

# 다중 방송 채널에 데이터 할당을 위한 두 단계 저장소-적재 알고리즘

권혁민<sup>†</sup>

## 요약

데이터 방송 시스템에서 서버는 방송 채널을 통하여 데이터들을 지속적으로 전파하고, 이동 클라이언트는 자신이 원하는 데이터가 방송 채널에 나타나기를 기다리기만 하면 된다. 그러나 방송 채널은 많은 데이터들에 의해 공유되어야 하므로, 원하는 데이터를 수신하기까지 예상 지연시간이 증가할 수 있다. 본 논문은 전체 데이터들의 예상 지연시간을 최소화하기 위하여 다중 방송 채널에 적절하게 데이터를 할당하기 위한 주제를 연구하여 TLBP(Two Level Bin-Packing)로 명명된 새로운 데이터 할당 기법을 제안한다. 본 논문은 우선 평균 예상지연시간의 이론적 하한 값을 소개하고, 이 값에 기초하여 저장소의 용량을 결정한다. TLBP 기법은 저장소-적재 알고리즘을 이용하여 전체 데이터들을 다수 개의 그룹으로 분할하고, 각 그룹의 데이터들을 각 채널에 배정한다. TLBP는 저장소-적재 알고리즘을 두 단계로 적용함에 의해, 동일 방송 채널에 할당된 데이터들의 액세스 확률의 차이를 방송 스케줄에 반영할 수 있어 성능을 향상시킬 수 있다. TLBP와 세 가지의 기존 기법과 성능을 비교하기 위하여 시뮬레이션이 수행되었다. 시뮬레이션 결과에 의하면 TLBP는 합리적인 실행부담을 가지면서도 평균 예상지연시간의 성능에 있어서 다른 기법보다 우수한 성능을 보인다.

## Two Level Bin-Packing Algorithm for Data Allocation on Multiple Broadcast Channels

Hyeok Min Kwon<sup>†</sup>

## ABSTRACT

In data broadcasting systems, servers continuously disseminate data items through broadcast channels, and mobile client only needs to wait for the data of interest to present on a broadcast channel. However, because broadcast channels are shared by a large set of data items, the expected delay of receiving a desired data item may increase. This paper explores the issue of designing proper data allocation on multiple broadcast channels to minimize the average expected delay time of all data items, and proposes a new data allocation scheme named two level bin-packing(TLBP). This paper first introduces the theoretical lower-bound of the average expected delay, and determines the bin capacity based on this value. TLBP partitions all data items into a number of groups using bin-packing algorithm and allocates each group of data items on an individual channel. By employing bin-packing algorithm in two step, TLBP can reflect a variation of access probabilities among data items allocated on the same channel to the broadcast schedule, and thus enhance the performance. Simulation is performed to compare the performance of TLBP with three existing approaches. The simulation results show that TLBP outperforms others in terms of the average expected delay time at a reasonable execution overhead.

**Key words:** Broadcast Schedule(방송 스케줄), Data Allocation(데이터 할당), Bin-Packing(저장소-적재), Mobile Computing(이동 컴퓨팅)

※ 교신저자(Corresponding Author): 권혁민, 주소: 충북 제천시 세명로 117번지 세명대학교 전산정보관 318호 (390-711), 전화: 043)649-1269, FAX: 043)649-1747, E-mail: hmkwon@semyung.ac.kr

접수일: 2011년 4월 8일, 수정일: 2011년 5월 26일  
완료일: 2011년 8월 17일

<sup>†</sup> 정희원, 세명대학교 정보통신학부 부교수

## 1. 서 론

최근 들어, 이동(mobile) 컴퓨터와 무선 네트워크 기술의 비약적인 발전으로 인하여 이동 컴퓨팅 환경에서 정보 서비스가 보편화되고 있다. 이에 따라 이동 클라이언트로 데이터를 효과적으로 전송하기 위한 기법들이 큰 관심을 끌고 있는데, 방송(broadcast) 방식에 기초한 데이터 전달 기법은 이동 컴퓨팅 환경에서 정보 전달을 위한 주요 기술로 인정받고 있다 [1-4]. 방송 방식의 데이터 전달 기법을 채택한 정보 시스템에서 방송 서버는 다수의 클라이언트들에게 데이터들을 지속적으로 전파하고, 각 클라이언트는 자신이 원하는 데이터가 방송 채널에 나타나면 이를 검색한다. 방송 시스템의 성능은 그 시스템이 지원하는 클라이언트 수에 영향을 받지 않기 때문에 시스템의 확장성(scalability)이 매우 우수하다. 따라서 방송 시스템은 기상예보, 도로교통 정보, 주식시장 정보, 전자 뉴스 등의 정보제공(information feeding) 응용과 같이 방대한 규모의 클라이언트를 지원해야 하는 응용 분야에도 매우 적합하다.

그러나 방송 시스템에서 서버는 제한된 방송 대역폭을 통하여 많은 데이터들을 일방적으로 방송하므로 클라이언트가 어떤 데이터를 액세스하려면 그 데이터가 방송되기를 기다려야 한다. 방송 시스템의 주요 성능지수는 평균 예상지연(average expected delay: AED) 시간인데, 이는 클라이언트가 어떤 데이터를 액세스하는데 필요한 평균 대기시간을 의미한다. AED 성능은 채널에 데이터를 할당하는 정책 및 방송 스케줄링(data broadcast scheduling) 기법에 크게 영향을 받기 때문에 이에 대한 많은 연구들이 수행되었다. 기존에는 단일 방송채널 환경에서 평균 예상지연시간을 줄이기 위한 연구들이 수행되었는데, 이들은 각 데이터의 액세스 확률에 근거하여 자주 사용하는 데이터를 더 빈번하게 방송하는 방법을 제시하여 AED 성능을 개선했다 [1,4-6].

[5]의 연구는 데이터의 액세스 확률과 크기를 고려하여 각 데이터의 최적 방송주기를 계산하여 스케줄링 정책에 반영했다. 그러나 최적 주기만을 고려할 경우 크기가 큰 데이터가 방송되는 순간에 다른 크기가 작은 데이터들의 방송이 많이 지연되어 전체 AED 시간이 커지는 문제가 있다. [6]의 연구는 이 문제를 해결하기 위하여, 현재 방송하려는 데이터의

크기가 커서 다른 데이터들을 최적 주기내에 방송하지 못할 경우 최적 주기내에 방송할 수 없는 데이터들 중에서 전체 AED 시간이 최소가 되는 데이터를 선정하여 방송하고, 현재 방송하려고 했던 크기가 큰 데이터의 방송을 연기할 수 있는 새로운 기법을 제안했다. 이 기법은 기존에 최적 주기만을 고려한 기법보다 우수한 AED 성능을 보인다 [6].

최근에는 다중 채널 환경을 위한 연구들이 많이 수행되었는데, 이들은 대부분 전체 데이터들을 어떻게 분할하여 각 채널에 할당할 것인가에 치중하여 연구를 수행하였다 [7-11]. 이 연구들은 평균 예상지연시간이 최소화될 수 있는 방향으로 전체 데이터들을 액세스 확률에 근거하여 채널 수만큼 다수 개의 그룹으로 분할하여 각 채널에 할당하였다. 그리고 같은 채널에 할당된 데이터들을 라운드-로빈(round-robin) 방식으로 동일 빈도로 방송하는 방법을 채택했다. 그런데 이동 컴퓨팅 환경의 특성상 방송해야 될 데이터 수는 방송 채널의 수에 비해서 매우 크기 때문에 한 채널에는 많은 수의 데이터가 배정된다. 따라서 동일 채널에 할당된 데이터들 간에 액세스 확률의 차이는 커질 수밖에 없다. 이와 같은 경우에 같은 채널의 데이터들을 인기도에 관계없이 동일 빈도로 방송한다면 전체 데이터들의 AED 성능은 심각하게 저하될 것이다.

본 논문은 이와 같은 기존 기법들의 단점을 개선하기 위하여 동일 채널의 데이터들도 다른 빈도로 방송할 수 있게 하는 데이터 할당 기법에 대해 연구한다. [12]의 연구는 K개의 방송 채널을 통하여 N개의 데이터들을 방송할 때 AED의 이론적 하한 값이  $AED_{optimal} = (\sum_{q=1}^N \sqrt{p_q})^2 / 2K$  라는 사실을 증명했다. 여기서  $p_q$ 는 데이터  $d_q$ 의 액세스 확률을 의미한다. 만일 각 채널이 이상적으로 데이터들을 방송한다면,  $AED_{optimal}$ 을 균등하게 배분하여 각 채널의 예상지연시간이  $AED_{optimal}/K$ 가 되도록 각 채널에 데이터를 배정할 때 최적 성능을 발휘할 것이다. 본 논문은 이 휴리스틱(heuristic)을 제안된 기법의 데이터 할당 정책에 적용하고, 저장소-적재(bin-packing) 알고리즘에 [13] 기초하여 각 채널에서 방송할 데이터들을 결정한다. 그리고 동일 채널에서 방송될 데이터들을 저장소-적재 알고리즘을 이용하여 한번 더 분할하여 각 그룹별로 다른 빈도로 방송한다. 제안된 기법은 전체 데이터들을 두 단계로 세밀하게 분할하

기 때문에 동일 채널에 할당된 데이터들을 인기도를 반영하여 다른 빈도로 방송할 수 있다. 따라서 제안된 기법은 다른 기법들에[7-11] 비해 크게 향상된 AED 성능을 발휘할 수 있고, 대부분의 환경에서 거의 최적에 필적하는 성능을 보일 수 있다. 본 논문은 제안된 기법의 성능을 평가하기 위하여 시뮬레이션 모델에 기초하여 제안된 기법과, 기존 기법, 그리고 이론적 하한 값의 성능을 비교한다.

본 논문의 구성은 다음과 같다. 2장에서 기존에 제안된 기법에 대하여 기술하고, 3장에서 본 논문이 제안하는 새로운 데이터 할당 기법을 기술한다. 4장에서는 제안된 기법의 성능을 평가, 분석하고, 마지막으로 5장에서 결론을 맺는다.

## 2. 관련 연구

이 장에서는 본 논문의 성능 평가에서 사용된 기법들을 중심으로 기존에 제안된 기법들에 대하여 기술한다. 단일채널 환경을 위해 제안된 평형(Flat) 기법은 채널에 배정된 데이터들을 라운드-로빈(round-robin) 방식으로 주기적으로 방송한다[4]. 따라서 모든 데이터들은 액세스 확률에 관계없이 동일 빈도로 방송된다. 이 기법은 다중채널 환경으로 용이하게 확장될 수 있는데, 다중 채널의 평형 기법은 각 채널에 동일 개수의 데이터를 배정하고, 각 채널은 이들을 평형 방송한다. 평형 기법은 단순하여 구현이 용이하다는 장점이 있지만, 액세스 확률이 편향(skew)되어 있을 경우에 다른 기법들에 비하여 심각하게 성능이 저하된다.

최근 들어 다중 방송채널 환경을 위한 많은 연구들이 수행되었는데, 대표적인 기법으로 BP(Bin-Packing)[7],  $VF^K$ (Variable-Fanout with the constraint K)[8], 동적 프로그래밍(Dynamic Programming: DP)[9], 그리고 제한된 동적프로그래밍(Restricted Dynamic Programming: RDP) 기법들이[10,11] 있다. 이 연구들은 모두 다음과 같은 동일한 시스템 모델을 가정하고 있다. 방송 데이터들은 동일한 크기를 가진  $N$  개의 데이터로 구성되며, 유사한 액세스 확률을 가진 데이터들을 동일 채널에 배정하기 위하여 각 데이터들은 액세스 확률의 내림차순으로 정렬되어 있다고 가정한다. 그리고 방송 서버에는 동일 대역폭을 갖는  $K$ 개의 방송 채널이 존재하고,

같은 채널에 할당된 데이터들은 평형 방식으로 동일 빈도로 방송된다.

BP 기법은 각 채널에 배정되는 데이터들의 액세스 확률의 합이 동일하게 되도록 채널에 데이터들을 할당한다[7]. 따라서 액세스 확률이 높은 데이터가 할당되는 채널에는 적은 수의 데이터가 배정되어 이들은 상대적으로 높은 빈도로 방송된다. BP 기법은 데이터의 액세스 확률을 고려하여 채널에 데이터를 배정하지만, 액세스 확률이 높은 데이터를 확률 대비 너무 빈번하게 방송하여 액세스 확률이 낮은 데이터들의 지연시간이 크게 증가하는 단점이 있다.

$VF^K$  기법은 가변적인 팬아웃(fanout) 트리를 구성하여 다중 채널에 데이터를 할당한다[8]. 이 기법은 각 데이터들을 트리의 루트(root) 노드의 자식 노드로 배치하여 초기 트리를 구성한다. 그리고 나서 루트의 자식 노드로서 중간 노드를 한 개 추가하고, AED 시간이 최소화되도록 액세스 확률이 낮은 데이터들을 추가된 중간 노드의 자식노드로 이동한다. 즉 트리의 다음 레벨로 이동한다. 이와 같은 과정을 트리의 깊이가 채널 수와 같아질 때까지 반복적으로 수행한다. 트리 구성이 완료되면 트리의 동일 레벨에 있는 단말 노드의 데이터들을 동일 채널에 배정한다.  $VF^K$  기법은 채널 수가 2의 지수승일 경우에는 거의 최적 성능을 발휘하지만, 그렇지 않을 경우에는 각 채널의 성능이 균형되지 못하여 AED 시간이 증가되는 단점이 있다. 그리고 채널에 데이터를 배정하기 위하여  $O(KN^2 \log K)$ 의 시간복잡도(time complexity)로 보여 실행 부담이 크다는 단점이 있다.

DP 기법은 각 채널의 데이터가 평형 방송된다는 가정하에 예상지연시간이 최소화되도록 전체 데이터들을  $K$  개의 그룹으로 분할하여 각 채널에 배정한다[9]. 따라서 DP 기법은 각 채널의 데이터들이 평형 방송되는 시스템 모델에서 AED 성능으로는 최적(optimal) 알고리즘이다. 그러나 DP 기법은 가능한 모든 분할 방법을 고려하여 일일이 AED 성능을 계산하므로 이의 실행을 위한 시간복잡도가  $O(KN^2)$ 이 되어 실행부담이 크다는 단점이 있다. [9]의 연구에서는 이 실행부담을 줄이기 위한 Greedy 알고리즘도 함께 제안했다. Wang과 Chen의 연구에서도 DP의 수행시간 측면의 단점을 개선하기 위하여 이론적으로 최적 AED 성능을 추정하여 데이터를 분할하는 RDP 기법을 제안했다[10,11]. RDP 기법은 채널에 데

이터를 배정하기 위하여  $O(N \log K)$ 의 시간복잡도를 보이며 거의 최적에 가깝게 데이터를 분할한다.

단일채널 환경에서 모든 데이터가 동일 크기이고 각 데이터마다 동일 간격으로 주기적으로 방송될 수 있다면, 각 데이터의 상대적인 방송빈도를 자신의 액세스 확률의 제곱근 값에 비례하게 방송하는 것이 가장 이상적이다[14]. [12]의 연구는 이 연구를 다중 채널 환경으로 확장하여,  $N$  개의 데이터를  $K$  개의 채널을 통하여 방송할 때, 전체 데이터들의 예상지연 시간이 식 (1)로 결정된다는 것을 증명했다. 여기서  $f_i$ 는 데이터  $d_i$ 의 액세스 횟수를 의미한다. 그러나 주어진 전체 조건들을 모두 만족시키면서 모든 데이터들을 방송하는 것은 거의 불가능하다. 예를 들어, 특정 채널에 데이터  $d_1 \sim d_3$ 가 할당되어 있고, 이들의 액세스 확률의 제곱근 값이 각각  $1/2, 1/3, 1/6$ 이라면 주어진 조건을 만족하면서 이들을 방송하는 것은 불가능하다. 따라서 식 (1)을 AED의 이론적 하한(lower-bound) 값이라 한다[12]. 비록 이 성능이 현실적으로 실현되기는 거의 불가능하지만, 특정 기법의 상대적인 성능을 평가할 때는 매우 유용하게 사용된다.

$$AED_{optimal} = \frac{1}{2K} \left( \sum_{i=1}^N \sqrt{\frac{f_i}{\sum_{j=1}^N f_j}} \right)^2 \quad (1)$$

### 3. 새로운 데이터 할당 기법

이 장에서는 본 논문이 제안하는 새로운 데이터 할당 기법에 대하여 기술한다. 전형적인 저장소-적재(Bin-Packing) 알고리즘은 일련의 개체들과 이들의 크기 및 저장소(Bin)의 용량이 주어졌을 때, 최소 개수의 저장소를 사용하면서 이들을 저장소에 적재하는 문제를 다룬다[13]. 본 논문은 저장소-적재와 유사한 개념을 도입하여 각 방송 채널이 방송할 데이터들을 결정한다. 본 논문은 각 채널에서 방송되는 데이터들을 그룹화하기 위하여 저장소와 서브저장소(Sub Bin: SBin)의 두 종류의 저장소를 운영하고, 전체 데이터들을 두 단계로 나누어 저장소에 적재한다. 따라서 본 논문이 제안하는 기법을 TLBP(Two Level Bin-Packing)로 명명한다. TLBP에서 저장소의 개수는 고정되어 있고, AED 시간이 최소화될 수 있도록 저장소의 용량을 결정하고 전체 방송 데이터들을 적절히 분할하여 저장소에 적재한다.

### 3.1 기본 시스템 모델

본 논문은 다음과 같은 시스템 모델과 용어를 사용한다.

(1) 전체 방송 데이터는 동일한 크기를 가진  $N$  개의 데이터로 구성되는데, 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.

(2) 방송 서버에는 동일 대역폭을 갖는  $K$  개의 방송채널이 존재하는데, 이를  $CH_i (i=1 \sim K)$ 로 표현하며,  $CH_i$ 에서 방송될 데이터의 저장소를  $Bin_i$ 로 표현한다.

(3) 각 채널은 큐 형태로 관리되는  $J$  개의 서브저장소를 운영하는데,  $CH_i$ 의 서브저장소를  $SBin_{i,j} (j=1 \sim J)$ 로 표현한다.  $Bin_i$ 와  $SBin_{i,j}$  용어는 문맥에 따라 각각의 저장소에 할당된 데이터 집합으로도 사용한다.

(4) 임의의 데이터는 하나의 SBin에만 할당될 수 있으며,  $Bin_i = \sum_{j=1}^J SBin_{i,j}$ 이며,  $Bin_i$ 와  $SBin_{i,j}$ 에 할당된 데이터 개수를 각각  $|Bin_i|$ 와  $|SBin_{i,j}|$ 로 표현한다.

(5) 한 방송채널에서 하나의 데이터가 방송되는데 걸리는 논리적인 시간을 1 tick으로 정의하고, AED 시간 단위로서 tick을 사용한다.

### 3.2 기본 개념 및 각 채널의 방송 스케줄링 기법

TLBP는 두 단계로 데이터를 분할하여 저장소(Bin)와 서브저장소(SBin)에 할당한다. 우선 채널의 Bin 용량을 적절히 결정하고, 이에 맞추어 전체 데이터를  $K$  개의 그룹으로 분할하여  $Bin_1 \sim Bin_K$ 에 적재한다. 그리고 각  $Bin_i (i=1 \sim K)$ 별로 SBin의 용량을 결정한다. 이 용량에 맞추어 각  $Bin_i$ 에 할당된 데이터들을  $J$  개의 그룹으로 분할하여  $SBin_{i,1} \sim SBin_{i,J}$ 에 적재한다. Bin과 SBin의 용량을 결정하는 방법과 이들에게 데이터를 할당하는 방법은 다음 절에서 자세히 설명한다. 각  $Bin_i$ 는  $SBin_{i,j} (j=1 \sim J)$ 를 결정하기 위하여 사용하는 임시 저장소로,  $CH_i$ 가 방송하는 실제 데이터는  $SBin_{i,j} (j=1 \sim J)$ 에 할당된다.

각 채널의 방송 스케줄러는 방송할 데이터의 선정이 필요할 때마다 표 1의 알고리즘을 실행한다. 우선 어떤 SBin의 데이터를 방송할 것인가를 결정하기 위하여, 표 1의 1)과 같이 라운드-로빈 방식으로 SBin을 결정한다. 그리고 그 SBin의 맨 앞에 있는 데이터

를 선정하여 방송하고, 그 데이터를 SBin의 맨 뒤로 이동한다. 여기서 j는 SBin 번호로서 방송 스케줄링 알고리즘을 기동시킬 때 이 값을 0으로 초기화한다. 방송 스케줄러는 각 SBin을 J tick 주기마다 선택하며, 동일 SBin에 할당된 데이터들을 공평하게 선정하여 방송한다. 따라서 동일 SBin에 할당된 데이터들은 방송채널 대역폭의 1/J을 사용하며 동일 빈도로 방송된다.

표 1. CH의 방송 스케줄링 알고리즘

- 1) choose a sub-bin  $SBin_{i,j}$  such that  $j=(j \text{ mod } J)+1$
- 2) choose a data item  $d_q$  from the front of  $SBin_{i,j}$
- 3) broadcast  $d_q$
- 4) Dequeue item  $d_q$  from the front of  $SBin_{i,j}$
- 5) Enqueue  $d_q$  at the rear of  $SBin_{i,j}$

### 3.3 두 단계 저장소-적재 알고리즘

[12]에서 증명된 식 (1)의  $(f_i/\sum_{j=1}^N f_j)$ 는 전체 데이터의 액세스 횟수 대비  $d_i$ 의 액세스 횟수를 의미하므로 본 논문의 시스템 모델에서  $p_i$ 에 대응된다. 따라서 식 (1)은 다음과 같이 표현된다.

$$AED_{optimal} = \frac{1}{2K} \left( \sum_{i=1}^N \sqrt{p_i} \right)^2 \quad (2)$$

본 논문은 최적 성능인  $AED_{optimal}$  실현을 목표로 하고 있다. 만일 각 채널이 자신에게 할당된 데이터들을 이상적으로 방송한다면,  $AED_{optimal}$ 을 각 채널이 균등하게 배분하여 담당할 때 최적 성능을 발휘할 것이다. 본 논문은 이 휴리스틱(heuristic)을 TLBP 기법의 데이터 할당 정책에 적용한다. 이를 위해서는 각 채널이  $AED_{optimal}/K$ 의 예상지연시간을 갖도록 데이터가 할당되어야 한다. CH가 이상적으로 데이터를 방송할 때의 AED를  $AED_{optimal}^i$ 로 표현하자.  $AED_{optimal}^i$ 은  $(1/2) \times (\sum_{d_q \in Bin_i} \sqrt{p_q})^2$ 로 계산된다[14]. 그런데  $\sum_{d_q \in Bin_i} \sqrt{p_q}$ 의 값이  $(\sum_{i=1}^N \sqrt{p_i})/K$ 로 결정되면,  $AED_{optimal}^i$ 은  $AED_{optimal}/K$ 이 된다. 즉, 각  $Bin_i$ 에 할당된 데이터들의 액세스 확률의 제곱근의 합이 동일하다면 진술한 휴리스틱하에서 최적 분배가 이루어지는 것이다.

본 논문은 이와 같은 개념으로 데이터를 분배하기 위하여 채널의 Bin 용량을  $(\sum_{i=1}^N \sqrt{p_i})/K$ 로 설정하

고, 다음-적합(Next-Fit) 방식으로 Bin 값이 Bin의 용량과 같아지거나 초과할 때까지 Bin에 데이터들을 저장한다. Bin 값은  $\sum_{d_q \in Bin} \sqrt{p_q}$ 를 의미한다. 저장소-적재 알고리즘에서 다음-적합 방식은 어느 한 순간에는 하나의 활성(active) Bin이 존재하며, 현재 개체를 이 Bin에 저장할 수 있으면 저장하고, 그렇지 않으면 다음에 활성화시킬 Bin으로 넘기는 방식이다. TLBP 기법은  $Bin_1 \sim Bin_K$ 를 차례대로 활성화시키는데, 활성 Bin에 Bin의 용량만큼 데이터를 할당한 후에 다음 Bin을 활성화시킨다. 그리고 데이터는  $d_1 \sim d_N$ 의 순서로 Bin에 할당시켜 나간다. 각 Bin의 데이터들은 액세스 확률의 내림차순으로 정렬을 유지한다. 본 연구는 전체 데이터들이 액세스 확률의 내림차순으로 정렬되어 있다고 가정하고 데이터는  $d_1 \sim d_N$ 의 순서로 할당하므로, 각 Bin에 데이터가 저장되는 순서를 유지하기만 하면 Bin의 정렬 특성을 쉽게 유지할 수 있다. 이와 같은 방식으로  $Bin_1 \sim Bin_K$ 에 데이터를 할당하는 알고리즘이 표 2에 제시되어 있는데, 여기서  $NextDataAllocI$ 는 Bin에 할당할 다음 데이터의 인덱스를 의미한다. 표 2의 알고리즘은 각  $Bin_i$ 에 데이터를 할당하면서 Bin 값을 계산해서  $BinVal_i$ 에 저장하는데, 이는 표 3에서 SBin 용량을

표 2. 첫 번째 단계의 저장소-적재 알고리즘

```

//precompute  $\sqrt{p_i}$  and  $\sum_{i=1}^N \sqrt{p_i}$ 
SumSr = 0
for each data i from 1 to N
     $sr_i = \sqrt{p_i}$ 
    SumSr = SumSr +  $sr_i$ 
end for
//allocate all data items into  $Bin_1 \sim Bin_K$ 
BinCapa = SumSr / K
NextAllocDataI = 1
for each channel i from 1 to K-1
    BinVal_i = 0
    for each data j from NextAllocDataI to N
        allocate  $d_j$  into  $Bin_i$ 
        BinVal_i = BinVal_i +  $sr_j$ 
        if (BinVal_i >= BinCapa)
            NextAllocDataI = j + 1
            break;
        end if
    end for
end for
BinVal_K =  $\sum_{i=NextAllocDataI}^N (sr_i)$ 
allocate remainder( $d_{NextAllocDataI} \sim d_N$ ) into  $Bin_K$ 
    
```

표 3. 두 번째 단계의 저장소-적재 알고리즘

```
//allocate each data in BIN into its SBins
for each Bin  $i$  from 1 to  $K$ 
     $SBinCapa = BinVal_i / J$ 
     $NextAllocDataI =$  first data index in  $Bin_i$ 
     $LastDataIndex =$  last data index in  $Bin_i$ 
    for each Sub Bin  $j$  of  $Bin_i$  from 1 to  $J-1$ 
         $SBinVal = 0$ 
        for each data  $k$  in  $Bin_i$  from  $NextAllocDataI$ 
            to  $LastDataIndex$ 
            allocate  $d_k$  into  $SBin_{i,j}$ 
             $SBinVal = SBinVal + sr_k$ 
            if ( $SBinVal >= SBinCapa$ )
                 $NextAllocDataI = k + 1$ 
                break;
            end if
        end for
    end for
    allocate remainder( $d_{NextAllocDataI} \sim d_{LastDataIndex}$ )
    in  $Bin_i$  into  $SBin_{i,J}$ 
end for
```

계산할 때 사용한다.

표 3은 각  $Bin_i(i=1\sim K)$ 의 데이터들을  $J$  개의 그룹으로 분할하여  $SBin_{i,1}\sim SBin_{i,J}$ 에 할당한다.  $Bin_i$ 의 데이터들을  $J$  개의 그룹으로 분할하는 개념은 표 2와 유사하다. 표 2에서 계산된  $BinVal_i$  값을  $CH_i$ 의 SBin으로 균등하게 배분하기 위하여  $CH_i$ 의 SBin 용량을  $BinVal_i/J$ 로 결정한다. 그리고  $SBin_{i,1}\sim SBin_{i,J}$ 를 차례대로 활성화하면서,  $Bin_i$ 의 데이터들을 다음-적합 방식으로 적절한 SBin에 할당시켜 나간다.

데이터 분할 및 할당 기법의 우수성은 데이터를 분할하여 각 채널에 배정하는데 걸리는 시간과 AED 성능에 의해 결정된다. 표 2에서 Bin에 데이터를 할당하는 부분이 이중 반복문 형태를 띠고 있지만, 실제적으로는  $d_1$ 부터  $d_N$ 까지 차례대로 각 Bin에 할당시키는 것이다. 따라서 표 2의 알고리즘은  $O(N)$ 의 시간복잡도를 보인다. 다음으로 표 3에서 특정  $Bin_i$ 에 저장된 데이터들을  $SBin_{i,1}\sim SBin_{i,J}$ 에 할당하는 부분을 살펴보자. 이 부분도 이중 반복문 형태이지만,  $Bin_i$ 의 데이터들을 차례대로 자신의 SBin에 할당시키므로  $O(|Bin_i|)$ 의 시간복잡도를 보인다. 따라서 표 3의 알고리즘은  $O(\sum_{i=1}^K |Bin_i|)$ 의 시간복잡도를 보이는데, 각 데이터는 하나의 Bin에만 저장될 수 있기 때문에  $O(\sum_{i=1}^K |Bin_i|)$ 는  $O(N)$ 이 된다. 따라서 전체 알고리즘의 시간복잡도는  $O(N)$ 으로 결정된다.

TLBP 기법에서 평균 지연시간의 성능은  $\sum_{i=1}^K \sum_{j=1}^J (AED_{i,j})$ 로 계산된다. 여기서  $AED_{i,j}$ 는  $SBin_{i,j}$ 에 할당된 데이터들의 AED를 의미한다.  $SBin_{i,j}$ 에는  $|SBin_{i,j}|$  개의 데이터들이 배정되어 이들이  $CH_i$  대역폭의  $1/J$ 을 사용하면서 라운드-로빈 방식으로 방송되므로 이들의 방송주기는  $J \times |SBin_{i,j}|$ 가 된다. 그러므로 이들을 액세스하려면 방송주기의 반인  $J \times |SBin_{i,j}|/2$  tick 시간만큼 대기해야 한다. 따라서  $AED_{i,j}$ 는  $(J \times |SBin_{i,j}|/2) \sum_{d_q \in SBin_{i,j}} p_q$ 로 계산되고, TLBP 기법의 AED 성능은 식 (3)과 같이 결정된다.

$$AED = \sum_{i=1}^K \sum_{j=1}^J ((J \times |SBin_{i,j}|/2) \sum_{d_q \in SBin_{i,j}} p_q) \quad (3)$$

#### 4. 성능 결과 및 분석

이 장은 TLBP 기법의 성능을 평가하고 그 결과를 분석한다. Flat은 가장 기본이 되는 기법이며, DP는 각 채널의 데이터가 평형 방송되는 시스템 모델에서 최적 성능을 발휘하며, BP는 본 논문과 같이 저장소 개념을 사용하기 때문에 이 세 기법을 비교 대상으로 선정하였다. 성능평가 대상이 되는 각 기법은 C 언어로 구현하였으며, 각 실험은 인텔 Core2 Duo 2.8GHz의 프로세서, 2GB RAM, Windows XP가 설치된 PC 환경에서 실시되었다. 각 실험에서 사용된 입력변수들은 표 4에 제시되어 있다.

서버가 방송하는 각 데이터는 그 크기가 동일하고, 각 데이터의 액세스 확률은 불균등한 액세스 형태를 모델링하기 위해 많이 사용되는 zipf 분포 모델을 [1,4,5] 따른다고 가정한다. zipf 분포 모델에서 각 데이터의 액세스 확률은  $\theta$  값에 의해 결정되는데,  $\theta$  값은 각 데이터들의 액세스 확률이 어느 정도 편향되어 있는지를 나타내는 인수로서  $\theta$  값이 커질수록 불균등한 액세스의 정도가 심화된다. 각 기법은 이 액세스 확률에 근거하여 전체 데이터들을 분할하여 각 채널에서 방송할 데이터들을 결정한다. 데이터 할당 기법의 주요 평가기준은 어떤 데이터를 액세스하기 위하여 필요한 평균 지연시간과 채널에 데이터를 배정하기 위한 실행시간이다. 따라서 본 논문은 이 두 가지 성능을 중심으로 성능을 평가한다. 성능 평가에서는 기존 기법들과의 비교와 더불어, 제안된 기법이 최적 평균 지연시간의 성능에 얼마나 근접하는

지를 파악하기 위하여 식 (2)에 의해 계산되는 이론적 하한 값과의 성능 비교도 실시한다. 본 논문은 이 이론적 하한 값의 성능을 발휘하는 가상적인 기법을 OPT(optimal)로 표현한다.

표 4. 성능 평가를 위한 입력 변수

입력변수	의 미	설 정
N	데이터 항목의 수	1000~10000
K	채널의 수	1~64
$\theta$	zipf 분포의 $\theta$ 값	0.0~1.0
J	채널당 SBin의 수	1~64

4.1 채널 수의 변화에 따른 평균 지연시간

본 실험은 채널의 수를 1~64까지 변화시키면서, 각 기법의 평균 지연시간의 성능추이를 살펴보았다.  $\theta$  값은 편향적인 액세스 형태를 모델링하기 위한 대표 값으로 많이 사용되는 0.95로 설정하였으며[1,4, 5], J 값은 10으로 설정하였다. K의 변화에 따른 평균 지연시간의 성능이 그림 1에 있는데, TLBP, DP, BP 그리고 Flat 기법 순으로 우수한 성능을 보인다. TLBP는 다른 기법들에 비해 매우 우수한 성능을 보일 뿐만 아니라, OPT에 거의 필적하는 성능을 발휘하는 것을 알 수 있다. 참고로 K가 3 이상이 되면 TLBP와 OPT는 1% 이내의 성능 차이를 보인다.

K가 1일 경우에 DP, BP, 그리고 Flat 기법은 2500 tick의 성능을 보인다. 이는 한 채널에 5000 개의 데이터가 할당되어 동일 빈도로 방송되므로, 어떤 데이터를 액세스하려면 평균적으로 방송주기의 반인 2500 tick을 기다려야 하기 때문이다. 채널 수가 적을

경우에는 한 채널에 많은 데이터들이 할당되어 이들 간에 액세스 확률의 차이가 커지게 된다. 그런데 DP, BP, 그리고 Flat 기법은 같은 채널의 데이터들을 인기도에 관계없이 항상 동일 빈도로 방송한다. 그러나 TLBP 기법은 어떤 채널에서 방송해야 될 데이터 집합, 즉 Bin에 저장되어 있는 데이터들을 분할하여 다수 개의 SBin에 할당하고, 각 SBin 별로 평형 방송을 하는 방법을 채택함으로써 동일 채널의 데이터들을 다른 빈도로 방송하는 것이 가능하다. 따라서 TLBP는 다른 기법에 비하여 우수한 성능을 보인다. 참고로 K가 4인 경우에 TLBP는 DP 기법보다 11% 이상 우수한 성능을 보이며, BP나 Flat 기법보다 100% 이상 향상된 성능을 보인다.

채널 수가 늘어남에 따라 TLBP와 DP 기법의 상대적인 성능 차이가 줄어든다. 이는 K가 증가하면 동일 채널에 할당된 데이터들 간에 액세스 확률의 차이가 줄어들어 이 데이터들을 동일 빈도로 방송하는 DP의 단점이 어느 정도 완화되기 때문이다. K가 32 이상으로 증가하면, DP는 TLBP 기법과 거의 유사한 성능을 보이기 시작한다. DP와 Flat은 데이터 분할 그룹의 수는 같지만, DP는 액세스 확률이 높은 데이터들이 배정되는 채널에는 더 적은 수의 데이터들을 할당하여 이들의 상대적인 방송빈도를 높일 수 있는데 비하여 Flat은 무조건 각 채널에 N/K 개의 데이터를 배정한다. 따라서 Flat 기법은 K가 증가하더라도 모든 데이터들을 동일 빈도로 방송한다. 따라서 K가 증가함에 따라 DP는 Flat보다 훨씬 우수한 상대적인 성능을 보인다. BP는 액세스 확률을 반영하여 데이터들을 분할하므로 Flat 기법보다 우수한 성능을 보이기는 하지만, 각 채널에 배정된 데이터의 개수에 따른 지연시간의 차이를 고려하지 않아 그 성능 차이가 그다지 크지는 않다.

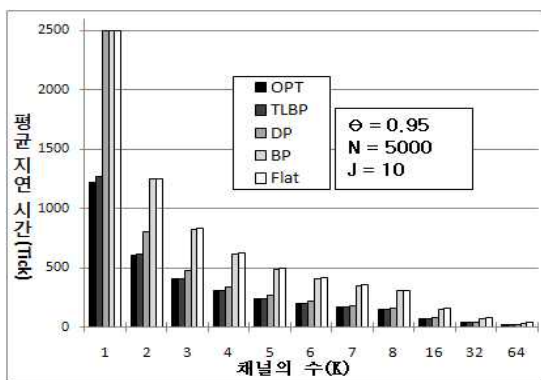


그림 1. 채널 수에 따른 평균지연시간

4.2  $\theta$  값 및 SBin 수의 변화에 따른 평균 지연시간

이 절은  $\theta$  및 J 값의 변화에 따른 평균 지연시간의 성능 결과를 살펴본다. 우선  $\theta$  값을 변화시키면서 살펴본 결과가 그림 2에 제시되어 있다.  $\theta$  값이 0이 되면, 각 데이터의 액세스 확률이 동일하게 되어 클라이언트가 액세스하는 데이터들의 분포는 균등 분포를 이루게 된다. 따라서 이 경우에는 각 기법들의 데이터 할당 정책의 차이에도 불구하고 모든 기법들이 동일한 성능을 보인다. 그러나  $\theta$  값이 증가함에 따라

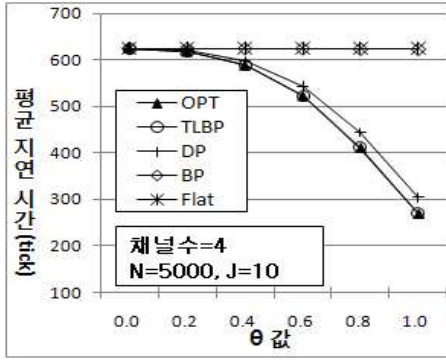


그림 2.  $\theta$  값에 따른 평균지연시간

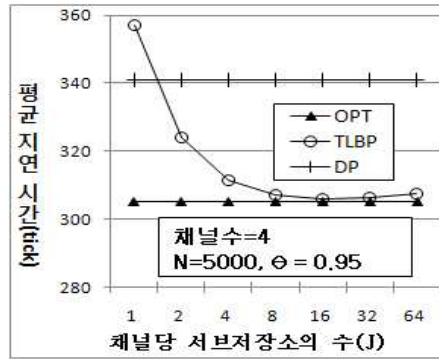


그림 3. J 값에 따른 평균지연시간

각 데이터들의 액세스 확률의 차이가 점점 커지게 되고, 이 액세스 확률을 고려하여 적절하게 채널에 데이터를 배정하는 TLBP와 DP는 다른 기법보다 훨씬 우수한 성능을 보이기 시작한다. 이 실험에서 TLBP는 전  $\theta$  구간에서 OPT에 거의 필적하는 성능을 보인다.

각 채널에 일정하게  $N/K$  개의 데이터를 할당하는 Flat 기법은  $\theta$  값에 관계없이 항상  $N/2K$  tick의 성능을 보인다. BP 기법은 액세스 확률이 높은 데이터가 할당된 채널에는 적은 수의 데이터를 배정하여 이들을 상대적으로 더 자주 방송한다. 그러나 BP 기법은 액세스 확률이 높은 데이터들을 확률 대비 너무 자주 방송함으로써 상대적으로 액세스 확률이 낮은 데이터들의 지연시간이 증가하게 된다. BP 기법이 Flat과 유사한 성능을 보이는 이유를 해석적으로 분석하기 위해, BP에서 각  $CH_i$ 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 한다. 따라서  $AED = \sum_{i=1}^K (N_i/2) \times S_i$ 가 된다. 그런데 데이터 분할 정책에 의해  $S_i = 1/K$ 이 되어  $AED = (1/2K) \sum_{i=1}^K N_i$ 가 된다. 그런데  $\sum_{i=1}^K N_i$ 는  $N$ 으로 계산되므로 BP 기법의 AED는 대략  $N/2K$ 가 되어, Flat과 거의 유사한 성능을 보이는 것이다.

J 값을 변화시키면서 살펴본 TLBP 기법의 성능이 그림 3에 있는데, OPT와 DP는 성능비교 차원에서 그 결과를 함께 제시한다. TLBP 기법은 각 채널의 데이터들을 이상적으로 방송한다는 전제하에 Bin 용량을 결정하고, 전체 데이터들을 분할하여 각 채널에 할당한다. 그리고 각 채널의 데이터들을 다시 J 개의 그룹으로 분할하여 SBIN에 할당하고, 각 SBIN 별로

평형 방송한다. 그런데 J 값을 작게 설정할 경우에는 각 채널의 데이터들을 최적으로 방송하지 못하게 된다. 극단적으로 J가 1일 경우에 TLBP는 같은 채널의 데이터들을 동일 빈도로 방송한다. 따라서 J가 작을 경우에는 TLBP와 OPT의 성능 차이가 큰 편이다. DP는 각 채널의 데이터가 동일 빈도로 방송된다는 전제하에 최적으로 데이터를 채널에 할당한다. 따라서 J가 1일 경우에 DP 기법은 TLBP보다 우수한 성능을 보인다.

J가 증가함에 따라 TLBP 기법의 성능은 향상되는데, J가 16 이상이 되면 더 이상 성능이 향상되지 않는다. 오히려 J가 32 이상이 되면 성능이 저하되기 시작하는데, 이는 채널에 배정된 데이터 수에 비해 SBin의 수가 너무 많기 때문이다. 적절하게 J 값을 설정하는 것이 성능에 영향을 미치는데, 이 값은 Bin에 할당된 데이터 수와  $\theta$  값을 고려하여 결정해야 한다. 본 논문의 실험에서는 J 값을 10~20 정도로 설정할 때 우수한 성능을 보인다. Bin에 배정된 데이터 수에 따라 각 Bin 별로 SBin의 수를 다르게 설정하는 것이 가장 우수한 성능을 발휘하지만, 그 성능 차이가 크지 않기 때문에 본 논문은 모든 Bin의 J 값을 동일하게 설정하였다.

### 4.3 채널에 데이터를 배정하기 위한 실행시간

이 절은 채널에 데이터를 할당하기 위한 실행시간을 비교 분석한다. 실행시간 결과는 각 실험을 10번 실행하여 측정된 시간의 평균 값으로 제시하였다. 본 실험에서  $\theta$  값은 실행시간에 큰 영향을 주지 않기 때문에 0.95로 고정하였다. 그리고 채널 수를 10으로 설정하고 데이터 항목의 수에 따른 실행시간의 변화



를 살펴본 결과가 그림 4에 제시되어 있다. Flat 기법은 전체 데이터를 균등하게 분할하여 각 채널에 배정하므로  $N$ 이 증가하더라도 항상 일정한 성능을 보이는데, 다른 기법에 비해 훨씬 적은 실행시간을 보인다.  $O(N)$ 의 시간복잡도를 보이는 BP와 TLBP 기법은  $N$ 이 증가함에 따라 이에 비례하여 실행시간도 증가한다. 비록 두 기법이 동일한 시간복잡도를 보이지만, TLBP 기법은 Bin 용량을 결정하기 위하여 각 데이터의 액세스확률의 제곱근 값과 이 합을 계산해야 하므로 BP보다 더 많은 실행시간을 필요로 한다. DP 기법은  $O(KN^2)$ 의 시간복잡도를 보이므로  $N$ 이 증가함에 따라,  $N^2$ 에 비례하여 실행시간이 증가하게 될 것이다. 그림 4에 제시된 실행시간은 대체로 이와 일치하는 결과를 보인다. DP는 다른 기법에 비해 실행 부담이 매우 큰 편인데, 이 단점은  $N$ 이 증가함에 따라 더욱 악화된다.  $N$ 을 5000으로 고정하고  $K$ 에 따른 실행시간의 변화를 살펴본 결과가 그림 5에 제시되어 있다. Flat과 DP 기법은 대체로  $K$ 에 비례하는 실행시간을 필요로 하는 반면, BP와 TLBP 기법은

$K$ 와 무관하게 거의 일정한 실행시간을 필요로 한다. TLBP 기법은 전체 데이터들을 두 단계로 세밀하게 분할하기 때문에 BP와 Flat 기법에 비해서는 더 많은 실행부담을 보인다. 그러나 그림 1과 그림 2에서 보는 것과 같이 TLBP 기법은 BP나 Flat보다 훨씬 우수한 평균 지연시간의 성능을 보인다. TLBP는 BP와 Flat 기법에 비해서는 실행부담을 회생하면서 평균 지연시간의 성능을 향상시킨 기법으로 볼 수 있다. TLBP는 그림 4의 환경에서 DP 기법과 비교하여,  $1/100,000 \sim 1/10,000$  정도로 훨씬 적은 실행 부담을 보인다. 그리고 그림 1과 그림 2에서 보는 것과 같이 우수한 평균 지연시간의 성능을 보인다. 뿐만 아니라 TLBP 기법은 대부분의 환경에서 평균 지연시간의 이론적 하한 값에 거의 필적하는 성능을 보인다. 이와 같은 점들을 고려한다면 TLBP 기법은 합리적인 실행 부담을 보이면서도 우수한 평균 지연시간의 성능을 발휘하여 매우 바람직한 성능 특성을 보인다고 할 수 있다.

5. 결 론

방송 방식의 데이터 전달 기법은 방대한 클라이언트들을 지원해야 하는 환경이나 이동 컴퓨팅 환경에서 정보 전달을 위한 주요 기술로 인정받고 있다. 그러나 많은 데이터를 방송해야 할 경우에 방송 채널의 순차성으로 인하여 클라이언트가 원하는 데이터를 수신하기까지의 예상지연시간이 크게 증가할 수 있다. 본 논문은 이 지연시간을 줄이기 위하여 다중 방송채널 환경을 위한 데이터 할당 기법에 대한 연구를 진행하였다. 본 논문은 이론적 평균 예상지연시간의 성능을 계산한 후, 이를 각 채널이 균등하게 배분하여 담당할 때 최적의 AED 성능을 발휘할 것이라는 휴리스틱을 적용하였다. 그리고 저장소-적재 알고리즘을 두 단계로 적용하여 TLBP로 명명된 새로운 데이터 할당 기법을 제안하고 성능 평가를 실시하였다. 제안된 기법은 다른 기법과 비교하여 채널에 데이터를 할당하기 위한 합리적인 실행 부담을 보이면서도 우수한 AED 성능을 발휘하는데, 특히 물리적 채널의 수가 적을 경우에 다른 기법보다 훨씬 우수한 성능을 보인다. 그리고 TLBP 기법은 AED의 이론적 하한 값에 거의 필적할 정도로 우수한 성능을 보인다. 본 논문은 단일 데이터를 액세스하는 환경에서 지

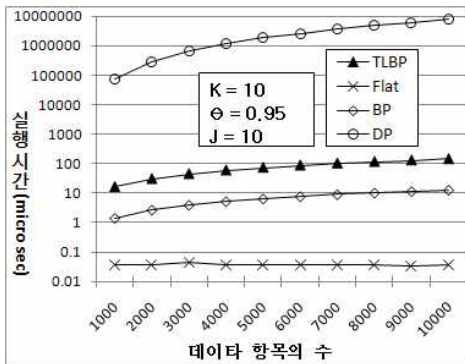


그림 4. 데이터 수에 따른 실행시간

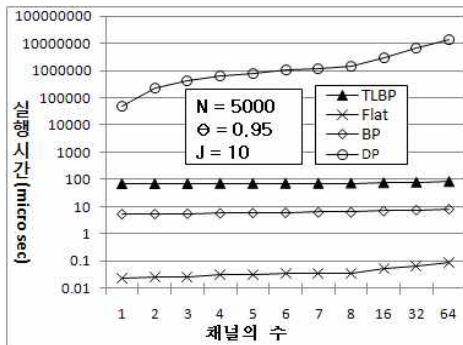


그림 5. 채널 수에 따른 실행시간

연시간을 줄이기 위한 연구를 진행하였다. 향후 과제로는 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," Proc. ACM OOPSLA Conf., pp. 94-105, 1997.

[ 3 ] M. Franklin and S. Zdonik, "Data in Your Face: Push Technology in Perspective," Proc. ACM SIGMOD Conf., pp. 516-519, 1998.

[ 4 ] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communications environments," Proc. ACM SIGMOD Conf., pp. 199-210, 1995.

[ 5 ] N.H. Vaidya and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environments," *Wireless Networks*, Vol.5, No.3, pp. 171-182, 1999.

[ 6 ] J.H. Min, J.Y.Hwang, and D.W. Paik, "A Data Scheduling Algorithm Considering the Request Probability for Data Broadcasting," *Journal of Korea Multimedia Society* Vol.11, No.3, pp. 398-403, 2008.

[ 7 ] 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.

[ 8 ] 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.

[ 9 ] 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, 2002.

[10] 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, 2005.

[11] S. Wang and H.L. Chen, "Near-Optimal Data Allocation Over Multiple Broadcast Channels," *Computer communications*, Vol.29, No.9, pp. 1341-1349, 2006.

[12] C. Hsu, G. Lee, and A.L.P. Chen, "A Near Optimal Algorithm for Generating Broadcast Programs on Multiple Channels," Proc. ACM 10th Int'l Conf. CIKM, pp. 303-309, 2001.

[13] D. Johnson, A. Demers, J. Ullman, M. Garey, and M. Graham, "Worst-Case Performance Bounds for Simple one Dimensional Bin Packing Algorithms," *SIAM J. on Computing*, Vol. 3, No.4, pp. 299-325, 1974.

[14] J.W. Wong, "Broadcast Delivery," Proc. of the IEEE, Vol.76, No.12, pp. 1566-1577, 1988.



권 혁 민

1984년 서울대학교 제어계측공학과 학사  
 1994년 한국과학기술원 정보및통신공학과(석사)  
 1998년 한국과학기술원 정보및통신공학과(박사)

1984년~1991년 대우전자 중앙연구소 컴퓨터개발부 선임연구원  
 1999년~현재 세명대학교 정보통신학부 부교수  
 관심분야: 데이터베이스, 모바일 컴퓨팅.