

# 이동 컴퓨팅 환경에서 데이터 방송 스케줄링 기법의 성능 평가

## (Performance Evaluation of Data Broadcast Scheduling Methods in Mobile Computing Environments)

오 상 수 † 신 동 천 \*\*

(Sang Su Oh) (Dong Cheon Shin)

**요 약** 이동 컴퓨팅 환경의 데이터 방송 시스템에서 클라이언트에게 최소의 응답시간을 제공할 수 있도록 방송을 스케줄링 하는 것은 매우 중요하다. 본 논문에서는, 방송 스케줄링 기법들의 성능을 응답시간 측면에서 모의 실험을 통하여 평가하고 분석한다. 기존 연구와는 달리, 클라이언트의 요구 형태 변화를 고려한 여러 가지 일반적인 응용 환경을 설정하여 평가한다.

**Abstract** In data broadcasting systems under mobile computing environments, it is very important to schedule the data broadcast in order to provide clients with the minimum response time. In this paper, in terms of the response time, we evaluate and analyze the performance of several broadcasting methods using the simulation approach. Unlike the previous work, we evaluate those methods by establishing several general and realistic application environments, in which the pattern change in client's needs is taken into account.

### 1. 서 론

이동 컴퓨팅 환경이란 클라이언트의 위치에 구애됨 없이 언제, 어디서나, 기존의 온라인 컴퓨팅 환경과 같은 수준의 서비스를 사용할 수 있는 환경을 말한다. 이동 컴퓨팅 환경은 HPC(Hand held PC), PDA(Personal Digital Assistants)나 펜 컴퓨팅(Pen Computing) 기기들의 개발과 함께 비약적인 무선통신 기술의 발전으로 생성된 새로운 컴퓨팅 패러다임으로서, 기존의 컴퓨팅 환경과는 달리 새로운 편리성-이동성-을 제공함으로써, 실생활에 커다란 변화를 가져오고 있다. 이를 뒷받침하기 위하여 무선 네트워크에 연결되는 휴대용 기기들의 소형화와 배터리 소모 등 클라이언트의 물리적 구조에 관한 연구와 함께 무선 네트워크 환경적 특성에서 기인한 잦은 단절, 데이터 손실에 관한 연구가 진행되고 있다[1,2,3,4].

그 중 무선 네트워크 환경에서 클라이언트와 서버간의 데이터 양 및 통신 능력을 고려해 볼 때, 서버측의 가용 대역폭에 비하여, 클라이언트가 가진 가용 대역폭은 상대적으로 적을 뿐만 아니라, 서버측에 요구하는 클라이언트의 요구 데이터도 서버측에서 클라이언트측으로 전송하는 데이터 양보다 작다. 이러한 무선통신의 송·수신 특성에 기인한 통신환경을 비대칭(asymmetric)통신환경이라고 한다[4,5,6]. 이러한 환경에서는 클라이언트가 서버측으로 직접 데이터를 요청하는 기존의 방법보다, 클라이언트의 요청 없이도 서버측에서 데이터를 클라이언트에게 전달하는 방식인 데이터 방송(data broadcast)기법이 제기되었다[5,6].

이 기법은 특히, 다수의 클라이언트가 관심을 갖는 공용정보-날씨, 뉴스, 주식, 교통, 스포츠 등-의 서비스에 있어서 클라이언트의 배터리나 대역폭의 자원 제약성을 완화시키는 특성을 가지고 있다. 이러한 데이터 방송 시스템의 주요 성능평가 기준은 응답시간이며, 응답시간은 일반적으로 클라이언트는 방송되는 항목 중 특정 항목을 액세스하는데 관심 있다고 가정할 때, 클라이언트가 일정 항목에 관심을 두고 질의를 제출하여, 방송 채널로부터 데이터를 얻는데 까지 걸린 시간을 의미한다.

† 비 회 원 LG전자 정보통신중앙연구소 연구원  
sangsu@lgic.co.kr

\*\* 중 심 회 원 중앙대학교 정보시스템학과 교수  
dcshin@cau.ac.kr

논문접수 : 2000년 6월 21일  
심사완료 : 2001년 2월 9일

그러나, 서비스를 제공하는 시스템의 처리능력이나 이용 가능한 전송 대역폭이 클라이언트 수에 비하여 극히 제한적인 상황에서, 다수의 요구를 처리해야하는 서버측은 클라이언트의 요구정보를 취합하여, 일정 정보만을 다음 방송 스케줄에 포함시키게 된다. 이때 각 클라이언트 요구시점과 방송 스케줄을 결정하는 시점간에는 엄연한 시간적 간격이 존재하며, 개별요구가 스케줄에 반영되었는지 여부도 불투명한 상태는 예측 불가능한 응답시간을 유발한다. 이러한 상황은 서비스를 제공받는 클라이언트들이 응답시간에 대하여 민감하다고 가정할 때, 방송 데이터의 스케줄을 결정하는 과정은 전체 방송 시스템의 서비스 질을 좌우하는 중요한 과정이라고 할 수 있다.

방송 스케줄링 기법은 방송 데이터의 선정 및 데이터의 방송순서를 결정하는 문제를 다룬다. 방송 데이터를 선정은 사용자의 요구를 감안하여 서비스 데이터 중 일부의 데이터를 선택하는 과정이며, 이를 위하여 서버는 사용자의 액세스 패턴이나 서버의 처리 용량을 감안하여 선정한다[10]. 서버는 선정된 데이터의 효율적인 전송을 위한 데이터 방송순서를 결정한다. 이때, 서버는 사용자 액세스 정보를 바탕으로 액세스가 높은 데이터의 순서[7,8]나 액세스 빈도에 따른 빈번한 데이터 방송[5,6]을 결정한다.

방송 스케줄링 기법의 성능 평가는 방송 스케줄링 기법의 제기와 함께 제기된 스케줄링 기법의 성능 평가를 중심으로 이루어졌다. 액세스 빈도에 따른 데이터 방송 구성 성능 평가[7,8]와 데이터 액세스 빈도에 따른 데이터 중복전송 방송구성 성능 평가[5,6,12]가 진행되었다. 그러나, 기존 연구의 성능 평가는 실험환경에서 실제 운용환경에서 고려되어야 하는 대역폭의 제약, 캐쉬사용[8]이나 사용자 액세스 패턴 변화[5,6]를 충분히 고려하지 않고 있다. 특히, 액세스 패턴은 클라이언트 단말이 갖는 정보 소비 경향을 반영하는 것으로서, 이를 간과한 실험결과는 실제 운용환경에서 일어날 수 있는 클라이언트의 요구패턴에 중속된 서비스 응용을 위한 적절한 방송 스케줄링 선정에 어려움을 가질 수 있다.

따라서, 본 논문은 기존 연구에서 제기되었던 대역폭의 제약, 캐쉬사용, 사용자 액세스 패턴변화, 패턴의 변화주기 및 변화정도를 고려한 실험환경을 설정하여 실제 운용환경과 유사한 실험환경을 구성하였다. 이러한 환경에서 사용자 변화를 충분히 고려한 클라이언트 모델을 통하여 각 스케줄링 기법의 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 평가 대상이 되는 방송 스케줄링 기법을 기술하고, 3장에서는 성

능평가 모델을 소개한다. 4장은 성능 평가 결과를 분석하고, 마지막으로 5장에서는 결론을 맺는다.

## 2. 방송 스케줄링

이 절에서는 데이터 방송 스케줄링기법의 성능평가 대상인 FCFS, RxW, Broadcast Disks(이후 BD라 칭함)기법을 소개한다.

### ■ FCFS 기법

FCFS 스케줄링 기법[8]은 도착순서에 의한 데이터 방송 스케줄링이 이루어지는 순차 데이터 서비스 스케줄링에 속한다. 즉, 클라이언트에 의하여 요구되는 데이터의 요구율이나 데이터 요구 빈번성에 대한 고려 없이, 직관적으로 전체 요구 데이터 중 측정시점에서 요구시점이 가장 우선인 데이터를 선정하게 된다. 따라서, 요구된 데이터는 서버의 임시 저장소에 요구순서로 배열하게 되며 배열순서에 따른 순차적인 데이터 방송이 이루어진다. 이 방식은 운영체제의 배치 프로세스에 일반적으로 사용되는 방식으로, 모든 데이터에 대한 요구를 요구시간에 따라서 순차적으로 처리하며, 응답 시간의 예측이 용이한 장점을 지니고 있다.

### ■ RxW 기법

RxW 스케줄링 기법[8]은 우선순위기반(priority oriented) 스케줄링 기법으로 Aksoy가 제기하였다. RxW 스케줄링 기법은 클라이언트의 요구율과 요구시간을 근간으로 하여, 데이터를 선택, 방송하는 스케줄링 기법이다. 스케줄링을 위하여 사용하는 선택값은 클라이언트의 요구율을 알 수 있는 요구횟수(data demand frequency) R와 지연시간(wait time) W의 곱이다. R의 값은 서버큐에 저장되는 클라이언트들의 요구횟수를 측정하여, 각 데이터마다 요구 횟수값을 가진다. W값은 데이터를 처음 요구한 시점에서 현재 시간값의 차이값을 각 데이터마다 관리를 하게 된다.

그림 1은 서버큐가 관리하는 클라이언트 요구정보를 기술한 것으로, PID는 요구된 데이터의 식별자 값이며, 1stArv은 요구도착 시간을 나타내며, R은 요구된 횟수를 나타낸다.

PID	1stArv	R
b	10	1
n	20	1
a	50	30
y	90	25
z	15	8

그림 1 서버 큐의 구조

서버는 각 데이터의 1stArr에서 현재 시간의 차를 R와 곱한 RxW를 계산하며, 스케줄링시 계산된 RxW를 비교하여, RxW에 따라 데이터를 배열하고, 값의 순서에 따라 방송하게 된다.

RxW 스케줄링 기법은 O(N) 오버헤드를 가지는 알고리즘으로서, 간편한 연산으로 사용자 요구율과 지연시간을 스케줄링에 반영함으로써 전체 사용자의 응답시간을 개선할 수 있음을 보이고 있다. 그러나, FCFS에 비하여 스케줄링시 RxW값을 비교하는 과정을 거치기 때문에 스케줄링 시간이 길어진다.

■ BD 기법

Acharya에 의하여 제기된 방송 데이터 스케줄링 기법[5,6,12]이다. Acharya는 데이터 방송시 일정한 주기로 반복되어 데이터 전송이 이루어지는 것이 클라이언트의 응답시간 개선에 효율적임을 설명하였으며, 평균적인 클라이언트 응답시간을 위해서 많은 클라이언트가 접근할 것으로 예상되는 데이터를 상대적으로 자주 방송할 필요가 있음을 지적하였다. 따라서, 방송을 위한 데이터는 선호도(priority)를 통하여 Hot, Cold의 여부를 선별한 후 다음 과정을 거쳐 데이터를 전송하는 기법이다.

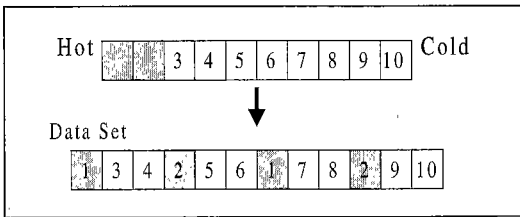


그림 2 BD의 구성

스케줄링의 시작은 Hot과 Cold를 선호도 순서로 나열한 후, Hot 데이터 집합과 Cold 데이터 집합으로 분류하여, Hot 집합의 데이터를 Cold 집합의 데이터보다 빈번히 방송하기 위한 Data Set을 구성한다. 그림 2는 1~10까지의 요구된 데이터를 요구 선호도에 따라 나열한 후, Hot 데이터를 1, 2로 구성된 데이터 집합으로 선정하고, Data set을 구성시 1, 2를 Cold 데이터 집합의 값보다 1회 더 방송하는 방송 순서로 구성한다.

BD의 구성상 장점은 Hot 집합의 데이터를 방송주기 내 여러 번 방송하는 특성을 가지고 있다. 따라서, 주기 내 빈번히 요구되는 데이터의 액세스 확률이 높아짐에 따라 전체 클라이언트 액세스 성능이 향상될 수 있음을 제시하였다. 이 기법의 스케줄링 시간은 위 두 기법에

비하여 스케줄링 소모시간이 큰 특징을 지니며, 방송 주기가 길다. 따라서 사용자 요구 변화가 심한 운용환경에서 성능의 저하가 예상된다. BD에 관한 연구로 [12]에서는 사용자의 액세스 패턴의 변화를 가정한 BD의 성능 평가연구를 진행하였다. 그러나, 네트워크 환경에서 사용자가 요구한 데이터의 직접 전송방식을 취하고 있다는 점에서 기존의 방송환경과 틀을 달리한다.

### 3. 성능 평가 모델

#### 3.1 기본 운용 환경

본 논문에서 가정하는 서비스 환경은 각 스케줄링이 제기된 환경을 기반으로 하고 있으며, 공통된 가정은 다음과 같다.

- 데이터 크기는 일정하다.

데이터베이스의 데이터는 동일크기의 데이터 항목(item)단위로 나뉘어지며, 각 항목단위로 서비스가 이루어진다고 가정한다. 이는 실제 증권정보 및 기상정보의 경우 일정 단위의 문자 데이터를 기본단위(item, packet) 상정하여, 서비스 이용료를 부과하고 있는 서비스 제공자를 고려한 설정임과 동시에 전송시 유발되는 전송시간의 일관성을 보장하기 위함이다.

- 각 클라이언트는 자체 캐쉬를 가진다.

서버측에서 전송되는 방송 데이터는 클라이언트 입장에서 최적이지 아닐 수 있다. 왜냐하면, 방송 데이터의 선정은 전체 클라이언트 요구를 충족시키기 위한 것이기 때문이다. 따라서 클라이언트 자체에 데이터를 저장할 수 있는 저장공간을 가지고 있으면, 방송 데이터 의존성을 감소시킬 수 있을 뿐 아니라 개별 클라이언트의 액세스 패턴을 따르는 자체 저장소를 유지함으로써 응답시간을 향상시킬 수 있다.

- 데이터의 일관성 고려는 하지 않는다.

방송 데이터는 읽기 전용 데이터로서 방송 데이터의 클라이언트 캐쉬내 저장에 따른 데이터 일관성 문제가 야기될 수 있다. 그러나, 방송 데이터의 캐쉬내 데이터의 일관성 문제나 방송전·후의 데이터 일관성 문제는 데이터 스케줄링 문제와는 또 다른 연구과제로 제시되고 있다[4,10]. 따라서 본 환경에서 데이터 일관성 문제는 고려치 않는다.

- 데이터 방송은 일정주기를 갖는다.

이미 기존 연구에서 주기적 데이터 방송은 비 주기적 방송에 비하여 전체 클라이언트들의 응답시간측면에서 효율적임이 증명되었다[5]. 따라서 본 서비스 환경에서도 주기적 데이터 방송을 설정하였다.

- 클라이언트는 요구 데이터를 수신 받을 때까지 새

로운 데이터를 요구하지 않는다.

클라이언트는 일정시점에 데이터 요구는 단일 항목에 국한될 뿐 아니라, 선 요구 데이터를 수신 받기 전에는 새로운 데이터에 대한 요구를 발생시키지 않는다. 이는 각 프로세스 단위로 운용되는 클라이언트가 서버 프로세스를 유발함으로써 발생될 수 있는 복잡성을 감소시킨다.

· 채널 할당 방식은 방송 및 요구 채널의 혼용 (hybrid allocation) 방식을 가정한다[10].

이는 데이터 방송환경이 특정 데이터만 고정적으로 방송되는 환경은 클라이언트의 요구를 반영하지 못하므로, 대화형의 시스템 구현이 일반화되는 현 컴퓨팅 환경에 적절치 못하다. 따라서, 네트워크 환경설정에 따라 서버의 데이터 방송은 클라이언트의 요구에 의하여 구성되며, 요구된 데이터는 방송 채널을 통하여 클라이언트에게 전송된다. 또한 송·수신 대역폭의 특성상 비대칭 통신환경이기 때문에 서버측의 데이터 방송 대역폭과 데이터 요구 대역폭의 차를 고려한 네트워크 모델이 요구된다. 이에, 본 논문에서는 데이터 방송 환경에서 비대칭적인 송·수신 대역폭을 지니며, 이를 통하여 클라이언트 송·수신이 가능한 네트워크환경을 설정한다.

· 데이터 전송은 송신채널과 수신채널이 독립된 단일 채널로 한다.

Vaidya가 제시한 다 채널 방식[9]의 데이터 전송은 각 클라이언트의 응답시간을 효과적으로 감소시킬 수 있으나, 각 채널이 동일한 데이터를 방송 시기만 달라하여 방송하기 때문에 서버측은 방송 대역폭의 효율적인 이용을 위한 별도의 연산이 필요하며, 클라이언트의 요구패턴이 동적으로 변하며, 각 셀내의 클라이언트수가 불규칙적인 상황에서 각 채널의 방송 데이터의 스케줄 설정이 어렵다는 점에서 본 논문에서 가정한 네트워크 환경은 송신 채널과 수신채널이 독립된 단일 채널로 국한한다.

· 서버는 클라이언트의 데이터 액세스 패턴 및 요구 변화를 감지한다.

클라이언트의 동적인 요구변화 특성에 맞는 방송 데이터의 스케줄링을 위하여, 대부분의 방송 시스템에서는 클라이언트가 요구정보 전송시, 요구 데이터의 식별자나, 요구시간 등의 부가정보를 실어(piggyback) 전송한다고 가정한다. 이 정보를 이용하여 서버는 클라이언트의 요구 데이터 및 요구 항목을 식별할 수 있게 된다.

**3.2 성능평가**

그림 3은 본 연구에서 사용한 성능평가 모델을 나타낸다.

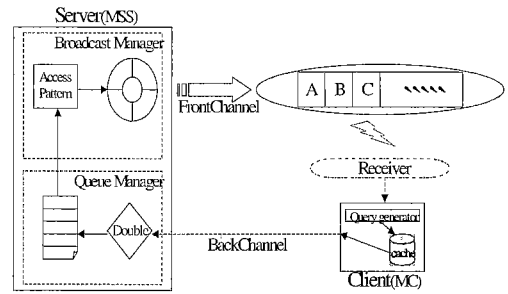


그림 3 성능 평가 모델

**3.2.1 클라이언트**

다수의 클라이언트가 존재하는 상황을 위해서 각 클라이언트는 프로세스로 구성하였으며, 시뮬레이션 시간동안 정해진 과정을 거치며, 각각의 프로세스가 독립적인 클라이언트 역할을 수행한다.

**가) 질의 생성**

클라이언트는 시뮬레이션 모델에서 개별 프로세스 단위로 동작하게 된다. 클라이언트 프로세스의 첫 동작은 클라이언트 데이터 요구 단계로, 질의 생성기(query generator)로부터 시작된다. 질의 생성기는 Zipf 분포를 따르는 클라이언트의 질의 패턴을 반영한 서비스 데이터의 식별자값을 생성하여 식별자의 데이터 값을 요구하는 과정을 나타낸다. Zipf 분포는 기존 연구에서 클라이언트의 질의 분포가 특정 데이터에 집중되어 나타나는 불균등(non-uniform)한 데이터 액세스 환경을 도출하기 위하여 사용된 분포이다[13]. Zipf 분포의 형태는 0.0~1.0까지 변하는  $\theta$  값에 의하여 결정되며, 1.0에 가까울수록 불균등 정도가 큰 특성을 가진다. 이후 프로세스는 클라이언트의 캐쉬에 데이터가 있는지를 검사하게 된다. 이때 캐쉬에 해당 데이터가 존재하면, 클라이언트의 데이터 요구 프로세스는 끝나게 된다.

본 논문에서는 클라이언트의 액세스 변화를 도출하기 위하여, 요구변화시점을 위한 변수값으로 Frequency와 요구변화정도를 위한 Offset 변수를 사용하였다. Frequency는 클라이언트의 액세스 변화 주기를 나타내고, Offset은 변화 정도를 나타낸다. 클라이언트 프로세스는 Frequency 값에 따라 Zipf 분포에서 얻어지는 데이터가 Offset만큼 변경된 데이터를 소유하게 된다. 결국 Offset과 Frequency의 값에 따라서 클라이언트의 요구 데이터값이 결정된다.

**나) 방송 정취 및 데이터 전송**

만일 캐쉬에 데이터가 없을 시 클라이언트 프로세스

는 일정기간(listen time)동안 서버측에서 방송하는 내용을 듣게 된다. 이 과정은 서버측으로 요구를 전달(upload)할 수 있는 대역폭이 상대적으로 적기 때문에 대역폭의 효율적 이용을 위한 측면에서의 효과와, 서버측에 과도한 요구를 감소시키며 효과적으로 적용된다. 이 과정에서도 얻지 못한 데이터는 서버에 직접 요청하는 과정을 거쳐 데이터 방송시 데이터를 수신하고 프로세스는 질의 간격에 따라 일정 시간 간격을 갖는다.

다) 데이터 수신 및 캐쉬관리

클라이언트는 질의한 데이터의 수신을 위해 방송되는 데이터 중 해당 데이터가 방송되었는지를 검사하며, 계속 기다리게 된다. 이 과정에서 서버측에서 데이터를 방송하는 시간과 도착을 검사하는 시간과의 차를 두고 검사를 진행한다. 클라이언트는 데이터 수신시 클라이언트에게 데이터값의 표시(display)와 동시에 캐쉬에 데이터의 저장을 시도한다. 캐쉬관리는 일반적으로 사용되고 있는 LRU(least recently used) 정책을 따른다.

표 1 운용환경 변수

변수명	내용	단위
DatabaseSize	데이터베이스 크기	Item
DataItemSize	Item 크기	Byte
PullDataSize	요구 데이터 크기	Byte
Frontchannel	데이터 방송채널 대역폭 크기	Bps
Backchannel	데이터 요구채널 대역폭 크기	Bps
Data_transfer_time	서버측으로 데이터 전송시간	(1Item/Time)
SEND_TM	서버에 요청하는데 걸리는 시간	Time (시뮬레이션 시간)
Send_Delay_Time	Backchannel 회선에서 대기시간	Time (시뮬레이션 시간)

표 2 클라이언트 변수

변수명	내용	단위
Number_Client	사용자수	명
CacheSize	캐쉬 크기	Item
Cache_Access_time	캐쉬 액세스 시간	(1Item/Time)
WAIT_TM	Broadcast 대기 시간	Time (시뮬레이션 시간)
INTARV	재질의 시간간격	Time (시뮬레이션 시간)
THETA	Zipf' $\theta$ 값의 크기	$\theta = (1-THETA)$
Frequency	데이터 요구변경 주기	Time (시뮬레이션 시간)
Offset	데이터 요구변경 범위	Item

라) 네트워크 환경

본 데이터 방송 네트워크 환경은 서버측에서 클라이언트측으로의 데이터 전송을 담당하는 방송채널(Front Channel)과 클라이언트측에서 서버측으로 데이터 요구를 전송하는 요구채널(BackChannel)이 있다. 이 두 채널의 대역폭은 방송채널에 비하여 요구채널이 가지는 대역폭은 상대적으로 적은 비대칭 통신환경을 가정하고 있으므로 환경 제약을 적용하기 위하여 방송채널과 요구채널의 비율을 100 : 1로 구성하였다[12]. 또한, 클라이언트의 요구가 요구채널을 통하여 서버측으로 전달되므로, 요구채널의 가용 대역이 사전 프로세스에 의하여 점유되었을 시 일정시간 후 재시도를 하는 과정이 요구된다.

3.2.2 서버

서버 프로세스는 데이터 방송 서버를 하나로 가정하므로, 하나의 서비스 서버가 존재하며, 다수 클라이언트 프로세스가 요청하는 데이터를 방송을 통하여 전송한다. 세부적인 서버 프로세스 동작은 다음과 같다.

가) 서버 큐

서버의 큐는 클라이언트에 의해 요청된 데이터를 분류하고 저장하는 역할을 수행한다. 서버에게 요청되는 클라이언트의 요구 데이터의 크기는 서비스 데이터보다 상대적으로 적은 형태를 가정하고 있다. 예를 들면, 서비스 데이터 식별자값이나 인덱스 정보를 서버측으로 전송할 것이다. 서비스 데이터의 요청을 받은 서버큐는 서버큐내에 요구된 데이터가 이미 존재하는지를 검사하게 된다. 이 과정에서 중복된 데이터는 클라이언트 요구 빈도값을 증가시킨다. 존재하지 않는 데이터 경우 새로운 노드를 생성하여 서버 큐에 저장된다.

나) 방송 관리자

서버 큐에서 관리하고 있는 데이터는 방송주기마다 방송을 위한 데이터 선정작업과 제시된 스케줄링 방식에 따라서 데이터 방송 순서를 결정하게 된다. 이 작업에서 서버 큐에 저장된 각 데이터에 관한 정보를 이용하여 데이터를 선정하게 된다. 본 논문에서는 각 스케줄링 방식에 맞추어 각 데이터의 요구빈도와 최초 요구시간의 정보를 이용하여 데이터의 선정이 이루어지며 FCFS, RxW, BD의 스케줄링 방식에 의한 스케줄링이 이루어진다. 스케줄링 되어진 데이터는 방송 대역폭(FrontChannel)에 따라 순차적으로 전송된다. 이때 가정환경이 방송환경이라는 점을 감안하여, 데이터는 방송 수신상태에 있는 모든 클라이언트에게 전송되어야 한다.

3.2.3 변수값 설정

가) 운용환경 모델 변수값 설정

서비스 데이터의 크기(DatabaseSize)는 기존 논문에서 보편적으로 사용한 3000개의 서비스 항목을 중심으로 1000, 3000, 5000개의 서비스 데이터 항목수를 가지는 데이터베이스 환경을 설정하였다[6,11,12]. 이에 각 항목의 크기(DataItemSize)는 기존 연구[11]에서 증권정보 서비스 환경 설정에서 사용된 128byte를 기준으로 삼았다. 이때 증권정보를 얻기 위해 서버측에 전송되는 요구 데이터의 크기(PullDataSize)는 해당 증권정보의 식별자값 및 요구 데이터의 부가정보를 고려하여 설정되었다.

방송채널(FrontChannel)의 크기는 기존 연구[6,7,11,12]에서 10kbps ~ 19.2kbps로 설정하고 있다. 이에 본 환경은 19.2 kbps 대역폭을 가지는 방송채널을 고려하였으며, 대역폭의 비대칭성에 관한 고려를 적용한 기존 연구의 제시[12]에 따라 방송대역폭의 1%로 요구 대역폭의 크기(BackChannel)를 설정하였다. 이에 따라 1개당 항목의 방송소요 시간(Data\_transfer\_tome)은 0.1

초이며, 요구 데이터의 전송시간(SEND\_TM)도 0.1초의 시간이 소요된다. 또한 동시 수용 가능한 요구 데이터는 12개로 클라이언트 12명이 동시에 요구 데이터를 전송할 수 있다. 이때 선 점유된 요구채널은 1초의 전송시간(Send\_Delay\_Time)이 소요된 후에 다음 요구 데이터를 수용할 수 있다.

나) 클라이언트 모델 변수값 설정

각 클라이언트는 시뮬레이션 프로그램의 시작과 동시에 클라이언트 수(Number\_Client)만큼 개별 프로세스로 설정되어 독립적인 클라이언트의 역할을 수행한다. 방송 수신 가능영역인 셀내의 클라이언트의 수는 방송 항목의 수에 따른 비율로 1000명까지로 설정하였다. 각 클라이언트는 캐쉬를 소유하고 있으며, 캐쉬의 크기(CacheSize)는 10~300 항목을 저장할 수 있는 크기를 변화시켜 가며 실험을 진행하였다. 또한, 캐쉬내의 데이터의 액세스에 소요되는 시간(Cache\_Access\_time)을 방송 전송시간의 10%로 설정하였다. 그리고, 방송 내용을 청취하는 시간(WAIT\_TM)을 10~ 40까지 변화시키며 성능을 측정하였다. 사용자의 질의 제출 간격(INTARV)은 지수분포를 따르는 개별 요구 제출시간을 설정한다.

그리고, 클라이언트의 데이터 액세스 패턴(THETA) 값에 의하여 결정된다. 이 값을 0.0 ~ 1.0까지 변화시키며 클라이언트의 요구 패턴을 조절한다. 또한, 요구변화의 빈도(Frequency)와 정도(Offset)는 빈도값의 경우 시뮬레이션 시간이 20000시간을 고려하여 0~10000까지 변화시키며 실험을 진행하였으며, 서비스되는 데이터베이스 항목수를 고려하여 변화정도 값을 0~ 2500까지 변화시키며 실험을 진행하였다.

다) 서버 모델 변수값 설정

서버의 데이터 방송시간(Broadcast\_Period)은 방송하

표 3 서버 변수

변수명	내용	단위
Broadcast_Period	데이터 방송시간	Time (시뮬레이션 시간)
ProgramSize	방송 Item 크기	Item
Disk1Size	Disk1의 크기	Item
Disk2Size	Disk2의 크기	Item
FCFS_INTVAL	FCFS 스케줄링 시간	Time (시뮬레이션 시간)
RxW_INTVAL	RxW 스케줄링 시간	Time (시뮬레이션 시간)
BD_INTVAL	BD 스케줄링 시간	Time (시뮬레이션 시간)

표 4 운용환경 변수값

변수명	변수값	단위
DatabaseSize	1000, 3000, 5000	Item
DataItemSize	128K	Byte
PullDataSize	2	Byte
Frontchannel	19200	Bps
Backchannel	192	Bps
Data_transfer_time	0.1	(1Item/Time)
Cache_Access_time	0.01	(1Item/Time)
SEND_TM	0.1	Time
Send_Delay_Time	0.1	Time

표 5 클라이언트 변수값

변수명	변수값	단위
Number_Client	10~1000	명
CacheSize	10~ 300	Item
Cache_Access_time	0.01	(1Item/Time)
WAIT_TM	0~ 40	Time
INTARV	5~ 40	Time
THETA	0.0~ 1.0	
Frequency	0~ 10000	Time
Offset	0~2500	Item

려는 데이터의 개수(ProgramSize)에 의하여 결정된다. 방송채널의 대역폭이 고정되어 있기 때문에 방송량에 따라 방송시간이 정해진다. 또한 BD의 경우는 각 디스크에 포함되는 데이터 양이 상이하며, 데이터 이용법칙(20 : 80 Rule)에 따라서 빈번히 방송되는 디스크 크기(Disk1Size)를 방송 데이터의 개수에 20%로 할당하였다. 또한 두 개의 디스크는 2:1 비율로 방송된다. 이는 기존의 실험환경에서 설정된 방식과 유사하다[12].

표 6 서버 변수값

변수명	변수값	단위
Broadcast_Period	10 ~ 40	Time
ProgramSize	100 ~ 400	Item
Disk1Size	ProgramSize / 5	Item
Disk2Size	ProgramSize - Disk1Size	Item
FCFS_INTVAL	1	Time
RxW_INTVAL	2	Time
BD_INTVAL	4	Time

또한 각 스케줄링의 스케줄링 오버헤드를 반영하기 위하여 각 스케줄링 기법의 스케줄링 알고리즘을 총 서비스 데이터베이스 항목을 스케줄링 하는데 소요된 시간을 비율로 산정하여 적용하였다.

### 3.3 모의 실험 환경 및 성능 평가 척도

모의 실험을 위한 시스템 환경은 인텔사의 PIII 칩에 램 128MB를 장착한 Window NT 4.0 운영체제환경에서 CSIM18 시뮬레이션 언어[14]를 사용하여 평가하였다. CSIM은 프로세스 시뮬레이션에 널리 사용되는 언어로, 기존 연구들[5,6,8,12]에서도 동일한 시뮬레이션 언어로 모델을 구성하였다.

성능 평가 척도는 앞서 제시한 바와 같이 각 변수값에 대한 사용자의 평균 액세스 시간이다. 또한 캐쉬의 크기에 따른 캐쉬 히트율을 측정된 실험에서는 평가 척도로 캐쉬 히트율이 척도로 제시되었다.

## 4. 실험 결과 분석

### 가. 실험 1: 사용자수의 변화

실험1은 Offset과 Frequence를 '0'으로 설정하였을 시 각 스케줄링의 성능을 나타내고 있다. 즉, 사용자 요구변화가 일어나지 않는 상태에서 사용자 요구패턴이 Zipf  $\theta = 0.95$ 로 데이터 요구 집중도가 높은 상황이다.

100명 미만의 사용자를 처리할 경우, FCFS의 성능

이 가장 좋은 것으로 나타난다. 이러한 결과를 낳은 이유는 사용자의 수가 너무 적어 방송 데이터양이 설정한 400개의 항목 이하의 요구를 제출하기 때문에 적은 FCFS의 성능이 가장 우수하다. 이와 대조적으로 BD의 경우 스케줄링 시간이 긴데다, 방송 데이터양이 스케줄링의 특성상 480개의 항목을 방송하게 되므로 한 주기인 방송시간이 길어지게 된다. 따라서 소량의 데이터를 전송하기 위한 준비시간이 길기 때문에 가장 성능이 떨어지게 된다. RxW인 경우 BD보다는 성능이 좋으나, FCFS보다는 성능이 떨어진다. 이유는 소량의 데이터 처리에서 FCFS보다 스케줄링 시간이 긴데다, 방송을 청취하는 사용자 수가 상대적으로 적고, 청취율이 낮아 선점된 데이터 성능을 제대로 반영하지 못하고 있다.

200~300명 사이의 사용자가 이용하는 상황에서는 FCFS와 RxW의 성능우위가 변하게 되었다. 이러한 이유는 사용자 수가 증가하여 방송을 청취하는 청취율이 높아져 선호도를 기반으로 하는 데이터 전송상의 장점이 부각된 것이라 볼 수 있다. BD의 경우 상대적으로 평균응답시간이 향상되는 추세를 보이고 있다. 이는 전체 사용자의 증가로 방송 청취율이 높아지며, 빈번히 요구되는 데이터를 중복 방송하는 효과가 나타나기 때문이다. 400명 이상의 사용자가 이용하는 상황에서 BD의 성능이 가장 우수한 것으로 나타났다. 사용자의 요구수가 400이상인 경우 FCFS와 RxW는 방송 데이터의 응답률 면에서 중복 데이터 방송이 이루어지는 BD에 비하여 떨어지기 때문에 평균응답시간이 저조한 값을 가지게 되었다. FCFS가 RxW보다 현저히 성능이 저조한 이유는 상대적으로 선호도가 낮은 데이터가 우선 방송되는 상황이 빈번히 발생할 가능성이 존재하기 때문에 전체 사용자의 평균응답시간이 가장 높게 나타나고 있다.

위 결과를 종합하여 고려할 때 사용자의 요구변화가 없거나 드문 상황에서는 사용자의 수가 증가할수록, FCFS와 RxW보다 상대적으로 BD의 성능이 좋아지는 결과를 얻을 수 있다.

### 나. 실험 2: 요구패턴 변화(Zipf' $\theta$ )

요구패턴은 사용자의 데이터 요구 경향을 반영하는 것으로서 본 실험에서는 요구 경향 변화에 대한 각 스케줄링의 적응성을 고찰하는데 초점을 둔다.

실험2는 Zipf'  $\theta$  값을 변화시키며 각 스케줄링의 평균응답시간을 측정하였다.  $\theta$  값이 1.0~ 0.8의 값을 가질 경우 BD의 성능이 가장 우수하나, 이후 평균응답시간이 급격히 상승함을 볼 수 있다. 이는 BD의 구성 관점에서

파악될 수 있는데, 매우 빈번히 요구되는 데이터를 방송 주기내 두 번이상 반복하여 방송하기 위하여 데이터를 스케줄링 하기 때문에  $\theta$  값이 작아지는 상황은 사용자의 데이터 요구의 균일성(uniform)이 높아지는 상황이다. 따라서 BD 요구빈도의 편차가 크지 않은 데이터를 중복 방송하는 상황이 야기되며, 이것은 전체 사용자의 평균 응답시간이 커지는 원인으로 작용된다.

RxW인 경우는 FCFS와 비교하여 0.7까지 우위를 보였으나, 0.7이하로 내려가자 스케줄링 시간의 부담으로 인하여 FCFS보다 높은 평균응답시간을 가져왔다. RxW의 경우도 데이터의 요구 균일성이 떨어지는 상황에서 요구율의 편차가 낮아져 스케줄링의 장점요인이 감소하며, FCFS에 비해 부담요인인 스케줄링 시간의 영향으로 인하여 FCFS보다 평균응답시간이 높아졌다. 그러나, BD과 같은 급격한 성능하락은 일어나지 않고 있다.

FCFS의 경우 스케줄링 과정은 Zipf'  $\theta$  값에 영향을 받지 않지만 평균응답시간의 증가를 보인 원인은 캐쉬 히트율의 감소에 원인이 있다.

캐쉬관리기법이 LRU(least recently used)이기 때문에 사용자의 요구패턴이 균일해질수록 빈번한 데이터 교체가 일어나기 때문이다. 따라서 FCFS의 평균응답시간은 점차 높아지는 결과를 낸다.

다. 실험 3: 요구빈도 및 변화량에 따른 변화

본 실험은 Offset값과 Frequency값을 변화시켜 결과를 측정하였다. Offset을 500, Frequency를 100으로 설정하여 실험하였다. 본 값의 의미는 시뮬레이션 시간 100초를 간격으로 데이터 변경 범위가 500 item 만큼의 변경이 발생한다는 설정이다. 이러한 가정은 사용자의 요구패턴은 특정 데이터에 집중하지만, 시간대별 요구 데이터 내용이 변하는 상황을 가정하였다.

이 같은 상황에서 BD는 FCFS, RxW에 비하여 평균 응답시간이 높게 나타났다. 이 같은 상황은 BD가 가지는 스케줄링 특성 및 방송주기에서 원인을 찾을 수 있다. BD의 경우 데이터의 스케줄링시점에서의 요구 데이터의 요구율을 기준으로 BD를 작성한다. 따라서, 방송될 당시에 Frequency의 기준에 부합되어 데이터 요구 범위가 상이해 지면 방송을 청취하는 사용자들의 청취도가 낮아진다. 또한, 스케줄링의 시간이 길고, 방송주기가 길기 때문에 빈번한 사용자 요구 변화를 맞추기가 어렵다. 이에 비해 FCFS와 RxW는 BD에 비해 단 시간의 스케줄링 시간 및 방송주기로 인하여 전체 사용자의 평균응답시간 측면에서 유리한 점을 지니고 있다. 따라서 사용자 요구의 빈도 및 요구 범위의 변화에 적응력에서 우수하다는 것을 알 수 있다.

라. 실험 4: 캐쉬 성능평가

데이터 방송환경에서 캐쉬의 역할은 사용자의 요구가 많은 데이터를 캐쉬내에 관리함으로써 서버측으로의 데이터 액세스를 감소시키는 역할을 한다. 그러면 데이터 방송환경에서 사용자 캐쉬의 크기는 얼마나 설정되어야 하는가? 에 관한 의문을 제기할 수 있다. 다음 그림의 결과는 다른 변수를 고정시킨 다음 캐쉬의 크기를 변화시키며, 캐쉬의 히트율(Hit Ratio)을 측정한 것이다.

그림 7에 나타난 바와 같이 캐쉬의 크기에 비하여 캐쉬의 히트율은 일정크기 100이상부터 증가하지 않고 있다. 즉 캐쉬크기와 히트율은 비례하지 않고 있다. 이유는 사용자의 데이터 액세스 패턴이 특정 데이터에 집중되어 나타나기 때문이다. 즉, 데이터 방송 환경에서의 데이터 액세스 패턴을 나타내는 Zipf 분포의  $\theta$  값이 0.95로 설정되어 불균등(non-uniform)한 데이터 액세스 형태를 보이기 때문에 100이상의 데이터를 저장하는 캐쉬는 히트보다 교체가 빈번히 일어나게 된다.

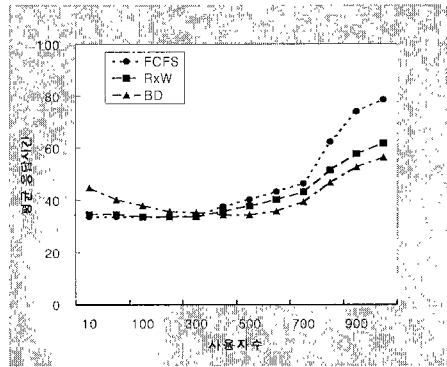


그림 4 사용자수의 변화

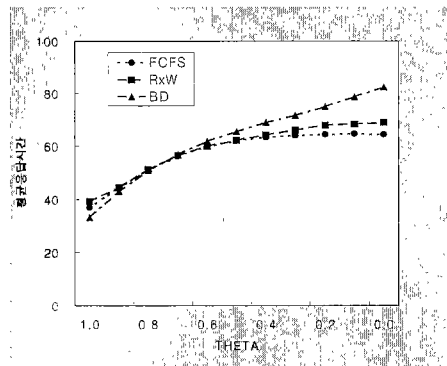


그림 5 요구패턴 변화 (Zipf'  $\theta$ )



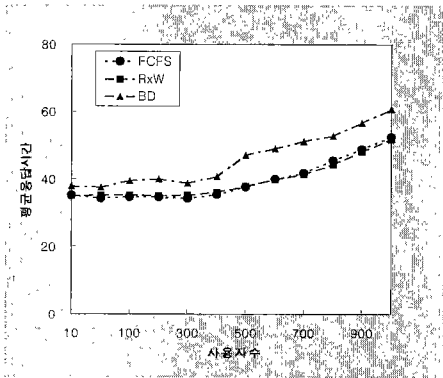


그림 6 요구 빈도 및 변화량

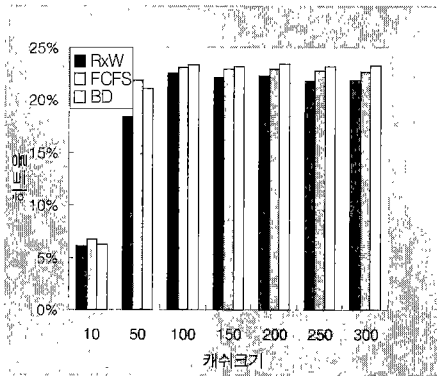


그림 7 캐시 크기 vs. 히트율

결과적으로 데이터 액세스 패턴이 불균등한 환경에서 사용자의 캐시는 일정만큼으로 효과적인 성능을 발휘할 수 있다. 따라서, 데이터 방송 서비스를 받는 사용자는 자신의 데이터 액세스 패턴에 따른 적절한 캐시 크기를 결정하면, 질의 수행성능을 향상시킬 수 있다.

### 5. 결론

본 논문에서는 기존 연구에서 간과되었던 데이터 방송 환경에서의 사용자 요구 변화를 고려한 시뮬레이션 환경을 구축하고, 동적으로 사용자 요구를 변화시켜 시뮬레이션 함으로서 실제 운용환경에 가까운 현실성을 부여하였다. 이는 기존 연구에서 제시되었던 방송 데이터의 스케줄링 방식들이 갖는 장점 및 단점을 인식하여, 이를 토대로 각 응용의 특성에 적합한 스케줄링 방식선택에 방향성을 제시하기 위함이다.

실제 시뮬레이션 환경 설정시 각 사용자의 단말에 제

약조건으로 캐시 크기 변화를 가정하였으며, 사용자의 동적 액세스 패턴의 변화를 측정하기 위한 방법으로 패턴변화 주기와 변화정도를 고려하였다. 이를 토대로 각 스케줄링의 환경적 특성에 따른 우수성 및 특징을 살펴 보았다. 실험 결과에 따르면, BD 방식은 요구 데이터가 한정적이며 데이터 요구변화가 심하지 않은 응용에 적합하며, RxW 방식은 요구패턴의 변화나 요구변화가 빈번히 일어나는 데이터 방송 환경에서 좋은 성능을 보이고 있음을 알 수 있었다.

향후 연구에서는 본 논문이 고려하지 않았던 방송 데이터 항목의 크기나 중요도에 따른 스케줄링 방법 및 데이터의 일관성을 고려한 시뮬레이션 운용환경을 구성에 관한 연구가 필요할 것이라 생각된다.

### 참고 문헌

- [1] G. Forman and J. Zahorian, "The Challenges of Mobile Computing," IEEE Computers, Vol.27, No.4, pp. 38-47, 1994.
- [2] Special section on Data Management Issues in Mobile Computing, ACM SIGMOD Record, Vol. 24, No. 4, pp. 3-39, 1995
- [3] R. Alonso and H. Korth, "Database System Issues in Nomadic Computing" Proc. ACM SIGMOD, pp. 388-392, 1993.
- [4] Daniel Barbara, "Mobile Computing and Database-A Survey," IEEE Transaction on Knowledge and Data Engineering, Vol. 11, No.1 pp. 108~117, 1999.
- [5] S. Acharyaya, R. Alonso, M. J Franklin and S. Zdonik. "Broadcast Disks: Data Management for Asymmetric Communications Environments" In Proc. ACM SIGMOD Intl Conf, on Management of Data, 1995.
- [6] S. Acharyaya, M Franklin , and S Zdonik, "Balancing Push and Pull for Data Broadcast," Proc. ACM SIGMOD Int'l Conf. Management of Data, Phoenix, Ariz, pp. 183-194, May 1997.
- [7] A. Datta, A. Celik, J. Kim, Debra E. Vander Meer, "Adaptive Broadcast Protocols to Support Power Conserving Retrieval by Mobile Users," Proc of the 13th IEEE International Conference on Data Engineering, April, 1997
- [8] D. Aksoy, M. Franklin "Scheduling for Large-scale On-demand Data Broadcasting" The IEEE INFORCOM Conference, San Francisco, CA 1998.
- [9] Nitin H. Vaidya, Sohail Hameed "Scheduling data broadcast in asymmetric communication environments," Wireless Networks 5, pp.171- 182, 1999.

- [10] E. Pitoura, G Samaras, "Data Management for Mobile Computing," Kluwer-Academic Publishers, 1998.
- [11] T. Imielinski and S. Viswanathan, B. R. Badrinath, "Energy-Efficient Indexing On Air," Proc. ACM SIGMOD, pp.25-36, 1994.
- [12] Qinglong Hu, Dik Lun Lee and Wang-Chien Lee, "Performance evaluation of a wireless hierarchical data dissemination system," Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking, pp. 163 - 173, 1999.
- [13] Donald E. Knuth. The Art of Computer Programming, volume 3. Addison-Wesley, Reading , Massachusetts, 2nd, 1981.
- [14] H.D. Schwerman. "CSIM: A C-based process oriented simulation language," Proc, 1986 Winter Simulation Conf, 1986.



오 상 수

1998년 중앙대학교 정보시스템학과 졸업(학사). 2000년 중앙대학교 대학원 정보시스템학과 졸업(석사). 2000년 ~ 현재 LG 전자 정보통신 중앙연구소 연구원. 관심분야는 이동 컴퓨팅, 데이터웨어하우스.

신 동 천

정보과학회논문지 : 데이터베이스

제 28 권 제 1 호 참조