정보학연구 제7권 제1호 2004년 3월(pp.57-67)
The Journal of Information Technology, Vol.7, No.1

RxW$^{\alpha}$: 풀 기반의 데이타 방송 시스템을 위한 새로운 방송 스케줄링 기법

권혁민*

요 약

데이타 방송은 방대한 규모의 클라이언트에게 데이타를 전파하기 위한 효과적인 기술로서 큰 관심을 끌고 있다. 데이타 방송 시스템의 주된 동기는 자신이 지원하는 클라이언트의 수는 시스템의 성능에 전혀 영향을 미치지 않고 임의로 증가될 수 있다는 것이다. 방송 시스템의 성능은 클라이언트 캐싱 전략과 데이타 방송 스케줄링 기법에 의해 크게 영향을 받는다. 본 논문은 후자의 문제를 취급하여 데이타 방송 시스템에 적합한 RxWα로 명명된 새로운 방송 스케줄링 기법을 제안한다. 그리고 모의 실험 모델을 통하여 RxWα 기법의 성능을 평가한다. 성능 평가 결과에 의하면 RxW$^{\alpha}$ 기법은 적당한 수준의 응답시간 성능을 보이면서도 응답시간의 분산 성능을 크게 개선할 수 있다.

Ⅰ. 서론

최근 들어, 인터넷의 폭발적인 인기와 함께 전 세계에서는 온라인 데이타 서비스에 대한 요구가 급격하게 증가하고 있고, 이에 따라 정보 제공(information-feed) 응용들이 큰 관심을 끌고 있는 실정이다. 정보 제공 응용의 예로는 날씨, 주식, 뉴스, 교통 등의 정보 서비스 응용들이 있는데, 이런 종류의 응용들은 정보를 생성하여 관리하고 제공하는 단일 또는 몇몇의 서버와 정보를 검색하고 이를 이용하는 다수의 클라이언트로 구성되는 정보소비(information- consumption) 모델의 형태를 띠고 있다[1].

방송(broadcast)에 기초한 데이타 전달 기법은 정보소비 모델 형태의 정보제공 응용에서 정보 전달을 위한 주요 기술로 인정을 받고 있다[2-4]. 방송 기술을 채택한 데이타 방송 시스템에서 방송 서버는 다수의 클라이언트에게 특정 데이타들을 지속적으로 전파하고 각 클라이언트는 자신이 원하는 데이타가 방송 채널에 나타나면 이를 검색한다. 데이타 방송 시스템의 성능은 그 시스템이 지원해야 하는 클라이언트의 수에 영향을 받지 않기 때문에 시스템의 확장성(scalability)이 매우 우수하다. 따라서 방송 시스템은 웹이나 정보제공 응용과 같이 방대한 규모의 클라이언트를 지원해야 하는 응용 분야에 매우 적합하다.

데이타 방송 시스템의 성능은 다음 두 가지의 핵심 기술에 의해 크게 영향을 받는다고 알려져 있다[1, 2, 5-9]: (1) 클라이언트 캐쉬 관리 기법과 (2) 데이타 방송 스케줄링(data broadcast scheduling) 기법. 클라이언트 캐쉬 관리 기법은 클라이언트 측에 적용되는 기술로서 클라이언트

* 세명대학교 소프트웨어학과 조교수

의 지역 메모리에 시스템이 허용하는 한도 내에서 특정 데이타들을 캐싱하여 방송 채널에서의 대기시간을 줄이기 위한 기술이다. 데이타 방송 스케줄링 기법은 서버측에 적용되는 기술로서 방송할 데이타를 선정하고, 이들의 방송 순서 및 방송 빈도(broadcast frequency)를 결정하는 문제를 다룬다.

데이타 방송 스케줄링 기법을 구현하는데는 기본적으로 두가지의 접근법이 있다: 풀 기반(pull-based) 시스템과 푸쉬 기반(push-based) 시스템. 푸쉬 기반의 방송 스케줄러를 채택하는 방송 서버는 사용자의 프로파일(user profile)에 의해 제공되는 정보를 바탕으로 클라이언트가 필요로 하는 데이타를 예상하고, 이 예상을 바탕으로 데이타의 방송 스케줄을 구성한다. 이 방식에서 클라이언트는 자신이 원하는 데이타를 서버에 요청하는 것이 아니라, 방송 채널을 감시하고 있다가 자신이 원하는 데이타가 방송되면 해당 데이타를 수신하여 액세스한다. 반면, 풀 기반의 환경에서는 클라이언트가 명시적으로 서버에게 자신이 필요로 하는 데이타를 요청한다. 이 요청들은 방송 서버의 서비스 큐에 모아지고 서버는 특정 스케줄링 알고리즘에 의해 다음에 방송할 데이타를 결정한다. 본 저자는 다른 연구에서[20] 클라이언트 캐쉬 관리 기법에 대한 연구를 한 바 있다. 본 연구에서는 풀 기반의 데이타 방송 방식을 채택한 시스템을 위한 방송 스케줄링 기법들에 대한 연구를 진행해 나갈 예정이다.

방송 스케줄러는 방송 시스템의 성능에 큰 영향을 미치기 때문에 이에 대한 많은 연구들이 수행되었다[2-8, 10, 11, 13-17]. 이 연구들은 대부분 평균 응답시간의 최소화에 맞추어 연구를 진행했다. 그러나 사용자의 관점에서는 평균 응답시간뿐만 아니라 평균 응답시간의 분산(variance) 성능도 매우 중요한 것으로 알려져 있다[12]. 이는 응답시간의 분산 성능은 방송시스템의 서비스 질에 대한 사용자의 느낌에 큰 영향을 주기 때문이다. 예를 들어, 어떤 데이타에 대한 응답시간이 각각 1, 1, 7분인 경우와 3, 3, 3분인 경우는 평균 응답시간은 동일하다. 그러나 사용자는 후자의 경우가 훨씬 서비스 질이 좋다고 느낀다는 것이다. 평균 응답시간의 분산 성능이 중요함에도 불구하고 그동안 응답시간의 분산 성능에 관한 연구는 별로 없었다. 비록 [12]에서 분산 성능에 관한 연구가 있었지만 이는 푸쉬 기반의 스케줄링 기법이기 때문에 동적으로 클라이언트의 액세스 패턴이 변화되거나 데이타의 액세스 확률을 미리 파악할 수 없는 환경에서는 적용이 불가능하다. 이와 같은 점을 고려하여 본 연구에서는 풀 기반의 방송 스케줄링 기법에 대한 평균 응답시간과 응답시간의 분산 성능에 대한 연구를 진행한다.

본 논문의 구성은 다음과 같다. 2 장에서 기존에 개발된 방송 스케줄링 기법에 대하여 기술하고, 3 장에서는 분산 성능을 고려한 새로운 알고리즘을 기술하고, 4장에서는 성능평가 모델을 기술한다. 그리고 5 장에서는 성능평가 결과를 분석하고, 마지막으로 6 장에서 결론을 맺는다.

II. 방송 스케줄링 기법

이 장에서는 본 논문의 성능 평가에서 사용된 기법을 중심으로 기존에 개발된 방송 스케줄링 기법에 대하여 살펴본다. 방송 스케줄러는 각 클라이언트들의 공통 관심사항을 적절하게 취합하여 방송할 데이타를 결정한다. 방송 스케줄러는 클라이언트의 평균 대기시간이 최소화되도록

스케줄링해야 하는데, 방송 스케줄러는 어떤 정보를 바탕으로 방송할 데이타를 선정하는 가에 따라 풀 기반(pull-based) 및 푸쉬 기반(push-based)의 기법으로 분류할 수 있다.

클라이언트 캐쉬가 존재하지 않을 경우에는 각 데이타를 액세스 확률의 루트 값에 비례하게 방송하는 것이 가장 이상적이라는 사실이 밝혀졌다[13]. 푸쉬 기반의 기법은 대부분 이 이론에 근거하여 개발되었는데, [2, 5]에서 연구된 *MBD* 모델과 [14]에서 연구된 *MAD*(mean aggregate delay) 기법과 [12, 15]에서 연구된 버켓을 이용한 스케줄링 기법들이 이 범주로 분류될 수 있다. [2, 5]의 연구는 데이타들을 주기적으로 전파하기 위하여 방송 채널을 저장 용량과 방송 빈도(broadcast frequency)가 다른 다수의 디스크로 구성하는 다중 방송 디스크(Multiple Broadcast Disks: MBD)로 모델링했다. *MBD* 모델에서는 각 데이타의 액세스 확률을 파악하여 비슷한 확률을 가진 데이타들을 그룹화하여 다수의 그룹으로 나누고 각 그룹을 각각 다른 디스크에 저장한다. 그리고 디스크에 저장된 데이타들의 액세스 확률을 고려하여, 각 디스크의 상대적인 방송 빈도를 결정한다. [14]에서 연구된 *MAD* 기법에서는 각 데이타 항목 i에 대하여 $\lambda_i^{\gamma} * w_i(n)$의 값을 계산하여 이 값이 가장 큰 데이타를 다음에 방송할 데이타로 선정한다. 여기서 λ_i는 데이타 항목 i의 요청율을 의미하며, $w_i(n)$은 그 데이타를 방송하고 나서 현재까지 흐른 시간을 의미한다. *MAD* 기법에서는 많은 실험을 거쳐 γ 값을 0.5로 설정할 때 가장 최적의 성능이 발휘된다는 사실을 밝혀냈다. [12, 15]에서는 각 데이타의 액세스 확률 및 그 데이타를 방송하고 나서 경과된 시간, 그리고 그 데이타의 크기를 바탕으로 방송할 데이타를 선정하는 스케줄링 기법을 제안했다. 기본적으로 푸쉬 기반의 방송 스케줄링 기법들은 각 데이타의 액세스 확률이나 요청율을 미리 파악할 수 있어야 한다는 제약이 있고, 동적으로 클라이언트의 액세스 패턴이 변할 경우에는 적용하기가 어렵다는 단점이 있다.

이와 같이 데이타의 액세스 확률을 미리 파악하기가 어렵거나 데이타의 액세스 패턴이 동적으로 변경되는 환경에서는 클라이언트에서 전송되는 데이타 요청 정보를 바탕으로 다음에 방송할 데이타를 선정하는 풀 기반의 스케줄링 기법을 적용하는 것이 바람직하다. 풀 기반으로 분류될 수 있는 스케줄링 기법으로는 *FCFS*(First Come First Served), *MRF*(Most Requests First), 그리고 *LWL*(Longest Wait First) 기법들이 있다[7, 8, 16]. 이들은 클라이언트들의 데이타 요청에 관한 정보를 서버의 서비스 큐에서 관리한다. *FCFS* 기법은 클라이언트가 요청한 순서대로 데이타를 방송한다. 이 기법은 *FIFO* 형태로 서버의 서비스 큐를 관리하는데, 불필요한 데이타 방송을 없애기 위하여 어떤 데이타 요청이 들어 왔을 때 이미 그 데이타가 서비스 큐의 엔트리에 존재할 경우에는 그 요청을 무시한다. *MRF* 기법은 서비스 큐에서 요청된 데이타 항목 번호와 요청 횟수를 관리한다. 그리고 이들 중에서 요청 횟수가 가장 많은 데이타를 선정하여 방송한다. *LWF* 기법은 서비스 큐에서 요청된 데이타들의 전체 대기시간을 관리한다. 전체 대기시간이란 어떤 특정 데이타의 각 요청의 대기시간의 합을 의미한다. *LWF* 기법은 전체 대기시간이 가장 긴 페이지를 선정하여 방송한다. 이 기법들 중에서 *LWF* 기법이 가장 우수한 성능을 발휘하는 것으로 알려져 있다 [7, 8, 16]. 그렇지만 *LWF* 기법은 전체 대기시간이 시간이 경과함에 따라 지속적으로 변하기 때문에 스케줄링할 때 실행 부담이 있다는 단점이 있다.

[7, 8]에서는 이 실행 부담의 단점을 완화하기 위하여 *LWF* 기법의 발견적 해결 알고리즘(heuristic algorithm)으로 볼 수 있는 *RxW*라고 명명된 알고리즘을 개발했다. *RxW* 기법은 서비스 큐에서 클라이언트에서 요청된 데이타들에 대해서 그들의 요청 횟수 및 그 요청의 처음 도착시간 정보를 유지 관리한다. 그리고 데이타의 요청 횟수 R과 그 데이타가 처음 요청된 시간에서 현재까지 흐른 시간 W의 곱으로 계산되는 *RxW* 값에 근거하여 다음에 방송할 데이타를 선정한다. 가장 큰 *RxW* 값을 가진 데이타 항목이 다음에 방송할 데이타로 선정된다. *RxW* 기법에서는 자주 요청되었거나 요청된 시간이 오래 경과된 데이타가 다음에 방송될 데이타로 선정될 가능성이 높은 기법이다.

Ⅲ. 분산 성능을 고려한 스케줄링 기법: RxW$^{\alpha}$

데이타 방송 스케줄링 기법은 방송 시스템의 성능에 큰 영향을 미치기 때문에 이 분야에서 많은 연구들이 있어 왔지만, 이들은 대부분 시스템의 평균 응답시간을 최소화하는 방향으로 연구가 수행되었다. [12]에서는 방송 스케줄링 기법과 사용자 응답시간의 평균 및 분산과의 연관 관계를 분석하고 각각을 최소화할 수 있는 조건식을 유도하였다. 일반적으로 응답시간의 평균의 최소화와 분산의 최소화는 동시에 이룰 수 없는 목표이지만, 이들은 평균과 분산을 서로 타협하여 두가지 모두를 적정하게 줄일 수 있는 알파 알고리즘을 제시하였다. 알파 알고리즘은 방송 데이타를 결정할 때 각 데이타 항목 i에 대하여 다음 식 1에 의해 표현되는 F_i 값을 계산하여 가장 큰 F_i 값을 갖는 데이타를 선정하여 방송한다.

$$F_i = (Q - R_i)^{\alpha} * p_i / l_i \qquad \text{(식 1)}$$

이 수식에서 Q는 현재시간을 의미하며, R_i는 데이타 항목 i가 방송된 마지막 시간을 의미한다. p_i는 데이타 항목 i의 액세스 확률을 의미하며, l_i는 그 데이타의 크기를 의미한다. 즉, 이 기법에서는 방송된 지 오래된 페이지일수록, 액세스 확률이 높을수록, 그리고 데이타의 크기가 작을수록 우선권을 가지는 기법이다. [12]에서는 α 값을 2로 설정할 경우에 가장 최적의 응답시간 성능을 보이며, 이 값이 3일 경우에 최적의 분산 성능을 보인다는 사실을 수식을 통해 증명했고 실험을 통해 이를 입증했다. 그렇지만 이 기법은 푸쉬 기반의 스케줄링 기법으로서 각 데이타의 액세스 확률을 미리 파악할 수 없을 경우에는 적용하기가 어렵다는 단점이 있다.

풀 기반의 스케줄링 기법에서 클라이언트가 이용할 수 있는 주된 정보는 데이타에 대한 요청 횟수와 그 요청이 도착한 시간이다. 물론 이러한 정보를 바탕으로 스케줄링에 반영할 수 있는 다른 유용한 정보를 유도할 수도 있겠지만 이는 본 논문의 범위를 벗어난다. 본 논문에서 제안하는 기법은 이 두 가지의 정보만을 이용하는 기법으로서 *RxW* 기법에 기초하고 있다. *RxW* 기법은 R과 W에 같은 가중치를 주고 있는데 비하여, 본 논문에서는 식 1의 개념에 착안하여 R과 W에 서로 다른 가중치를 주는 *RxW*$^{\alpha}$로 명명된 기법을 제안한다. *RxW*$^{\alpha}$ 기법은 다음 수식으로 표현되는 *RxW*$^{\alpha}$ 값에 기초하여 가장 큰 *RxW*$^{\alpha}$ 값을 가지는 데이타를 다음에 방송할 데이타로 선정한다.

$$RxW^{\alpha} = R * (Q - first_arrival_time)^{\alpha} \qquad \text{(식 2)}$$

여기서 R은 해당 데이타의 요청 횟수를 의미

하며, Q는 현재 시간을 의미한다. *first_arrival_time*은 해당 데이타가 방송되고 나서 그 데이타가 다시 요청될 때 그 요청시간으로 설정되는 값이다. R은 해당 데이타가 방송되고 나면 다시 0으로 초기화된다. α 값은 성능 튜닝을 위하여 조정될 수 있는 변수로서 본 논문에서는 α 값을 변화시키면서 이 값이 응답시간 및 응답시간의 분산 성능에 미치는 영향을 파악할 것이다.

Ⅳ. 성능 평가 모델

이 장에서는 방송 스케줄링 기법의 성능을 파악하기 위하여 성능 평가 모델을 제시한다. 성능 평가 비교 대상으로는 푸쉬 기반의 대표적인 스케줄링 기법인 *FCFS*와 *RxW* 기법을 선정하였다. 그리고 본 논문에서는 *MRF*를 약간 변형한 *MRFL*로 명명된 기법을 선정하였다. *MRFL* 기법에서는 요청 횟수가 가장 많은 데이타가 방송 데이타로 선정되는데, 만일 요청 횟수가 최대인 데이타 항목이 다수 개일 경우에는 처음 요청이 가장 오래된 데이타를 방송 데이타로 선정하는 기법이다. 성능 평가 모델은 MCC에서 개발한 CSIM[18] 언어를 이용하여 구현하였다.

본 논문은 각 데이타 항목은 길이가 일정한 고정 길이의 페이지라고 가정한다. 본 연구의 성능 평가 모델에서 방송 채널은 일정한 방송 대역폭을 갖는 서버로 모델링되었는데, 방송 채널의 대역폭은 [2, 5, 7-9] 연구에서 사용된 tick이라는 논리적 시간 단위를 이용하여 표현한다. tick은 방송 채널에서 하나의 데이타 항목을 방송하는데 걸리는 시간을 의미한다. 서버의 방송 스케줄러는 서버의 큐에 취합된 정보를 바탕으로 성능 평가 대상이 되는 각 기법의 알고리즘에 따라 방송할 데이타를 선정하는데, 스케줄러의 스케줄링 부담(overhead)은 다른 연구들과[2, 5, 7-9] 마찬가지로 고려하지 않는다. 본 논문은 방대한 규모의 클라이언트들을 목표로 하고 있기 때문에, 이들 클라이언트들을 구현하는데 폐쇠 큐잉 모델(closed queuing model)을 적용하는 것은 현실적으로 불가능하다. 따라서 본 논문은 개방(open) 큐잉 모델을 사용하는데, 클라이언트들은 이들의 데이타 요청 시간간의 간격이 지수 분포(exponential distribution) 형태를 갖는 단일 요청 스트림으로 모델링되었다. 클라이언트에서 서버로 보내는 데이타 요청 정보는 그 데이타의 양이 크지 않기 때문에 그 비용을 고려하지 않는다.

본 논문에서 사용된 입력변수들은 표 1에 제시되어 있는데, 이들의 설정 값은 대부분 [2, 5, 7-9]의 연구를 바탕으로 하고 있다.

〈표 1〉 성능 평가 모델의 입력 변수

입력변수	의미	설정
DbSize	데이타베이스의 크기	10000 pages
λ	요청의 평균 도착률 (지수 분포)	1 ~ 512 requests/tick
RegionSize	zipf 분포에서 동일한 액세스 확률을 갖는 페이지 수	50 pages
θ	zipf 분포의 θ 값	0.95

서버의 데이타베이스는 DbSize 수의 페이지로 구성되며, 서버는 이 페이지들 중에서 방송할 데이타를 선정한다. 클라이언트들은 데이타 요청들간의 도착간 시간(inter-arrival time)이 평균 λ 값을 갖는 지수 분포를 이루도록 데이타를 요청한다. 클라이언트들이 액세스하는 데이타는 불균등한(non-uniform) 액세스 형태를 모델링하기 위해 많이 사용되는 zipf 분포 부하 모델[5, 9]에 따라 선정된다. 본 논문의 zipf 부하

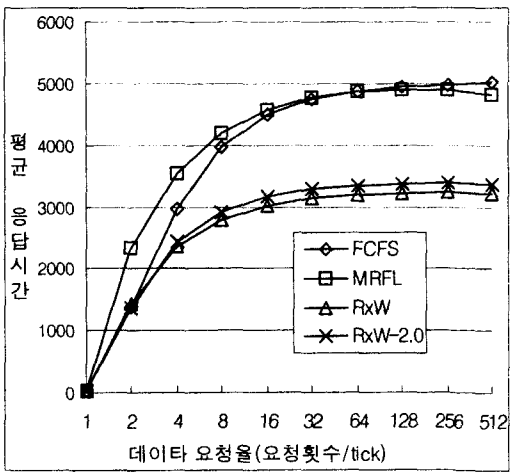
모델에서는 서버의 데이타베이스 내의 페이지를 RegionSize 만큼의 페이지로 나누어 다수의 영역으로 구분하는데, 각 영역은 서로 겹치지는 않는다. 따라서 표 1의 입력 변수 DbSize와 RegionSize를 살펴보면, DbSize 내의 페이지들이 200개의 영역으로 구분됨을 알 수 있다. 각 영역은 1부터 시작하여 200까지의 일련번호를 갖는 영역 번호로서 구분하는데, 이 영역에 zipf 분포가 적용된다. 즉, 각 영역의 액세스 확률은 $1/r^{\theta}$(r은 영역번호를 의미)에 비례한다. 이 zipf 부하 모델에서 클라이언트가 어떤 페이지를 액세스할 것인가를 결정하기 위해서는 우선 zipf 분포에 맞게 액세스할 영역을 먼저 선정한다. 그리고 나서, 그 영역에서 어떤 페이지를 액세스할 것인가는 균등(uniform) 분포에 따라 선정한다. 본 연구에서 θ 값은 0.95로 설정되어 있다.

V. 성능 결과 및 분석

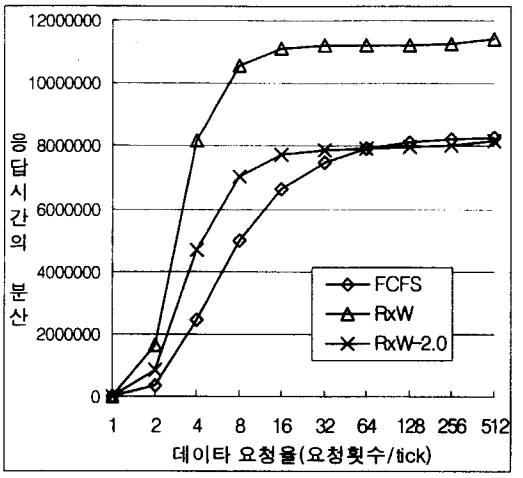
이 장에서는 4 장의 성능 평가 모델을 바탕으로 각 기법의 성능을 평가하여 그 결과를 제시하고 분석한다. 본 논문의 주요 성능 평가 지수는 평균 응답시간과 응답시간의 분산 성능이다. 실험은 복사 방법(replication approach)을[19] 사용하였는데, 이 절에서 제시된 결과 값은 5개의 서로 다른 임의 수를 사용하여 실시된 실험 결과의 평균값으로, 각 실험은 2 천만 개의 페이지를 액세스할 때까지 실시하였다.

5.1. 실험 1: 데이타 요청율 변화에 따른 성능

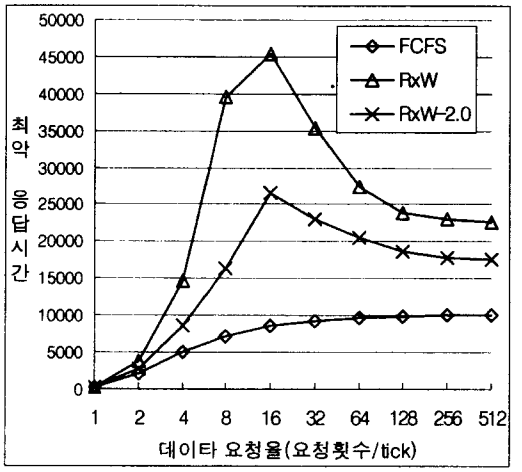
본 실험에서는 데이타의 요청율을 변화시키면서 각 기법의 성능추이를 살펴보았다. 클라이언트에서 하나의 데이타 페이지를 요청하고 난 이후 이를 액세스하는데 걸리는 평균 응답시간의 결과가 그림 1에 제시되어 있다. 이 그래프에서 가로축은 tick당 요청되는 데이타 항목의 수를 의미한다. RxW$^{\alpha}$ 기법의 대표격으로 RxW$^{2.0}$을 선정하였는데, 이는 α 값을 2.0으로 설정할 때 적당한 수준의 응답시간을 보이면서 우수한 분산 성능을 보이기 때문이다. α 값을 변화시키면서 살펴본 성능 결과는 5.2절에 제시되어 있다. 그래프 범례에서 RxW-2.0은 RxW$^{2.0}$ 기법의 성능을 의미한다. 그림 1의 결과를 살펴 보면, RxW와 RxW$^{2.0}$ 기법은 FCFS와 MRFL 기법에 비하여 대체로 30 ~ 40% 정도의 우수한 응답시간의 성능을 보인다. 이는 MRF와 FCFS 기법은 각각 데이타 요청횟수 또는 대기시간만을 고려하여 다음에 방송할 데이타를 선정하는데 비하여, RxW와 RxW$^{2.0}$ 기법은 이 두 요인에 적절한 가중치를 주어 방송할 데이타를 선정한다는 차이점에서 비롯된다. 요청횟수와 대기시간에 동일한 가중치를 주는 RxW 기법은 RxW$^{2.0}$ 기법에 비하여 4% 이내의 향상된 성능을 보인다.



(그림 1) 평균 응답시간 (tick)



(그림 2) 응답 시간의 분산 성능



(그림 3) 최악 응답시간 (tick)

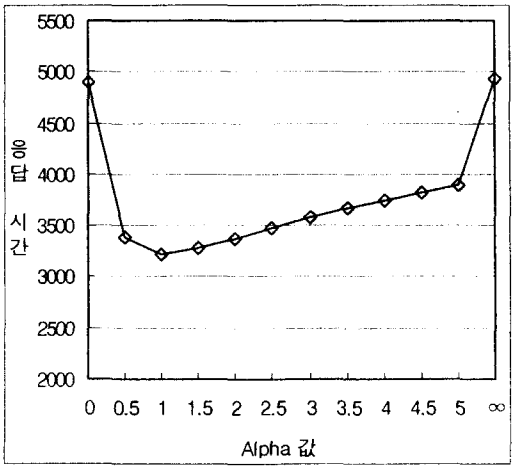
(그림 2)와 (그림 3)에는 각각 응답시간의 분산 성능과 최악 응답시간의 결과가 제시되어 있다. 응답시간의 분산 성능에 있어서는 대체로 FCFS, RxW$^{2.0}$, 그리고 RxW 기법순으로 우수한 성능을 보인다. FCFS 기법은 대기시간만을 고려하여 방송할 데이타를 선정하기 때문에 다른 기법에 비하여 우수한 분산 성능을 보인다. 그렇지만 데이타 요청율이 어느 이상이 되면 RxW$^{2.0}$과 거의 유사하거나 더 낮은 성능을 보인다. 이는 FCFS 기법의 평균 응답시간이 RxW$^{2.0}$에 비하여 훨씬 크기 때문이다. 응답시간이 클 경우에는 비록 공평하게 대기시간만을 고려하여 스케쥴링하더라도 분산 값은 커지게 마련이다. 요청 횟수보다 대기시간에 더 큰 가중치를 주는 RxW$^{2.0}$은 RxW 기법에 비하여 훨씬 우수한 분산 성능을 보인다. 데이타 요청율을 128로 설정할 경우에, RxW$^{2.0}$ 기법은 RxW에 비하여 40% 이상 우수한 분산 성능을 보인다. 최악 응답 시간의 성능에 있어서도 FCFS, RxW$^{2.0}$, RxW 기법 순으로 우수한 성능을 보인다.

방송 시스템의 서비스 질은 평균 응답시간에 의해서만 결정되는 것이 아니라, 응답시간과 응답시간의 분산 및 최악 응답시간의 성능에 의하여 종합적으로 결정된다. 위의 성능 결과에 의하면 RxW$^{2.0}$ 기법은 RxW 기법에 비하여 유사한 응답시간의 성능을 보이면서도 매우 우수한 분산 성능을 보인다. 그리고 FCFS 기법과 비교해서는 유사한 분산 성능을 보이면서도 매우 우수한 응답시간의 성능을 보인다. 따라서 RxW$^{2.0}$ 기법은 사용자가 느끼는 서비스의 질에 있어서 다른 기법에 비하여 매우 바람직한 성능 특성을 보인다고 관단된다. MRFL 기법은 다른 기법에 비하여 훨씬 큰 최악 응답시간 및 분산 결과를 보이기 때문에 그래프에서 생략되어 있다. 참고로 MRFL 기법은 RxW$^{2.0}$ 기법에 비하여 대략 7 ~ 8배 정도의 분산 값을 보이며, 3 ~ 5배 정도의 최악 응답시간을 보인다.

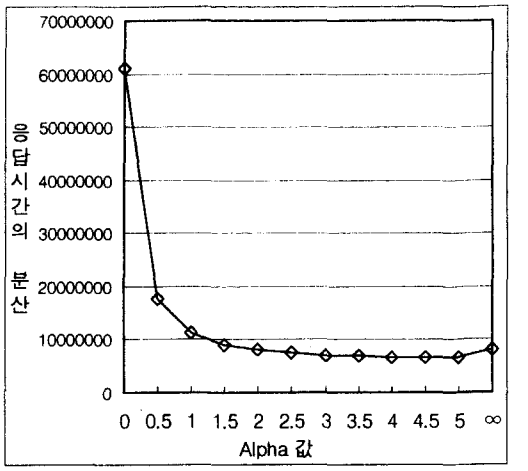
5.2. 실험 2: Alpha 값의 변화에 따른 성능

(그림 4)와 (그림 5)는 RxW$^{\alpha}$ 기법에서 α 값을 변화시키면서 살펴본 평균 응답시간과 응답시간의 분산 성능 결과이다. 이 실험에서 데이

타의 요청율은 tick당 128개의 데이타가 요청된다고 설정했다. RxW$^{\alpha}$ 기법은 α 값이 작을수록 MRFL 기법과 유사해 지며 α 값이 커질수록 FCFS 기법과 유사해지는 특징이 있다. 극단적으로, α 값을 0으로 설정할 경우에는 스케쥴링 시에 요청횟수만이 고려되므로 MRFL 기법과 동일해 지며, α 값을 ∞로 설정할 경우에는 대기시간에 의해서만 방송 데이타가 선정되므로 FCFS 기법과 동일해 진다. 그림 4에 의하면 α 값을 증가시키면, 초기에는 평균 응답시간의 성능은 증가하지만, α 값이 1 이상이 되면 응답시간의 성능이 저하되기 시작한다. 즉, 스케쥴링 시 요청횟수와 대기시간에 같은 가중치를 주는 것이 평균 응답시간의 성능에서는 가장 바람직하다. 반면, α 값이 증가하면 응답시간의 분산 성능은 급격하게 개선되는데, α 값이 3.0 이상이 되면 그 개선 정도가 미미해 진다. 즉, α 값이 어느 이상이 되면, 분산 성능은 개선되지 않고 평균 응답시간의 성능만을 저하시키는 특성을 보인다. 본 실험 결과에 의하면 α 값을 2로 설정하는 것이 대체로 만족할 만한 응답시간의 성능을 보이면서도 우수한 분산 성능을 보인다고 볼 수 있다.



(그림 4) 평균 응답시간(tick)



(그림 5) 응답 시간의 분산 성능

Ⅵ. 결론

방송 방식에 기초한 데이타 전달 기법은 정보 소비 모델 형태의 정보제공 응용 및 이동 컴퓨팅 환경의 응용에서 정보 전달을 위한 주요 기술로 인정을 받고 있다. 본 논문에서는 이와 같은 방송 기술을 채택한 데이타 방송 시스템의 방송 서버에 적용 가능한 스케쥴링 기법인 RxW$^{\alpha}$ 기법을 제안했다. 그리고 모의 실험을 통하여 RxW$^{\alpha}$ 기법의 성능과 RxW, FCFS 및 MRFL 기법의 성능을 비교하였다. RxW$^{\alpha}$ 기법에서 환경에 따라 α 값을 적절하게 설정하면, 평균 응답시간의 성능을 크게 희생시키지 않고 응답시간의 분산 성능 및 최악 응답시간의 성능을 크게 개선할 수 있다. 따라서 RxW$^{\alpha}$ 기법은 사용자가 느끼는 방송 시스템의 서비스 질을 크게 향상시킬 수 있는 장점이 있다.

본 논문의 미래 연구 과제로서 방송 스케쥴링 기법과 클라이언트 캐쉬 관리 기법이 조합된 연구를 진행해 나갈 것이다. 그리고 클라이언트에

서 서버로의 상향링크(uplink)가 존재하는 환경에서 클라이언트의 요청을 바탕으로 방송할 데이타를 동적으로 변화시키는 방송 스케줄링 기법에 대한 연구를 진행할 예정이다.

참고 문헌

[1] K. Stathatos(1999), *Air-caching: Adaptive hybrid data delivery,* Ph.D. Thesis, Maryland University.

[2] S. Acharya(1998), *Broadcast disks: dissemination-based data management for asymmetric communication environments,* Ph. D. thesis, Brown University.

[3] M. Franklin and S. Zdonik(1997), A framework for scalable dissemination-based systems, In the *International Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA 97),* pp. 94-105, Atlanta, GA.

[4] M. Franklin and S. Zdonik(1998), Data in your face: Push technology in perspective, *Proceeding of the 1998 ACM SIGMOD International Conference on Management of Data,* pp. 516-519, Seattle, WA.

[5] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik(1995), Broadcast disks: Data management for asymmetric communications environments, *Proceeding of the 1995 ACM SIGMOD International Conference on Management of Data,* pp. 199-210, San Jose, CA.

[6] S. Acharya, M. Franklin, and S. Zdonik (1997), Balancing push and pull for data broadcast, *Proceeding of the 1997 ACM SIGMOD International Conference on Management of Data,* pp. 183-194, Tucson, Arizona.

[7] D. Aksoy and M. Franklin(1998), Scheduling for large-scale on-demand data broadcasting, *Proceeding of IEEE INFOCOM,* pp. 651-659, San Francisco, CA.

[8] D. Aksoy and M. Franklin(1999), RxW: Scheduling approach for large-scale on-demand data broadcasting, *IEEE/ACM Transactions on Networking, Vol. 7,* No. 6.

[9] V. Liberatore(2000), Caching and scheduling for broadcast disk systems, In the *Second Workshop on Algorithm Engineering and Experiments ALENEX 00,* San Francisco, CA.

[10] K. Stathatos, N. Roussopoulos, and J. S. Baras(1997), Adaptive data broadcast in hybrid networks, Proceeding of *the 23th International Conference on Very Large Data Bases,* pp. 326-335, Athens, Greece.

[11] C. J. Su, L. Tassiulas, V. Tsotras(1999), Broadcast scheduling for information distribution, *Wireless Networks, Vol. 5,* pp. 137-147.

[12] Shu Jiang and Nitin H. Vaidya(2002), Response time in data broadcast systems: Mean, variance and tradeoff, *Mobile*

Neworks and Applications, Vol. 7, No. 4, pp. 37-47.

[13] M. Ammar and J. Wong(1985), The design of teletext broadcast cycles, *Performance Evaluation, 5.*

[14] Chi-Jiun Su, Leandros Tassiulas, and Vassilis J. Tsotras(1999), Broadcasting scheduling for information distribution, *Wireless Networks, Vol. 5,* pp. 137-147.

[15] Nitin H. Vaidya and Sohail Hameed(1999), Scheduling data broadcast in asymmetric communication environments, *Wireless Networks, Vol. 5,* pp. 171-182.

[16] H. D. Dykeman, M. H. Ammar, and J. W. Wong(1886), Scheduling algorithms for videotex system under broadcast delivery, In Proceedings of *ICC '86,* pp. 1847-1851.

[17] S. Acharya, S. Muthukrishnan(1998), Scheduling on-demand broadcasts new metrics and algorithms, Proc. *ACM/IEEE Int. Conf. MobiCom 1998,* Dallas.

[18] H. Schwetman(1992), *CSIM users' guide for use with CSIM revision 16,* Microelectronics and Computer Technology Corporation.

[19] A. M. Law and W. D. Kelton(1991), *Simulation modeling & analysis,* McGraw-Hill.

[20] 권혁민(2002), "데이타 방송 환경을 위한 클라이언트 캐쉬 관리 기법의 성능 평가", 「정보학연구」, 제5권 4호, pp. 9-21.

RxW$^{\alpha}$: A new broadcast scheduling scheme for pull-based data broadcast systems

Hyeok-Min Kwon*

Abstract

Data broadcast has attracted a lot of attention as an efficient way of disseminating data to very large client populations. The main motivation of data broadcast systems is that the number of clients that they serve can grow arbitrarily large without any effect on their performance. The performance of broadcast system depends mainly on client caching strategies and on data broadcast scheduling mechanisms. This paper addresses the latter issue and proposes a new broadcast scheduling scheme, named RxW$^{\alpha}$, that is suitable to data broadcast systems. This paper also evaluates the performance of RxW$^{\alpha}$ on the basis of a simulation model. The performance results indicate that RxW$^{\alpha}$ is capable of improving the performance in the variance of response time significantly, showing a reasonable performance in the response time.

Key words : broadcast system, broadcast scheduling

* Department of Software, Semyung University