

<http://dx.doi.org/10.7236/JIWIT.2012.12.1.17>

JIWIT 2012-1-3

다중 방송 채널 환경을 위한 유사 최적화 데이터 할당 기법

A Near Optimal Data Allocation Scheme for Multiple Broadcast-Channel Environments

권혁민*

Hyeokmin Kwon

요약 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 채널에 적절하게 데이터를 할당하기 위한 주제를 연구하여 NODA(near optimal data allocation)로 명명된 새로운 데이터 할당 기법을 제안한다. 제안된 기법은 우선 각 방송 채널이 방송해야 하는 데이터들을 결정하기 위하여 평균 예상지연시간의 이론적 하한 값에 기초하여 전체 데이터들을 K 개의 그룹으로 분할한다. 그리고 나서 NODA는 동일 방송 채널에 할당된 데이터들을 다른 빈도로 방송하기 위하여, 각 그룹의 데이터들을 확장된 동적 프로그래밍 알고리즘을 사용하여 B 개의 그룹으로 분할한다. 제안된 기법은 동일 채널에 할당된 데이터들을 그들의 인기도를 반영하여 방송할 수 있기 때문에 평균 예상지연시간을 최소화할 수 있다.

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. This paper explores the issue of designing proper data allocation on multiple broadcast channels to reduce this wait time, and proposes a new data allocation scheme named near optimal data allocation(NODA). The proposed scheme first partitions all data items in K group based on the theoretical lower-bound of the average expected delay to determine data items which each broadcast channel has to broadcast. Then, NODA further partitions each group of data items in B groups using extended dynamic programming algorithm to broadcast data items allocated on the same broadcast channel in different frequencies. The proposed scheme is capable of minimizing the average expected delay time since it can broadcast data items allocated on the same channel reflecting their popularities.

Key Words : Data allocation, broadcast schedule, dynamic programming.

1. 서론

이동 컴퓨팅(mobile computing) 환경에서 방송 방식

에 기초한 데이터 전달 기법은 클라이언트에게 효과적으로 데이터를 전송하는 기술로 인정받고 있다^[1-11]. 방송 방식의 데이터 전달 기법을 채택한 정보 시스템에서 서

*총신회원, 세명대학교 정보통신학부
접수일자 2012.1.5, 수정일자 2012.2.5.
게재확정일자 2012.2.10

Received: 5 January 2012 / Revised: 5 February 2012 /

Accepted: 10 February 2012

*Corresponding Author: hmkwon@semyung.ac.kr

Dept. of Information and Communication System, Semyung University, Korea

버는 방송 채널을 통하여 순차적, 그리고 반복적으로 데이터들을 방송하고 이동 클라이언트는 자신이 원하는 데이터가 방송 채널에 나타나면 이를 액세스한다.

방송 시스템에서 데이터 전송 비용이나 클라이언트가 어떤 데이터를 액세스하는데 걸리는 지연 시간은 클라이언트의 수에 영향을 받지 않기 때문에 시스템 확장성이 매우 우수하다. 그러나 클라이언트들은 방송 채널의 데이터들을 순차적으로 액세스해야 하기 때문에 클라이언트가 어떤 데이터를 수신하기 위해서는 원하지 않는 데이터의 방송이 지나가기를 기다려야 한다. 방송에 기초한 정보 시스템의 주요 성능지수는 평균 예상지연(average expected delay: AED) 시간인데, 이는 클라이언트가 특정 데이터의 액세스를 원하는 시간부터 실제 이를 액세스하는데 걸린 평균 대기시간을 의미한다.

그 동안 이 대기 시간을 줄이기 위하여 단일 방송채널 환경에서 방송 스케줄링 기법에 관한 연구들이^[1-5] 수행되었다. 방송 스케줄링 기법에서 방송 서버는 특정 시점에 어떤 데이터를 방송할 것인가를 결정하기 위하여 방송 스케줄을 구성해야 한다. 효율적인 방송 스케줄은 클라이언트의 평균 대기시간을 최소화 할 수 있는데, 기존의 연구들은 각 데이터의 액세스 확률에 근거하여 자주 사용되는 데이터를 더 빈번하게 방송하는 방법을 제시하여 AED 성능을 개선했다. 그러나 최적으로 방송 스케줄을 구성하더라도, 서버가 방송해야 할 데이터의 수가 증가함에 따라 예상 지연시간은 크게 증가할 수밖에 없다. 이의 효과적인 해결책 중의 하나는 물리적인 방송 채널의 수를 증가시켜 각 채널의 방송 스케줄을 짧게 구성하는 것이다.

다중 방송채널 환경에서는 각 채널의 방송 스케줄을 구성하는 문제와 더불어 전체 데이터들을 어떻게 분할하여 각 방송 채널에 할당한 것인가의 문제를 효율적으로 해결해야 한다. 다중 방송채널 환경에서 단순하게 전체 데이터들을 동일한 수로 분할하여 각 채널에 배정하면 평균 지연시간은 채널 수에 선형적으로 비례하여 줄어들 것이다. 그러나 일반적으로 클라이언트에서 각 데이터의 액세스 확률은 다르게 마련이다. 만일 액세스 확률이 높은 데이터들이 배정되는 채널에 더 적은 수의 데이터를 배정하면, 이들을 더 높은 빈도로 방송하는 것이 가능하며 AED 성능을 크게 개선할 수 있을 것이다.

최근에는 이와 같은 관점에서 전체 데이터들을 분할해 각 방송 채널에 할당하기 위한 데이터 할당 기법에 대

한 많은 연구들이 수행되었다^[6-9]. 이들은 대부분 같은 채널에 할당된 데이터들은 평형(FLAT) 방식으로 동일한 빈도로 방송하는 모델을 가정하고, AED가 최소화되는 방향으로 전체 데이터들을 액세스 확률에 근거하여 채널 수 만큼 다수 개의 그룹으로 분할하여 각 채널에 배정하였다. 그런데 이동 컴퓨팅 환경의 특성상 방송해야 될 데이터의 수는 물리적 채널의 수에 비해서 매우 크기 때문에 한 채널에 많은 수의 데이터가 배정될 가능성이 많다. 이 경우에는 동일 채널에 할당된 데이터들 간에 액세스 확률의 차이는 커질 수밖에 없다. 따라서 같은 채널의 데이터들을 그들의 인기도에 관계없이 동일 빈도로 방송한다면 AED 성능은 심각하게 저하될 것이다.

TLDP(two level dynamic programming) 기법은^[10] 이와 같은 단점을 극복하기 위하여 전체 데이터들을 두 단계로 분할하여 동일 채널의 데이터들도 그들의 인기도를 반영하여 방송 빈도를 다르게 할 수 있다. 따라서 TLDP는 다른 기법보다 훨씬 우수한 AED 성능을 보인다^[10]. 그러나 TLDP 기법은 첫 번째 단계의 데이터 분할이 최적이지 아닐 뿐만 아니라, 각 채널에서 방송할 데이터 집합을 결정하기 위하여 전체 데이터들을 분할하는데 시간이 많이 걸린다는 단점이 있다.

본 논문은 이와 같은 기존 기법들의 단점을 개선하기 위하여, 각 방송 채널을 동일 대역폭을 갖는 다수 개의 서브채널로 분할하는 개념을 사용한다. 그리고 서브채널 별로 평형 방송을 하는 시스템 모델을 고안하여 NODA(near optimal data allocation)로 명명된 새로운 데이터 할당 기법을 제안한다. 본 논문은 우선 다중 방송채널 환경에서 평균 지연시간의 이론적 하한 값을 계산하는 수식을 소개한다. 그리고 이 이론적 하한 값을 각 채널이 균등하게 배분하여 담당하도록 전체 데이터들을 방송 채널의 수 만큼의 그룹으로 분할하여 각 방송 채널에 할당한다. 그리고 나서 TLDP 기법에서 제안된 확장된 동적 프로그래밍 알고리즘을^[10] 이용하여, 각 채널에 할당된 데이터들을 다시 분할하여 자신의 서브채널에 할당한다. 제안된 기법은 서브채널별로 평형 방송을 하는 방식을 통하여 동일 방송채널에 할당된 데이터들도 인기도를 반영하여 다른 빈도로 방송할 수 있다. 따라서 제안된 기법은 다른 기법들에^[6-9] 비해 크게 향상된 AED 성능을 발휘할 수 있고, 대부분의 환경에서 최적(optimal)에 거의 필적하는 성능을 보일 수 있다. 본 논문은 제안된 기법의 성능을 평가하기 위하여 시뮬레이션 모델에 기초하

여 제안된 기법과, 기존 기법, 그리고 이론적 하한 값의 성능을 비교한다.

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

II. 관련 연구

이 장에서는 기존에 제안된 기법들에 대하여 기술한다. 관련 연구 및 본 논문이 제안하는 기법은 다음 세 항목의 시스템 모델과 용어를 공통적으로 사용한다.

a) 방송 데이터들은 $d_q(1 \leq q \leq N)$ 로 표현되는 동일 크기의 N 개의 데이터로 구성되며, d_q 의 액세스 확률을 p_q 로 표현한다. 그리고 유사한 액세스 확률을 가진 데이터를 동일 채널에 배정하기 위해 데이터는 p_q 의 내림차순으로 정렬되어 있다고 가정한다.

b) 방송 서버에는 동일 대역폭을 갖는 K 개의 방송채널이 존재하는데, 이를 $CH_c(1 \leq c \leq K)$ 로 표현한다. 그리고 임의의 한 데이터는 한 채널에만 할당될 수 있다.

c) 한 방송채널에서 하나의 데이터를 방송하는데 걸리는 논리적인 시간을 1 tick으로 정의하고, AED 시간 단위로 tick을 사용한다.

단일채널 환경을 위해 제안된 평형(FLAT) 기법은^[1] 채널에 배정된 데이터들을 라운드-로빈(round-robin) 방식으로 주기적으로 방송한다. 따라서 모든 데이터들은 액세스 확률에 관계없이 동일 빈도로 방송된다. 이 기법은 다중채널 환경으로 용이하게 확장될 수 있는데, 다중채널의 평형 기법은 각 방송채널에 동일 개수의 데이터를 배정하고, 각 채널은 이들을 평형 방송한다. 평형 기법은 단순하여 구현이 용이하다는 장점이 있지만, 모든 데이터들을 그들의 인기도에 관계없이 동일 빈도로 방송하므로 AED 성능이 다른 기법들에 비하여 심각하게 저하되는 단점이 있다.

최근 들어 다중 방송채널 환경을 위한 연구들이 많이 수행되었는데, 대표적으로 VF^K(Variable-Fanout with the constraint K)^[6], 동적 프로그래밍(Dynamic Programming: DP)^[7], 그리고 제한된 동적프로그래밍(Restricted Dynamic Programming: RDP)^[8-9] 기법들이

있다. 이 기법들은 자신의 정책에 따라 전체 데이터들을 K 개의 그룹으로 분할하여 각 방송채널에 할당하고, 각 채널은 자신에게 할당된 데이터들을 평형 방송한다.

VF^K 기법은^[6] 가변적인 팬아웃(fanout) 트리를 구성하여 다중 채널에 데이터를 할당한다. 이 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^[7] 각 채널의 데이터가 평형 방송된다는 가정 하에 예상 지연시간이 최소화되도록 전체 데이터들을 K 개의 그룹으로 분할하여 각 채널에 배정한다. 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 계산해야 하기 때문에 이의 실행을 위한 시간복잡도가 $O(KN^2)$ 이 되어 실행부담이 크다는 단점이 있다. 이 연구에서는 이 실행부담을 줄이기 위한 Greedy 알고리즘도 함께 제안했다. Wang과 Chen의 연구에서도^[8-9] DP의 수행시간 측면의 단점을 개선하기 위하여, 이론적으로 최적 AED 성능을 추정하여 데이터를 분할하여 각 방송채널에 할당하는 RDP 기법을 제안했다. RDP는 $O(N \log K)$ 의 시간복잡도를 보이며 거의 최적에 가깝게 데이터를 분할한다.

TLDP에서는^[10] 각 방송채널의 데이터들을 동일 빈도로 방송하는 기존 기법들의^[6-9] 단점을 극복하기 위하여, 동일 방송채널의 데이터들을 그들의 인기도를 반영하여 다른 빈도로 방송할 수 있는 스케줄링 기법을 제안했다. 이를 위해 TLDP에서는 DP 알고리즘을 확장한 EDP(extended dynamic programming) 알고리즘을 제안하여, 전체 데이터들을 두 단계로 세밀하게 분할하여 스케줄링 정책에 반영했다. 비록 TLDP가 FLAT이나 DP 기법에 비하여 훨씬 우수한 AED 성능을 발휘하지만^[10], 데이터를 분할하는데 필요한 실행 부담이 다른 기법에 비하여 크다는 단점이 있다.

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^[5]. Hsu et al.은 이 연구를 다중채널 환경으로 확장하여,

AED의 이론적 하한(lower-bound) 값이 식 (1)로 결정된다는 것을 증명했다^[11]. 여기서 f_i 는 d_i 의 액세스 횟수를 의미한다. 그러나 주어진 전체 조건들을 만족하면서 모든 데이터들을 방송하는 것은 실재는 거의 불가능하다. 비록 이 값이 실현될 수는 없지만 어떤 기법의 상대적인 성능을 평가할 때는 매우 유용하다.

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

III. 새로운 데이터 할당 기법

이 장에서는 본 논문이 제안하는 새로운 데이터 할당 기법에 대하여 기술한다. 제안된 기법은 거의 최적으로 데이터를 분할하여 서브채널에 할당해 주기 때문에 NODA(near optimal data allocation)라고 명명한다.

1. 기본 시스템 모델

기존 기법들은^[6-9] 같은 방송채널에 할당된 데이터들을 동일빈도로 평형 방송을 하는데 비하여, 제안된 기법은 서브채널의 개념을 도입하여 각 서브채널별로 평형 방송하는 모델을 사용한다. 이를 위해 본 논문은 2 장의 시스템 모델에 다음을 추가한 모델을 사용한다.

a) 각 방송채널은 동일 대역폭을 갖는 B 개의 서브채널로 구성된다고 가정하며, $CH_c(c=1 \sim K)$ 의 서브채널을 $SCH_{c,b}(b=1 \sim B)$ 로 표현한다.

b) $CH_c(1 \leq c \leq K)$ 에서 방송되는 데이터 집합을 G_c 로 표현하며, $SCH_{c,b}(c=1 \sim K, b=1 \sim B)$ 에서 방송되는 데이터 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.

c) $SCH_{c,b}$ 는 $G_{c,b}$ 의 데이터들을 라운드-로빈(round-robin) 방식으로 평형 방송하는데, 이를 위하여 $G_{c,b}$ 에 할당된 데이터들은 큐 형태로 관리된다.

d) 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 서브채널 $SCH_{c,b}$ 가 방송하는 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

본 논문은 각 방송채널을 가상의 B 개의 서브채널로 간주한다. 따라서 시스템에는 $K \times B$ 개의 서브채널들이 존재하며, NODA는 전체 데이터들을 적절히 분할하여 각 서브채널에 할당한다. 다수 개의 서브채널에 데이터를 할당하는 문제는 그림 1에서 보는 것처럼 데이터들을 분할하는 문제로 볼 수 있다. 본 논문은 그림 1과 같이 두 단계로 데이터를 분할하는데, 우선 전체 데이터들을 채널 수 만큼 K 개의 그룹으로($G_1 \sim G_K$) 분할하여 각 방송 채널에서 방송할 데이터들을 결정한다. 그리고 나서 각 $G_c(c=1 \sim K)$ 를 다시 서브채널의 수만큼 B 개의 그룹으로($G_{c,1} \sim G_{c,B}$) 분할하여 각 서브채널에 순서대로 할당한다. 각 단계에서 데이터를 분할하는 방법은 다음 절에서 자세히 설명한다.

본 논문은 각 방송채널이 담당할 데이터를 결정하기 위하여 데이터가 분할되는 지점을 $ccut_c(c=1 \sim K)$ 로 표현하며, 각 G_c 를 서브채널에 할당하기 위하여 분할되는 지점을 $sccut_{c,b}(b=1 \sim B)$ 로 표현한다. 알고리즘 기술의 편의상 분할점(cut point)을 각 데이터 그룹의 마지막 데이터의 인덱스로 표현하자. 예를 들어 그림 1에서 $ccut_1$ 은 3이고 $sccut_{c,1}$ 은 p 이다.

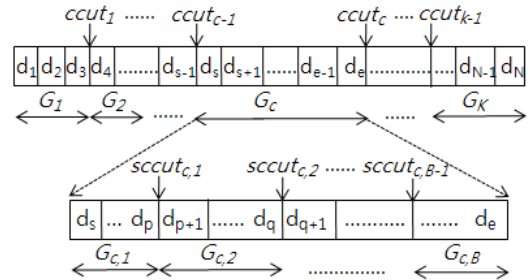


그림 1. N 개의 데이터를 $K \times B$ 서브채널에 할당하기 위한 데이터 분할 문제

Fig. 1. Data partition problem for allocating N datum on $K \times B$ sub-channels

본 논문은 각 서브채널이 자신에게 할당된 데이터들을 라운드-로빈(round-robin) 방식으로 평형 방송하는 모델을 사용한다. 그러나 본 논문에서 서브채널은 개념상 존재하는 것이지만 물리적으로 방송채널을 분할하는 것은 아니다. 제안된 기법은 각 방송채널 별로 방송 스케줄러를 운영하는데, 각 방송 스케줄러가 기동되면 그림 2의 알고리즘을 실행하여, 지속적으로 데이터들을 방송하여

서브채널의 개념을 실현한다. 여기서 b 는 서브채널 번호로서 0으로 초기화되어 있다고 가정한다. 각 채널의 방송 스케줄러는 방송할 데이터의 선정이 필요할 때마다 우선 어떤 서브채널의 데이터를 방송할 것인가를 결정한다. 그리고 선택된 서브채널의 맨 앞에 있는 데이터를 선정하여 방송하고, 그 데이터를 맨 뒤로 이동한다. 방송 스케줄러는 각 서브채널을 B tick 주기마다 선택하며, 동일 서브채널에 할당된 데이터들을 라운드-로빈 방식으로 공평하게 방송한다. 따라서 동일 서브채널에 할당된 데이터들은 방송채널 대역폭의 $1/B$ 을 사용하며 동일 빈도로 방송된다.

```

while (true)
  choose a data group  $G_{c,b}$  such that  $b=(b \text{ mod } B)+1$ 
  choose a data item  $d_i$  from the front of  $G_{c,b}$ 
  broadcast  $d_i$  and wait for finishing the broadcast
  Dequeue item  $d_i$  from the front of  $G_{c,b}$ 
  Enqueue  $d_i$  at the rear of  $G_{c,b}$ 
End while
    
```

그림 2. CH_c 의 방송 스케줄링 알고리즘
Fig. 2. Broadcast Scheduling Algorithm of CH_c

3. 유사 최적화 데이터 할당 기법

이 절에서는 전체 데이터들을 두 단계로 분할하여 각 서브채널에 할당하는 방법에 대하여 기술한다. Hsu et al.^[11]이 증명한 식 (1)에서 $(f_i / \sum_{j=1}^N f_j)$ 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 식 (2)로 표현되는 이론적 최적 성능인 $AED_{optimal}$ 실현을 목표로 하고 있다. 각 방송채널에 데이터를 분배할 때 균형을 갖추지 못하고 특정 채널에 너무 많은 데이터를 배정하면, 해당 채널에 과부하가 발생하여 전체 AED 성능이 심각하게 저하될 것이다. 이와 같은 점을 고려한다면, 각 방송채널이 $AED_{optimal}$ 을 균등하게 배분하여 담당할 때 최적 성능을 발휘할 것이다. 이를 위해서는 각 방송채널이 $AED_{optimal}/K$ 의 예상지연시간을 갖도록 데이터가 분배되어야 한다. 본 논문은 이 휴리스틱(heuristic)을 각 방송채널에서 방송할 데이터의 집합 $G_1 \sim G_K$ 를 결정할 때 사용한다. 우선 각 방송채널은 자신

에게 배정된 데이터들을 이상적으로 방송한다고 가정하는데, 이는 다음에 설명할 가상의 서브채널에 최적으로 데이터를 할당함에 의해 실현된다. 특정 방송채널 CH_c 가 자신에게 분배된 데이터들을 이상적으로 방송할 때의 AED를 $AED_{optimal}^c$ 로 표현하자. $AED_{optimal}^c$ 은 $(1/2) \times (\sum_{d_i \in G_c} \sqrt{p_i})^2$ 로 계산된다^[5]. 그런데 $\sum_{d_i \in G_c} \sqrt{p_i}$ 의 값이 $(\sum_{i=1}^N \sqrt{p_i})/K$ 로 결정되면, $AED_{optimal}^c$ 은 $AED_{optimal}/K$ 이 된다. 즉, 각 방송채널에 분배된 데이터들의 액세스 확률의 제공근 합이 동일하다면 최적 분배가 이루어지는 것이다.

이와 같은 개념으로 각 방송채널에서 담당할 데이터들을 결정하기 위하여 그림 1의 각 $ccut_i (i=1 \sim K-1)$ 를 결정하기 위한 알고리즘이 그림 3에 제시되어 있다. 이 알고리즘은 우선 $\sqrt{p_i}$ 와 $\sum_{i=1}^N \sqrt{p_i}$ 값을 계산한 후, 각 방송채널이 담당해야 할 데이터들의 액세스 확률의 제공근 합, 즉 $(\sum_{i=1}^N \sqrt{p_i})/K$ 를 계산하여 $AvgChSumSr$ 에 저장한다. 그림 1에서 보는 것과 같이 $ccut_i$ 는 i 의 범위가 $1 \sim K-1$ 만이 의미가 있는데, 논문 기술의 편의상 $ccut_0 = 0$, $ccut_K = N$ 으로 설정한다.

```

//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
//determine  $ccut_0 \sim ccut_K$ 
AvgChSumSr = SumSr / K
 $ccut_0 = 0$ ,  $ccut_K = N$ 
for each cut point  $i$  from 1 to  $K-1$ 
  ChSumSr = 0
  for each data  $j$  from  $ccut_{i-1}+1$  to N
    ChSumSr = ChSumSr +  $sr_j$ 
    if (ChSumSr >= AvgChSumSr)
       $ccut_i = j$ 
      break;
    end if
  end for
end for
    
```

그림 3. 첫 번째 단계의 데이터 분할 알고리즘
Fig. 3. First stage of data partitioning algorithm

알고리즘 설명을 위하여 $ccut_i$ 을 결정하는 과정을 살펴보자. 우선 데이터들의 액세스 확률의 제공근 합을 저장하기 위한 $ChSumSr$ 을 0으로 초기화한다. 그리고 나서

d_{ccut_0+1} , 즉 d_1 부터 d_N 까지 순서대로 개념상 G_1 에 포함시키면서 $ChSumSr$ 을 재계산한다. $ChSumSr$ 값이 $AvgChSumSr$ 과 같아지거나 이를 초과하게 되면 맨 마지막에 포함시킨 데이터가 $ccut_1$ 로 결정된다. $ccut_1$ 이 결정되고 나면 다시 $ChSumSr$ 을 0으로 초기화하고 $d_{ccut_1+1} \sim d_N$ 을 순서대로 G_2 에 포함시키면서 $ccut_2$ 를 결정한다. 이와 같은 방식으로 $ccut_1$ 부터 $ccut_{K-1}$ 까지 순차적으로 구할 수 있다. 각 $ccut_i(i=0 \sim K)$ 가 결정되면 CH_c 가 방송하는 데이터 집합 G_c 는 $d_{ccut_{c-1}+1} \sim d_{ccut_c}$ 로 결정되는 것이다.

다음으로 각 방송채널의 서브채널에서 방송할 데이터들을 결정하기 위하여 각 $G_c(c=1 \sim K)$ 를 B 개의 그룹으로 분할하여 CH_c 의 각 서브채널에 할당해야 한다. 각 G_c 를 분할하는 방법은 TLDP 기법에서 제안된 EDP (extended dynamic programming) 알고리즘에^[10] 기초하고 있다. EDP 알고리즘 설명을 위해 그림 1과 같이 채널 CH_c 가 담당하는 데이터들이 d_s 에서 d_e 까지의 데이터들로 결정되었다고 가정하자. CH_c 의 어떤 서브채널 $SCH_{c,b}$ 에 $d_i \sim d_j$ 의 데이터들이 할당되어 있을 경우 $AED_{c,b}$ 값의 $1/B$ 을 $C_{i,j}$ 로 표현하자. $SCH_{c,b}$ 는 CH_c 대역폭의 $1/B$ 를 사용하여 자신에게 할당된 데이터들을 평형 방송하므로, 이 데이터들은 B tick당 한 개씩 주기적으로 방송된다. 따라서 $SCH_{c,b}$ 의 방송 주기는 $B \times (j-i+1)$ 가 된다. 그런데 어떤 데이터를 액세스하려면 평균적으로 그 채널의 방송주기의 반을 기다려야 한다. 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 개의 서브채널 중 앞부터 k 개의 서브채널에 $d_s \sim d_j$ 의 데이터들을 할당할 때의 최적 AED 값의 $1/B$ 을 $opt_left_{k,s,j}$ 로 표현하자. $opt_left_{1,s,j}$ 는 한 개의 서브채널만 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 (3)과 같이 표현된다. 그림 1에서 B-1 개의 분할점, $sccut_{c,1} \sim sccut_{c,B-1}$ 은 $opt_left_{k,s,j}(2 \leq k \leq B)$ 를 구하는 과정에서 우측 수식의 최소 값을 결정짓는 r 값들에 의해 결정된다.

$$opt_left_{k,s,j} = \min \{opt_left_{k-1,s,r} + C_{r+1,j}\} \\ \text{where } s+k-2 \leq r \leq j-1 \quad (3)$$

식 (3)에 의해 B-1 개의 분할점이 구해지면, $G_{c,1}$ 은 $d_s \sim d_{sccut_{c,1}}$ 의 데이터들로 결정되고, $G_{c,b}(b=2 \sim B-1)$ 는 $d_{sccut_{c,b-1}+1} \sim d_{sccut_{c,b}}$ 으로, $G_{c,B}$ 는 $d_{sccut_{c,B-1}+1} \sim d_e$ 의 데이터들로 결정된다. 이와 같은 방식으로 NODA는 TLDP에서^[10] 제안된 EDP 알고리즘을 이용하여 각 G_c 를 B 개의 그룹으로 분할하여 각 서브채널에 할당한다.

4. 알고리즘 실행속도 및 AED 성능에 관한 고찰

데이터 할당 기법의 우수성은 데이터를 분할하는데 필요한 실행시간과 AED 성능에 의하여 결정된다. 각 채널에서 방송할 데이터들을 결정하는 그림 3의 알고리즘은 이중 반복문의 형태를 띠고 있지만, 실제적으로는 $d_1 \sim d_N$ 까지 차례대로 스캔하면서 각 $ccut_i$ 를 결정한다. 따라서 그림 3의 알고리즘은 $O(N)$ 의 시간복잡도(time complexity)를 보인다. EDP를 이용해 데이터를 분할하는 알고리즘은 $O(\text{분할그룹의수} \times \text{데이터의수}^2)$ 의 시간복잡도를 보인다^[10]. 따라서 EDP 알고리즘을 이용하여 각 G_c 를 B 개의 그룹으로 분할하는 데는 $O(\sum_{c=1}^K B \times |G_c|^2)$ 의 시간복잡도를 보인다. 여기서 $|G_c|$ 는 G_c 에 포함된 데이터의 개수를 의미한다. 그런데 $N = \sum_{c=1}^K |G_c|$ 이므로 $N^2 \geq \sum_{c=1}^K |G_c|^2$ 가 성립한다. 따라서 NODA 기법은 대부분의 데이터가 특정 방송채널에 할당되는 최악의 경우에는 $O(BN^2)$ 시간복잡도를 보인다. 그러나 데이터의 액세스 확률의 차이가 크지 않아 각 방송채널에 대략 N/K 개의 데이터들이 할당되는 환경에서는 $O(\sum_{c=1}^K B \times |G_c|^2)$ 는 $O((B/K) \times N^2)$ 으로 계산된다. 따라서 NODA 기법에서 데이터를 분할하는데 필요한 시간복잡도는 $O((B/K) \times N^2)$ 에서 $O(BN^2)$ 의 사이 값으로 결정된다.

NODA 기법에서 AED는 $\sum_{c=1}^K \sum_{b=1}^B (AED_{c,b})$ 로 계산된다. $SCH_{c,b}$ 는 $G_{c,b}$ 의 데이터들을 B tick당 한 개씩 라운드-로빈 방식으로 평형 방송하므로, $SCH_{c,b}$ 가 자신의 모든 데이터들을 한 번씩 방송하는데 걸리는 방송 주기는 $B \times |G_{c,b}|$ 이다. 여기서 $|G_{c,b}|$ 는 $G_{c,b}$ 에 포함된 데이터의 개수를 의미한다. 그런데 어떤 데이터를 액세스하려면 평균적으로 방송 주기의 반을 기다려야 하므로, $SCH_{c,b}$ 가 방송하는 데이터를 액세스하려면 평균 $B \times |G_{c,b}|/2$ tick 시간만큼 대기해야 한다. 따라서 $AED_{c,b}$ 는 $(B \times |G_{c,b}|/2) \sum_{d_q \in G_{c,b}} (p_q)$ 로 계산되며, NODA 기법에

서 전체 데이터들의 평균 예상지연시간은 식 (4)와 같이 결정된다.

$$AED = \sum_{c=1}^K \sum_{b=1}^B \left(\frac{B \times |G_{c,b}|}{2} \sum_{d_q \in G_{c,b}} (p_q) \right) \quad (4)$$

IV. 성능 결과 및 분석

이 장에서는 NODA 기법의 성능을 평가하고 그 결과를 기술한다. FLAT은^[1] 가장 기본이 되는 기법이며, DP는^[7] 각 채널의 데이터가 평형 방송되는 시스템 모델에서 최적 성능을 발휘하며, TLDP 기법은^[10] 본 논문과 유사한 모델을 사용한다. 따라서 본 논문은 이 세 기법들을 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 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	채널당 서브채널의 수	10	max(10, ⌊N/(K×100)⌋)

NODA 기법에서 채널당 서브채널의 수를 의미하는 B 값은 성능 튜닝을 위해 조정할 수 있는 변수인데, 이 값은 θ 값, 방송채널의 수, 데이터 항목의 개수 등을 고려하여 적절히 결정해야 한다. 만일 방송채널의 수가 적을 경우에 B 값을 너무 작게 설정하면 한 서브채널에 너무 많은 데이터가 할당되어 최적 AED 성능을 발휘하지 못하게 된다. 본 논문은 이 단점을 개선하기 위하여 평균적으로 한 서브채널에는 100 개 이하의 데이터를 할당한다는 간단한 휴리스틱을 적용하여 성능을 튜닝한다. NODA는 B의 기본 값으로 10을 사용하는데, 서브채널당 평균 100 개 이상의 데이터가 할당될 경우에는 B의 값을 ⌊N/(K×100)⌋로 결정한다. 즉 B 값을 max(10, ⌊N/(K×100)⌋)

⌋)으로 설정한다. 여기서 ⌊x⌋는 x와 같거나 x를 초과하는 최소 정수를 의미한다.

서버가 방송하는 각 데이터는 그 크기가 동일하고, 각 데이터의 액세스 확률은 불균등한 액세스 형태를 모델링하기 위해 많이 사용되는 zipf 분포 모델을^[1, 4] 따른다고 가정한다. zipf 분포 모델에서 각 데이터의 액세스 확률은 θ 값에 의해 결정되는데, θ는 각 데이터들의 액세스 확률이 어느 정도 편향되어 있는지를 나타내는 인수로서 θ 값이 커질수록 불균등한 액세스의 정도가 심화된다. 각 기법은 이 액세스 확률에 근거하여 전체 데이터들을 분할하여 각 채널, 또는 서브채널에 할당한다. 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.

1. 채널에 데이터를 할당하기 위한 실행시간

이 절은 전체 데이터들을 분할하여 각 방송채널 또는 서브채널에 할당하기 위한 실행시간을 비교 분석한다. 실행시간 결과는 각 실험을 10 번 실행하여 측정된 시간의 평균 값으로 제시하였다. DP와 FLAT 기법의 실행시간은 θ 값에 영향을 받지 않지만, TLDP와 NODA 기법의 실행시간은 이에 영향을 받는다. θ 값이 커지면 각 채널에 배정되는 데이터 수의 차이가 더 커지게 되므로, TLDP와 NODA 기법의 실행시간은 더 증가하게 된다. 본 실험에서는 NODA의 실행시간이 많이 걸리는 환경을 고려하여 θ 값을 0.95로 비교적 크게 설정하였다. 그리고 채널 수를 10으로 설정하고 데이터 항목의 수에 따른 실행시간의 변화를 살펴본 결과가 그림 4에 있다.

NODA 기법은 시간복잡도가 $O((B/K) \times N^2) \sim O(BN^2)$ 의 사이 값으로 결정되지만, 대체로 $O((B/K) \times N^2)$ 의 시작 복잡도를 보인다. 반면 DP 기법은 $O(KN^2)$ 의 시간복잡도를 보이며, TLDP 기법은 $O((K+(B/K))N^2)$ 에 가까운 시간복잡도를 보인다. 이론상으로는 NODA, DP, 그리고 TLDP 순으로 빠른 실행시간의 결과를 보인다. 그리고 이 세 기법은 모두 N이 증가함에 따라, N^2 에 비례하여 실행시간이 증가한다. 그림 4에 제시된 실행시간은 대체로 이와 일치하는 결과를 보인다. 이 실험에서 NODA는 DP 기법과 비교하여 1/24 ~ 1/8 정도의 실행시간을 필요로 한다. 그림 4에서 구분이 잘 되지 않는 않지만, TLDP는 DP 기법보다 5% ~ 13% 정도 더 많은 실행시간을 필요로

한다. FLAT 기법은 전체 데이터를 균등하게 분할하여 각 채널에 배정하므로 N이 증가하더라도 항상 일정한 성능을 보이는데, 다른 기법에 비해 훨씬 적은 실행시간을 보인다.

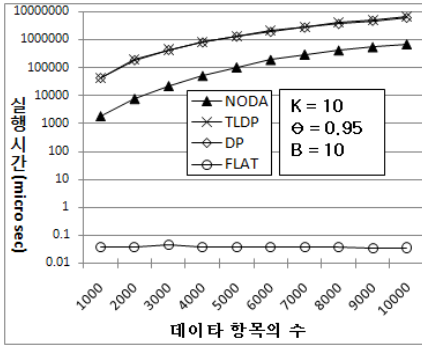


그림 4. 데이터 수의 변화에 따른 실행시간
Fig. 4. No. of data items vs. execution time

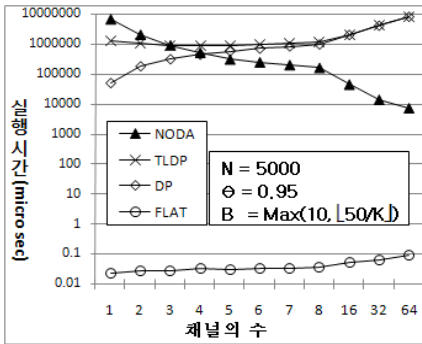


그림 5. 채널 수의 변화에 따른 실행시간
Fig. 5. No. of channels vs. execution time

N을 5000으로 고정하고 K에 따른 실행시간의 변화를 살펴본 결과가 그림 5에 있다. 이 실험에서 TLDP 기법은 B 값을 10으로 고정하였다. FLAT과 DP 기법은 대체로 K에 비례하는 실행시간을 필요로 한다. TLDP 기법에서 처음에는 K가 증가함에 따라 실행시간이 감소하다가, K가 5 이상으로 증가하면 실행시간이 증가하기 시작한다. 이는 TLDP가 $O((K+(B/K)N)^2)$ 의 시간복잡도를 보이는데서 기인된다. 즉, K 값이 1~4의 구간에서는 K가 증가할 때, K의 증가분보다는 B/K의 감소분이 더 크며, K가 5 이상이 되면 K의 증가분이 B/K의 감소분보다 더 커지기 때문이다.

평균적으로 $O((B/K) \times N^2)$ 의 시작복잡도를 보이는 NODA 기법은 K 값이 증가함에 따라, 지속적으로 실행

시간이 감소하는데, K가 1~4의 구간에서는 급격하게 감소한다. 이는 이 구간에서는 K 값이 증가함에 따라 B 값은 작아져, B/K 값이 크게 감소하기 때문이다. K 값이 작을 경우에는 NODA 기법이 가장 많은 실행시간을 필요로 하지만, K 값이 3 이상이 되면 TLDP보다 더 적은 실행시간을 보이며, 5 이상으로 증가하면 DP보다 더 적은 실행시간을 보이기 시작한다. 이와 같은 실행시간의 결과는 이론적으로 제시된 시간복잡도의 결과와 거의 일치한다.

2. 채널 수의 변화에 따른 평균 지연시간의 성능

본 실험에서는 채널 수를 1에서 64까지 변화시키면서 각 기법의 성능추이를 살펴보았다. 평균 지연시간의 성능 평가에서는 기존 기법들과의 비교와 더불어, 제안된 기법이 최적 평균 지연시간의 성능에 얼마나 근접하는지를 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. 본 실험에서 θ 값은 편향적인 액세스를 모델링하기 위한 대표 값으로 많이 사용되는 0.95로 설정하였다^[1, 4]. 클라이언트가 하나의 데이터를 액세스하는데 걸리는 AED 성능이 그림 6에 제시되어 있는데, NODA, TLDP, DP, 그리고 FLAT 기법 순으로 우수한 성능을 보인다. NODA는 다른 기법들에 비해 우수한 성능을 보일 뿐만 아니라, AED의 이론적 하한 값인 OPT에 거의 필적하는 성능을 발휘하는 것을 알 수 있다. 참고로 NODA 기법과 OPT의 성능 차이는 전 채널 구간에서 1% 이내에 불과하다.

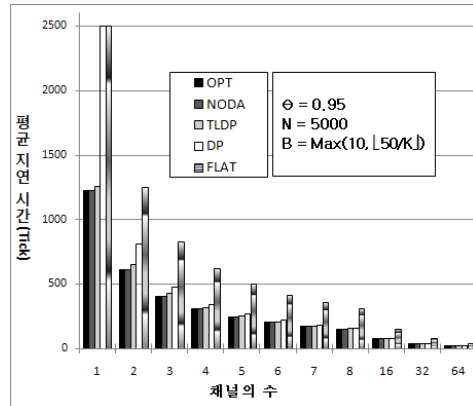


그림 6. 채널 수의 변화에 따른 평균 지연시간
Fig. 6. No. of channels vs. average delay time

채널 수가 작을 경우에는 한 방송채널에 많은 데이터들이 할당되어 이들 간에 액세스 확률의 차이가 커진다. DP와 FLAT 기법은 같은 방송채널의 데이터들을 동일 빈도로 방송하는데 비하여, NODA와 TLDP 기법은 동일 방송채널의 데이터들을 한번 더 분할하여 서브채널에 할당하고 각 서브채널별로 평형 방송한다. 따라서 NODA와 TLDP 기법은 동일 방송채널의 데이터들을 인기도를 반영하여 다른 빈도로 방송하는 것이 가능하다. 그러므로 K가 작을 경우에 NODA와 TLDP 기법은 DP와 FLAT 기법에 비하여 훨씬 우수한 AED 성능을 보인다. K가 증가함에 따라 NODA와 TLDP 기법과 DP의 상대적인 성능 차이가 줄어든다. 이는 K가 증가하면 한 채널에 적은 수의 데이터들이 할당되므로 동일 채널에 배정되는 데이터들 간에 액세스 확률의 차이가 줄어들어, 같은 채널의 데이터들을 동일빈도로 방송하는 DP의 단점이 어느 정도 완화되기 때문이다. K가 32 이상이 되면, NODA, TLDP, 그리고 DP 기법은 거의 유사한 성능을 보이기 시작한다.

NODA, TLDP, 그리고 DP 기법은 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 데이터들을 배정한다. 따라서 FLAT 기법은 K가 증가하더라도 모든 데이터들을 여전히 동일빈도로 방송한다. 그러므로 FLAT 기법에서는 채널 수가 증가하면, 평균 지연시간이 K에 선형적으로 비례하여 감소될 뿐이다. 따라서 채널 수가 증가하더라도 FLAT 기법은 다른 기법과의 상대적인 성능 차이가 큰 편이다.

NODA 기법은 TLDP와 유사한 시스템 모델을 사용하지만, TLDP보다 최대 6% 이상 향상된 AED 성능을 보인다. 그 이유는 다음의 두 가지 원인에서 비롯된다. 첫째, NODA는 가변적인 B 값을 사용하여 서브채널에 적절한 수의 데이터를 배정할 수 있는데, TLDP는 방송채널의 수에 관계없이 동일한 B 값을 사용하여 방송채널의 수가 적을 경우에는 서브채널에 너무 많은 데이터들을 할당한다. 둘째, NODA와 TLDP 기법에서 각 방송채널은 자신에게 할당된 데이터들을 거의 이상적으로 방송한다. 그러나 NODA 기법은 각 방송채널이 이상적으로 데이터를 방송한다고 가정하고 첫 번째 단계의 데이터 분할을 진행하는 반면, TLDP 기법은 첫 번째 데이터 분할 정책에서 각 방송채널은 자신에게 배정된 데이터들을 평

형 방송한다고 가정한다. 따라서 TLDP 기법에서 첫 번째 단계의 데이터 분할이 최적이지 못하다. 이 두 가지의 이유로 인하여 NODA는 TLDP 기법보다 우수한 성능을 보이는 것이다.

3. θ 및 N 값의 변화에 따른 평균지연시간의 성능

본 실험에서는 θ 및 N 값의 변화에 따른 평균 지연시간의 성능추이를 살펴보았다. K를 4, N은 5000, B를 $\max(10, \lfloor N/(K \times 100) \rfloor)$ 에 따라 13으로 설정하고, θ 값을 변화시키면서 살펴본 성능 결과가 그림 7에 제시되어 있다. 이 실험에서 NODA는 TLDP 기법보다 최대 3.1% 정도 우수한 성능을 보인다. 그러나 그 차이가 그래프 상에서 거의 구분되지 않기 때문에 TLDP의 성능 결과는 생략한다. θ 값이 0이 되면, 각 데이터의 액세스 확률이 동일하게 되어 클라이언트가 액세스하는 데이터들의 분포는 균등 분포를 이루게 된다. 따라서 이 경우에는 각 기법이 어떤 스케줄링 정책을 사용하더라도 동일한 성능을 보인다. 그러나 θ 가 증가함에 따라 각 데이터들의 액세스 확률의 차이가 점점 커지게 된다. 이 액세스 확률의 차이를 고려하여 적절하게 방송채널에 데이터를 배정하는 NODA와 DP 기법에서 θ 값이 증가함에 따라 평균 지연시간이 줄어들기 시작한다. 그러나 FLAT 기법은 데이터들의 액세스 확률에 상관없이 각 채널에 N/K 개의 데이터를 일정하게 할당하기 때문에 θ 의 변화에 관계없이 항상 $N/2K$ 의 동일한 성능을 보인다. 따라서 θ 가 증가함에 따라 NODA와 DP 기법은 FLAT보다 훨씬 우수한 성능을 보인다. NODA는 DP 기법에 비하여 우수한 성능을 보이며 전 θ 구간에서 거의 OPT에 필적하는 성능을 보인다.

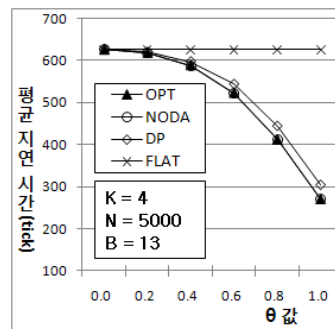


그림 7. θ 값의 변화에 따른 평균지연시간
Fig. 7. θ value vs. average delay time

K를 4, θ 값을 0.95로, B를 13으로 설정하고, 데이터 항목의 수를 변화시키면서 살펴본 성능 결과가 그림 8에 제시되어 있다. 이 실험에서도 TLDP 기법은 NODA와 성능 차이가 크지 않아 그 결과를 생략한다. N 값이 증가함에 따라 각 기법의 평균 지연시간은 거의 선형적으로 증가한다. N이 1000일 경우에 NODA 기법은 DP 보다 8%, FLAT보다는 78% 정도의 우수한 성능을 보인다. N이 증가함에 따라 각 기법의 성능 차이는 점점 커지는데, N이 10000이 되면 NODA 기법은 DP 보다 13%, FLAT 보다는 115% 정도의 우수한 성능을 보인다. 그리고 NODA 기법은 전 구간에서 OPT와 성능 차이가 0.3% 이내에 불과할 정도로 거의 최적에 가까운 성능을 발휘한다.

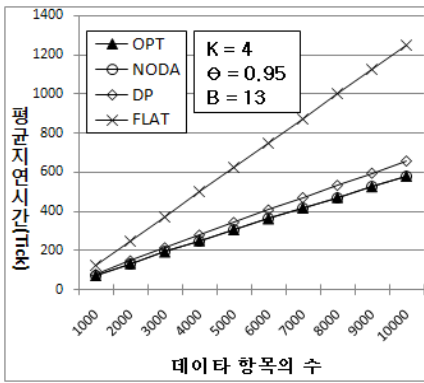


그림 8. 데이터 수의 변화에 따른 평균지연시간
Fig. 8. No. of data items vs. avg. delay time

그림 4와 그림 5에서 보는 것과 같이, 채널의 수가 작은 환경을 제외하면 NODA는 TLDP와 DP 기법보다 채널에 데이터를 할당하는 실행부담이 훨씬 적은 편이다. 그리고 그림 6에서 보는 것과 같이 NODA는 TLDP와 DP 기법보다 우수한 평균 지연시간의 성능을 보인다. 뿐만 아니라 NODA 기법은 대부분의 환경에서 이론적 최적 성능을 의미하는 OPT에 거의 필적하는 평균 지연시간의 성능을 보인다. FLAT은 다른 기법에 비하여 훨씬 적은 실행부담을 보이지만, 평균 지연시간이 매우 크다는 단점이 있다. 이와 같은 점들을 고려한다면 NODA 기법은 합리적인 실행 부담을 보이면서도 우수한 평균 지연시간의 성능을 발휘하여 매우 바람직한 성능 특성을 보인다고 할 수 있다.

V. 결론

본 논문은 방송 방식의 데이터 전달 기법을 채택한 다중 방송채널 환경의 정보 시스템에서 클라이언트가 하나의 데이터를 액세스하는데 필요한 평균지연시간을 최소화하기 위하여, 전체 데이터들을 분할하여 각 방송채널 및 서버채널에 할당하는 데이터 할당 기법에 대한 연구를 진행하였다. 본 논문이 제안한 NODA 기법은 이론적 평균 예상지연시간의 성능을 계산한 후, 이를 각 방송채널이 균등하게 배분하여 담당하도록 전체 데이터들을 분할하여 각 방송채널에 할당한다. 그리고 나서 각 방송채널은 B 개의 서버채널로 구성된다는 개념 하에, 각 방송채널에 할당된 데이터들을 다시 B 개의 그룹으로 분할하여 각 서버채널에 할당한다.

각 방송채널의 서버채널에 할당된 데이터들은 해당 방송채널 대역폭의 1/B을 사용하면서 평형 방송된다. 그리고 데이터들은 액세스 확률의 내림차순으로 각 서버채널에 순서대로 할당되기 때문에 액세스 확률이 높은 데이터들이 할당되는 서버채널에는 상대적으로 적은 수의 데이터들이 할당된다. 따라서 이들은 상대적으로 높은 빈도로 방송될 수 있다. 이와 같이 NODA는 서버채널의 개념을 사용하여 동일 방송채널의 데이터들을 그들의 인기도를 반영하여 방송할 수 있다. 본 논문은 시뮬레이션 모델을 설계 구현하여 성능평가를 실시하였는데, 제안된 기법은 다른 기법과 비교해 채널에 데이터를 할당하기 위한 합리적인 실행 부담을 보이면서도 우수한 AED 성능을 발휘한다. 특히 물리적 채널의 수가 적을 경우에 다른 기법보다 훨씬 우수한 성능을 보였다. 그리고 NODA 기법은 평균지연시간의 이론적 하한 값에 거의 필적하는 성능을 보였다.

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

참고 문헌

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] H.M. Kwon, "TLDP: A New Broadcast Scheduling Scheme for Multiple Broadcast-Channel Environments," *The Journal of the Institute of Webcasting, Internet and Telecommunication*, Vol. 11, No. 2, pp. 63-72, 2011.

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

저자 소개

권혁민(중신회원)



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

연구소 컴퓨터개발부 선임연구원

- 1999년 ~ 현재 세명대학교 정보통신학부 부교수

<주관심분야 : 트랜잭션 처리, 분산/병렬 데이터베이스, 모바일 컴퓨팅>