

논문 2011-2-10

TLDP: 다중 방송 채널 환경을 위한 새로운 방송 스케줄링 기법

TLDP: A New Broadcast Scheduling Scheme for Multiple Broadcast-Channel Environments

권혁민*

Hyeokmin Kwon

요 약 방송 기반의 데이터 전파는 모바일 컴퓨팅 환경에서 보편적으로 받아들여지는 통신 방식이다. 그러나 많은 데이터를 방송해야 할 경우에 방송 채널의 순차성으로 인하여 원하는 데이터를 수신하기까지의 예상 지연시간이 증가한다. 이 대기시간을 줄이기 위하여 본 논문은 다중 채널을 통하여 데이터를 방송하는 문제를 연구한다. 기존의 기법들은 각 채널의 평형 방송을 가정하고 데이터들을 액세스 확률에 근거하여 분할하고 다중 채널에 할당한다. 만일 동일 채널에 할당된 데이터들도 그들의 액세스 확률에 근거하여 방송빈도를 다르게 한다면 성능은 더 향상될 것이다. 이와 같은 관점에서 본 논문은 동일 채널에 할당된 데이터들의 액세스 확률의 차이를 방송 스케줄에 반영할 수 있는 두 단계 동적프로그래밍 기법으로 명명된 새로운 방송 스케줄링 기법을 제안한다.

Abstract Broadcast-based data dissemination has become a widely accepted approach of communication in the mobile computing environment. However, with a large set of data items, the expected delay of receiving a desired data increases due to the sequential nature of the broadcast channel. With the objective of minimizing this wait time, this paper explores the problem of data broadcast over multiple channels. In traditional approaches, data items are partitioned based on their access probabilities and allocated on multiple channels, assuming flat data scheduling per channel. If the data items allocated on the same channel are broadcast in different frequencies based on their access probabilities, the performance will be enhanced further. In this respect, this paper proposes a new broadcast scheduling scheme named two level dynamic programming(TLDP) which can reflect a variation of access probabilities among data items allocated on the same channel.

Key Words : Broadcast schedule, data allocation, dynamic programming.

I. 서 론

최근 들어, 모바일(mobile) 컴퓨터와 무선 네트워크 기술의 비약적인 발전으로 인하여 모바일 컴퓨팅 환경에서 정보 서비스가 보편화 되고 있다. 이에 따라 모바일 클라이언트에게 효과적으로 데이터를 보내기 위한 기법

들이 큰 관심을 끌고 있는데, 방송(broadcast) 방식에 기초한 데이터 전달 기법은 모바일 컴퓨팅 환경에서 정보 전달을 위한 주요 기술로 인정받고 있다^[1-5].

방송 시스템에서 서버는 방송 채널을 통하여 데이터들을 순차적으로 전파하고, 각 클라이언트는 자신이 원하는 데이터가 방송 채널에 나타나면 이를 액세스한다. 방송 시스템에서 어떤 데이터의 방송은 그 데이터를 원하는 모든 클라이언트들의 요구를 충족시킬 수 있기 때문에 시스템의 확장성(scalability)이 매우 우수하다. 따

*정회원, 세종대학교 정보통신학부
접수일자: 2010.12.21, 수정일자: 2011.3.8
게재확정일자: 2011.4.15

라서 방송 시스템은 기상예보, 도로교통 정보, 주식시장 정보, 전자 뉴스와 같이 방대한 규모의 클라이언트를 지원해야 하는 응용 분야에 매우 적합하다. 뿐만 아니라, 방송 시스템은 클라이언트에서 서버로의 통신 대역폭이 제한적이어서 클라이언트가 자신의 요청을 서버로 전달하는데 비용이 많이 들거나 또는 불가능할 가능성이 많은 모바일 컴퓨팅 환경에 매우 적합하다.

방송 서버는 제한된 방송 대역폭을 통하여 모든 데이터들을 방송해야 하기 때문에 클라이언트가 원하는 데이터를 액세스하기 위해서는 원하지 않는 데이터가 지나가기를 기다려야 한다. 방송 시스템의 주요 성능지수는 평균 예상지연 (average expected delay: AED) 시간인데, 이는 클라이언트가 특정 데이터의 액세스를 원하는 시간부터 실제 이를 액세스하는데 걸린 평균 대기시간을 의미한다. 이 AED 성능은 데이터 방송 스케줄링(data broadcast scheduling) 기법에 크게 영향을 받는다고 알려져 있는데^[1-13], 방송 스케줄링 기법에서는 방송할 데이터를 선정하고, 이들의 방송 순서 및 방송 빈도(broadcast frequency)를 결정하는 문제를 다룬다.

방송 스케줄링 기법은 기본적으로 풀 기반(pull-based)과 푸시 기반(push-based) 기법으로 분류할 수 있다^[5, 11]. 풀 기반 기법에서 클라이언트는 명시적으로 서버에게 자신이 필요로 하는 데이터를 요청한다. 이 요청들은 서버의 서비스 큐에 모아지고 서버는 특정 스케줄링 알고리즘에 의해 다음에 방송할 데이터를 결정한다. 푸시 기반의 시스템에서 방송 서버는 각 데이터의 액세스 확률에 근거하여 데이터의 방송 스케줄을 구성한다. 이 방식에서 클라이언트는 자신이 원하는 데이터를 서버에 요청하는 것이 아니라, 방송 채널을 감시하고 있다가 자신이 원하는 데이터가 방송되면 해당 데이터를 수신하여 액세스한다. 본 논문은 클라이언트에서 서버로의 통신 대역폭이 매우 제한적이거나 불가능한 환경을 고려하여 푸시 기반의 방송 스케줄링 기법에 한하여 연구를 진행한다.

그 동안 단일 방송채널 환경에서 AED 시간을 줄이기 위한 연구들이^[1, 4, 7] 진행되었는데, 이 연구들은 각 데이터의 액세스 확률에 근거하여 자주 사용되는 데이터를 더 빈번하게 방송하는 방법을 제시하여 AED 성능을 개선하고자 하는 노력을 했다. 최근에는 다중 채널 환경을 위한 많은 방송 스케줄링 기법들이^[8-13] 제안되었는데, 이들은 대부분 데이터의 액세스 확률에 근거하여 전체 데

이터를 채널의 수만큼 다수 개의 그룹으로 분할하여 이를 각 채널에 배정한다. 그리고 각 채널에 할당된 데이터들은 동일한 빈도로 방송하는 방법을 채택했다. 만일 물리적 채널의 수가 적을 경우에는 한 채널에 할당되는 데이터의 수가 많아지게 되는데, 이 경우에는 동일 채널에 할당된 데이터들의 액세스 확률의 차이는 커질 수밖에 없다. 이와 같은 환경에서 같은 채널의 데이터들을 동일 빈도로 방송한다면 AED 성능은 심각하게 저하될 것이다. 만일 전체 데이터들을 다수 개의 채널에 할당한 후에, 동일 채널에 할당된 데이터들도 액세스 확률의 차이를 고려하여 그들의 방송 빈도를 다르게 한다면 AED 성능은 대폭 개선될 것이다.

본 논문은 이와 같은 점을 고려하여 기존에 제안된 동적 프로그래밍 알고리즘을^[11] 확장하고, 이를 두 단계로 적용하여 AED 성능을 대폭 개선할 수 있는 다중 채널 환경을 위한 새로운 방송 스케줄링 기법을 제안한다. 본 논문의 구성은 다음과 같다. 2 장에서 기존에 개발된 스케줄링 기법에 대하여 기술하고, 3 장에서는 본 논문이 제안하는 새로운 방송 스케줄링 기법을 기술한다. 4 장에서는 본 논문이 제안한 기법의 성능을 평가, 분석하고, 마지막으로 5 장에서 결론을 맺는다.

II. 관련 연구

이 장에서는 기존에 제안된 푸시 기반의 방송 스케줄링 기법에 대하여 기술한다. 평형(FLAT) 기법은^[4] 가장 단순한 기법으로 구현이 용이하다는 장점이 있다. 이 기법은 모든 데이터들을 동일한 빈도로 주기적으로 방송한다. 평형 기법은 각 데이터들의 액세스 확률이 비슷할 경우에는 어느 정도의 성능이 기대되지만, 액세스 확률이 심하게 편향(skew)되어 있을 경우에는 다른 기법들에 비하여 심각하게 성능이 저하된다.

다중 방송 디스크(Multiple Broadcast Disks: MBD)연구에서는^[1, 4] 데이터의 액세스 확률에 근거하여 그들의 방송빈도를 결정하기 위한 모델을 제시했다. 이 연구에서는 방송 채널을 저장 용량과 액세스 속도가 다른 다수의 디스크로 구성했다. MBD 모델은 비슷한 액세스 확률을 가진 데이터들을 그룹화하여 몇 개의 그룹으로 나누고 각 그룹을 각각 다른 디스크에 저장한다. 그리고 각 그룹의 액세스 확률에 근거하여 각 디스크의 상대적인

방송빈도를 결정하여 각 디스크에 있는 데이터들을 단일 방송 채널을 통하여 방송한다. MBD 모델에서 같은 디스크에 배정된 데이터들을 동일 채널에 배정하면 쉽게 다중 채널 환경으로 확장할 수 있다.

최근 들어 다중 방송채널 환경을 위한 많은 방송 스케줄링 기법들이 제안되었는데, 대표적인 기법으로는 BP(Bin-Packing)^[8], VF^K(Variable-Fanout with the constraint K)^[9], 동적 프로그래밍(Dynamic programming: DP)^[11], 그리고 제한된 동적프로그래밍(Restricted Dynamic programming: RDP) 기법들이^[12, 13] 있다. 단일 채널을 위한 스케줄링 기법은 방송할 데이터의 방송 빈도 및 그들의 방송 순서를 어떻게 결정할 것인가의 문제를 다루는데 비하여, 다중 채널을 위한 스케줄링 기법은 방송할 데이터를 어떻게 분할하여 채널에 할당할 것인가에 주안점을 두고 연구를 진행했다. 이 기법들은 모두 유사한 액세스 확률을 가진 데이터들을 동일 채널에 배정하기 위하여 각 데이터를 액세스 확률의 내림차순으로 정렬을 한 후에 채널에 데이터를 배정하며, 같은 채널에 배정된 데이터들을 평형방식으로 동일 빈도로 방송한다.

BP^[8] 기법은 각 채널에 배정되는 데이터들의 액세스 확률의 합이 동일하게 되도록 채널에 데이터들을 할당한다. 따라서 액세스 확률이 높은 데이터가 할당되는 채널에는 더 적은 수의 데이터가 배정되어 이들은 상대적으로 더 높은 빈도로 방송된다. BP 기법은 데이터의 액세스 확률을 고려하여 채널에 데이터를 배정하지만, 액세스 확률이 높은 데이터를 확률에 비하여 너무 빈번하게 방송하는 단점이 있다.

VF^K 기법은^[9] 가변적인 팬아웃(fanout) 트리를 구성하여 다중 채널에 데이터를 할당한다. 이 기법은 각 데이터를 트리의 루트(root) 노드의 자식 노드로 배치하여 초기 트리를 구성한다. 그리고 나서 루트의 자식 노드로서 중간 노드를 한 개 추가하고, 데이터들의 AED 시간이 최소화되도록 액세스 확률이 낮은 데이터들을 추가된 중간 노드의 자식노드로 이동한다. 즉 트리의 다음 레벨로 이동한다. 이와 같은 과정을 트리의 깊이가 채널의 수와 같아질 때까지 반복적으로 수행한다. 트리의 구성이 완료되면 트리의 동일 레벨에 있는 단말 노드의 데이터들을 동일 채널에 배정한다. VF^K 기법은 특정 환경에서는 거의 최적의 성능을 발휘하지만, 항상 최적의 성능을 내는 것은 아니다. 이 기법은 채널의 수가 2의 지수승이 아

닐 경우에는 각 채널의 성능이 균형되지 못하여 예상 지연시간이 증가되는 단점이 있다. 그리고 채널에 데이터를 배정하기 위하여 $O(KN^2 \log K)$ 의 시간복잡도(time complexity)로 보이므로 이를 위한 실행 부담이 크다는 단점이 있다. 여기서 K는 채널의 수, N은 전체 데이터의 수를 의미한다.

Hsu et al.^[10]은 다중 방송채널 환경에서 이론적으로 최적의 AED 성능과 관련된 수식을 유도했다. Yee et al.^[11]은 이 연구를 바탕으로 동적 프로그래밍(Dynamic Programming: DP)에 기초한 스케줄링 기법을 제안했다. DP 기법은 각 채널의 데이터가 평형 방송된다는 가정하에 AED 시간이 최소화되도록 전체 데이터들을 채널 수만큼 K 개의 그룹으로 분할하여 각 채널에 배정한다. DP 기법은 가능한 모든 분할 방법을 고려하여 일일이 AED 성능을 계산해야 하기 때문에 $O(KN^2)$ 의 시간복잡도를 보여 비교적 실행부담이 큰 편인데, Yee et al.^[11]의 연구에서는 이를 줄이기 위한 Greedy 알고리즘도 함께 제안했다. Wang과 Chen의 연구에서도^[12, 13] DP의 수행시간 측면의 단점을 개선하기 위하여 이론적으로 최적의 AED 성능을 추정하여 데이터를 분할하는 RDP 기법을 제안했다. RDP 기법은 채널에 데이터를 배정하기 위하여 $O(N \log K)$ 의 시간복잡도를 보이며 거의 최적에 가깝게 데이터를 분할한다.

III. 새로운 방송 스케줄링 기법

기준에 제안된 다중 채널을 위한 대부분의 스케줄링 기법들은^[8, 9, 11-13] 같은 채널에 배정된 데이터들을 동일한 빈도로 방송하기 때문에 채널 수가 적을 경우에는 AED 성능이 심각하게 저하되는 단점이 있다. 이 장에서는 이 단점을 개선하기 위하여 동일 채널의 데이터들도 액세스 확률을 고려하여 그들의 방송 빈도를 다르게 할 수 있는 Two Level Dynamic Programming(TLDP)으로 명명된 새로운 스케줄링 기법에 대하여 기술한다.

1. 시스템 모델

TLDP 기법의 설명을 위해 본 논문은 다음과 같은 용어를 사용하며, 다음의 가정을 하고 있다.

- a) 전체 데이터는 동일 크기의 N 개의 데이터로 구성되어 이를 $d_q(1 \leq q \leq N)$ 로 표현하며, d_q 의 액세스

- 확률을 p_q 로 표현한다. 그리고 전체 데이터들은 p_q 의 내림차순으로 정렬되어 있다고 가정한다.
- b) 방송 서버에는 동일 대역폭을 갖는 K 개의 방송 채널이 존재하는데, 이를 $CH_c(c=1\sim K)$ 로 표현한다. 본 논문은 시간 단위로서 방송 채널에서 한 데이터를 방송하는데 걸리는 시간을 의미하는 tick이라는 논리적인 시간을 사용한다.
 - c) 각 채널은 큐 형태로 관리되는 B 개의 버킷을 운영하는데, CH_c 의 버킷을 $bucket_{c,b}(b=1\sim B)$ 로 표현한다. 각 버킷에는 적절히 데이터가 배정되는데, 각 채널은 자신의 버킷에 할당된 데이터들을 방송한다.
 - d) CH_c 에서 방송되는 데이터 집합을 $G_c(1\leq c\leq K)$ 로 표현하며, $bucket_{c,b}$ 에 배정된 데이터의 집합을 $G_{c,b}$ 로 표현한다. $G_c = \sum_{b=1}^B G_{c,b}$ 이며, 임의의 데이터는 하나의 $G_{c,b}$ 에만 포함될 수 있다.
 - e) d_i 를 액세스하는데 걸리는 지연시간을 w_i 로 표현하면, 전체 데이터들의 AED는 $\sum_{i=1}^N (w_i \times p_i)$ 로 정의된다. 본 논문은 특정 버킷 $bucket_{c,b}$ 에 할당된 데이터들의 AED를 $AED_{c,b}$ 로 표현하는데, 이는 $\sum_{d_i \in G_{c,b}} (w_i \times p_i)$ 로 정의된다.

2. 기본 개념

TLDP 기법은 DP를^[11] 확장한 EDP(extended dynamic programming) 알고리즘(3.3 절에서 설명 예정)을 이용하여 전체 데이터들을 $G_1\sim G_K$ 까지 K 개의 그룹으로 분할하여 각 채널에서 방송할 데이터들을 결정한다. 그리고 각 G_c 를 다시 EDP 알고리즘을 이용하여 버킷의 수만큼 $G_{c,1}\sim G_{c,B}$ 로 분할하여 이를 순서대로 $bucket_{c,1}\sim bucket_{c,B}$ 에 할당한다.

1. choose a bucket $bucket_{c,b}$ such that $b=(b \text{ mod } B) + 1$
2. choose a data item d_i from the front of $bucket_{c,b}$
3. broadcast d_i
4. Dequeue item d_i from the front of $bucket_{c,b}$
5. Enqueue d_i at the rear of $bucket_{c,b}$

그림 1. CH_c 의 방송 스케줄링 알고리즘
Fig. 1. Broadcast scheduling algorithm of CH_c .

각 채널의 방송 스케줄러는 방송할 데이터 선정이 필

요할 때마다 그림 1과 같이 우선 round-robin 방식으로 버킷을 결정한 후, 그 버킷의 맨 앞에 있는 데이터를 선정하여 방송하고, 그 데이터를 해당 버킷의 맨 뒤로 이동한다. 여기서 b 는 버킷 번호로서 0으로 초기화되어 있다고 가정한다. 각 버킷은 순서대로 공평하게 선택되기 때문에 동일 버킷에 배정된 데이터들은 해당 채널 대역폭의 $1/B$ 을 사용하며 동일 빈도로 방송된다.

3. 확장된 동적 프로그래밍 알고리즘

다수 개의 채널에 데이터를 배정한 후, 이를 다시 다수 개의 버킷에 할당하는 문제는 그림 2에 보는 것처럼 데이터를 분할하는 문제로 볼 수 있다. 우선 데이터들을 $G_1\sim G_K$ 집합으로 분할하고, 각 G_c 를 다시 버킷의 수만큼 $G_{c,1}\sim G_{c,B}$ 로 분할한다. 본 논문은 채널에 데이터를 분배하기 위해 분할되는 지점을 $ccut_c(c=1\sim K-1)$ 로 표현하며, G_c 를 $bucket_{c,b}(b=1\sim B)$ 에 할당하기 위하여 분할되는 지점을 $bcut_{c,b}(b=1\sim B-1)$ 로 표현한다. 알고리즘 기술의 편의상 분할점(cut point)을 각 데이터 그룹의 마지막 데이터의 인덱스로 표현하자. 예를 들어 그림 2에서 $ccut_1$ 은 3이고 $bcut_{c,1}$ 은 p 이다.

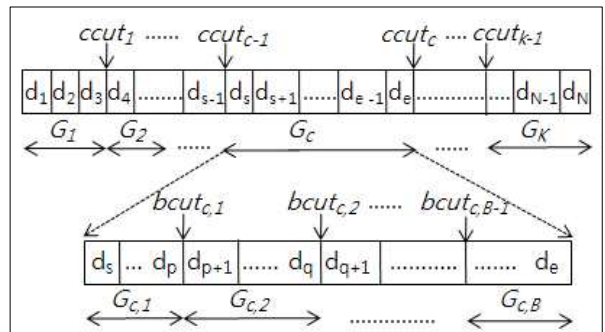


그림 2. N 개의 데이터를 $K \times B$ 버킷에 할당하기 위한 데이터 분할 문제

Fig. 2. Data partition problem for allocating N datum on $K \times B$ buckets

TLDP에서 전체 데이터를 $G_1\sim G_K$ 그룹으로 분할하는 개념과 각 G_c 를 $G_{c,1}\sim G_{c,B}$ 로 분할하는 개념이 동일하므로, 본 논문은 G_c 를 분할하는 방법을 중심으로 알고리즘을 설명한다. 이를 위해 우선 전체 데이터들이 $G_1\sim G_K$ 그룹으로 분할되어 있다고 가정하자. 각 G_c 를 B 개의 버킷에 최적으로 배정하기 위하여 각 분할점 $bcut_{c,b}(b=1\sim B-1)$ 를 구하는 EDP 알고리즘이 그림 3에 제시되어 있다. 이 알고리즘 설명을 위해 그림 2와 같이 특정 채널

CH_c 에 d_s 에서 d_e 까지 배정되어 있다고 가정하자. CH_c 의 특정 버킷 $bucket_{c,b}$ 에 d_i 에서 d_j 까지 배정되어 있을 경우 이 버킷에 배정된 데이터들의 $AED_{c,b}$ 값의 $1/B$ 를 $C_{i,j}$ 로 표현하자. 이 데이터들은 CH_c 대역폭의 $1/B$ 를 사용하므로 B tick당 한 개씩 주기적으로 방송된다. 그리고 어떤 데이터를 액세스하려면 평균적으로 그 버킷의 방송주기의 반을 기다려야 하므로 이들의 평균 지연시간은 $B \times (j-i+1)/2$ 로 계산된다. 따라서 $AED_{c,b}$ 는 $B \times ((j-i+1)/2) \sum_{q=i}^j p_q$ 가 되며 $C_{i,j}$ 는 $((j-i+1)/2) \sum_{q=i}^j p_q$ 가 된다. B 버킷 중 앞부터 k 개의 버킷만을 고려하여, 이 k 개의 버킷에 d_s 에서 d_j 까지 할당할 때의 최적 AED 값의 $1/B$ 를 $opt_left_{k,s,j}$ 로 표현하자. $opt_left_{1,s,j}$ 는 한 개의 버킷만 고려하면 되므로 $((j-s+1)/2) \sum_{q=s}^j p_q$ ($s \leq j \leq e$)로 초기화된다. 이 값과 $C_{i,j}$ 에 근거하여 가능한 모든 분할점들을 고려하여 $opt_left_{2,s,j}$ 를 구할 수 있고, 순차적으로 $opt_left_{B,s,j}$ 까지 구할 수 있는데 이를 일반화 하면 식 (1)과 같이 표현된다.

$$opt_left_{k,s,j} = \min \{opt_left_{k-1,s,r} + C_{r+1,j}\} \quad (1)$$

where $s+k-2 \leq r \leq j-1$

```

procedure EDP(s, e, count)
// s: start data index, e: end data index
// count: count of Data Partition Groups
for all cut points j from s to e
     $opt\_left_{1,s,j} = \frac{j-s+1}{2} \sum_{q=s}^j p_q$ 
end for
for all stages k from 2 to count
for all cut points j from s+k-1 to e
     $opt\_left_{k,s,j} = \min \{opt\_left_{k-1,s,r} + C_{r+1,j}\}$ 
    where  $s+k-2 \leq r \leq j-1$ 
     $lastp_{k,s,j}$  = the corresponding r such that
     $opt\_left_{k,s,j}$  is minimized
end for
end for
end procedure
    
```

그림 3. 확장된 동적 프로그래밍 알고리즘
Fig. 3. Extended dynamic programming algorithm

그림 3에서 $lastp_{k,s,j}$ 는 k 개의 버킷에 $d_s \sim d_j$ 를 최적으로 배정하기 위한 마지막 분할점을 의미한다. B 개의 버킷에 $d_s \sim d_e$ 데이터를 최적으로 할당하기 위한 마지막 분

할점 $lastp_{B,s,e}$ 는 그림 2의 $bcut_{c,B-1}$ 에 대응되는데, 이는 $opt_left_{B-1,s,j}$ 와 $C_{i,j}$ 에 기초하여 $opt_left_{B,s,e}$ 를 구하는 과정에서 결정된다(식 (1) 참조). 이 값이 결정되면, 마지막 버킷 $bucket_{c,B}$ 에는 $d_{bcut_{c,B-1}+1} \sim d_e$ 까지의 데이터들이 배정된다. 아직 버킷에 배정되지 않은 데이터 $d_s \sim d_{bcut_{c,B-1}}$ 를 $(B-1)$ 개의 버킷에 최적으로 배정하기 위한 마지막 분할점은 $opt_left_{B-1,s,bcut_{c,B-1}}$ 를 결정짓는 $lastp_{B-1,s,bcut_{c,B-1}}$ (이는 $bcut_{c,B-2}$ 에 대응)로 결정된다. 이와 같이 마지막 분할점이 결정되고 나면 이 분할점에 기초하여 이전의 분할점들이 역순으로 순차적으로 결정된다.

다음으로 전체 데이터를 $G_1 \sim G_K$ 그룹으로 분할되는 문제를 고려하자. DP는^[11] 각 채널의 데이터들이 평형방송된다는 가정하에 최적으로 데이터를 분할하여 각 채널에 배정한다. 본 논문은 각 채널의 데이터가 평형방송되는 것은 아니지만, DP와 동일한 가정하에 채널에 데이터를 배정한다. 이와 같은 가정을 한다면, 전체 데이터를 $G_1 \sim G_K$ 로 분할하는 문제는 EDP 알고리즘에서 s 는 1, e 는 N 으로 count는 K 로 고정되는 특수한 경우에 해당한다. 다만 이 경우에 $C_{i,j}$ 정의는 특정 채널에 $d_i \sim d_j$ 까지 배정되어 평형방송될 때의 AED 로 변경되는데, 이 값은 $C_{i,j} = ((j-i+1)/2) \sum_{q=i}^j p_q$ 로 계산된다. $opt_left_{k,1,j}$ 의 정의도 k 개의 채널에 $d_1 \sim d_j$ 까지 배정되어 평형방송될 때의 최적 AED 로 변경되며 $opt_left_{1,1,j}$ 는 $(j/2) \sum_{q=1}^j p_q$ ($1 \leq j \leq N$)로 초기화된다. 이와 같이 용어의 정의만을 변경하면 동일한 수식으로 계산되므로 전체 데이터를 $G_1 \sim G_K$ 그룹으로 분할할 때도 EDP 알고리즘을 그대로 사용하는 것이 가능하다.

4. 다중 버킷에 데이터를 할당하는 알고리즘

그림 4는 EDP를 사용하여 각 버킷에 데이터를 배정하는 알고리즘이다. 우선 EDP에서 $opt_left_{k,s,j}$ 를 구할 때 필요한 $C_{i,j}$ 를 빠르게 계산하기 위하여 액세스 확률의 부분합(partial sum)을 미리 계산해 놓는다. 확률의 부분합은 ps_i 로 표현하는데, 이는 $ps_i = \sum_{q=1}^i p_q$ 로 정의한다. 이 값이 미리 계산되어 있으면 $C_{i,j} = ((j-i+1)/2) \times (ps_j - ps_{i-1})$ 이 되어 $C_{i,j}$ 를 빠르게 계산할 수 있다. 그림 2에서 보는 것과 같이 $ccut_c$ 는 c 의 범위가 $1 \sim K-1$ 만이 의미가 있는데 알고리즘 구현 편의를 위하여 $ccut_0=0$,

$ccut_K=N$ 으로 설정한다. 그리고 전체 데이터를 K 개의 그룹으로 나누기 위하여 EDP를 호출하여 각 분할점을 구하고 $ccut_c(c=1\sim K-1)$ 를 설정한다. $bcut_{c,b}$ 도 b의 범위가 $1\sim B-1$ 만이 의미가 있는데 구현 편의를 위해 $bcut_{c,0}$ 는 $ccut_{c-1}$ (이전 채널의 마지막 데이터)로, $bcut_{c,B}$ 는 $ccut_c$ (현재 채널의 마지막 데이터)로 설정한다. 그리고 각 G_c 를 $G_{c,1}\sim G_{c,B}$ 로 분할하기 위하여 EDP를 호출하여 분할점을 구해 각 $bcut_{c,b}$ ($b=1\sim B-1$)를 설정하고 각 버킷에 데이터를 할당한다.

```

// 채널의 수: K, 버킷의 수: B
// precompute partial sum of p_i
ps_1 = p_1
for each data i from 2 to N
    ps_i = ps_{i-1} + p_i
end for
// partition all data into G_1 ~ G_K
ccut_0 = 0, ccut_K = N
call procedure EDP(1, N, K)
for each channel c from K to 2
    ccut_{c-1} = lastp_{c,1,ccut_c}
end For
//partition each G_c into G_{c,1} ~ G_{c,B}
for each channel c from 1 to K
    bcut_{c,0} = ccut_{c-1}, bcut_{c,B} = ccut_c
    call procedure EDP(ccut_{c-1}+1, ccut_c, B)
    for each bucket b of CH_c from B to 2
        bcut_{c,b-1} = lastp_{b,ccut_{c-1}+1,bcut_{c,b}}
        allocate data d_{bcut_{c,b-1}+1} ~ d_{bcut_{c,b}} into bucket_{c,b}
    end For
    allocate remainder(d_{bcut_{c,0}+1} ~ d_{bcut_{c,1}}) into bucket_{c,1}
end For
    
```

그림 4. 다중 버킷에 데이터를 할당하는 알고리즘
Fig. 4. Data allocation algorithm on multiple buckets

5. 알고리즘 실행속도 및 AED 성능에 관한 고찰

TLDP에서 각 채널의 스케줄러가 방송 데이터를 결정할 때 버킷에 데이터가 배정되어 있으면 $O(1)$ 의 시간복잡도(time complexity)를 보인다. 따라서 TLDP의 실행속도 측면에서는 EDP 알고리즘을 이용해 데이터를 분할하여 버킷에 배정하는 그림 4의 수행시간에 대한 평가만이 의미가 있다. EDP에서 $opt_left_{k,s,j} = \min_k opt_left_{k,s,j} + C_{r+1,j}$ 를 계산할 때 모든 r 값을 고려해야 하므로 데이터 개수에 비례하는 실행시간을 필요로 한다. $opt_left_{k,s,j}$ 의

계산이 데이터 분할그룹의 수 그리고 데이터 수만큼 이 중 반복문을 통하여 계산되므로 EDP는 $O(\text{분할개수} \times \text{데이터의수}^2)$ 의 시간복잡도를 보인다. 그림 4의 실행시간은 대부분 EDP를 호출해 실행하는 부분에 의해 결정되는데, 우선 전체 데이터를 $G_1 \sim G_K$ 로 분할하는 부분에서 $O(KN^2)$ 의 시간복잡도를 보인다. 그리고 각 $G_c(c=1\sim K)$ 를 $G_{c,1} \sim G_{c,B}$ 로 분할하기 위하여 EDP가 K번 호출되는데 이 부분에서 $O(\sum_{i=1}^K BN_i^2)$ 의 시간복잡도를 보인다. 여기서 N_i 는 G_i 의 데이터 개수를 의미한다. 그런데 $N = \sum_{i=1}^K N_i$ 이므로 $N^2 \geq \sum_{i=1}^K N_i^2$ 이 성립한다. 따라서 그림 4는 최악의 경우에도 $O((K+B)N^2)$ 의 시간복잡도를 보인다.

기존의 DP 알고리즘이 $O(KN^2)$ 의 시간복잡도를 보이는데 비하여, TLDP 기법은 데이터를 더 세밀하게 분할하여 버킷에 할당하기 때문에 더 많은 수행시간을 필요로 하는 단점이 있다. 그러나 한번 버킷에 데이터들을 배정하고 나면 그 데이터들은 고정이 되어, 방송할 데이터를 선정할 때마다 필요한 실행 부담이 아니므로 그다지 큰 문제가 되지 않을 것이다.

TLDP에서 AED는 $\sum_{c=1}^K \sum_{b=1}^B (AED_{c,b})$ 로 계산된다. 특정 $bucket_{c,b}$ 에는 $d_{bcut_{c,b-1}+1} \sim d_{bcut_{c,b}}$ 의 데이터들이 할당되어 이들이 B tick마다 한 개씩 주기적으로 방송된다. 그러므로 이들을 액세스하려면 $bucket_{c,b}$ 의 방송주기의 반인 $B \times (bcut_{c,b} - bcut_{c,b-1})/2$ tick 시간만큼 대기해야 한다. 따라서 $AED_{c,b}$ 는 $B \times ((bcut_{c,b} - bcut_{c,b-1})/2) \sum_{i=bcut_{c,b-1}+1}^{bcut_{c,b}} (p_i)$ 로 계산된다. 그러므로 TLDP 기법에서 평균 예상지연시간은 다음 식과 같이 결정된다.

$$AED = \sum_{c=1}^K \sum_{b=1}^B (B \times \frac{bcut_{c,b} - bcut_{c,b-1}}{2} \sum_{i=bcut_{c,b-1}+1}^{bcut_{c,b}} (p_i)) \quad (2)$$

IV. 성능 결과 및 분석

이 장에서는 TLDP의 성능을 평가하고 그 결과를 기술한다. 비교 대상으로는 FLAT, BP, 그리고 DP 기법을 선정하였다. 각 기법의 알고리즘은 C 언어로 구현하였으며, 본 논문에서 사용된 입력변수들은 표 1과 같다.

표 1. 성능 평가를 위한 입력 변수

Table 1. Input parameters for performance evaluation

입력변수	의미	기본값	설정
N	데이터 항목의 수	5000	5000
K	채널의 수	4	1 ~ 64
θ	zipf 분포의 θ 값	0.95	0.0 ~ 1.0
B	버킷의 수	10	1 ~ 64

서버가 방송해야 될 데이터 항목의 수는 N이며, 각 데이터 항목은 그 길이가 일정하다고 가정한다. 그리고 각 데이터의 액세스 확률은 불균등한(non-uniform) 액세스 형태를 모델링하기 위해 많이 사용되는 zipf 분포 모델[6, 7]을 따른다고 가정한다. zipf 분포 모델에서 각 데이터의 액세스 확률은 식 (3)으로 표현되는데, 각 스케줄링 기법은 이 액세스 확률에 근거하여 데이터들을 분할하여 각 채널 또는 버킷에 배정한다. 여기서 θ 값은 각 데이터들의 액세스 확률이 어느 정도 편향되어 있는지를 나타내는 인수로서, θ 값이 0일 경우에는 각 데이터의 액세스 확률은 모두 동일하며 θ 값이 커질수록 불균등한 액세스의 정도가 심화된다.

$$p_i = \frac{(1/i)^\theta}{\sum_{j=1}^N (1/j)^\theta} \quad (1 \leq i \leq N) \quad (3)$$

방송 스케줄링 기법의 주요 성능 평가지수는 전체 데이터의 평균 지연시간이므로 본 논문에서는 이 성능을 중심으로 성능을 평가한다.

1. 채널 수의 변화에 따른 성능 결과

본 실험에서는 채널 수를 1에서 64까지 변화시키면서 각 기법의 성능추이를 살펴보았다. θ 값은 편향적인 액세스를 모델링하기 위한 대표 값으로 많이 사용되는 0.95로 설정하였으며^[1, 4, 6], TLDP의 경우 버킷 수를 10으로 설정하였다. 클라이언트가 하나의 데이터를 액세스하는데 걸리는 AED 성능이 그림 5에 제시되어 있는데, TLDP, DP, BP, 그리고 FLAT 기법 순으로 우수한 성능을 보인다. 채널 수가 1일 경우에는 TLDP를 제외한 다른 기법들은 모두 2500 tick의 성능을 보인다. 이는 한 채널에 5000 개의 데이터가 할당되어 이들이 동일 빈도로 주기적으로 방송되므로, 클라이언트가 어떤 데이터를 액세스하기 위하여 평균 2500 tick을 기다려야 하기 때문이

다. 채널 수가 1일 경우에 TLDP는 다른 기법에 비하여 2배 정도의 매우 우수한 성능을 보인다.

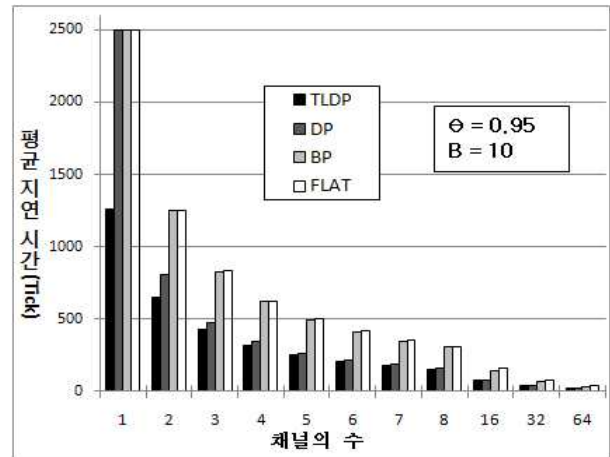


그림 5. 채널 수의 변화에 따른 평균 지연시간

Fig. 5. Number of channels vs. average delay time

채널 수가 적을 경우에는 한 채널에 많은 데이터들이 할당되어 이들 간에 액세스 확률의 차이가 커지게 된다. DP, BP, 그리고 FLAT 기법은 같은 채널의 데이터들을 동일 빈도로 방송하는데 비하여 TLDP 기법은 동일 채널에 할당된 데이터를 다시 분할하여 그들의 방송 빈도를 다르게 한다. 따라서 채널 수가 적을 경우에는 TLDP는 다른 기법에 비하여 훨씬 우수한 성능을 보이는 것이다. 참고로 채널의 수가 4인 경우에 TLDP는 DP 기법보다 8% 이상 우수한 성능을 보이며, BP나 FLAT 기법보다는 97% 이상 향상된 성능을 보인다.

채널 수가 늘어남에 따라 TLDP와 다른 기법들의 성능 차이가 줄어들는데, 이는 다른 기법들도 채널 수가 증가함에 따라 세밀하게 데이터를 분할할 수 있기 때문이다. 더 많은 그룹으로 데이터를 분할함에 따라 동일 채널에 배정된 데이터들간의 액세스 확률의 차이가 작아지므로 이들이 평형 방식으로 방송되는 단점이 완화되어 TLDP와 다른 기법간의 성능 차이가 줄어드는 것이다. 채널 수가 16 이상으로 증가하게 되면, DP는 TLDP 기법과 거의 유사한 성능을 보이기 시작한다. DP, BP, 그리고 FLAT 기법에서는 데이터 분할 그룹의 수는 같지만, DP 기법은 데이터들의 AED 성능이 최소화되도록 분할하는 정책을 사용하기 때문에 BP나 FLAT 기법보다 훨씬 우수한 성능을 보인다. BP 기법은 액세스 확률을 반영하여 분할하므로 FLAT 기법보다는 우수한 성능을 보이는

하지만, 각 채널에 배정되는 데이터의 개수에 따른 지연 시간의 차이를 고려하지 않아 그 성능 차이가 그다지 크지는 않다.

2. θ 값 및 버킷 수의 변화에 따른 성능 결과

본 실험에서는 θ 값 및 버킷 수의 변화에 따른 성능 추이를 살펴보았다. θ 값을 변화시키면서 살펴본 성능 결과가 그림 6에 제시되어 있다. zipf 분포에서 클라이언트 액세스의 불균형의 정도는 θ 값에 의하여 결정되는데, 만일 θ 값이 0이 되면, 각 데이터의 액세스 확률이 동일하게 되어 클라이언트가 액세스하는 데이터들의 분포는 균등(uniform) 분포를 이루게 된다. 따라서 이 경우에는 모든 기법들이 동일한 성능을 보인다.

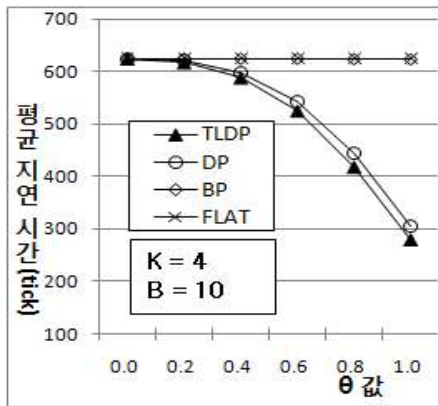


그림 6. θ 값에 따른 평균지연시간
Fig. 6. θ value vs. avg. delay time

그러나 θ 값이 증가함에 따라 각 데이터들의 액세스 확률의 차이가 점점 커지게 되고, 이 확률을 고려하여 적절하게 방송빈도를 결정하는 TLDP와 DP 기법은 다른 기법에 비해 우수한 성능을 보이기 시작한다. FLAT 기법은 액세스 확률에 상관없이 각 채널에 N/K 개의 데이터를 일정하게 할당하기 때문에 θ 값의 변화에 관계없이 항상 $N/2K$ 의 동일한 성능을 보인다. BP 기법은 각 채널에 배정되는 데이터의 액세스 확률의 합이 동일하게 되도록 데이터를 분할한다. 따라서 확률이 높은 데이터가 할당된 채널에는 적은 수의 데이터가 배정되어 이들은 상대적으로 더 높은 빈도로 방송된다. 따라서 직관적으로는 θ 값이 커짐에 따라 BP가 FLAT 기법보다 월등한 성능을 보일 것 같지만 실제로는 그렇지 않다. BP 기법은 액세스 확률이 높은 데이터를 더 자주 방송하는 방법을 채택하고 있지만, 이들을 확률 대비 너무 자주 방송함

으로써 상대적으로 액세스 확률이 낮은 데이터들의 지연 시간이 길어지기 때문이다. BP와 FLAT 기법이 유사한 성능을 보이는 이유를 해석적으로 분석하기 위해, BP에서 각 채널 CH_i 에 배정된 데이터 수와 액세스 확률의 합을 N_i 와 S_i 라 하자. CH_i 의 방송주기는 N_i 이므로 CH_i 의 데이터를 액세스하려면 평균 $N_i/2$ 를 기다려야 하므로 $AED = \sum_{i=1}^K (N_i/2) \times S_i$ 가 된다. 그런데 데이터 분할 정책에 의해 $S_i = 1/K$ 이 되어 $AED = (1/K) \sum_{i=1}^K N_i/2$ 가 된다 그러므로 BP 기법의 AED는 대략 $N/2K$ 가 되어, FLAT 기법과 거의 유사한 성능을 보이는 것이다.

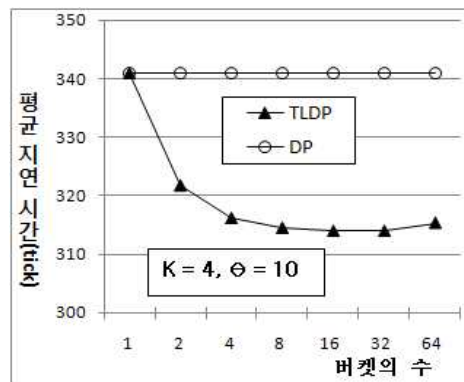


그림 7. 버킷 수에 따른 평균지연시간
Fig. 7. No. of buckets vs. avg. delay time

버킷 수(B)에 따른 성능 결과가 그림 7에 제시되어 있는데, DP 기법은 버킷을 사용하지 않지만 성능비교 차원에서 그 결과를 함께 제시한다. B가 1일 경우에 TLDP는 DP 기법과 동일하게 데이터를 분할하기 때문에 이들은 서로 동일한 성능을 낸다. 그러나 B 값이 증가함에 따라 TLDP의 성능은 우수해 지는데, B가 16 이상이 되면 더 이상 성능이 향상되지 않는다. 오히려 B가 32 이상이 되면 성능은 대체로 저하되기 시작하는데, 이는 채널에 배정되는 데이터에 비해 버킷의 수가 너무 많기 때문이다. 적절하게 B 값을 설정하는 것이 성능에 영향을 미치는데, 이 값은 θ 값, 채널의 수, 데이터 항목의 개수 등을 고려하여 결정되어야 한다. 본 논문의 실험에서는 B의 값을 5~20 정도로 설정할 때 우수한 성능을 보인다. 채널에 배정되는 데이터의 수에 따라 각 채널별로 버킷의 수를 다르게 설정하는 것이 가장 우수한 성능을 발휘하기는 하지만 그 성능 차이가 그다지 크지 않기 때문에 본 논문에서는 B의 값을 고정한 결과 값을 제시한다.

V. 결론

방송 방식의 데이터 전달 기법은 방대한 클라이언트들을 지원해야 하는 환경이나 모바일 컴퓨팅 환경에서 정보 전달을 위한 주요 기술로 인정받고 있다. 그러나 많은 데이터를 방송해야 할 경우에 방송 채널의 순차성으로 인하여 원하는 데이터를 수신하기까지의 예상 지연시간이 크게 증가할 수 있다. 본 논문은 이 지연시간을 줄이기 위하여 다중 방송채널 환경을 위한 방송 스케줄링 기법에 대한 연구를 진행하였다. 기존의 제안된 대부분의 기법들은 같은 채널에 배정된 데이터들을 동일 빈도로 방송하기 때문에 물리적 채널의 수가 적을 경우에는 예상 지연시간의 성능이 심각하게 저하되는 단점이 있다. 본 논문은 이와 같은 단점을 개선하기 위하여 두 단계로 세밀하게 데이터를 분할하여 동일 채널에 할당된 데이터들도 그들의 방송 빈도를 다르게 할 수 있는 TLDP로 명명된 새로운 방송 스케줄링 기법을 제안했다. 그리고 성능 평가를 실시하여 TLDP 기법의 성능과 DP, BP 및 FLAT 기법의 성능을 비교하였다. 성능평가 결과에 의하면, 제안된 기법은 방송 스케줄러의 주요 성능지수인 평균 지연시간에 있어서 다른 기법에 비하여 우수한 성능을 보인다. 특히 물리적인 채널의 수가 적을 경우에는 TLDP 기법은 다른 기법보다 훨씬 우수한 성능을 보인다.

본 논문은 단일 데이터를 액세스하는 환경에서 지연시간을 줄이기 연구를 진행하였다. 본 논문의 미래 연구로서는 동시에 다수의 데이터를 액세스하는 환경과 클라이언트 캐쉬가 존재하는 환경을 고려한 스케줄링 기법에 관한 연구를 진행할 것이다.

참 고 문 헌

- [1] S. Acharya, "Broadcast Disks: Dissemination-based Data Management for Asymmetric Communication Environments," *Ph.D. thesis*, Brown University, 1998.
- [2] M. Franklin and S. Zdonik, "A Framework for Scalable Dissemination-Based Systems," *In the Int. Conf on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA 97)*, pp. 94-105, Atlanta, GA, October 1997.
- [3] M. Franklin and S. Zdonik, "Data in Your Face: Push Technology in Perspective," *Proc. of the 1998 ACM SIGMOD Int. Conf on Management of Data*, pp. 516-519, Seattle, WA, June 1998.
- [4] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric communications environments," *Proc. of the 1995 ACM SIGMOD Int. Conf on Management of Data*, pp. 199-210, San Jose, CA, May 1995.
- [5] S. Acharya, M. Franklin, and S. Zdonik, "Balancing Push and Pull for Data Broadcast," *Proc. of the 1997 ACM SIGMOD Int. Conf on Management of Data*, pp. 183-194, Tucson, Arizona, May 1997.
- [6] V. Liberatore, "Caching and Scheduling for Broadcast Disk Systems," *Journal of Experimental Algorithmics*, Vol. 6, No. 5, 2001.
- [7] N.H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," *Wireless Networks*, Vol. 5, No. 3, pp. 171-182, May 1999.
- [8] K. Prabhakara, K. A. Hua, and J. Oh, "Multi-Level Multi-Channel Air Cache Designs for Broadcasting in a Mobile Environment," *Proc. Int'l Conf Data Eng. (ICDE)*, 2000.
- [9] W.C. Peng and M.S. Chen, "Efficient channel allocation tree generation for data broadcasting in a mobile computing environment," *Wireless Networks*, Vol. 9, No. 2, pp. 117-129, 2003.
- [10] C. Hsu, G. Lee, A.L.P. Chen, "A near optimal algorithm for generating broadcast programs on multiple channels," *Proc. ACM 10th Int'l Conf CIKM*, Atlanta, Georgia, pp. 303-309, 2001.
- [11] W.G. Yee, S. Navathe, E. Omiecinski, and C. Jermaine, "Efficient Data Allocation over Multiple Channels at Broadcast Servers," *IEEE Trans. on Computers*, Vol. 51, No. 10, pp. 1231-1236, Oct. 2002.

- [12] S. Wang and H.L. Chen, "An $O(N \log K)$ Restricted Dynamic Programming Algorithm for Data Allocation over Multiple Channels," *IEICE trans. on communications*, Vol. E88-B, No. 9, pp. 3756-3764, Sep. 2005.
- [13] S. Wang and H.L. Chen, "Near-optimal data allocation over multiple broadcast channels," *Computer communications*, Vol. 29, No. 9, pp. 1341-1349, Elsevier Science, 2006.

저자 소개

권혁민(정회원)



- 1984년 서울대학교 제어계측공학과 학사
- 1994년 한국과학기술원 정보및통신공학과 석사
- 1998년 한국과학기술원 정보및통신공학과 박사
- 1984년 ~ 1991년 대우전자 중앙

연구소 컴퓨터개발부 선임연구원

- 1999년 ~ 현재 세명대학교 정보통신학부 부교수
- <주관심분야 : 트랜잭션 처리, 분산/병렬 데이터베이스, 모바일 컴퓨팅>