

이동 컴퓨팅 환경에서 데이터 특성을 고려한 실시간 혼성 방송 알고리즘

(Real-Time Hybrid Broadcasting Algorithm Considering Data Property in Mobile Computing Environments)

윤혜숙[†] 김영국^{††}
(Hyesook Yoon) (Young-Kuk Kim)

요약 최근 수년 동안, 데이터 방송기술은 많은 수의 클라이언트를 갖는 이동 컴퓨팅 환경에서 매우 효과적인 데이터 전달 시스템으로 주목받고 있다. 특히 혼성 방송은 주기적인 반복데이터 방송과 온디맨드 방송을 통해, 제한된 대역폭 상에서 많은 이동 클라이언트를 수용하면서 동시에 동적인 데이터 액세스를 지원할 수 있다. 그러나 기존의 혼성 방송 알고리즘에서 데이터 액세스 변화를 파악하기 위해 사용하는 기법에는 문제가 있다. 즉, 많은 요청을 받은 데이터 아이템을 반복 방송에 추가시키면 액세스 요구가 적어져 이 데이터 아이템이 방송되고 있기 때문에 요청이 적어진 것인지 아니면 수요가 줄어서 요청이 적어진 것인지 모른다는 것이다. 기존 연구에서는 이 문제 때문에 냉각값을 곱해 서서히 제거하는 방법을 사용하거나 일부러 방송에서 누락시켜 수요를 창출하는 방법을 사용한다. 그러나 이 두 방법은 모두 데이터의 특성을 고려하지 않는 인공적인 기법이다. 본 논문에서 이러한 문제를 해결하기 위해 데이터 유형을 분류하여 방송하는 데이터 방송 알고리즘 RTAHB-DT(Real-Time Adaptive Hybrid Broadcasting based on Data Type)를 제안하고, 시뮬레이션을 사용해 제안 알고리즘의 성능을 분석한다.

키워드 : 데이터 혼성 방송, 실시간 방송, 이동 컴퓨팅 환경

Abstract For recent years, data broadcast technology has been recognized as a very effective data delivery mechanism in mobile computing environment with a large number of clients. Especially, a hybrid broadcast algorithm in real-time environment, which integrates one-way broadcast and on-demand broadcast, has an advantage of adapting the requests of clients to a limited up-link bandwidth and following the change of data access pattern. However, previous hybrid broadcasting algorithms has a problem in the methods to get a grip on the change of data access pattern. It is caused by the diminution of requests for the data items which are contained in periodic broadcasting schedule because they are already broadcasted. To solve this problem, existing researches may remove data items in periodic broadcasting schedule over a few cycles multiplying cooling factor or find out the requests of data items with extracting them on purpose. Both of them are the artificial methods not considering the property of data. In this paper, we propose a real-time adaptive hybrid broadcasting based on data type(RTAHB-DT) to broadcast considering data property and analysis the performance of our algorithm through simulation study.

Key words : data hybrid broadcast, real-time broadcast, dynamic mobile computing,

1. 서론

이동 컴퓨팅은 강력한 이동 컴퓨터와 신뢰성 있는 빠

른 네트워크이라는 두 기술의 뒷받침으로 빠른 속도로 확산되고 있다. 이동 컴퓨팅 환경에서, 사용자는 네트워크에 고정되거나 알려진 위치에 있을 것을 요구받지 않으므로, 많은 이동 사용자가 PCS 폰이나 노트북, PDA 등의 이동 클라이언트를 가지고 다니며 필요하면 즉시 정보를 액세스한다.

이러한 사용자의 요구와 비대칭 통신 형태의 네트워크 환경에 적합한 정보 배달 메커니즘으로 무선 데이터 방송 메커니즘에 대한 연구가 활성화되었다[1,2,3]. 특히

· 본 연구는 정보통신부 및 정보통신연구진흥원의 대학IT연구센터

육성·지원사업의 연구결과를 수행되었음

† 정 회 원 : 특허청 전기전자심사국 사무관

hsyoon@cs.cnu.ac.kr

†† 정 회 원 : 충남대학교 전기정보통신부 교수

ykim@cnu.ac.kr

논문접수 : 2004년 10월 27일

심사완료 : 2005년 3월 15일

혼성 데이터 방송 기법은 이동 사용자가 원하는 데이터가 시간에 따라 계속 변하는 동적 환경에서 많은 사용자가 원하는 정보를 적절하게 선택하여 배달할 수 있다 [4-7]. 그러나 혼성 방송 시스템은 주기적인 푸시(push) 방송과 마찬가지로 사용자의 액세스 요구가 변화하는 것을 파악하기 어렵다는 문제가 있다. 즉, 많은 요청을 받은 데이터 아이템을 반복 방송하면 사용자의 액세스 요구가 적어져 이 데이터 아이템이 방송되고 있기 때문에 요청이 적어진 것인지 아니면 수요가 줄어서 요청이 적어진 것인지 모른다는 것이다. 그러나 대부분 현재 방송중인 데이터는 계속 필요로 하는 데이터일 가능성이 많다. 특히 실시간 혼성 방송에서는 일단 방송 스케줄에 포함되면 마감시간에 따라 방송 빈도가 결정되므로 주기적인 방송에 포함되지만 하면 요청횟수가 적어진다. 따라서 다음 데이터 방송에 이전 주기에 요청된 데이터 아이템만을 고려하여 데이터를 선택한다면 상향 트래픽이 커지는 결과를 초래하기 쉽다. 이러한 문제를 해결하기 위해 기존 혼성 방송 알고리즘에서는 우선순위를 낮추기 위한 냉각값(cooling factor)을 곱해[5,6] 서서히 제거하는 방법을 사용하거나 일부러 방송을 하지 않고 그동안 요청되는 횟수를 측정해 아이템의 실제 필요정도를 측정하기도 한다[6,8]. 그러나 일괄적으로 모든 방송 데이터에 냉각값을 곱하거나 방송하지 않는 것은 데이터의 특성을 고려하지 않는 인공적인 방법으로 냉각값 선택이나 누락 데이터 선정에 따라 시스템 성능이 크게 좌우될 수 있다. 특히 같은 환경, 즉 동적 액세스 이동 컴퓨팅 환경에서 실시간 데이터를 방송하는 TCAHB(Time-Critical Adaptive Hybrid Broadcasting) 알고리즘[5]은 고정된 냉각값을 사용하고 일정 한도 내에서만 대역폭 분할을 결정하는 방식을 사용해 적용될 수 있는 응용이 한정된다.

본 논문에서는 이러한 문제를 해결하기 위해 데이터 유형을 분류하여 방송하는 데이터 방송 알고리즘 RTAHB-DT를 제안한다. RTAHB-DT 알고리즘에서는 데이터를 사용패턴에 따라 두 가지 유형으로 분류하여 냉각값의 적용과 방송데이터 선택을 달리하고, 또한 주기적인 반복 방송과 온디맨드 방송을 위한 대역폭도 데이터 유형의 분포를 고려하여 동적으로 할당할 수 있다. 이러한 데이터 유형을 방송 데이터 집합 (d_1, d_2, d_3, d_4)를 예로 하여 설명하면 다음과 같다. 데이터는 요청 횟수가 높을수록 우선순위가 높은 상태에서 데이터 방송에 포함된다[6]. 아래 데이터 패턴에서 핫(hot)과 콜드(cold)의 구분은 해당 방송주기에 주기적인 방송 스케줄에 포함되면 핫 데이터이고 포함되지 않으면 콜드 데이터로 표기한다.

< 데이터 사용 패턴 >

d_1 : hot→hot→cold→cold→cold→cold→cold→cold→cold
수요가 낮으며 일시적인 요청

d_2 : hot→cold→cold→hot→cold→hot→hot→hot→cold
수요가 낮으며 꾸준히 요청

d_3 : hot→hot→hot→hot→hot→cold→cold→cold→cold
수요가 많으며 일시적인 요청

d_4 : hot→hot→cold→cold→hot→hot→cold→hot→hot→hot
수요가 많으며 꾸준히 요청

데이터 d_1 과 d_3 는 각기 시작한 우선순위가 틀려 방송에 머무르는 사이클 수는 다르나 둘 다 일정 시기 동안 요청되고 더 이상 요청이 들어오지 않는다는 점에서 일치한다. d_1 은 요청횟수가 적으므로 낮은 순위로 시작하여 주기적인 방송에서 빨리 빠져나오고 d_3 는 요청횟수가 많아 높은 순위로 시작했다 서서히 빠져 나온다. 반면에 d_2 과 d_4 는 우선순위는 다르나 지속적으로 요청되는 데이터로 스케줄에서 누락되면 다시 요청 횟수가 늘어나 방송 스케줄에 포함되고 냉각기간을 거쳐 또 빠져나오게 되어 반복적으로 상향 트래픽 수요를 창출하는 역할을 한다. 본 논문에서는 d_1 과 d_3 와 같은 데이터를 동적 데이터로 d_2 과 d_4 와 같은 데이터 정적 데이터로 분류하여 냉각값을 달리 적용하여 사용자 요구에 맞는 데이터를 배포할 수 있다. 한편 요청횟수가 높고 마감시간이 짧은 데이터를 주기적인 반복 방송에 포함시켜 방송하고 상대적으로 요청횟수가 낮고 마감시간이 긴 데이터를 온디맨드 방송에 포함시켜 특정 데이터에 대한 기아(starvation) 상태를 방지할 수 있다. 이것은 매 방송 주기에 결정되기 때문에 주기적인 방송과 온디맨드 방송에 할당해야 할 대역폭 역시 동적으로 변화한다. 본 논문의 RTAHB-DT 알고리즘은 이를 위한 대역폭 분할 방법을 제공한다.

논문의 나머지 구성은 다음과 같다. 2절에서는 이동 컴퓨팅 환경과 관련 연구에 관해 기술한다. 3절에서는 기반시스템인 이동 데이터베이스 모델을 교통정보시스템을 대상으로 살펴보기로 한다. 4절에서 실시간 혼성 스케줄링 기법 RTAHB-DT의 설계를 설명하고 5절에서 성능 평가와 결과 분석을 수행한다. 6절에서 결론을 짓기로 한다.

2. 관련 연구

이 절에서는 먼저 본 논문의 전반적인 환경인 혼성 데이터 방송 모델을 설명하고 이에 대한 이전 연구에 대해 살펴보기로 한다.

2.1 혼성 데이터 방송 모델

이동 컴퓨팅 환경에서 데이터를 배포하기 위한 방송 기법은 다음의 3가지 모델로 나눌 수 있다. 즉, 푸시 기반(push-based)의 단 방향 방송, 풀 기반(pull-based)

의 양 방향 방송과 푸시와 풀을 통합한 혼성(hybrid) 방송이다[6]. 방송전용 모델인 푸시 기반의 단 방향 기법에서, 서버는 특별히 요청하지 않아도 주기적으로 데이터를 클라이언트로 방송한다[1,3,9]. 방송을 위한 하향 대역폭은 한정되어 있으므로 가능한 많은 사용자가 요구하는 데이터를 선별하여 이들 데이터를 적절한 빈도로 방송하기 위해서는 클라이언트와의 상호작용이 중요하다. 최근 수 년 동안 제안된 풀 기반의 방송 알고리즘이나 혼성 방송 알고리즘은 이러한 점을 극복하기 위한 것이다[5-7,9,10].

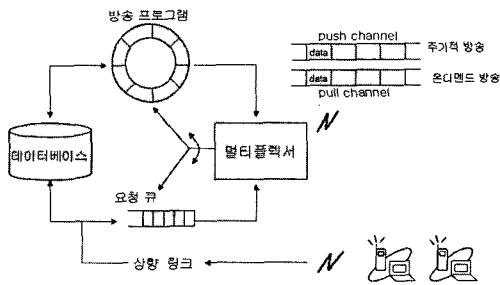


그림 1 혼성 데이터 방송 시나리오

그림 1은 혼성 방송 모델의 시나리오를 보여준다. 이동 클라이언트는 필요한 데이터가 방송되지 않으면 상황 링크를 통해 요청한다. 요청된 데이터는 요청 큐에 저장되고 요청횟수와 마감시간을 고려하여 푸시 기반의 주기적 방송과 풀 기반의 온디맨드 방송으로 나누어 방송한다.

2.2 관련 연구

데이터 혼성 방송 스케줄링 기법은 푸시기법과 풀 기법의 트레이드오프(trade-off)의 타협점으로 연구되었다[4]. [11]에서 소개된 HBoD 모델은 기존의 방송디스크 모델을 확장한 것으로 두 개의 분리된 채널, 서버에서 사용자로의 방송을 위한 고용량 채널과 사용자가 서버로 요청을 전달할 수 있는 저용량의 채널을 통합하고 시분할 멀티플렉싱을 이용하여 주기적인 방송을 할 때 사용가능한 대역폭의 고정된 비율을 사용하고, 나머지 대역폭은 온디맨드 처리에 사용한다.

확장된 방송디스크(IPP)는 풀 기반과 푸시 기반 방송 디스크 기법의 통합 모델이다[4]. 이 연구는 두개의 독립적인 데이터 채널, 백채널(back channel)과 후론트 채널(front channel)을 가정한다. 후론트 채널은 푸시 기반 연산에 사용되고 백 채널은 풀 기반 연산을 위한 매체로 동작한다. 사용가능한 대역폭은 두 채널에서 공유한다. 서버에 거의 경쟁이 없는 경우나 서버 과부하의 두 극단적인 경우에 대해 순수한 풀과 순수한 푸시 방

안이 각각 최적 옵션이라고 결론짓는다. 그렇지 않은 경우에 대해 IPP(Interleaved Push and Pull) 혼성 방송을 제안한다.

적용형 혼성 체계는 각 데이터 배달 모드의 특성을 개발하고 클라이언트 요구에 더 잘 부합되는 형태로 통합시키는 기법을 사용한다[5]. 혼성 시스템의 반응시간 기대치 T 는 T_{pull} 와 T_{push} 의 비중 평균이다. 적용형 혼성 체계는 최적 성능을 위해 데이터 푸시와 풀 간의 적절한 균형을 유지하도록 이 비중 평균이 작아지는 지역에서 해답을 찾는다. 고정된 스케줄을 가정한 일반적인 주기적 방송 체계와는 달리, 이 적용형 기법에서 방송의 내용과 크기는 작업 부하에 더 잘 맞추기 위해 끊임없이 갱신된다.

TCAHB 모델은 위에서 소개한 HBoD 모델과 유사하나 사용자 프로필, 즉 클라이언트의 데이터 액세스 분포 형태가 동적이라고 가정하고 액세스 분포가 변화에 따라 주기적인 방송을 위해 데이터를 선택할 때 이전 방송 주기 동안 요청된 횟수와 대역폭 크기와 데이터 요청과 관련된 마감시간 제한조건을 고려한다[6]. 실시간 데이터의 배포를 위해 온디맨드 방송은 EDF를 바탕으로 스케줄되며, 주기적인 방송은 데이터 아이템에 대한 잠재적인 요청을 만족시키는데 필요한 최소의 빈도를 갖는 요소를 방송한다.

SGID 메커니즘은 대부분의 정보 방송 서비스가 사실상 데이터를 동적으로 생성시킨다는 점을 고려하여 데이터 방송 시나리오를 지원할 뿐 아니라 동적 정보서비스도 배포하는 확장된 혼성 데이터 방송이다[7].

[5-7]에서 소개한 기존 혼성 방송 알고리즘은 모두 방송 주기 별로 사용자가 요구하는 액세스 패턴에 따라 데이터를 선택하여 방송한다. 그러나 이들 시스템에서 방송 데이터의 선택 시 우선순위는 이전 주기 요청 횟수와 데이터의 크기 등으로 전체적인 방송 데이터 액세스 패턴은 고려하지 않으며 요청 횟수의 내용도 신뢰하기 어렵다는 단점이 있다.

3. 실시간 혼성 방송 모델 기반 시스템

3.1 절에서는 본 논문에서 제안하는 실시간 혼성 방송 알고리즘이 수행되는 이동 컴퓨팅 모델과 용어를 설명한다. 3.2절에서는 이러한 환경에 적합한 응용 예로 교통정보 시스템을 설명한다.

3.1 이동 시스템 모델 및 용어

이동 컴퓨팅 환경에 사용되는 공통 구조 모델이 그림 2에 나타나 있다[12]. 이 모델에서, 지리상의 영역은 셀이라고 부르는 구역으로 나뉘어져 있다. 각 셀은 기지국 제어기에 의해 커버되고 서비스된다. 이동 클라이언트(Mobile Unit :MU)와 기지국 컴퓨터(Stationary Computer

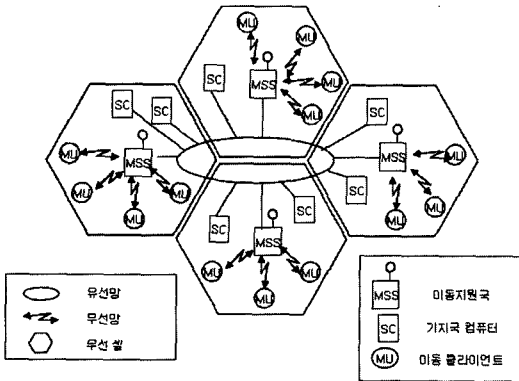


그림 2 이동 컴퓨팅 모델

:SC)의 두 형태의 컴퓨터가 존재한다. SC는 고정망에 연결되어 있으며, 몇몇 SC는 MU와 통신하기 위한 무선 인터페이스가 장착되어 있어 무선기지국 혹은 이동 지원국(Mobile Support Station : MSS)이라고 부른다. 이동 환경에서는 데이터 방송에 할당된 하나의 방송 채널이 있다고 가정한다. 사용자는 그들이 원하는 데이터 아이템을 얻기 위해 이 채널을 항상 감시한다. 한편 MU가 MSS로 정보 요청을 보낼 수 있는 백 채널이 있으며, 그 대역폭은 방송대역폭에 비해 상당히 작다.

이러한 이동 구조를 바탕으로 하는 이동 컴퓨팅 시스템 모델은 방송 및 데이터베이스 서버, 이동 클라이언트 집단과 유선 네트워크를 포함한 이동 네트워크로 구성된다[4]. 서버에서 클라이언트로의 프론트 채널(front channel)은 고대역폭의 무선 네트워크를 통해 데이터가 배포되고 클라이언트에서 서버로의 백채널(back channel)은 저용량의 유무선 네트워크로 연결된다. 서버는 이동 클라이언트에 방송할 데이터베이스를 유지하고 동적으로 변하는 정보를 연계된 외부의 데이터 공급자로부터 공급받으며, 이동 네트워크의 무선 채널을 통해 데이터를 배포한다. 또한 서버는 공급자로부터 받은 데이터를 데이터베이스에 반영하기 위해 갱신 트랜잭션을 수행한다. 어떤 변경이든 서버에서 일어나고 배포되며 배포된 데이터는 읽기 전용이라고 가정한다. 서버는 데이터를 주기적으로 방송하는데 인덱스 데이터를 포함하여 주기적인 데이터와 온디맨드 데이터가 차례로 방송되는 기간을 방송주기(broadcast cycle)라 하고 한 주기 동안 방송되는 데이터의 논리적 단위를 bcast라고 한다[9]. 방송 주기의 길이는 가변적이고 하나의 방송 주기는 중간에 빈 시간 없이 즉시 다음 주기로 바뀐다. 방송되는 데이터 아이템의 값은 방송 주기 초반의 데이터베이스 상태 즉, 사이클 초기까지 커밋된 트랜잭션에 의해 생성된 값과 일치한다.

클라이언트는 가능한 배터리 소모를 줄여야 하기 때

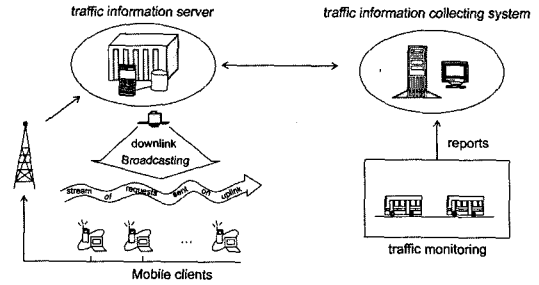


그림 3 교통정보시스템

문에 원하는 특정 아이템만을 읽도록 하기 위해 두 가지 모드를 갖는다. 즉, 방송에 귀 기울이는 조정(tune) 모드와 응답을 기다리고 있는 대기(idle) 상태의 응답 모드이다. 이를 위해 방송되는 데이터에는 클라이언트가 채널에 관심 있는 데이터가 나타났는지 알 수 있도록 인덱스 정보가 제공된다. 이동 클라이언트는 정보를 읽는 트랜잭션을 발생시킨다. 각 트랜잭션은 간단한 데이터 요청 집합이며, 일정 순서로 수행된다고 가정한다. 이동 클라이언트가 제기한 트랜잭션에서 요청하는 데이터 아이템이 캐시에 존재하면 즉시 데이터 아이템을 얻는 것이며, 그렇지 않으면 데이터 요청은 데이터 아이템이 방송되기를 기다려서 공중에서 얻어야 한다.

3.2 교통정보시스템

이동 컴퓨팅 시스템 모델의 예로 교통정보시스템을 그림 3에서 보여준다. 교통정보시스템은 실시간 교통정보를 다수의 이동 클라이언트 혹은 고정 클라이언트에 제공해주며 이를 위해 다양한 연계 교통정보 수집시스템에서 정보를 수집한다. 시스템의 구성요소는 방송 및 데이터베이스 서버, 수개의 정보수집장치, 다수의 이동 클라이언트 혹은 고정 클라이언트와 유무선 네트워크로 구성된다. 방송서버는 비대칭 통신 채널을 통해 이동 클라이언트와 통신한다. 방송서버는 이동 클라이언트에 방송할 데이터베이스를 유지하고 실시간으로 변하는 교통정보를 교통방송이나 교통정보 센터와 같이 연계된 외부의 데이터 공급자로부터 공급받는다. 매 방송 주기마다 교통정보 서버는 주기적인 데이터 아이템과 온디맨드 데이터 아이템을 방송한다. 이동 클라이언트는 서버로부터 데이터 아이템을 찾는 트랜잭션을 생성하며, 각 트랜잭션은 몇 개의 데이터 아이템을 요구한다. 예를 들어, 특정 시간 내에 어떤 장소에 도착하려는 운전자는 PDA나 휴대폰을 사용하여 현재의 교통 흐름 속도나 교통사고 발생과 같은 도로 교통 상황을 체크하거나 실시간 교통상황을 감안한 최단거리를 찾으려할 것이다. 이동 클라이언트는 캐시나 공중에서 필요한 데이터를 검색해 제공한다. 만일 캐시나 공중에서 액세스할 수 없는 경우 상향 링크를 통해 데이터를 요청한다.

4. 실시간 혼성 방송 알고리즘

이 절에서는 본 논문에서 제안하는 실시간 혼성 알고리즘 RTAHB-DT(Real-Time Adaptive Hybrid Broadcasting Algorithm based on Data Type)을 소개한다. 4.1절에서 전체적인 알고리즘 순서와 데이터 요청 횟수와 방송 지속성 특성에 의해 데이터 아이템을 분류하는 방법을 설명한다. 4.2절에서는 이들 아이템을 방송하기 위해 대역폭을 분할하는 기법을 설명하고, 4.3절과 4.4절에서 차례로 주기적인 방송 데이터의 선택과 온디맨드 방송 데이터의 선택을 위한 알고리즘을 설명한다.

4.1 RTAHB-DT 개요

그림 4는 RTAHB-DT에 의해 방송 주기 당 수행되는 기능을 흐름도 형식으로 정리한 것이다. 방송 주기는 선택된 주기적인 방송 데이터와 온디맨드 방송 데이터가 모두 방송되는 기간으로 클라이언트는 이 기간 동안 필요한 데이터를 요청할 수 있다. 서버는 방송 주기 말미에 다음과 같은 기능을 수행하게 된다.

- 이전 방송 주기에 선택된 모든 데이터 아이템에 대해 방송에 포함된 내력을 알려주는 지표인 history factor 값을 계산하고, 데이터를 변환한다.
- 요청된 데이터 분포에 따라 대역폭을 분할한다.
- 주기적인 방송 데이터를 선택한다.
- 온디맨드 방송 데이터를 선택한다.

데이터 아이템을 방송에 포함시키는 일차적인 기준은 클라이언트로부터의 요청이며, 포함된 데이터 아이템은 일정 주기가 지나면 방송에서 제외된다. 만일 제외된 아이템에 대한 수요가 여전히 있으면 다시 요청이 늘어나 이 아이템은 방송에 포함될 것이다. 이것은 데이터 아이템에 따라 다른 양상으로 나타난다. 어떤 아이템은 일시적인 요청을 받으나 다른 아이템은 지속적인 요청을 받을 수 있다. 다음은 이와 같은 관점에서 데이터를 분류한 것이다.

- 정적 데이터(static data) : 클라이언트에서 오랜 기간

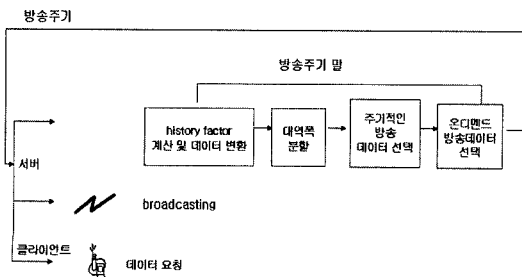


그림 4 RTAHB-DT 알고리즘 흐름

필요로 하는 데이터로 history factor h가 일정 임계값 이상이 되면 동적데이터로 변환된다.

- 동적 데이터(dynamic data) : 클라이언트에서 일정 시기에만 필요로 하는 데이터로 모든 데이터의 초기값은 동적 데이터로 시작한다. history factor h가 일정 임계값 미만으로 떨어지면 정적 데이터로 변환된다.

어떤 데이터 아이템이 방송에 포함되었던 내력을 알려주는 history factor h는 이전 방송 주기 동안 연속 방송된 간격이 짧아질수록 커지는 값으로 다음과 같이 정의한다.

- 방송 프로그램에 추가되면 h_i++
- 일정 주기(=d) 동안 방송에 포함되지 않으면 $h_i = 0$
- $h_i \geq h_{cutoff}$ 이면, 동적 데이터로 아니면 정적 데이터로 변환된다. 임계값은 사이클 1부터 시작할 수 있으며 시뮬레이션을 통해 적절한 값에 대해 알아본다. 그림 5는 이러한 데이터 변환 관계를 보여준다.

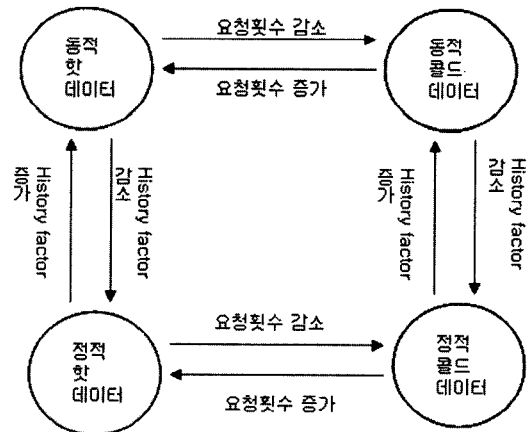


그림 5 데이터 분류 간 변환관계

4.2 대역폭 분할

서버는 각 방송 주기 마지막에 폴 채널과 푸시 채널에 할당할 방송 대역폭을 갱신한다. 우선 주기적인 방송에 필요한 푸시 채널 용량을 계산하여 할당하고 나머지를 온디맨드 방송을 위해 폴 채널에 할당한다. 주기적인 방송에 대한 대역폭 할당 비율은 방송할 데이터 아이템 $f_1, f_2, f_3, \dots, f_n$ 이 주어졌을 때, 이들 아이템을 마감시간 전에 방송할 수 있는 최소의 대역폭 B_p 를 결정하는 문제로 귀결될 수 있다. 즉, 식 (1)과 같이 이러한 방송 데이터 아이템의 집합을 Φ 라 하고, 각 데이터 아이템의 파일 크기가 s_i 이고 마감시간이 d_i 라고 하면 스케줄될 수 있는 최소의 B_p 값을 결정하는 문제가 된다.

$$\Phi = \{(s_1 \cdot B_p \cdot d_1), (s_2 \cdot B_p \cdot d_2), \dots, (m_n \cdot B_p \cdot d_n)\} \quad (1)$$

[13]에 정의된 바와 같이 주기적인 태스크 시스템 Φ 이 스케줄이 가능하려면 모든 태스크 밀도의 합이 1 이하가 필요충분조건이 된다. 따라서 B_p 은 식 (2)와 같이 임계값을 계산할 수 있다.

$$\sum_{i \in \Phi} \frac{s_i}{B_p \cdot d_i} \leq 1 \quad B_p \geq \sum_{i \in \Phi} \frac{s_i}{d_i} \quad (2)$$

즉, 최소한 필요한 대역폭 량은 방송할 데이터의 밀도의 합에 의존한다. 사용가능한 대역폭 량은 정해져 있으므로 모든 데이터의 밀도의 합이 이 양을 초과하면 스케줄이 불가능해지므로 방송할 데이터를 주어진 대역폭 한도 내에서 선택해야 한다. 주어진 대역폭은 온디맨드 방송에도 할당되어야 하므로 대역폭을 적절히 분할해야 할 것이다. 대역폭을 분할하기 위해 매번 별도로 계산하기 보다는 이전 주기 동안 방송된 데이터와 새롭게 요청된 데이터를 고려하여 결정하는 것이 효율적이다. 이를 위해 총 대역폭 B 를 n 개의 대역폭으로 세분화한다. 즉 세분화된 대역폭 b 는 B/n 로 계산된다. 현재 대역폭이 $B_{p,old}$ 이고 대역폭 증가분인 a 로 계산하면 다음 주기의 대역폭 B_p 는 식 (3)과 같이 정의할 수 있다. 식(4)에서 a 는 주기적인 방송 데이터에서 냉각값에 의해 줄어든 데이터 때문에 남는 대역폭과 새로 큐에 들어온 데이터에 필요한 대역폭 간의 차이를 나타낸다. 식 (5)는 온디맨드 방송을 위한 대역폭의 계산식이다.

$$B_p = B_{p,old} + a/b \cdot b \quad (3)$$

$$a = -B_{p,old}(s \cdot cf_s + (1-s) \cdot cf_d) + (B - B_{p,old}) \cdot q_{cutoff} \quad (4)$$

q_{cutoff} : number of data items requested

more than the cutoff threshold

s : static data rate contained \in periodic broadcasting data

$$0 \leq s, cf_s, d, cf_d \leq 1$$

$$B_o = B - B_p, \text{ on-demand broadcast bandwidth} \quad (5)$$

4.3 주기적인 방송 데이터 선택

주기적인 방송 모드에서는 pfair 알고리즘[13]을 사용하여 데이터를 마감시간 내에 반복 방송하는 방법으로 실시간 요구사항을 만족시킨다. 주기적인 방송 모드에서는 대역폭 효율과 액세스 시간을 개선하기 위해 마감시간이 짧고 크기가 작은 데이터를 선택하는 것이 유리하다. 우선순위 계산에 이러한 점을 반영하기 위해 정적 데이터와 동적 데이터에 대해 각각 다음과 같이 연산한다. 다음 주기(i -번째)의 우선순위는 이전 주기를 기준으로 한다.

$$\cdot \text{priority}_i = \text{priority}_{i-1} \quad (6)$$

요청횟수가 임계값을 넘은 주기적인 방송에 포함된

데이터의 우선순위

$$\cdot \text{priority}_i = \text{priority}_{i-1} \cdot cf_s \quad (7)$$

요청횟수가 임계값을 넘지 않는 주기적인 방송에 포함된 정적데이터의 우선순위

$$\cdot \text{priority}_i = \text{priority}_{i-1} \cdot cf_d \quad (8)$$

요청횟수가 임계값을 넘지 않는 주기적인 방송에 포함된 동적데이터의 우선순위

$$\cdot \text{priority}_i = \frac{\text{Number of Requests}}{\text{density}} \cdot cf_s \quad (9)$$

$$\text{density} = \frac{s}{d}$$

큐에 새로 요청된 정적 데이터의 우선순위

$$\cdot \text{priority}_i = \frac{\text{Number of Requests}}{\text{density}} \cdot cf_d \quad (10)$$

큐에 새로 요청된 동적 데이터의 우선순위

이미 방송 중인 데이터는 마감시간에 따라 반복 방송하므로 우선순위 큐에 요청이 많지 않으나 임계값 이상의 요청이 들어오면 식(6)과 같이 우선순위를 그대로 유지한다. 식(7)과 (8)은 이미 방송 중인 데이터에 한해 동적 냉각값을 곱해 새로운 우선순위로 반영한다. 초기값은 동적 데이터에서 시작하므로 초기에는 대부분의 데이터의 방송 데이터로의 진출입이 느리지만, 일단 정적 데이터로 분류되면 동적 데이터에 비해 더 빠르게 진출입할 수 있다. 식(9)와 (10)은 큐에 새로 요청되는 데이터 아이템의 우선순위를 계산하는 식이다. 식 (9)와 (10)에서 우선순위 초기값에 냉각값을 곱하는 이유는 동적데이터와 정적데이터의 진출입을 데이터 유형에 따라 다르게 하기 위해서이다.

4.4 온디맨드 방송 데이터의 선택

주기적인 방송 데이터를 선택하고 스케줄하는 작업을 끝내면 온디맨드 방송을 위한 대역폭에 할당할 데이터를 선정한다. 온디맨드 방송에 포함할 데이터는 이전 주기에 큐에 요청된 데이터 중 주기적 방송에 포함되지 않은 데이터 위주로 한다. 클라이언트가 서버로 데이터를 요청했을 때 주기적인 방송에서 데이터를 발견하지 못하면 계속 기다리다 결국 마감시간이 지나 나중에 도착해도 이미 쓸모없는 경우를 예상할 수 있다. 따라서 온디맨드 방송은 요청 빈도수는 적지만 마감시간이 중요한 데이터를 방송하는데 사용한다.

이러한 온디맨드 방송 스케줄링에는 데이터를 요청한 시점에서 데이터의 특성과 마감시간을 고려하는 EDF(Earliest Deadline First) 방식이 사용된다. 마감시간은 요청 시점부터의 상대적인 시간으로 식 (11)과 같이 계산하며, 요청 큐에 남아있는 데이터를 위와 같이 각 아이템의 마감시간을 키로 하여 정렬하고 아이템을 선택한다.

$$\text{deadline}_x = \min_x [d_x - \text{currentTime} - \min(\text{requestTime}_x)] \quad (11)$$

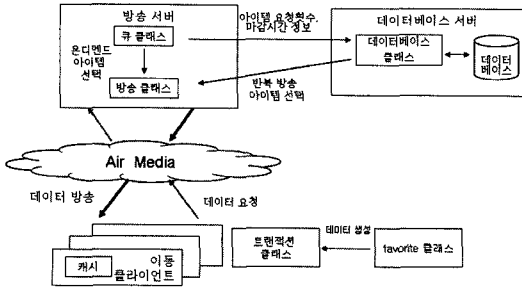


그림 6 성능 평가 모델

d_x : 데이터아이템 i 의 마감시간

$\min(requestTime_x)$: 이전 주기 동안 데이터 아이템 x 에 대한 가장 빠른 요청 시간

5. 성능분석 및 평가

성능 평가에서는 다양한 워크로드 하에서 동적인 데이터의 비율이 증가함에 따라 냉각값과 history factor의 임계값의 최적 값인지 테스트하였다. 또 특정 냉각값에서 TCAHB 알고리즘과 비교해보기로 한다. TCAHB 알고리즘은 같은 동적 액세스 이동 컴퓨팅 환경에서 실시간 데이터를 혼성 방송하나, 고정된 냉각값을 사용하고 일정 한도 내에서만 대역폭 분할을 결정하므로 제안 기법에 비해 비탄력적임을 실험을 통해 비교한다. 실험은 Intel PentiumIV 1.4GHz CPU를 탑재한 윈도우 환경에서 C++를 사용해 수행하였다.

5.1 성능 평가 모델

그림 6의 성능 평가 모델은 본 시뮬레이션의 바탕이 된다. 크게 방송서버, 데이터베이스 서버, 에어 미디어와 이동 클라이언트의 4개 엔티티와 큐 클래스, 방송 클래스, 데이터베이스 클래스, 트랜잭션 클래스와 favorite 클래스의 5개 클래스로 구성된다. 방송 서버에는 큐 클래스와 방송 클래스의 2개의 클래스가 있다. 큐 클래스는 상향 링크 채널 용량을 고려하여 클라이언트로부터 요청받은 데이터 아이템을 저장하기 위한 우선순위 큐를 가지며, 방송 클래스와 데이터베이스 클래스에게 가장 짧은 마감시간을 갖는 데이터 아이템과 요청 횟수를 알려준다. 방송 클래스에서는 데이터베이스와 우선순위 큐를 참조하여 주기적인 스케줄과 온디맨드 스케줄을 결정한다. 데이터베이스 클래스에서는 매 방송 주기마다 데이터 유형과 우선순위를 갱신한다. 이동 클라이언트의 트랜잭션 클래스는 트랜잭션을 제기한다. 트랜잭션은 필요한 데이터 아이템 번호와 마감시간으로 함께 정의한다. 이때 필요한 데이터는 favorite 클래스에서 제공한다. 동적 데이터 비율을 제외하고 나머지 데이터는 zipf 분포를 가지며 이 분포로 제공한다. 트랜잭션은 마감시

표 1 성능 파라미터와 기본값

성능 파라미터	기본 값	
데이터베이스 크기	1000 items	
데이터 아이템 크기	10 pages	
데이터 아이템의 상대적 마감시간	200, 500, 700 단위시간	
시뮬레이션 시간	1000000 단위시간	
partitioned bandwidth	0.05~0.95	
Cooling factor	cf_s	0.5~0.9
	cf_a	0.3~0.9
상향 링크 용량	단위시간 당 2번 요청	
워크 로드(workload)	1~5	
cutoff_threshold	h_{cutoff}	12~100 cycles
	q_{cutoff}	cycle time / deadline
동적 데이터 비율	10~60%	

간이 지나서 완성하게 되면 유용성이 현저히 떨어져 쓸모없게 된다고 가정한다. 따라서 트랜잭션이 마감시간을 넘기게 되면 중단한다.

5.2 성능 평가 파라미터

본 시뮬레이터의 시간 단위는 하향 링크 채널 용량을 표현하기 위해 한 페이지를 방송하는데 걸리는 시간으로 정의한다. 표 1에 주요 파라미터와 기본값이 나타나 있다. 주요 성능측정치로 deadline miss ratio와 두 번째 성능 측정치는 충족된 요청의 평균 액세스 시간(access latency)로 한다. 대역폭 분할은 전체 대역폭을 20으로 나누어 수행하고 실험에서 반복방송의 비율이 최소 0.95까지 허용하였다. partitioned bandwidth는 이를 나타낸다. 또 워크로드는 단위시간 당 제기된 트랜잭션의 수를 나타낸다. 대부분의 방송 시스템에서 사용자가 요청하는 데이터는 지프(zipf) 분포를 따른다.

5.3 성능 실험 결과 및 분석

5.3.1 다양한 워크로드(workload)에 따른 성능 비교

그림 7의 두 그래프는 동적 데이터 비율이 증가할 때 TCAHB와 RTAHB-DT의 Access Latency와 Deadline Miss Ratio를 다양한 워크로드 상에서 보여준다. 두 스케줄링 알고리즘 모두 동적 데이터 비율이 증가할수록 또 워크로드가 커질수록 성능이 저하된다. 이러한 결과는 동적 데이터 비율이 증가하면 요청되는 데이터의 종류가 많아지므로 반복 방송의 비중이 작아지는 것이 유리하며, 워크로드는 처리해야 할 용량이 전체적으로 증가하기 때문이다. 이러한 면에서 같은 워크로드에서 RTAHB-DT가 TCAHB 보다 더 좋은 성능을 보인다고 분석할 수 있다. 한편 RTAHB-DT는 워크로드가 1인 경우 가장 좋은 성능을 보이고 워크로드가 3, 5로 증가하면 성능 개선이 떨어지는 이유는 데이터 분류에 오버헤드가 발생하고 워크로드의 증가에 따라 더 다양한 데이터가 요구되기 때문에 적응도가 떨어지기 때문인 것으로 분석된다.

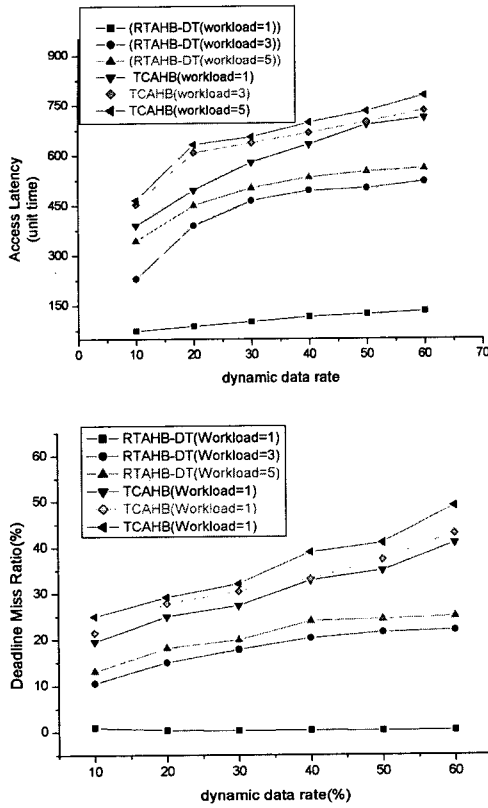


그림 7 다양한 워크로드 하에서 TCAHB와 RTAHB-DT 성능 비교

5.3.2 냉각값 차이에 따른 성능 비교

그림 8은 워크로드가 1일 때 정적 데이터의 냉각값(CFS)과 동적데이터의 냉각값(CFD)의 차이가 각각 0.2(0.5-0.3, 0.7-0.5, 0.9-0.7), 0.4(0.7-0.3, 0.9-0.5), 0.6(0.9-0.3)일 때의 성능을 측정 한 것이다. 이 그림에서 일정하게 가장 좋은 성능을 보이는 경우는 CFS-CFD가 0.9-0.3과 0.7-0.3인 경우이다. 그림 9와 그림 10은 워크로드가 3과 5일 때의 성능이다. 여기서는 CFS-CFD가 0.9-0.3과 0.7-0.3인 경우 동적 데이터 비율이 낮을 때 좋은 성능을 보이나 동적 데이터 비율이 커질수록 CFD 값이 작을수록 유리한 것으로 나타난다. 이것은 데이터 요구량이 커지고 동적 데이터 부분이 많아질수록 주기적인 데이터에서 빨리 순환시키는 것이 유리하기 때문인 것으로 분석된다. 또 워크로드가 높아지면 냉각값이 작을수록 deadline miss ratio의 성능이 주로 개선되고 access latency는 비슷하다. 이것은 데이터 교체 속도가 빨라져 데이터 종류가 다양해지기 때문이다.

5.3.3 History Factor 임계값(HCT)에 따른 성능 비교

그림 11, 12, 13은 각각 워크로드가 1, 3, 5일 때

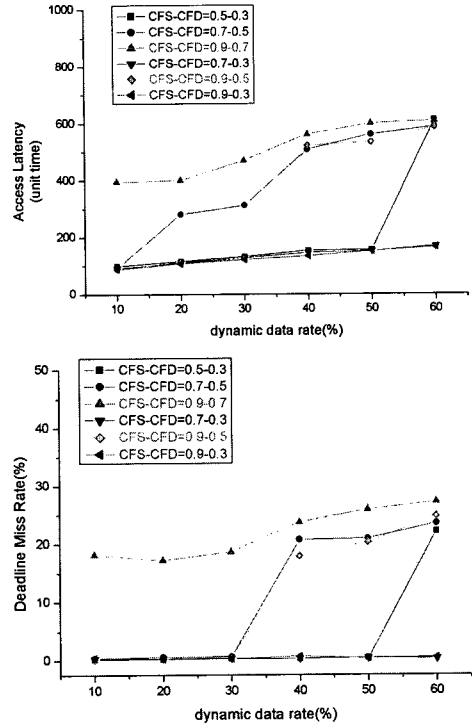


그림 8 workload = 1일 때 냉각값 차이에 따른 성능

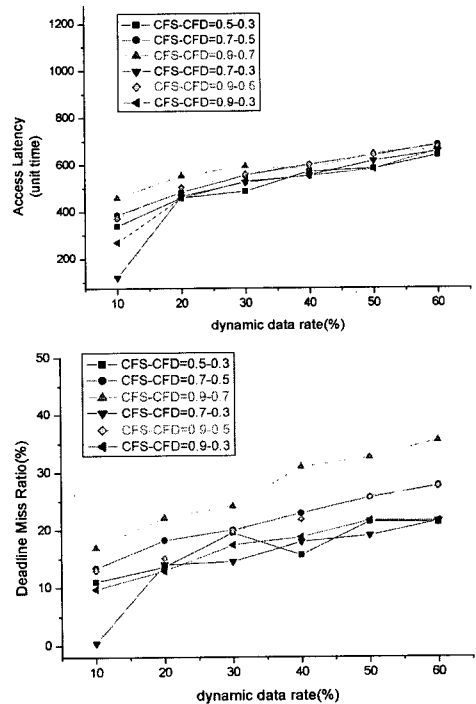


그림 9 workload = 3일 때 냉각값 차이에 따른 성능

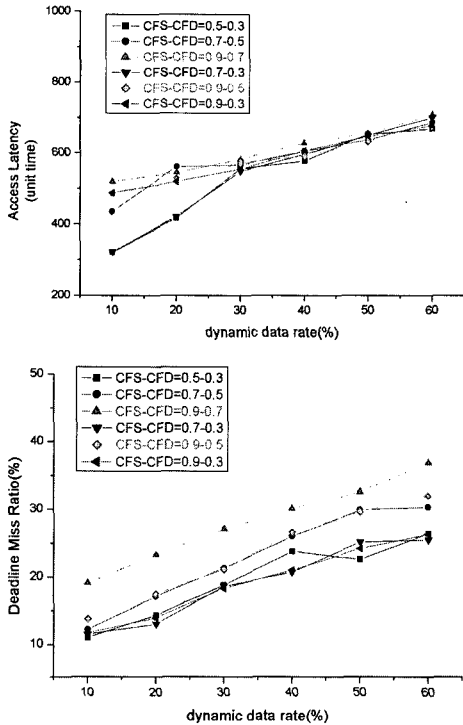


그림 10 workload = 5일 때 냉각값 차이에 따른 성능

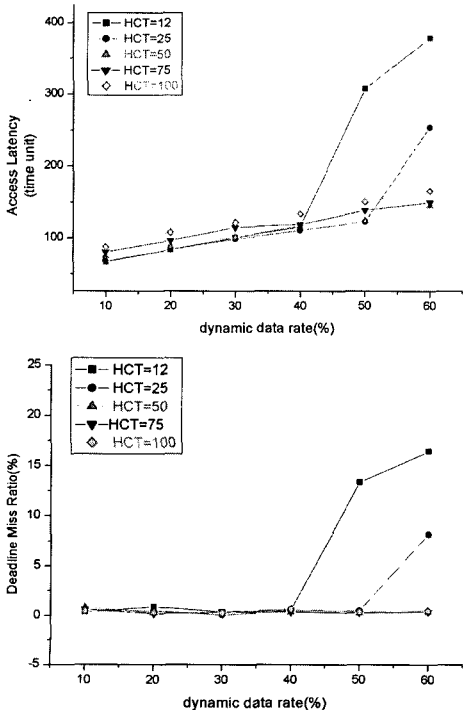


그림 11 workload = 1일 때 HCT 차이에 따른 성능

history factor의 임계값을 변화시키면서 성능을 측정하는 것이다. 그림 11에서 보면 HCT 25 이하일 때 동적 데이터 비율이 상대적으로 커지면 성능이 급격히 저하되고 50 이상에서는 큰 차이가 없다. 이것은 워크로드가 크지 않을 때 동적 데이터가 비율이 높아질수록 정적 데이터로의 변환을 단기간에 하면 그 영향이 크다는 것을 의미한다. 그림 12와 13에서는 HCT 값에 따른 성능 차이는 크지 않으나 access latency는 HCT가 작을수록 좋고 deadline miss ratio는 HCT가 클수록 성능이 좋은 것으로 나타난다. 이것은 정적 데이터로의 변환이 빠를수록 반복 데이터의 비중이 커져 요구량이 많은 데이터의 배달이 빨라져 평균 액세스 시간이 좋아지는 반면에 데이터 종류는 상대적으로 적어져 트랜잭션이 중단되는 경우가 많기 때문으로 분석된다.

5.3.4 History Factor의 임계값에 따른 성능 비교

그림 14와 15는 워크로드가 1과 3일 때 HF를 리셋시키는 임계값이 변할 때 성능이 어떻게 변하는지 보여준다. 두 그림 모두에서 임계값이 1이나 2일 때가 성능이 낮게 나타나고 5이상일 때는 임계값 변화에 따른 성능 차이가 미미하다. 이것은 정적 데이터로 분류된 데이터를 한두 번 방송에서 벗어난다고 해서 다시 동적 데이터로 전환하면 다시 정적 데이터로 분류하는데 오버헤드가 소모되기 때문인 것으로 분석된다. 그러나 5이상인 경우 정적 데이터에서 벗어난 것으로 판단해도 별 무리가 없는 것으로 보인다.

6. 결론

휴대폰이나 팜탑과 같은 무선기기를 통한 이동 컴퓨팅 환경이 대중화되면서 데이터베이스 서버에서 많은 무선기기에 내재한 클라이언트에서 요구하는 데이터를 마감시간 내에 전달하는 것이 중요한 이슈로 자리 잡았으며, 혼성 방송 기법은 대규모 클라이언트 집단을 수용하면서 사용자의 데이터 액세스 요구가 동적인 환경에 적합하다는 사실이 여러 연구에서 밝혀졌다.

본 논문에서는 동적인 환경에 더 잘 적용할 수 있도록 데이터의 특성을 고려하여 우선순위를 부여하고 재계산하는 혼성방송 알고리즘인 RTAHB-DT를 제안하였다. 본 알고리즘을 대상으로 시뮬레이션 한 결과 전반적으로 기존 혼성 방송 알고리즘 TCAHB보다 좋은 성능을 보이고 특히 동적 데이터 비율이 증가해도 TCAHB에 비해 성능 저하가 심하지 않았다. 이것은 동적 데이터 비율이 높아질수록 온디맨드 방송을 위한 대역폭을 늘리기 때문인 것으로 판단된다. 또한 동적 데이터 비율이 커질수록 동적데이터에 대한 냉각값을 낮추고 정적 데이터에 대한 냉각값과의 차이를 많이 두면 성능이 향상되었다.

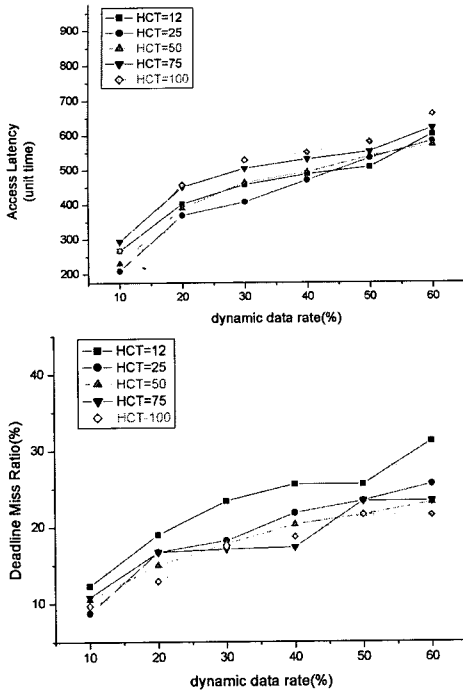


그림 12 workload = 3일 때 HCT 차이에 따른 성능

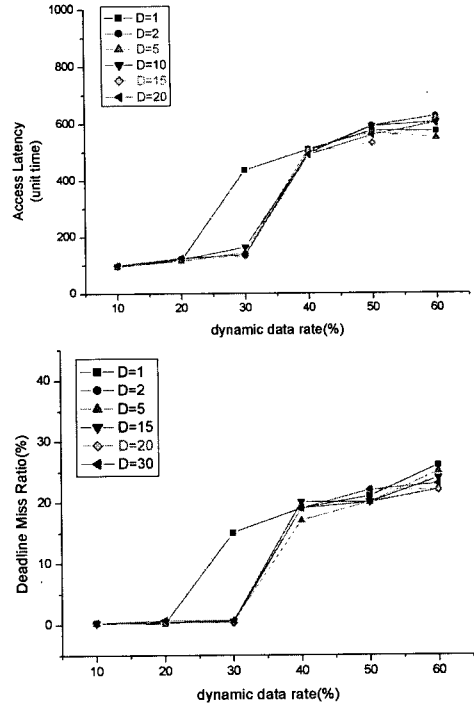


그림 14 workload = 1일 때 History Factor의 임계값에 따른 성능

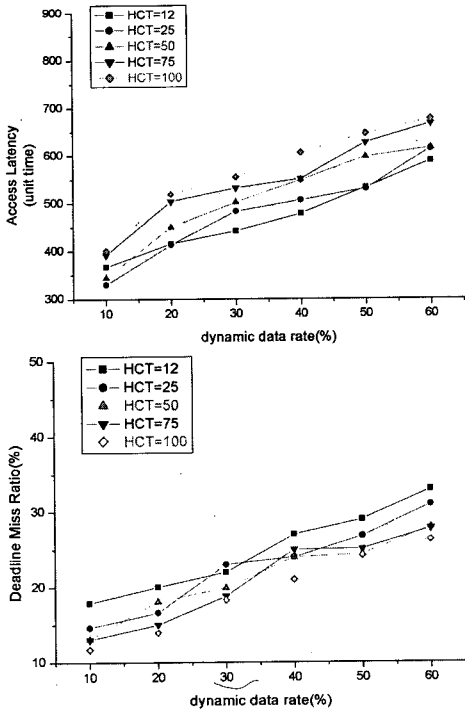


그림 13 workload = 5일 때 HCT 차이에 따른 성능

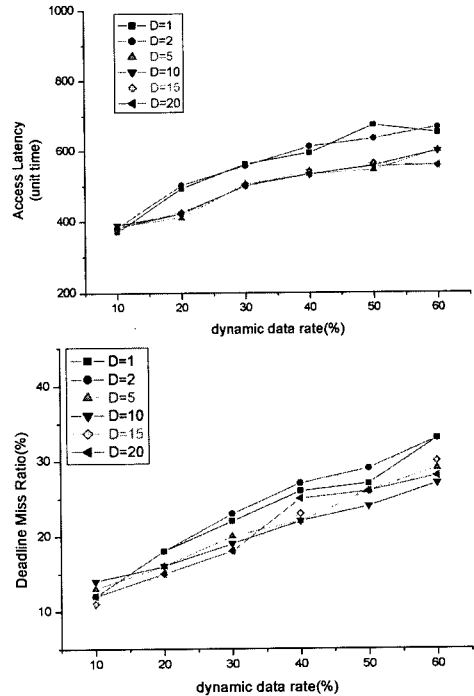


그림 15 workload = 3일 때 History Factor의 임계값에 따른 성능

참 고 문 헌

- [1] S. Acharya, R. Alonso, M. Franklin and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments" in Proceedings of ACM SIGMOD, pp. 199-210, 1995.
- [2] D. Babrbara, "Mobile Computing and Database - A Survey," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, pp. 108-117, January-February, 1999.
- [3] T. Imielinski, et al. "Data on Air: Organization and Access," IEEE TKDE, 9(3): pp. 353-372, 1997.
- [4] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Data Broadcast" in Proceedings of ACM SIGMOD, Tucson, Arizona, pp. 183-194, May 1997.
- [5] J. Fernandez and K. Ramamritham, "Adaptive Dissemination of Data in Time-Critical Asymmetric Communication Environments," Proceedings of the 11th Euromicro Conference on Real-Time Systems, pp. 491-505 Sep. 1998.
- [6] K. Stathatos, N. Roussopoulos and J.S. Baras, "Adaptive Data Broadcast in Hybrid Networks," 23rd International Conference on Very Large DataBases, Athens, Greece, pp. 326-335, Sept. 1997.
- [7] C. Hu and M. Chen, "Adaptive Information Dissemination: An Extended Wireless Data Broadcasting Scheme with Loan-Based Feedback Control," IEEE Transactions on Mobile Computing, vol 2, No. 4, pp. 322-336, October-December, 2003.
- [8] C. Hu an M. Chen, "Dynamic Data Broadcasting with Traffic Awareness," Proc. 22nd IEEE Int'l Conf. Distributed Computing Systems, pp. 112-122, July 2002.
- [9] E. Pitoura, "Supporting Read-Only Transactions in Wireless Broadcasting" in Proceedings of the DEXA '98 Workshop on Mobility in Databases and Distributed Systems, pp. 428-433, August 1998.
- [10] D. Aksoy & M. Franklin, "Rxx: A scheduling approach for large-scale on-demand broadcast," ACM/IEEE Transactions on Networking. 846-860, July 1999.
- [11] P. Xuan, S. Sen, OGonzalez, J. Fernandez and K. Ramamritham, "Efficient and Timely Dissemination of Data in mobile Environments," Proceedings of the Third IEEE Real Time Technology and Applications Symposium, Montreal Canada, pp. 38-48, June 1997.
- [12] M. Karakaya, O. Ulusoy, "An Efficient Broadcast Scheduling Algorithm for Pull-Based Mobile Environments," Wireless Networks, Vol5(No 3), pp. 183-193 May 1999.
- [13] S. Baruah and S. Lin, "Pfair Scheduling of Generalized Pinwheel Task Systems," IEEE Transactions on Computers Vol 47(No 7), pp. 812-816, July 1998.



윤혜숙

1986년 서울대학교 계산통계학과 학사
 1988년 서울대학교 계산통계학과 석사
 1988년~1995년 4월 한국통신 연구원
 2004년 8월 충남대학교 컴퓨터과학과 박사.
 2002년 9월~2005년 4월 충남대학교 전기정보통신공학부 BK 전임교수. 2005년 5월~현재 특허청 전기전자심사국 통신사무관. 관심분야는 실시간 데이터베이스시스템, 이동컴퓨팅시스템



김영국

1985년 서울대학교 계산통계학과 학사
 1987년 서울대학교 계산통계학과 석사
 1995년 버지니아대학교 컴퓨터과학과 박사. 1995년 VTT(Technical Research Centre of Finland) 및 SINTEF Telecom & Informatics, Norway 방문연구원. 2002년 8월~2003년 7월 UC Davis 객원교수. 1996년~현재 충남대학교 전기정보통신공학부 부교수. 관심분야는 실시간 데이터베이스 시스템, 전자상거래시스템, 이동데이터베이스시스템