하며, M_i 의 D_j 에 대한 데이터 액세스 확률은 논문[3]과 같이 $P_{ij} = 0.005f \times (1 + 0.0001 \times j)$ 의 식에 의해서 계산되었다. 각각의 데이터의 업데이트 확률은 지수 분포에 따라 계산되었으며, 계산식은 아래 식(7)과 같다.

$$f_x(x) = \lambda e^{-\lambda x}, (x \geq 0) \quad (7)$$

본 논문에서 사용된 기본적인 파라미터는 <표 2>에 설명되어 있다. 데이터 접근시간을 계산하기 위해서 우리는 한 홉(hop) 사이에서 데이터가 전송되는 시간을 기본 시간 계산 단위로 하였다. 접근시간은 한 모바일 노드에서 요청이 발생하고 데이터가 그 노드에 전달되기까지의 시간으로 계산하였으며 접근시간의 정의는 논문[15]에 정의되어 있는 바와 유사하다.

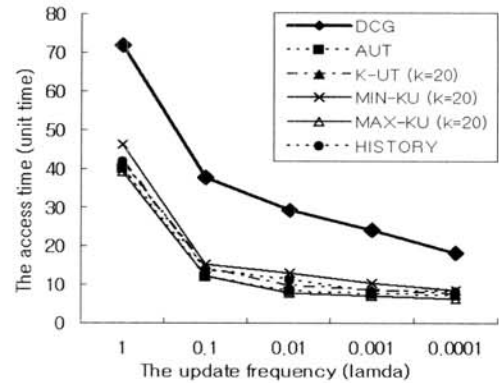
성능 평가를 위해 본 논문에서는 지금까지 모바일 애드혹 네트워크 환경에서 제안된 캐쉬 관리 기법 중 가장 효율적이라고 알려진 논문 [3]의 DCG 기법과 비교 평가를 수행하였다. DCG 기법은 이진연결 컴퍼넌트(Bi-connected Component)를 활용하여 모바일 노드를 그룹핑한다. 이러한 방법을 바탕으로 각각의 모바일 애드혹 네트워크 노드의 캐쉬 활용성을 극대화함으로써 모바일 애드혹 네트워크 환경에서 높은 데이터 접근성을 보장하는 기법이다. DCG 기법에서는 질의 수행을 위해 업데이트 여부를 원본 데이터를 가진 노드에게 질의 수행시마다 확인 방식으로 캐쉬 일관성을 확보한다. 지금까지 제안된 모바일 애드혹 네트워크 환경에서 일관성 확보 기법은 이와 동일하거나, 거의 유사한 기법에 기반하기 때문에 이중 가장 캐쉬 활용성이 높아 데이터 접근시간적인 측면의 성능이 좋은 DCG 기법과 본 논문에서 제안한 기법을 비교 평가하였다.

<표 2> 시뮬레이션 파라미터

파라미터	값
노드의 개수 (N)	40
통신 범위 (R)	5~25(m)
네트워크 크기	100(m)*100(m)
캐쉬 크기 (데이터 개수)	5~35
업데이트율 (λ)	0.0001~1
노드의 최대속도	1~9(m/sec)
k의 기본값	20

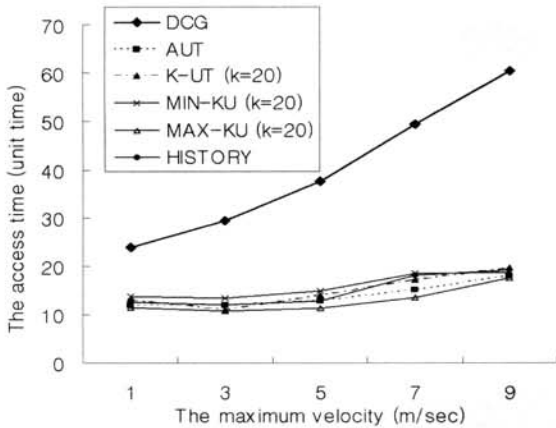
4.2 실험 결과

(그림 4)는 본 논문에서 다양한 업데이트율을 적용하



(그림 4) 업데이트율에 따른 접근시간

였을 때, 측정된 데이터 접근시간을 보여준다. 상대적인 평가를 위해 논문[3]에서 제안된 DCG 기법이 활용되었다. 최대값, 최소값을 활용하는 시간보증기법에서는 최근 20개(k=20)의 업데이트 주기를 활용하여 계산을 수행하였다. 본 논문에서는 업데이트율을 식 6의 λ 를 활용하여 조정하였다. 작은 λ 가 의미하는 바는 업데이트가 자주 일어나는 것을 의미한다. (그림 4)에서 볼 수 있듯이 본 논문에서 제안하는 기법은 모두 기존의 DCG 기법에서보다 접근시간이 좋은 것을 알 수 있다. 본 논문에서 제안한 기법중 최대값을 시간보증으로 사용한 기법인 MAX-KU가 접근시간에서 가장 좋은 성능을 보인다. 제안된 기법이 모두 기존의 DCG 기법보다 좋은 성능을 보이는 이유는 DCG의 경우 복사본을 사용할 때 임시 데이터 활용을 하는데 롤백 문제가 일어날 수도 있고, 롤백이 일어나지 않더라도 원본 데이터를 복제본 데이터와 비교 확인하기 위한 시간이 필요하기 때문이다. 단 본 논문에서 롤백이 일어날 경우의 접근시간은 측정하지 않았다. 롤백이 일어날 경우 접근시간은 롤백이 일어난 모든 프로세싱을 실패(Fail)로 처리해야하기 때문에 접근시간을 구하는 것 자체가 의미가 없을 수 있기 때문이다. (그림 4)에서 측정된 접근시간은 시간 제약응용에서 매우 중요한 의미를 지닌다. 예를 들면 1장에서 설명한 재난상황의 응용에서 각각의 모바일 노드들이 위험 지역의 위치 정보를 저장하고 있다고 할 때, 재난 현장에서 빠져나가기 위한 퇴로를 결정하는 응용서비스를 생각해 볼 수 있다. 위험 지역 위치 정보는 각각에 노드에 의해서 생성된 정보이며, 이러한 정보를 기반으로 퇴로를 결정하는 것은 빠른 응답을 요구한다. AUT와 HISTORY 기법의 경우, MAX-KU나 MIN-KU 사이의 접근시간을 보인다. MAX-KU의 경우 캐쉬에 저장된 데이터는 가장 오랜시간 동안 유효(Valid)하다고 인정되기 때문에 캐쉬에 오래 저장된다. 캐쉬에 데이터가 저장되는 시간이 상대적으로 길기 때문에, MIN-KU에 비해 캐쉬의 활용성은 높다. 그러나, 오랜 저장은 캐쉬 일관성 측면에서는 손해를 불러올 수도 있다. 이러한 문제를 해결하기 위한 기법이 AUT와 HISTORY기반의 기법이다. 그래프를 통해 알 수 있듯이, AUT와 HISTORY기반의 기법은 MAX-KU



(그림 5) 다양한 속도에 따른 접근시간

와 MIN-KU 사이의 성능을 보여준다. AUT는 과거 일정기간의 평균 업데이트 주기를 HISTORY는 과거 일정기간의 업데이트 중 최근을 주기를 중요시한다는 차이가 있기 때문에, 활용되는 응용의 특성에 따라 적절하게 선택되어 사용될 수 있을 것이다.

(그림 5)는 다양한 속도에 따른 접근시간을 보여준다. 그림을 통해 볼 수 있듯이 제안한 기법은 모바일 노드의 이동에 기존의 DCG 기법에 비해 덜 민감하다. 따라서, 이동속도가 증가하더라도 접근시간이 크게 증가하지 않는 것을 볼 수 있다. 이는 제안하는 기법의 경우 이동성이 증가하여도 원본과 복제본의 일관성을 확인하는 시간이 없기 때문에 접근시간이 향상되는 것을 볼 수 있다.

본 논문에서 제안하는 기법의 데이터 일관성은 시간보증에 의해서 확보된다. 원본 데이터와 일치해야 일관성이 확보되는 기존의 기법과는 다르게 시간보증이 유효한 경우 일관성이 확보되는 약한 일관성 모델을 사용하기 때문에 데이터 일관성 측면에서는 손해가 없지만, 제안하는 기법들은 현재성의 손실이 존재한다. 현재성 손실을 측정하기 위해 본 논문에서 제안하는 기법들이 원본과 다른 데이터를 접근할 확률을 <표 3>에 나타내었다. 3.1절에서 설명한 바와 같이 현재성 손실이 일어날 확률이 적으며 이는 <표 3>를 통해 증명된다. <표 3>에 나타난 수치는 사용자가 캐쉬에 있는 데이터를 사용할 경우, 최신 버전을 사용할 확률과 업데이트가 일어난 구 버전 데이터를 접근할 확률을 나타낸다. 예를 들면 AUT 기법의 경우, 전체 캐쉬 데이터 접근 중 98.92%는 가장 최신의 데이터를 사용하였지만, 1.08%의 접근에서는 업데이트가 일어난 구 버전 데이터를 접근하였다. 이는 약한 일관성 모델을 사용하더라도, 100번의 질의 중 1~2번만 최신 데이터가 아닌 업데이트가 이미 일어난 데이터를 사용하는 것을 의미한다. 앞에서 설명한 재난상황에서의 퇴로 서비스의 경우, 현재 그 지역이 실제로 위험지역이 아니더라도, 그 지역을 피해 가야만 하도록 하는 상황이 구버전 데이터를 접근함으로써 일어날 수 있다. 이러한 상황은 1장에서 설명한 시간제약 응용은 재난상황을 위한 응용에서는 보다 효율적이다. 실제 그 지역이 지금 위험한지 아

<표 3> 기법별 최신 버전과 구버전 접근 확률

	AUT	K-UT	MIN-KU	MAX-KU	HISTORY
최신 버전접근(%)	98.92	98.50	99.29	97.74	98.77
구 버전접근(%)	1.08	1.50	0.71	2.26	1.23

닌지 확인하기 위해 응답시간을 늦추는 것보다는 실제 위험하지 않더라도 빠르게 퇴로를 알려주는 것이 보다 응용 구조사의 안전을 위해 적합한 기법이기 때문이다.

5. 결 론

본 논문에서는 모바일 애드혹 네트워크 환경에서 시간 제약 응용을 위한 캐쉬 일관성 확보기법을 제안하였다. 제안된 기법은 시간보증에 기반하여 캐쉬의 일관성을 보장하며, 불규칙적인 업데이트 패턴 등을 보완하기 위한 기법 또한 본 논문에서 제안되었다. 또한 실험을 통해 제안된 기법들이 1~2.2%의 현재성 손실이 있기는 했지만 접근시간 측면에서 기존의 기법들 보다 2배 이상 향상된 성능을 보여주는 것을 증명하였다. 이는 4장에서 설명한 바와 같이 재난상황을 위한 응용과 같이 시간제약을 지니는 응용에서 보다 빠른 응답시간을 확보할 수 있도록 도와준다. 제안된 기법 중 MAX-KU의 경우 접근시간 측면의 성능은 가장 좋았으나, 구 버전 접근이 상대적으로 많은 단점이 있었고, MIN-KU의 경우 접근시간 측면의 성능은 제안된 기법 중 가장 좋지 않았으나, 최신 버전을 가장 많이 접근 하는 것을 볼 수 있었다. 각각의 기법은 응용에 따라 선택 될 수 있으며, 보완 기법 또한 응용에 적합하게 선택될 수 있을 것이다. 시간보증 기반 캐쉬 일관성 모델에서 가장 중요한 부분은 보다 정확성 높은 시간보증 확보에 있다. 향후 연구로써 보다 정확성 높은 시간보증 기법에 대한 연구를 진행할 예정이며, 제안된 응용이 아닌 범용 응용에 사용할 수 있는 모바일 애드혹 네트워크를 위한 캐쉬 일관성 확보기법에 대한 연구를 진행할 예정이다.

참 고 문 헌

- [1] G. Cao, L. Yin, and C. R. Das. "Cooperative cache based data access in ad hoc networks", IEEE Computer, Vol.37, No.2, pp.32-39, 2004.
- [2] J.-L. Huang and M.-S. Chen. "On the effect of group mobility to data replication in ad hoc networks". IEEE Transaction on Mobile Computing, Vol.5, No.5, pp.492-507, 2006.
- [3] T. Hara. "Effective replica allocation in ad hoc networks for improving data accessibility". In Proceedings of IEEE INFOCOM, pp.1568-1576, 2001.
- [4] T. Hara and S. K. Madria. "Dynamic data replication

using aperiodic updates in mobile adhoc networks". In Proceedings of the International Conference on Database Systems for Advanced Applications, pp.869-881, 2004.

- [5] H. Hayashi, T. Hara, and S. Nishio. "Cache invalidation for updated data in ad hoc networks". In Proceedings of CoopIS/DOA/ODBASE, pp.516-535, 2003.
- [6] H. Hayashi, T. Hara, and S. Nishio. "Updated data dissemination in ad hoc networks". In Proceedings of Ubiquitous Mobile Information and Collaboration Systems, pp.28-42, 2004.
- [7] H. Hayashi, T. Hara, and S. Nishio. "On updated data dissemination exploiting an epidemic model in ad hoc networks". In Proceedings of the International Workshop on Biologically Inspired Approaches to Advanced Information Technology, pp.306-321, 2006.
- [8] 최재호, 이상근, 김정현, 김창현. "재난구조용 웨어러블 인터페이스 설계 및 구현", 차세대 컴퓨팅 추계 학술대회 논문집, pp.158-163. 2008.
- [9] 최재호, 오재오, 이명수, 이상근. "모바일 애드혹 네트워크에서 시간제약응용을 위한 캐쉬 무효화 기법", 한국정보처리학회 추계학술발표대회 논문집, 제15권, 제2호. 2008.
- [10] D. Barbara and T. Imielinski. "Sleepers and workaholics: Caching strategies in mobile environments". In Proceedings of ACM SIGMOD Conference on Management of Data, pp.1-12, 1994.
- [11] H. Guo, P.-A. Larson, R. Ramakrishnan, and J. Goldstein. "Relaxed currency and consistency: How to say "good enough" in SQL". In Proceedings of ACM SIGMOD Conference on Management of Data, pp.815-826, 2004.
- [12] D. J. Ram, M. U. Mahesh, N. S. K. C. Sekhar, and C. Babu. "Causal consistency in mobile environment". ACM SIGOPS, Vol.35, No.1, pp.34-40, 2001.
- [13] J. C.-H. Yuen, E. Chan, K. -Y. Lam, and H.-W. Leung. "Cache invalidation scheme for mobile computing systems with real-time data". ACM SIGMOD Record, Vol.29, No.4, pp.34-39, 2000.
- [14] K. Ramamritham, R. Sivasankaran, J. A. Stankovic, D. T. Towsley, and M. Xiong. "Integrating temporal, real-time, and active databases". SIGMOD Record, Vol.25, No.1, pp.8-12, 1996.
- [15] T. Imielinski, S. Viswanathan, and B. Badrinath. "Data on air: Organization and access". IEEE Transactions on Knowledge and Data Engineering, Vol.9, No.3, pp.353-372, 1997.



최재호

e-mail : redcolor25@korea.ac.kr

2003년 숭실대학교 컴퓨터학과(학사)
2005년 고려대학교 대학원 컴퓨터학과(석사)
2005년~현 재 고려대학교 대학원 컴퓨터학과 박사과정

관심분야: MANET 데이터관리, 스트리밍 데이터관리, XML 데이터 관리 등



오재오

e-mail : doublefive@korea.ac.kr

2008년 가톨릭대학교 컴퓨터학과(학사)
2009년~현 재 고려대학교 대학원 컴퓨터학과 석사과정

관심분야: 센서 데이터관리, 모바일 컴퓨팅 등



이명수

e-mail : lms9711@korea.ac.kr

2004년 고려대학교 컴퓨터학과(학사)
2006년 고려대학교 대학원 컴퓨터학과(석사)
2006년~현 재 고려대학교 대학원 컴퓨터학과 박사과정

관심분야: 센서 데이터관리, 모바일 컴퓨팅, 위치기반 서비스 등



이상근

e-mail : yalphy@korea.ac.kr

1994년 고려대학교 컴퓨터학과(학사)
1996년 고려대학교 대학원 컴퓨터학과(석사)
1999년 고려대학교 대학원 컴퓨터학과(박사)
2000년~2001년 Univ. of Tokyo 특별방문 연구원

2003년~현 재 고려대학교 컴퓨터학과 부교수

관심분야: Web DB, XML DB, 위치기반 정보시스템, 센서 데이터 관리 등