모바일 애드혹 네트워크에서 시간제약 응용을 위한 캐쉬 무효화 기법*

최재호, 오재오, 이명수, 이상근
고려대학교 컴퓨터 · 전파통신 공학과
e-mail:{redcolor25, doublefive, lms9711, yalphy}@korea.ac.kr

Cache Invalication Schemes for Time Constraint Applications in Mobile Ad hoc Networks

Jae-Ho Choi, Jae-Oh Oh, Myong-Soo Lee, SangKeun Lee
Division of Computer and Communication Engineering, Korea University

요 약

최근 모바일 장치의 증가와 무선 네트워크 환경의 발전은 모바일 애드혹 네트워크에 대한 관심을 증가시키고 있다. 최근까지의 연구는 모바일 애드혹 네트워크에서 데이터 접근성(Accessibility) 향상을 위해 복제의 활용이 효율적임을 증명하고 있다. 복제 기법이 모바일 애드혹 네트워크에서 실제로 활용되기 위해서는 원본과 복제본 사이에 일관성(Consistency)이 확보되어야만 한다. 기존의 연구에서는 일관성 확보를 위해 원본을 가진 노드에게 확인하는 기법을 활용하고 있다. 그러나, 시간제약을 가지는 응용의 경우 원본을 가진 노드에게 확인하는 기법은 효율적이지 못하며, 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, 접근시간(Access time) 측면에서 성능향상을 얻을 수 있었다. 실험 결과는 본 논문의 접근시간 측면에서의 효율적임을 증명하여 준다.

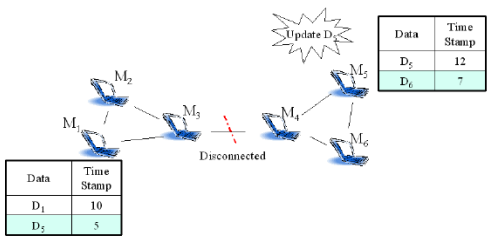
1. 서론

최근 모바일 장치와 무선 네트워크 환경의 발전은 모바일 애드혹 네트워크 통신에 대한 관심을 증가시키고 있다[2][9]. 이러한 모바일 애드혹 네트워크는 전쟁상황 이나 재난상황과 같은 인프라를 활용할 수 없는 상황에서 네트워크를 구성토록 해준다. 이러한 응용은 또한 응답시간이 중요한 시간제약을 가지는 응용이라는 특징이 있다.

지금까지 다양한 모바일 애드혹 네트워크 라우팅 프로토콜이 제안되었으며, 이러한 라우팅 프로토콜은 매우 중요한 연구 영역으로 고려되어왔다. 비록 라우팅 프로토콜이 매우 중요한 연구 분야이긴 하지만 네트워크 구성의 최종목적이 응용(Application)을 사용하기 위해서이기 때문에 캐쉬 관리, 데이터 복제관리, 업데이트 관리 등의 데이터 관리 연구 또한 매우 중요한 연구 분야이다. 이와 같은 이유로 최근 모바일 애드혹 네트워크에서 데이터 관리 연구 또한 많이 발표되었다[5][7][8].

지금까지 연구된 바에 따르면 캐쉬의 효율적인 활용은 데이터의 접근성(Accessibility)를 높이고, 데이터의 접근시간(Access time)을 줄이며, 좁은 대역폭의 효율적인 활용을 위해 매우 유용한 방법이다. 그러나 캐쉬에 저장된 데이터는 업데이트 될 수 있기때문에 복제본을 활용한 캐쉬가 사용되기 위해서는 반드시 캐쉬 일관성(Consistency) 문제가 해결되어야만 한다.

최근 연구에서 사용된 일관성 해결방법은 다음과 같다. 한 노드에서 복제되어 캐쉬에 저장된 데이터를 활용하기 위해서는 반드시 원본을 가지고 있는 노드에게 데이터의 업데이트 유무를 확인해야한다[5][7][8]. 지금까지의 기법은 가장 일관성있는 데이터를 사용할 수 있다는 장점이 있으나, 캐쉬 일관성을 유지하기 위해 데이터 사용 시 매번 통신비용을 감수해야하는 문제가 있으며, 비용을 감수하더라도 모바일 노드의 자유로운 이동성 때문에 때로는 일관성을 유지하지 못할 수도 있다. 이러한 문제로 인해 Time-to-live(TTL) 기반의 약한 일관성 기법이 제안되었다[2]. 그러나 TTL 기반의 기법은 특정한 인터벌에 제약을 받지 않고 업데이트가 일어나는 모바일 애드혹 네트워크 환경에 적합하지 않다.



(그림 1) '임시 데이터 활용'의 예

* 이 연구에 참여한 연구자(의 일부)는 '2단계 BK21'의 지원비를 받았음.

그림 1은 본 논문에서 해결하고자 하는 임시 데이터 활용의 예를 보여준다. M_1 노드는 시간 5에 데이터 D_5를 M_5 노드로부터 가져와서 활용한 뒤 캐쉬에 저장하였다. 저장 후 시간 7에 M_3 노드와 M_4 노드의 연결이 끊어졌으며, D_5는 시간 12에 다시 업데이트 되었다. 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[5][7][8]. 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(Rollback)'하게 되어 모두 취소된다. 이러한 '롤백'은 모바일 애드혹 네트워크에서 빈번하게 일어나게 되며, 전체 노드의 성능에 크게 악영향을 미치게 된다. 또한, 모바일 애드혹 네트워크의 가장 중요한 응용 중 하나인 응급 상황, 전쟁 상황 등을 위한 응용에서는 더욱 해결되어야만 한다. 왜냐하면, 하나의 프로세싱 '롤백'은 구조사나 전투원의 생명에 직접적으로 영향을 미칠 수 있기 때문이다. 본 논문에서는 약간의 데이터 현재성(Currency) 감소를 통해 '롤백' 현상을 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 무선환경에서 업데이트 관리를 위해 가장 널리 사용되는 기법은 무효화 리포트(Invalidation Report) 기반의 업데이트 관리 기법이다[1]. 무효화 리포트 기반의 기법은 주기적으로 무효화 리포트를 방송한다. 기존의 모바일 애드혹 네트워크에서 데이터 관리를 하는 많은 기법들이 무효화 리포트 기법 기반의 업데이트 관리를 하고 있으나[5][6][7][8], 이는 연결이 자주 끊어지는 모바일 애드혹 네트워크에는 적합하지 않다.

기존의 모바일 애드혹 네트워크에서 업데이트를 고려한 논문들은 최근값(The latest-value) 기반의 일관성 모델을 사용하고 있다[5][8]. 그러나 이러한 일관성 모델은 시간제약적인 응용에는 적합하지 않다. 실제로 일관성 모델은 응용에 따라 다양하게 사용될 수 있다[3][11][12]. 본 논문에서는 기존의 모바일 애드혹 네트워크 데이터관리와는 다르게 보다 빠른 접근시간 확보와 '롤백' 문제 해결을 위해 약한 일관성 모델을 사용하고자 한다. 논문 [14]에서는 자기중심적 무효화 기법인 Invalidation by Absolute Validity Interval (IAVI)를 제안하였다. IAVI에서 클라이언트들은 데이터에 표시된 일종의 시간표시(Time stamp)를 통해 캐쉬를 무효화한다. 본 논문에서도 IAVI와 같이 데이터에 정보를 포함하여 일관성을 유지하고자 한다.

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. 생성된 시간 보증은 실제 업데이트 시간과 일치하지 않을 수도 있다. 만약 M_1 노드가 D_1에 대해 시간보증을 7이란 기간 동안 했다고 해서 반드시 7이란 시간이 흐른 후에 D_5가 업데이트 되는 것은 아니다. 그러나, 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 위치 정보가 전파될 때 가장 중요한 것은 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 올바르지 않은 시간보증이 적용된 데이터를 사용할 확률은 다음과 같은 조건을 만족시켜야만 하므로 낮다. 첫 번째, 원본 데이터의 업데이트가 시간 보증 이전에 일어나야만 한다. 두 번째, 데이터를 사용하는 질의 또한 시간 보증 전에 나타나야만 한다. 세 번째 요청하는 데이터가 캐쉬에 저장되어 있어야만 한다.

3.2 시간보증 계산방법

시간보증을 계산하기 위해서 가장 기본 적인 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_i와 U_t는 각각 실제 업데이트 주기와 업데이트 횟수를 나타낸다.

$$TG = \frac{\sum_{i=1}^{n} P_i}{U_t} \qquad (1)$$

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.

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

시간 보증을 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 시간 보증은 식(3)과 식(4)를 통해서 계산된다.

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

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

시간 보증을 계산하는 마지막 방법은 최근 업데이트 주기를 보다 중요시하는 기법이다. 본 논문에서는 마지막 기법을 HISTORY라 하였다. 시간보증은 최근 시간 보증값과 최근 업데이트 주기를 통해 계산된다. HISTORY기법의 장점은 업데이트 주기가 일정한 패턴을 가지고 변화하는 경우 시간보증의 일관성 보장이 보다 정확해 진다는데 있다. HISTORY기법에서 시간보증은 식(5)에 의해서 계산된다.

$$TG = \alpha(the\ latest\ TG) + (1-\alpha)(the\ latest\ P_i) \quad (5)$$

3.3 알고리즘

이번 장에서는 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_c는 현재의 타임스탬프를 나타내며, TG_i는 데이터 d_i의 시간 보증을 나타낸다.

알고리즘 1. 공급자의 알고리즘

```
(A)When data di is updated
    Calculation TG (< di,tc >);
(B)When receives a request of data di
    Send data <di, TGi>; /* TG for di is not changed until TGi */
(C)CalculationTG(){
    /*implements using proposed calculation methods*/
    /*<di, TGi> 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 di is requested
    if di is placed in its local cache
        use di immediately;
    else send request di to neighbors;
(B)When receives response
    store <di, TGi> in local cache;
(C)if (tc = TGi)
    remove <di, TGi> from local cache;
```

4. 실험 결과

4.1 시뮬레이션 모델

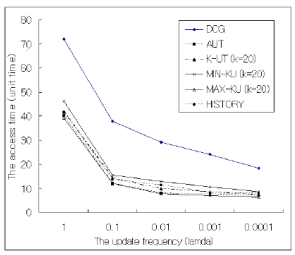
제안한 기법의 성능 평가를 위해 본 논문에서 고려된 시뮬레이션 환경은 기본적으로 [6][7]논문에서 고려된 환경과 동일하다. 본 논문의 시뮬레이션에서는 전체 모바일 애드혹 네트워크에 40개의 모바일 노드가 있다고 가정하였다. 각각의 노드($M_1,M_2,M_3,...,M_n$)들은 로컬 캐쉬 저장소를 가지로 있으며, 최대 속도 V로 100*100 크기의 영역을 랜덤하게 이동한다. 모바일 노드의 이동 모델링을 위해서는 목적지를 랜덤하게 선택하고 일정속도로 이동한 후 다시 랜덤한 목적지를 선택하는 모델이 사용되었다. 각각의 모바일 노드들과 같은 수의 원본 데이터 ($D_1,D_2,D_3,...,D_n$)가 네트워크 상에 존재하며, M_i,의 D_j,에 대한 데이터 엑세스 확률은 [6]번 논문과 같이 P_{ij}=0.005f*(1+0.0001*j)의 식에 의해서 계산되었다. 각각의 데이터의 업데이트 확률은 지수 분포에 따라 계산되었으며, 계산식은 아래 식(6)과 같다.

$$f_x(x) = \lambda e^{-\lambda x}, (x \ge 0) \quad (6)$$

본 논문에서 사용된 기본적인 파라미터는 표 1에 설명되어 있다. 데이터 접근시간을 계산하기 위해서 우리는 한 홉(hop) 사이에서 데이터가 전송되는 시간을 기본 시간 계산 단위로 하였다. 접근 시간은 한 모바일 노드에서 요청이 발생하고 데이터가 그 노드에 전달되기까지의 시간으로 계산하였으며 접근시간의 정의는 논문 [10]에 정의되어 있는 바와 유사하다.

<표 1> 시뮬레이션 파라미터

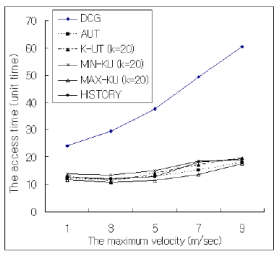
Parameter	Value
Number of nodes (N)	40
Communication range (R)	5 ~ 25 (m)
Size of the networks ($MAX_X \times MAX_Y$)	100(m) × 100(m)
Cache size (data items)	5 ~ 35
Update rate (λ)	0.0001 ~ 1
Maximum velocity of the nodes (v)	1 ~ 9 (m/sec)
Value of k	20



(그림 2) 업데이트율에 따른 접근시간

4.2 실험 결과

그림 2는 본 논문에서 측정한 데이터 접근 시간을 보여준다. 상대적인 평가를 위해 논문[4]에서 제안된 DCG 기법이 활용되었다. 최대값, 최소값을 활용하는 시간보증기법에서는 최근 20개(k=20)의 노드를 활용하여 계산하였다. 우리는 업데이트율을 식 6의 λ를 활용하여 조정하였다. 작은 λ가 의미하는 바는 업데이트가 자주 일어나는 것을 의미한다. 그림 2에서 볼 수 있듯이 최대값을 시간보증으로 사용한 기법이 접근시간에서 가장 좋은 성능을 보인다. 또한 제안된 기법은 모두 기존의 DCG만을 사용한 기법보다는 접근 시간 면에서 좋은 성능을 보인다. 그 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 3) 다양한 속도에 따른 접근시간

그림 3은 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 제안하는 기법들은 현재성의 손실이 존재한다. 현재성 손실을 측정하기위해 본 논문에서 제안하는 기법들이 원본과 다른 데이터를 접근할 확률을 표 2에 나타내었다. 3.1절에서 설명한 바와 같이 현재성 손실이 일어날 확률이 적으며 이는 표 2를 통해 증명된다.

<표 2> 현재성의 손실

	AUT	K-UT (k=20)	MIN-KU (k=20)	MAX-KU (k=20)	HISTORY
Stale data access	98.92	98.50	99.29	97.74	98.77
Up-to-date data access	1.08	1.50	0.71	2.26	1.23

5. 결론

본 논문에서는 모바일 애드혹 네트워크 환경에서 시간제약을 가지는 응용을 위해 데이터 접근시간향상 시키는 새로운 데이터 일관성 확보기법을 제안하였다. 제안된 기법은 시간보증에 기반하여 데이터 일관성을 확보한다. 향후 연구로써 시간제약을 가진 응용이 아닌 범용 응용을 위한 데이터 일관성 확보 기법 및 보다 정확성 높은 시간보증 확보 기법에 대해 연구할 예정이다.

참고문헌

[1] D. Barbara and T. Imielinski. Sleepers and workaholics: Caching strategies in mobile environments. In *Proceedings of ACM SIGMOD Conference on Management of Data*, pages 1-12, 1994.

[2] G. Cao, L. Yin, and C. R. Das. Cooperative cache based data access in ad hoc networks. *IEEE Computer*, 37(2):32-39, 2004.

[3] 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*, pages 815-826, 2004.

[4] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *Proceedings of IEEE INFOCOM*, pages 1568-1576, 2001.

[5] 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*, pages 869-881, 2004.

[6] H. Hayashi, T. Hara, and S. Nishio. Cache invalidation for updated data in ad hoc networks. In *Proceedings of CoopIS/DOA/ODBASE*, pages 516-535, 2003.

[7] H. Hayashi, T. Hara, and S. Nishio. Updated data dissemination in ad hoc networks. In *Proceedings of Ubiquitous Mobile Information and Collaboration Systems*, pages 28-42, 2004.

[8] 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*, pages 306-321, 2006.

[9] 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*, 5(5):492-507, 2006.

[10] T. Imielinski, S. Viswanathan, and B. Badrinath. Data on air: Organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353-372,1997.

[11] A. Kahol, S. Khurana, S. K. Gupta, and P. K. Srimani. A strategy to manage cache consistency in a disconnected distributed environment. *IEEE Transaction on Parallel Distributed Systems*, 12(7):686-700, 2001.

[12] D. J. Ram, M. U. Mahesh, N. S. K. C. Sekhar, and C. Babu. Causal consistency in mobile environment. *ACM SIGOPS*, 35(1):34-40, 2001.

[13] K. Ramamritham, R. Sivasankaran, J. A. Stankovic, D. T. Towsley, and M. Xiong. Integrating temporal, real-time, and active databases. *SIGMOD Record*, 25(1):8-12, 1996.

[14] J. C.-H. Yuen, E. Chan, K. yiu Lam, and H.-W. Leung. Cache invalidation scheme for mobile computing systems with real-time data. *ACM SIGMOD Record*, 29(4):34-39, 2000.