

장기간 접속 단절된 이동 클라이언트를 위한 효과적 캐시 유지 기법

(An Efficient Cache Maintenance Scheme for Long Disconnected Mobile Clients)

박 광 진 [†] 송 문 배 ^{**} 강 상 원 [†] 황 종 선 ^{***}
(Kwangjin Park) (Moonbae Song) (Sangwon Kang) (Chongsun Hwang)

요 약 접근 빈도가 높은 데이터를 클라이언트의 캐시에 저장하여 사용하는 것은 한정된 무선 대역폭을 사용하는 이동 컴퓨팅 환경에서 매우 중요한 이슈이다. 그러나 클라이언트의 접속 단절이 빈번하게 발생하는 무선환경에서 캐시 데이터의 일관성을 보장해주기 위해서는 적지 않은 노력이 필요하다. 최근에는 캐시 데이터의 일관성 검사를 위하여 서버의 주기적인 무효화보고(Invalidation Report, *IR*) 기법을 이용한 연구가 활발하게 진행되고 있다. 하지만 클라이언트가 오랜 접속 단절이 발생할 경우 모든 캐시 데이터를 버려야 하므로, 성능에 큰 장점을 얻을 수 없게 된다.

이에 본 논문에서는 이동 컴퓨팅 환경에서 빈번한 접속 단절로 인하여 오랫동안 무효화 보고를 받지 못하더라도 유효한 캐시 데이터를 최대한 유지시킬 수 있는 기법들을 제안한다. 먼저 클라이언트가 접속 단절 후 첫 재접속이 되었을 경우, 자신의 접속 단절 시간을 전송하도록 하여 서버가 동적으로 *IR*에 포함될 이전 일정 주기를 결정하도록 하였다. 이에 반하여 두 번째 기법에서는 서버가 특정 기간 동안의 *IR* 정보를 저장하도록 하여, 오랜 접속 단절 후에도 클라이언트가 직접 캐시 데이터의 일관성 검사를 요청할 수 있도록 하였다. 제안한 기법에 의해 현저히 줄어드는 클라이언트의 캐시 손실률과 그에 따른 업링크 수의 감소, 그리고 질의에 대한 응답시간의 감소로 인한 시스템 성능 향상은 시뮬레이션을 통해 확인된다.

키워드 : 이동 컴퓨팅, 무효화보고(*IR*), 캐시 일관성

Abstract In the mobile computing environment, caching data at the client's side is a useful technique for improving the performance. For caching to be effective, the cache contents must be consistent with those stored in the server. That is, unfortunately, difficult to enforce due to the frequent disconnection and mobility of clients. In the literature, the basic approach adopted is for the server to periodically broadcast invalidation reports(*IRs*) that contain information about objects that have been updated recently. However *IR*-based cache maintenance scheme has a main drawback. Any client who has been disconnected longer than w seconds has to discard all cached items even some of them may still be valid.

In this paper, we propose two main schemes which can adapt clients' losing *IR* messages by long disconnection. First, the client sends its disconnection time when connection is re-established, and the server uses this information to decide the size of broadcast window w dynamically. Second, the server maintains *IR* messages for a given period of time and a client who has been disconnected longer than w seconds can verify its cache consistency individually. Compared to previous *IR*-based schemes, our schemes can significantly reduce cache missing and uplink request and improve the throughput by reducing response time.

Key words : Mobile Computing, Invalidation Report(*IR*), Cache Consistency

[†] 비 회 원 : 고려대학교 기초과학기술연구소 연구원
kjpark@disys.korea.ac.kr
swkang@disys.korea.ac.kr

^{**} 정 회 원 : 고려대학교 기초과학기술연구소 연구원
mbsong@disys.korea.ac.kr

^{***} 종신회원 : 고려대학교 컴퓨터학과 교수
hwang@disys.korea.ac.kr
논문접수 : 2004년 1월 13일
심사완료 : 2005년 2월 15일

1. 서론

이동 컴퓨팅 환경의 발전으로, 사용자는 자신의 물리적인 위치와 관계없이 정보를 접근할 수 있게 되었다. 사용자는 “언제 어디서나” 정보의 접근이 가능하기를 바라며 그들이 원하는 요구 또한 다양해지고 있다. 비대칭 통신 환경, 빈번한 접속 단절, 제한된 자원 등의 속성을 가진 이동 컴퓨팅 환경에서 캐시의 사용은 점차 매우 중요한 비중을 차지하고 있다.

캐시의 사용은 클라이언트의 서버에 대한 의존도를 낮춰주고 클라이언트 자원을 효율적으로 사용할 수 있도록 도와주며 사용자의 데이터에 대한 접근 시간을 줄여주는 역할을 한다. 그림 1은 일반적인 이동 컴퓨팅 환경의 시스템 구조를 보여주고 있다.

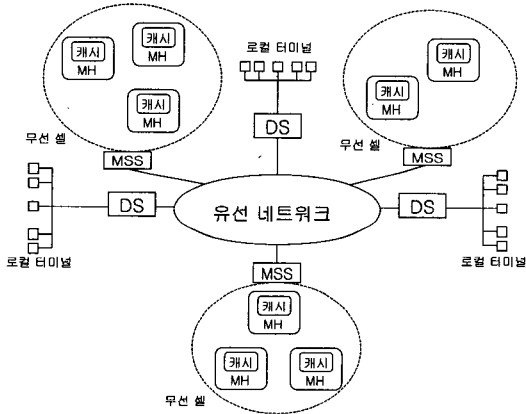


그림 1 이동 컴퓨팅 환경

본 논문에서의 MSS는 그림 2에서 보이는 것과 같이 다음과 같은 다섯 가지 주요 작업을 수행한다:

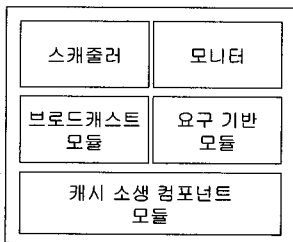


그림 2 MSS 구성 요소

- 브로드캐스트 모듈: 브로드캐스트 데이터 유지 및 해당 셀 내의 모든 클라이언트에게 데이터 전송
- 요구 기반 모듈: 이동 클라이언트의 질의에 대한 데이터 전송

- 캐시 소생 컴포넌트 모듈: 특정 주기 동안의 IR 정보 보관 및 공급
- 모니터: 각 주기마다 이동 클라이언트가 보내온 질의와 접속 단절 시간 정보를 감시
- 스케줄러: 브로드캐스트 데이터와 요구 기반 데이터의 사용 비중 결정. 예를 들어 서버의 작업 부하가 높을 경우 브로드캐스트 데이터의 사용 비중을 높이고 반대의 경우 요구 기반 데이터의 사용 비중을 높인다.

캐시 데이터를 사용하기 위해서는 데이터의 일관성에 대한 문제를 해결해야하는데, 접속 단절이 빈번하게 발생하고 클라이언트의 수가 유동적인 이동 컴퓨팅 환경에서는 주로 서버가 IR 메시지를 보내주며, 클라이언트는 이 메시지를 이용하여 캐시 데이터의 일관성을 유지하는 방식을 사용한다. 이 경우 서버는 클라이언트 캐시에 대한 상태정보를 유지하지 않으며 오로지 서버내의 갱신된 데이터 아이템에 대한 정보만 전달한다. 그러나 캐시 데이터의 일관성을 보장 해주기 위해 주기적인 IR 메시지를 보내는 방식은 클라이언트가 캐시 데이터를 사용하기 전 데이터의 유효성 여부를 판단하기 위해 다음 IR이 올 때까지 기다려야 하며, 장기간 접속 단절되어 IR 메시지를 놓친 클라이언트의 경우, 무조건 모든 캐시 데이터를 버려야하는 문제점이 있다.

위와 같은 문제점들을 극복하기 위해 Bit Sequence (BS) 기법[1]과 Updated Invalidation Report(UIR) 기법[2] 등이 제안되었다. BS 기법은 TS(Broadcasting Timestamp) 기법[3]이 갖는 문제점인 오랜 접속 단절시 발생하는 문제에 대처하기 위해 계층화된 비트열을 전송하였다. 이에 반하여 UIR 기법은 그림 4와 같이 IR 메시지 주기 사이에 적절한 정보(UIR)를 보냄으로써 TS 기법이(그림 3 참고) 갖는 질의 처리 시간의 지연 문제를 극복하였다. 하지만 이 기법들은 클라이언트가 받게되는 불필요한 정보의 양이 많아지거나, 오랜 접속 단절에 효율적으로 대처할 수 없다. 따라서 본 논문에서는 클라이언트로부터 서버로의 데이터 요구 메시지

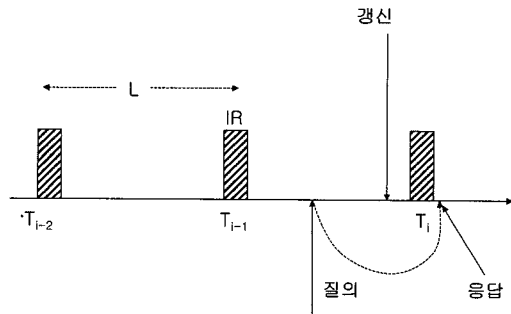


그림 3 TS 기법

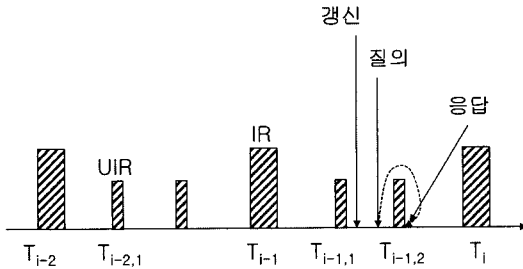


그림 4 UIR 기법

의 횟수를 줄이고, 또한 오랜 접속 단절에 대처할 수 있는 캐시 유지 기법을 제안한다. 이를 위하여 먼저 서버가 관리하는 데이터를 클라이언트의 접근 빈도에 따라 브로드캐스트 전용 및 요구 기반 처리 데이터로 나눈다. 이후 클라이언트에 대한 일정 기간 이상의 접속 단절이 발생하면, 다음 두 가지 기법 중 하나를 적용한다. 먼저 클라이언트가 접속 단절 후 첫 재 접속을 시도할 때 자신의 접속 단절 시간 정보를 서버에게 보내줌으로써, 서버가 IR 의 이전 일정 주기동안의 윈도우 크기 w 를 동적으로 결정하도록 한다. 두 번째 방식은, 서버가 이전 일정 주기동안의 IR 에 대한 정보를 별도의 공간(캐시 소생 컴포넌트)에 저장해두도록 한다. 이후 클라이언트가 자신의 캐시 데이터에 대하여 명시적으로 유효성을 검사하도록 한다.

본 논문의 구성은 다음과 같다: 2장에서는 통신환경과 데이터 전달 방식 그리고 기존 연구에서 사용되는 w 와 L 의 속성 및 해당 매개변수가 시스템에 미치는 영향에 대하여 기술한다. 3장에서는 시스템 모델과 구체적인 알고리즘을 기술하고, 4장에서는 제안하는 기법의 캐시 손실률과 그에 따른 성능향상을 TS 기법과 비교하고, 마지막으로 5장에서는 결론 및 향후 연구과제를 기술한다.

2. 기존 연구

2.1 비대칭 통신 환경

일반적인 이동 컴퓨팅 환경에서는 주로 비대칭 통신 환경을 사용한다. 비대칭 통신 환경은 서버에서 클라이언트(downstream)로의 정보 흐름의 양이 그 반대 경우인 클라이언트에서 서버로 전달(upstream)하는 정보 흐름의 양보다 훨씬 큰 것을 의미한다. 비대칭 통신 환경을 사용하는 경우는 소수의 서버와 다수의 클라이언트에 의한 경우나 서버가 데이터 아이템을 갱신할 경우, 또는 새로운 아이템을 생성한 뒤 클라이언트에게 알려주어야 할 경우 주로 사용하게 된다. 따라서 비대칭 통신 환경은 시스템의 특성뿐 아니라 어플리케이션에서의 데이터 흐름에 의해서도 결정된다. 비대칭 통신은 네트워크 어플리케이션에서 서버와 클라이언트에게 특정한 제약성

을 가져다줄 수 있다. 즉, 클라이언트의 수가 서버보다 훨씬 많은 시스템의 경우 클라이언트로부터 서버로 향한 정보 전달에 많은 제약성을 갖게 된다. 그 예로 WWW(World Wide Web)와 FTP(File Transfer Protocol) 서버는 주어진 시간에 허용할 수 있는 접속자의 수를 제한함으로써 그러한 문제를 해결하고 있다.

2.2 무효화 메시지

IR 은 갱신된 데이터 아이템에 대한 정보와, 마지막 변경된 가장 최신 값을 갖는 시간의 범위인 w , 그리고 IR 방송 주기인 L 로 크게 분류해볼 수 있다. 이들 각각의 변수는 클라이언트의 접속 단절시간과 갱신되는 데이터 아이템의 양 등과 관련하여 최적의 상태로 조정될 수 있다. 예를 들어, IR 의 방송 주기 L 을 길게 할 경우 IR 을 놓치는 클라이언트의 수는 줄어드는 반면 대기해야 할 클라이언트의 수가 증가하게되며, 반대로 IR 의 방송 주기 L 을 작게 할 경우 IR 을 듣지 못하고 모든 캐시 데이터를 버려야 하는 클라이언트의 수가 증가하게 된다. 마찬가지로 w 크기를 크게 잡으면, IR 에 포함되는 정보 데이터 양이 많아져 클라이언트의 불필요한 정보 청취를 가져올 수 있으며, 반대로 너무 작게 잡을 경우 IR 을 놓친 클라이언트의 수가 증가함으로써 질의 요청에 따른 서버의 작업 부하가 발생할 수 있다. 그러한 이유로 클라이언트의 접속 단절 시간과 w 크기 그리고 L 주기의 변화는 주기적인 IR 메시지를 사용하여 캐시의 일관성을 유지하는 방식을 사용할 때 매우 중요한 변수가 될 수 있다. 따라서 그러한 환경에서 발생할 수 있는 여러 가지 상황들 즉, 사용자 질의에 대한 대기 시간, 서버의 작업 부하, 네트워크 대역폭의 효과적인 사용 등을 최적화하기 위해 앞서 언급했던 여러 요소들을 고려한 적절한 w 크기 그리고 L 주기의 조정이 필요하다.

3. 시스템 환경

시스템 환경을 구성하기 위해서 여러 다양한 모델을 사용할 수 있다. 일반적인 이동 컴퓨팅 환경은 업링크와 다운링크 통신채널로 구성되어지며, 본 논문에서는 업링크와 다운링크 메시지 전송을 모두 허용하는 혼합 방식을 사용한다. 서버는 본 논문의 시스템 모델에서 데이터는 서버에 의해서만 갱신되며 브로드캐스트 채널과 업링크 채널은 서로 독립적으로 사용되며 클라이언트의 접속 단절은 주어진 시간 범위 내에서 발생한다고 가정한다.

3.1 기본 모델

본 논문에서는 전달할 데이터를 브로드캐스트 데이터와 요구 기반 데이터로 분류한다. 서버가 전달할 데이터는 브로드캐스트방식을 사용하여 클라이언트들에게 전달하고, 요구 기반 데이터는 클라이언트의 요청에 따라 다운링크 채널을 사용하여 해당 클라이언트에게 개별적

으로 전달하게 된다. 데이터 분류를 위한 기준은 [4]에서 제시한 방식을 사용하였다.

3.1.1 서버 측 모델

서버가 관리하는 데이터는 크게 세 가지로 브로드캐스트 데이터, IR 메시지, 요구 기반 데이터로 분류할 수 있다. D를 데이터 아이템들의 집합이라 하고 T_i 를 IR이 방송되어지는 시점의 i 번째 시간이라 가정할 때 서버는 IR_i 메시지 구성을 위해 다음과 같은 갱신된 데이터 목록 U_i 를 유지한다.

$$U_i = \{[j, t_j] \mid j \in D \text{ 이고 } T_i - w \leq t_j \leq T_i\}.$$

j 는 $T_i - w$ 동안에 갱신된 데이터이고, t_j 는 j 의 마지막 갱신시간이라 할 때, 서버는 단일 네트워크 채널을 사용하며 IR과 브로드캐스트(데이터 객체)데이터를 IR 단계와 데이터 전달(Data Delivery) 단계로 나누어 전달한다. 또한 서버는 브로드캐스트 데이터를 주기적 간격(L)으로 클라이언트에게 전달한다. 이때 브로드캐스트 데이터의 구조는 플랫폼(flat)한 구조이며, 데이터가 브로드캐스트 되어지는 시작 시점에서 먼저 IR 메시지를 전송한다. IR 메시지는 이전 주기에 서버에서 갱신된 데이터 아이템의 ID 번호로 구성된다. 브로드캐스트 서버는 다운링크채널을 통하여 IR 메시지와 데이터 객체를 전달하며 전달되는 데이터의 종류와 브로드캐스트 주기는 이전 브로드캐스트 주기에 클라이언트로부터 전달받은 데이터 요청 메시지와 시간정보에 의해 결정된다. 또한 업링크 채널은 서버가 클라이언트로부터의 요청을 받는데 사용되며, 요청 받은 메시지는 그림 5에서와 같이 서버의 요청 데이터 큐와 백업 데이터 큐에 저장되어 다음번 브로드캐스트 주기의 데이터 구성 및 전달 주기를 결정하는데 사용된다. 서버는 이전 주기에 클라이언트들로부터 받은 정보를 통해 다음과 같은 작업을 수행한다.

- (i) 데이터 항목 x에 대한 질의를 받은 경우
 x에 대한 질의 요청 카운터를 증가 시킨다.
 if (x의 요청수가 기준값을 초과하면) { // x가 브로드캐스트 데이터인 경우 다음 브로드캐스트 목록에 포함
 else 개별적인 질의 처리를 통해 답변
 }
- (ii) 주어진 값을 초과하여 클라이언트가 접속 단절되는 경우
 다음 브로드캐스트의 w 값 증가
 다음 브로드캐스트의 L 값 감소

3.1.2 클라이언트 측 모델

클라이언트는 서버로부터 IR 정보를 받아 캐시의 유효성 여부를 판단한다. 클라이언트는 i 번째 시간에 해당하는 질의 목록 Q_i 를 유지한다.

$$Q_i = \{[j] \mid [j] \text{ 는 } [T_{i-1}, T_i] \text{의 주기동안 클라이언트가 사용자로부터 받은 요구 기반 데이터에 대한 질의임}\}$$

클라이언트는 브로드캐스트 채널을 통해 서버의 브로드캐스트 데이터를 청취하며 업링크 채널을 통해 서버에게 데이터를 요청한다. 또한 각각의 클라이언트는 캐시를 유지하며 캐시 데이터를 사용하기 전에 서버의 IR 메시지를 사용하여 캐시의 유효성 여부를 판단한다.

구체적인 알고리즘은 다음과 같다.

- (i) 데이터 항목 x에 대한 질의를 받은 경우
 if ($x \in$ 브로드캐스트 데이터) { // x가 브로드캐스트 데이터인 경우
 if ($x \in$ 캐시 && 현 주기의 IR을 받은 경우) 캐시데이터 사용
 else 브로드캐스트 데이터 사용
 }
 else { // x가 요구 기반 데이터인 경우 - 서버에게 요청
 if ($x \in$ 캐시 && 현 주기의 IR을 받은 경우) 캐시데이터 사용
 else if ($x \in$ 캐시 && 현 주기의 IR을 받지 않은 경우) 다음 IR 기다림
 else 서버에게 데이터 요청
 }
- (ii) 주기적인 IR을 받은 경우, T_i 를 클라이언트가 가장 마지막에 받은 IR의 시간이라 할 때,
 if ($T_i - T_1 > w$) { 모든 캐시 데이터 버리기 // 적응적 장기간 접속 단절 대처 기법 알고리즘 적용
 }
 else {
 캐시 데이터 일관성 검사;
 if (연산이 대기중인 경우) 알고리즘 i 수행// 질의에 대한 응답
 }
 $T_i := T_i$;

3.2 장기간 접속 단절된 이동 클라이언트의 캐시 유지 기법

이 모델은 기존에 클라이언트가 w 주기 이상 접속 단절되어 IR메시지를 놓쳤을 경우 자신의 캐시 데이터를 모두 버려야 하는 문제점을 극복하고 캐시내의 유효한 데이터를 최대한 살리는데 목적이 있다. 클라이언트의 캐시 데이터를 표 1과 같이 유효(valid), 불분명(unknown), 스테일(stale) 데이터로 분류하였으며, 본 논문에서는 불분명(unknown)데이터를 버리지 않고 최대한

살리는데 초점을 맞추었다. 따라서 3.2.1절에서는 장기간 접속 단절된 클라이언트가 자신의 접속 단절 시간을 서버에게 보내주어 w 크기를 조정하여 줌으로써 IR 을 놓쳐 모든 캐시 데이터를 버리게되는 클라이언트의 숫자를 최소화할 수 있는 방식을 소개하며 3.2.2절에서는 IR 메시지를 놓친 클라이언트가 개별적으로 캐시 소생 컴포넌트(cache resuscitation component)에게서 받은 백업 데이터를 통해 자신의 캐시 데이터를 최대한 유지할 수 있는 기법을 소개하고자 한다.

표 1 IR 메시지 확인에 따른 데이터 분류

유효(valid)	IR 메시지를 통해 갱신되지 않았음을 확인한 데이터
불분명(unknown)	접속 단절로 인해 IR 메시지를 받지 못해 갱신 여부를 확인할 수 없는 데이터
스태일(stale)	IR 메시지를 통해 갱신되었음을 확인한 데이터

3.2.1 적응적 장기간 접속 단절 대처 기법

제안한 시스템 모델에서 장기간 접속 단절이 발생하는 클라이언트의 수가 증가하여 업링크 채널을 통한 클라이언트의 질의 요청이 빈번하게 발생하게 되면 업링크 채널할당에 따른 대역폭 문제, 질의 요청에 따른 서버의 작업부하, 그리고 클라이언트의 수가 증가할 경우의 확장성과 같은 다양한 문제가 발생할 수 있다. 따라서 본 논문은 장기간 접속 단절된 클라이언트가 재접속 이후 서버에게서 받은 IR 정보를 통해 자신이 w 크기 이상으로 접속 단절되었음을 확인하면, 질의 요청 시 업링크 채널을 통해 서버에게 자신의 접속 단절 시간 정보(D_T)를 함께 첨부하여 보내준다. 서버는 일정주기 동안 클라이언트가 보내준 단절시간 정보를 통해 w 와 L 그리고 D_T 와의 관계를 고려하여 다음 번 브로드캐스트 데이터의 w 크기를 동적으로 결정하도록 한다.

클라이언트는 다음과 같은 알고리즘을 실행한다.

```

if(클라이언트가  $w$  주기 이상 접속 단절되어  $IR$  메시지를 못 받은 경우){
    서버에게 자신의 접속 단절 시간정보 전송
    자신이 유지하고 있는 모든 캐시 정보를 불분명 상태로 전환
    if(클라이언트가 다음번 주기에서  $IR$ 을 받은 경우){
        if(조정된  $w$  크기로 인해 놓친  $IR$ 정보를 다시 얻은 경우){
            불분명 데이터 중 갱신되지 않은 데이터는 유효로 전환하고 갱신된 데이터는 캐시에서 버린다.
        }
    }
}
    
```

```

else 모든 캐시 데이터를 버림
}
else  $IR$  메시지가 올 때까지 채널 청취
else 다음번  $IR$ 을 청취하여 갱신된 데이터는 스태일로 갱신되지 않은 데이터는 유효로 표시
}
    
```

3.2.2 캐시 소생 컴포넌트를 사용한 기법

앞서 장기간 접속 단절 대처기법에 의해 클라이언트의 평균 접속 단절 시간이 길어질 경우 w 와 L 을 조정하여 클라이언트의 캐시 손실에 대한 문제점을 극복하는 것을 보았다. 그러나 이 경우 무리하게 w 와 L 을 늘림으로써 다음과 같은 두 가지 문제가 발생 할 수 있다. 첫째, w 가 너무 커질 경우 브로드캐스트 하게 될 데이터양이 증가하여 불필요한 정보를 받게 되는 클라이언트의 수가 증가할 수 있다. 둘째, L 이 너무 커질 경우 브로드캐스트 데이터를 기다리는 클라이언트의 대기 시간이 무리하게 길어질 수 있다. 따라서 이 경우 N 을 캐시 소생 컴포넌트가 유지하는 이전 주기 범위라 할 때 캐시 소생 컴포넌트에 $w*N$ 주기의 데이터를 유지하고 있다가 요청이 들어올 경우 해당 클라이언트에게 전달해주는 방식을 취한다. 즉 장기간의 접속 단절로 인해 자신의 캐시에 유지하고 있는 데이터의 유효성 여부를 확인할 수 없는 클라이언트는 캐시 소생 컴포넌트의 정보를 사용함으로써 기존의 IR 을 놓친 클라이언트가 자신의 캐시 데이터를 모두 버려야하는 문제점을 극복할 수 있도록 한다. 또한 백업 데이터는 그림 6의 (b)와 같이 IR 주기마다 갱신된 데이터 아이템의 번호를 데이터베이스내의 각각의 아이템에 일련번호를 부여하여 나타내도록 한다. 따라서 각각의 일련번호가 해당 아이템을 나타내도록 구성하여 전달하게 될 전체 메시지의 크기를 줄여줌으로써 캐시 소생 컴포넌트 사용에 따른 부가적인 메시지 부담을 최소화하였다. 그림 5와 같이 데이터와 백업 정보를 얻기 위한 클라이언트의 질의 요청은 업링크 채널을 통해 서버에게 전달되며 전달된 정보는 스위치를 통해 각각의 큐로 보내진 뒤 해당 정보를 다운링크 채널을 통해 얻게 된다.

서버 프로토콜

캐시 소생 컴포넌트의 백업 데이터의 시간 범위(B_i)는 다음과 같은 특성을 갖는다.

$$T_{i-N} \leq B_i \leq T_i$$

서버는 푸시 채널을 통해 전달되는 w 주기 데이터에 대하여 N 크기만큼의 백업 데이터를 캐시 소생 컴포넌트에 유지한다. 그림 6의 (a)는 T_i 를 현재 브로드캐스트 되는 IR 시간이라 할 때 T_{i-3} 주기까지의 백업 데이터를 유지하고 있는 서버의 상태를 보여주고 그림 (b)는

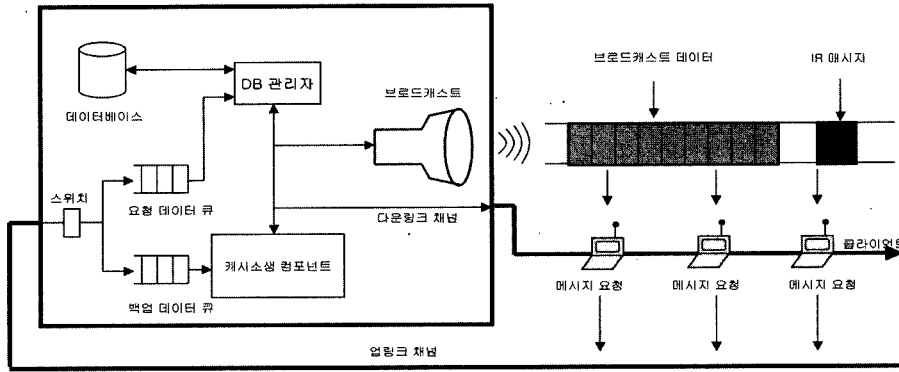
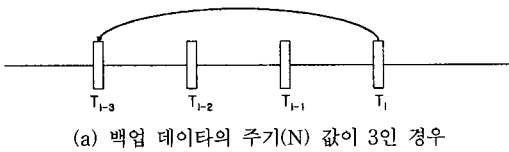


그림 5 혼합 데이터 전달 방식 모델



(a) 백업 데이터의 주기(N) 값이 3인 경우

N 값	갱신된 데이터 아이템 번호									
0 (T_i)	3	12	15	23	26	78	102	121	145	205
1 (T_{i-1})	22	34	58							
2 (T_{i-2})	100	124	153	200	301	314				
3 (T_{i-3})	25									

(b) N 값과 데이터의 갱신 목록 정보

그림 6 캐시 소생 컴포넌트 구성 요소

해당 갱신 목록 정보를 보여주고 있다. 그림과 같이 캐시 소생 컴포넌트에는 $w*N$ 주기 동안에 갱신된 정보를 백업 데이터에 유지한다. 이때 백업 데이터는 각각의 IR 주기에 갱신된 데이터 아이템 번호로 구성된다. 캐시 소생 컴포넌트의 데이터는 현재 서버의 브로드캐스트 데이터 목록이 전송된 뒤 갱신된다.

4. 성능 평가

여러 IR-기반 캐시 무효화 알고리즘들이 장기간 접속 단절된 클라이언트의 캐시 문제를 다루기 위해 제안되었다. 그중 본 논문은 응답지연시간이 우수한 TS 기법과 본 논문에서 제안한 기법을 서로 비교 분석하여 성능 평가를 수행하였다. 이후 성능 평가에서는 적응적 장기간 접속 단절 대처 기법과 캐시 소생 컴포넌트 기법을 함께 사용하여 이루어졌으므로 두 제안 기법을 ATS(Adaptive Broadcasting Timestamp) 기법으로 통합하여 표기하도록 하였다.

4장에서는 앞서 설명한 혼합 데이터 전달 방식 모델(그림 5 참고)을 기본 환경으로 하여 다양한 방식으로 성능 향상을 비교 분석한다. 4.1절에서는 4.2.1-4.2.3절에

서의 성능 분석을 위한 기본 가정을 정립하고, 관련 매개변수에 대하여 정의한다. 마지막으로 4.2절에서는 성능 평가의 결과를 그래프를 통해 나타내었다.

4.1 기본 가정 및 관련 매개변수

본 논문에서 사용하는 시스템 매개변수는 표 2와 같다.

표 2 성능평가 매개변수

매개변수	변수 범위
클라이언트 수	1 - 100개
데이터 개수	1000개
데이터 크기	256바이트
IR 리포트 주기	700 주기
클라이언트 접속 단절 시간	0 - 30(ms)
브로드캐스트 데이터 적중률	20 - 90(%)
윈도우 크기	0 - 10
클라이언트 캐시 크기	100
사용자 질의에 대한 캐시 적중률	1 - 100(%)
백업 데이터 범위	윈도우 크기*N
다운링크 채널 할당	144kbps(7.1ms/데이터)
업링크 채널 할당	36kbps

4.2 성능 평가에 대한 결과

이 절에서는 시뮬레이션 결과를 통하여 나온 TS 기법, UIR기법 그리고 BS 기법과 제안 기법인 ATS 기법간의 다양한 성능 비교에 대한 결과를 살펴보도록 한다. 4.2.1절에서는 무선 네트워크 환경에서의 캐시 손실률을 비교하고, 4.2.2절에서는 클라이언트의 질의 수를, 마지막으로 4.2.3절에서는 사용자 질의에 대한 클라이언트의 응답시간을 비교 평가하였다.

4.2.1 캐시 손실

질의 수와 사용자 질의에 대한 클라이언트의 응답시간 그리고 처리량 등은 클라이언트의 캐시 손실에 의해 많은 영향을 받는다.

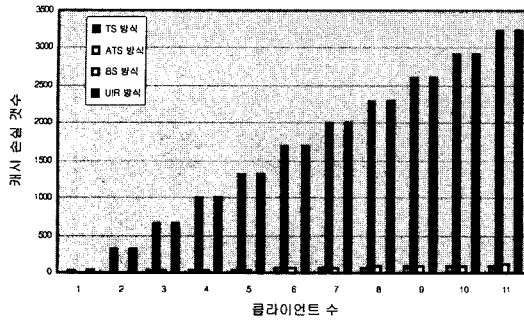


그림 7 평균 접속 단절 시간이 짧은 경우의 클라이언트의 캐시 손실 갯수

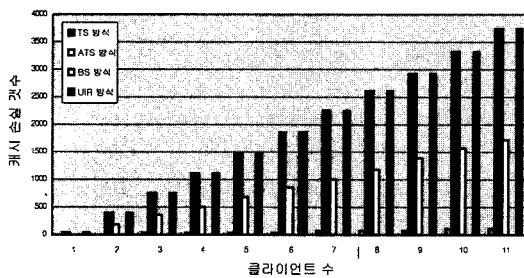


그림 8 평균 접속 단절 시간이 긴 경우의 클라이언트의 캐시 손실 갯수

그림 7은 클라이언트의 평균 접속 단절 기간이 짧은 경우 ATS 기법과 TS 기법 그리고 UIR 기법과 BS 기법의 클라이언트 수에 따른 캐시 손실률을 그래프로 보여주고 있다. 그래프를 통해 ATS 기법을 사용함으로써 클라이언트의 수가 증가해도 TS 기법에 비해 현저히 적은 량의 클라이언트 캐시 손실을 가져오는 것을 볼 수 있다. 이는 ATS 기법을 사용하여, w 주기를 초과하여 접속 단절된 클라이언트는 캐시 소생 컴포넌트의 백업 데이터를 이용하여 캐시 손실률을 줄여줌으로써 얻게 되는 결과이다. 또한 이 실험에서 BS 기법이 다른 기법에 비해 상대적으로 적은 량의 캐시 손실을 보이는 것을 알 수 있는데, 이는 앞서 서론에서 언급한 바와 같이 이전 시간의 모든 갱신정보를 무효화 메시지가 포함하고 있기 때문이다. 그러나 이 경우 클라이언트가 받게 되는 불필요한 정보의 양이 증가하게 되는 문제가 발생한다[2]. UIR 기법은 TS 기법과 같이 클라이언트가 w 주기를 초과할 경우 모든 캐시의 데이터를 버려야하므로 사용자 증가에 따른 캐시 손실률이 그림에서 볼 수 있듯이 같다. 그림 8은 클라이언트의 평균 접속 단절 시간이 긴 경우 ATS 기법과 TS 기법 그리고 UIR 기법과 BS 기법의 클라이언트 수의 증가에 따른 캐시 손실률을 그래프로 보여주고 있다. 그림에서 보여지는 것과

같이 클라이언트의 평균 접속 단절 시간이 긴 경우의 캐시 손실률도 ATS 기법이 TS 기법과 비교하여 우수한 성능을 보여주고 있다. 또한 앞서 그림 7에서의 실험과 같이 BS 기법이 다른 기법에 비해 상대적으로 적은 량의 캐시 손실을 보이는 것을 알 수 있다.

4.2.2 질의의 수

이 절에서는 브로드캐스트 데이터의 적중률이 낮을 때와 높을 때 발생하는 질의의 수를 각각의 기법들과 비교 평가하고 있다.

그림 9는 클라이언트의 브로드캐스트 데이터 적중률이 낮은 경우의 질의의 수를 각 기법과의 비교를 통해 보여주고 있다. 또한 그림 10은 브로드캐스트 데이터의 적중률이 높은 경우를 비교 평가하였다. 그림 10에서 보여지는 것과 같이 브로드캐스트 데이터의 적중률이 높은 경우 그 반대의 경우를 나타낸 그림 9와 비교하여 TS 방식과 ATS 방식 모두 현저히 낮은 수의 질의가 발생함을 알 수 있다. 이는 클라이언트 캐시 적중률이 낮아도 두 기법 모두 대부분의 데이터를 브로드캐스트 데이터에서 얻을 수 있기 때문이다. 따라서 이 경우 단순히 질의 숫자로 두 기법을 비교할 때 성능의 큰 차이를 보

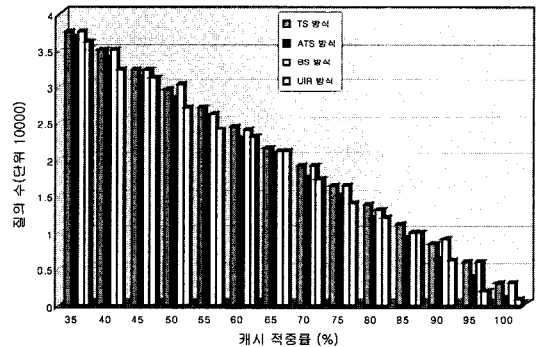


그림 9 브로드캐스트 데이터에 대한 적중률이 20%인 경우의 질의 수 비교

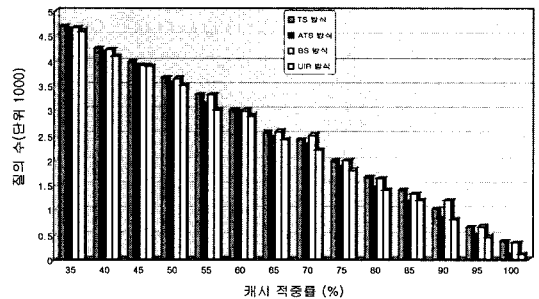


그림 10 브로드캐스트 데이터에 대한 적중률이 70%인 경우의 질의 수 비교

이지 않는다. 그러나 다음절에서 비교하게 될 전체 응답 시간을 서로 비교해 보면 제안한 기법이 더 우수하게 나타나는 것을 알 수 있다.

4.2.3 응답 지연 시간

본 논문은 질의에 대한 응답 시간을 캐시 데이터 사용과 브로드캐스트 데이터 사용 그리고 업링크 질의를 통한 응답 시간으로 분류하여 각각의 시간 값을 산출하였다. 즉 요청 받은 데이터가 클라이언트의 로컬 캐시에 있는 경우와 캐시에 없고 브로드캐스트 데이터에 있는 경우, 마지막으로 업링크 질의를 통해 얻는 경우로 분류하여 각각의 시간 값을 부여하였다. BS 기법은 다른 기법과는 달리 IR의 갱신 주기가 누적됨에 따라 전달해야 할 데이터의 양이 증가하므로, 동일한 환경에서의 평가가 어려우므로 응답 지연 시간에 대한 성능 평가 대상에서 제외하였다.

그림 11과 12에서 보여지는 것과 같이 브로드캐스트 데이터 적중률이 높은 경우의 응답시간이 그렇지 않은

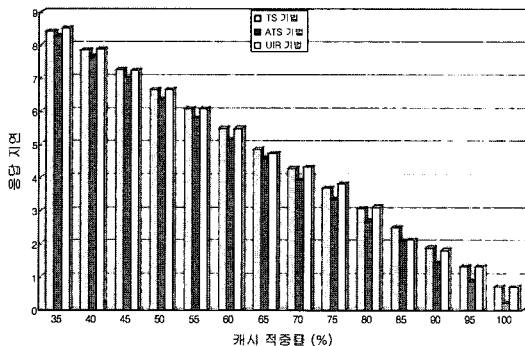


그림 11 브로드캐스트 데이터에 대한 적중률이 낮은 경우의 질의에 대한 응답 지연 비교

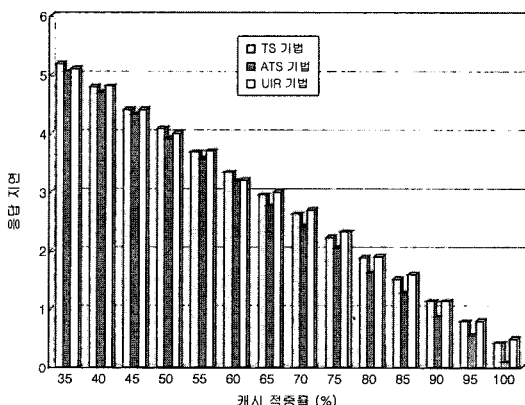


그림 12 브로드캐스트 데이터에 대한 적중률이 높은 경우의 질의에 대한 응답 지연 비교

경우보다 상대적으로 빠른 것을 볼 수 있다. 앞서 4.2.2 절에서 질의 수를 비교할 때 브로드캐스트 데이터 적중률이 높은 경우 TS 기법과 제안 기법 모두 비슷한 질의 숫자를 나타내는 것을 볼 수 있었다, 그러나 응답 시간을 비교하면 그림 12와 같이 캐시 적중률이 증가할수록 ATS 기법이 TS 나 UIR 기법에 비해 월등히 우수함을 보여주고 있다.

5. 결론

앞서 서론 부분에서 언급한 바와 같이 본 논문은 이동 컴퓨팅 환경에서 기존 캐시 유지 기법들이 갖는 단점인 장기간 접속 단절된 클라이언트의 캐시 손실을 최대한 막는데 그 초점을 맞추었다. 즉 적응적 장기간 접속 단절 대처 기법을 통해 w 주기를 넘어 접속 단절된 클라이언트의 수를 줄여줌으로써 자신의 캐시 데이터가 유효함에도 버려야하는 문제점을 최소화하였다. 또한 캐시 소생 컴포넌트 기법을 사용함으로써 개별적인 클라이언트에 대한 캐시 유효성을 보장해줌으로써 잦은 접속 단절과 같은 열악한 무선환경에서의 클라이언트 캐시를 최대한 살리고자 하였다. 따라서 캐시를 버려야하는 클라이언트의 숫자를 최소화하고 개별적인 클라이언트의 캐시 유효성을 증명해 줌으로써 캐시 손실로 인한 업링크 발생을 최소화하고 그에 따른 서버의 작업 부하를 줄여 주었다. 끝으로 시뮬레이션 결과를 통해 제안한 기법이 기존 TS 기법에 비해 브로드캐스트 데이터의 적중률에 관계없이 적은 질의 수와 낮은 응답 시간을 보이는 것을 볼 수 있었다.

참고 문헌

- [1] Jin, Jing, A. Elmagarmid, A.S. Helal and R. Alonso, "Bit-Sequence: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," *ACM/Baltzer Mobile Networks and Application*, Vol. 2, No. 3, pp. 115-127, 1997.
- [2] Guohong Cao, "Scalable Low-latency Cache Invalidation Strategy for Mobile Environment," *International conference on Mobile computing and networking*, pp. 200-209, 2000.
- [3] Daniel Barbara, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," *ACM SIGMOD international conference on Management of data*, pp. 1-12, 1994.
- [4] Swarup Acharya and Michael Franklin, "Broadcast Disks: Data Management for Asymmetric Communication Environments," *ACM SIGMOD*, pp. 199-210, 1995.
- [5] Qinglong Hu and Dik Lun Lee, "Adaptive Cache Invalidation Method in Mobile Environments," *IEEE International Symposium on High Perfor-*

- mance Distributed Computing*, pp. 264-273, 1997.
- [6] Swarup Acharya and Michael Franklin, "Balancing Push and Pull for Data Broadcast," *In Proceeding of the 1995 ACM SIGMOD*, 1997.
- [7] Swarup Acharya and Michael Franklin, "Dissemination-based data delivery using broadcast disks," *IEEE Personal Communication*, pp. 50-60, 1995.
- [8] Evaggelia Pitoura and Bharat Bhargava, "Maintaining Consistency of Data in Mobile Distributed Environment," *In proceeding of IEEE*, pp. 404-413, 1995.



박 광 진

2000년 고려대학교 이학사. 2002년 고려대학교 이학 석사. 2004년 고려대학교 이학 박사 수료. 2001년~현재 고려대학교 기초과학기술 연구소 연구원. 관심분야는 모바일 컴퓨팅, 이동객체 데이터베이스, 데이터 전달



송 문 배

1996년 군산대학교 이학사. 1998년 숭실대학교 이학 석사. 2003년 고려대학교 이학 박사 수료. 2001년~현재 고려대학교 기초과학기술 연구소 연구원. 관심분야는 모바일 컴퓨팅, 이동객체 데이터베이스, 시공간 데이터 마이닝



강 상 원

1998년 고려대학교 이학사. 2003년 고려대학교 이학 석사. 2001년~현재 고려대학교 기초과학기술 연구소 연구원. 관심분야는 모바일 컴퓨팅, 이동객체 데이터베이스, 시맨틱 캐시



황 중 선

1966년 고려대학교 이학사. 1970년 고려대학교 이학 석사. 1978년 University of Georgia 전산학 박사. 1995년~현재 고려대학교 컴퓨터학과 교수. 관심분야는 분산컴퓨팅, 모바일 컴퓨팅, 이동객체 데이터베이스