디지털 멀티미디어 방송(DMB)에서 클라이언트 요구 기반의 데이터 캐러셀 스케줄링

정회원 최 수 정*, 박 익 현*, 강 상 혁*

Data carousel scheduling based on user-request statistics for Digital Multimedia Broadcast (DMB)

Sujeong Choi*, Ik Hyun Park*, Sang Hyuk Kang* ***Regular Members***

요 약

이 논문에서는 DMB에서의 주문형 데이터 방송을 위한 새로운 데이터 캐러셀 스케줄링 알고리듬을 제안한다. 클라이언트로부터 리퀘스트를 받으면 서버는 그 리퀘스트에 대한 통계치를 이용하여 아이템을 두 집합으로 나눈다. hot 아이템을 한 캐러셀 내에서 여러 번 반복하여 방송하도록 하여 그 아이템에 대한 클라이언트의 대기시간을 줄이도록 하였으며, hot 아이템의 방송회수에 상한선을 두어 cold 아이템의 방송 기회를 늘이도록 하여 클라이언트의 리퀘스트에 대한 응답 성공률을 높이도록 하였다. 모의실험을 통해 캐러셀 기반의 다른 알고리듬과 비교하였을 때 원하는 결과를 얻어낼 수 있었다.

Key Words : DMB, carousel, on-demand broadcast

ABSTRACT

We propose a new data carousel scheduling algorithm for on-demand data broadcasting in DMB. The server divides data items into two sets named hot and cold, according to request statistics from clients. When constructing a data carousel, hot items are placed periodically with their upper limit of broadcasting frequency. If there are empty slots after placing hot items, cold items with high request ratio are placed until the carousel is full. A cold item is broadcast only once in the caroulsel. For the response on clients' requests, our proposed scheme is shown to have high success ratio with short waiting time.

Ⅰ. 서 론

디지털 오디오 방송(Digital audio broadcast; DAB) 망은 통신 기술의 발전으로 인하여, 디지털 멀티미디어 방송(DIgital multimedia boradcast; DMB)로 진화하고 있다[1]. 멀티미디어 방송은 오디오/비디오/데이터 등 여러 가지 형태의 정보를 복합적으로 전송함으로써 정보의 가치를 더욱 높일 수 있게 된다. 특히 '데이터' 방송은 오디오/비디오에 부가적인 데이터나 혹은 독자적인 정보를 가지는 데이터를 방송의 형태로 공급하는 형태인데, 데이터 캐러셀 등의 형태에 기반한 스케줄링 기법이 그 효율성을 좌우한다.

셀룰러 혹은 PCS, 무선 LAN 망을 이용한 역방향 채널을 통해 클라이언트가 방송국에 데이터를 전송하는 것이 가능하도록 하는 근래의 기술은 앞으로의 방송망은 이러한 양방향으로 진행될 것이라는 것을 말해준다. 이것은 서버에서 클라이언트로의

※ 본 논문은 2005 서울시산학연 협력사업의 지원을 받음.
* 서울시립대학교 전자전기컴퓨터공학부 통신망연구실({cris, mnie, shkang}@uos.ac.kr)
논문번호 : KICS2005-09-364, 접수일자 : 2005년 9월 6일

대역폭이 반대쪽의 대역폭보다 큰 비대칭적인 환경을 가진, 방송과 통신망의 융합을 이끌어 내게 되었다.

이와 같은 환경에서, 서버가 클라이언트의 요구에 대한 통계치를 고려하여 방송아이템을 스케줄링 하는 메커니즘이 필요하게 된다. 데이터 방송을 위한 스케줄링 메커니즘은 방송을 효율적으로 만들기 위해 연구되고 있다.

수년간, 데이터 방송 스케줄링 알고리듬에 대한 많은 연구결과들이 제시되었다. 방송 스케줄링 방법을 두 가지로 나눌 수 있는데, 하나는 전형적인 정적인 방송이고 다른 하나는 동적 방송[2]이다. 종종 전자는 push-based 혹은 주기적 방송이라 하기도 하고 후자는 pull-based 혹은 주문형 방송이라 불리기도 한다.

정적 방송에서, 데이터 프로그램의 스케줄은 고정된 채 클라이언트로부터의 피드백을 고려하지 않는다. 예를 들어 [3]에서는 "broadcast disks"를 제안하였는데, 클라이언트 수나 접근 패턴은 변하지 않고 서버로의 피드백도 고려하지 않고 있다.

주문형 방송은 서버가 클라이언트로부터의 리퀘스트를 실시간으로 반영한다. [4]나 [5]에서는 하나의 아이템을 전송할 때마다, 서버는 리퀘스트의 대기시간이나 각 아이템에 대한 선호도를 고려하여 가장 적절한 아이템을 결정하여 방송한다. 그리고 [4]는 크기가 다른 아이템을 가정하고 있다. 그러나 이 연구들은 캐러셀에 기반하지 않은 주문형 방송 시스템에서의 스케줄링에 관한 것이다.

최근에 제안되는 주기적 방송과 주문형 방송을 결합한 스케줄링 알고리듬은 각각의 방송 스케줄링 보다 효율적인 결과를 얻어 내고 있다[2, 6, 7]. 특히 [2]에서는 고정된 비율로 대역폭을 주기적 방식과 주문형 방식에 나누어서 할당하였고 [6]은 리퀘스트의 상대적 데드라인을 가정하고 있다. 또한, [7]은 주기적 방송과 주문형 방송을 결합하는 방법으로 방송할 아이템을 동적으로 결정하고 있다.

기존의 연구논문 대부분에서 아이템의 크기가 일정한 것으로 가정하였다. 그러나 이 논문에서는, 다양한 크기의 아이템을 가진 데이터 캐러셀 기반의 Frequency-based Broadcast Scheduling(FBS)을 제안한다. 본 논문에서 새롭게 제안된 방법의 가장 큰 특징은 캐러셀 기반의 스케줄링이다. 이것은 하나의 아이템을 방송할 때마다 스케줄링 계산을 하는 기존의 대부분의 논문([2], [4], [5], [6], [7])들과 달리 일정한 주기 동안의 방송 순서를 결정한다. 따라서 사용자는 캐러셀 전체의 데이터 구성을 알 수 있다. 각 캐러셀 주기동안, 서버는 클라이언트로부터 도착한 리퀘스트의 통계치를 이용하여 적절한 아이템을 정하고 결정된 아이템 캐러셀을 구성한다. 이 논문의 목적은 많은 클라이언트의 요구를 만족시키면서 많이 요청된 아이템은 자주 방송하여 클라이언트의 대기시간을 줄이고자 한다. 그러기 위해, 리퀘스트의 비율에 따라 아이템을 두 개의 집합으로 나누고 차등화된 방송 기회를 아이템에게 부여하였다. 시뮬레이션에서는 캐러셀 기반의 업데이트 알고리듬([9])과 새롭게 제안된 알고리듬을 비교하였는데, [9]에서의 여러 방법은 아이템의 중복을 고려하지 않고 있다. 또한 구체적으로 캐러셀을 구성하는 아이템을 결정하는 방법이 설명되지 않았기 때문에 별도의 아이템 결정 방법을 사용하였다. 그 결과, 본 논문에서 제안한 스케줄링이 상대적으로 낮은 대기시간을 유지하면서 리퀘스트에 대한 방송 성공률이 높다는 것을 보인다.

Ⅱ. FBS 알고리듬

이 논문에서 제안한 알고리듬은 클라이언트로부터도착한 리퀘스트를 이용하여 방송 아이템을 결정하는 주문형 데이터 방송을 가정하고 있으며, 하나의 캐러셀 내에 클라이언트의 만족도를 높일 수 있는 아이템을 선정하여 방송하기 위한 알고리듬이다. 데이터 캐러셀은 하나의 방송 주기를 구성하는 오브젝트(아이템)의 집합으로 정의한다[8]. 먼저 클라이언트로부터 전송된 리퀘스트는 서버의 큐에 쌓이고 서버는 이 리퀘스트에 대한 측정값을 이용하여 아이템을 선정하고 결정된 아이템으로 데이터 캐러셀을 구성한다. 주기적 방송에서라면 서버에 의해 결정된 캐러셀이 일정하게 반복된다. 그러나 이 논문에서는 클라이언트로부터의 리퀘스트를 반영하여 캐러셀을 주기적으로 업데이트하는 알고리듬을 제안하므로 캐러셀을 구성하는 오브젝트는 캐러셀에 따라 다를 수 있으며, 캐러셀 전체의 길이도 다소 변할 수 있다. 그러나, 캐러셀은 최대값을 가지고 있어서 그 이상의 길이로 구성될 수 없다.

이 논문에서 가정하고 있는 방송환경을 그림 1에 나타냈다. 클라이언트가 보낸 리퀘스트는 셀룰러, PCS, 무선 LAN과 같은 이동 통신 채널을 통해 서버에게 전송이 되며 고정된 길이를 가진 서버의 큐에 저장이 된다. 서버는 이 리퀘스트를 이용하여, 아이템을 선택하고 캐러셀을 구성한 뒤, 클라이언트에게 방송하는 것이다. 만약 큐에 있는 리퀘스트가

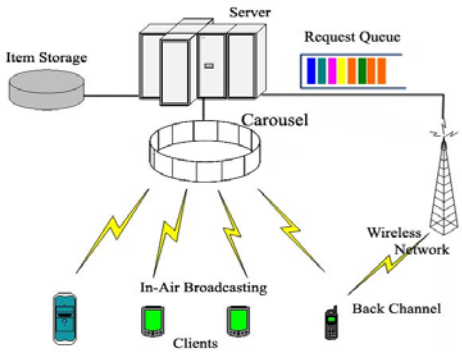

그림 1. DMB 서비스 시스템

요청한 아이템이 어느 정도의 특정 시간동안 방송되지 못한다면 그 리퀘스트는 응답에 실패한 것으로 정의한다. 어떤 아이템이 방송이 된다면, 그 아이템을 요청한 리퀘스트는 응답에 성공한 것으로 정의한다.

캐러셀의 최대 길이를 B tu(timeunit)라 표기하는데, 모의실험에서는 0.2초를 1tu으로 사용하도록 한다. 서버는 N 개의 아이템을 보유하고 있으며, 각 아이템에 대한 리퀘스트가 저장되는 서버의 큐 길이는 고정된 값, T이다. 또한 λ는 timeunit 당 도착하는 평균 리퀘스트 수를 나타내고 있다.

또한, 아이템 i의 크기는 s_i로 표기하고 큐에 있는 i에 대한 리퀘스트 중 도착한 이후에 가장 오랫동안 기다린 시간은 w_i라고 하겠다. 큐에 있는 리퀘스트 중 아이템 i에 대한 리퀘스트 수를 q_i라 표기하는데, 과거의 리퀘스트 수의 통계를 반영하기 위하여 이것을 아래와 같이 EWMA(exponentially weighted moving average)값으로 계산한다.

$$q_i \leftarrow \alpha q_i + (1-\alpha) q_{i,new} \tag{1}$$

이 때, $q_{i,new}$는 새롭게 측정된 리퀘스트 수이고 $0<\alpha<1$을 만족한다.

모의 실험을 통하여 이 논문에서 제안한 알고리듬의 성능을 평가하는데, 여기에서 성능지표로 응답시간(response time)과 방송 성공확률(success probability)을 사용하는데 아래와 같이 정의한다.

- Response time: 리퀘스트를 보낸 뒤 원하는 아이템이 방송될 때까지 클라이언트가 기다리는 시간의 평균
- Success probability: 리퀘스트에 대하여 아이템이 방송될 확률

2.1 데이터 캐러셀의 구성

아이템 i에 대한 리퀘스트 비, r_i를 다음과 같이 정의하도록 한다.

$$r_i \equiv \frac{q_i}{\sum_{j=1}^{N} q_j} \tag{2}$$

서버는 자신의 저장소에 있는 데이터 아이템을 리퀘스트 비 r_i의 경계값을 다음과 같이 r_i의 평균값으로 정의하고 그에 따라 두 집합으로 나눈다.

$$\theta \equiv \frac{\sum_{i=1}^{N} 1_{r_i>0} r_i}{\sum_{i=1}^{N} 1_{r_i>0}} \tag{3}$$

이때, $r_i>0$이면 $1_{r_i>0}=1$이 되고, $r_i<0$이면 $1_{r_i>0}=0$을 의미한다.

하나의 아이템이 $r_i \geq \theta$을 만족하면, 그 아이템을 hot 아이템이라 부르고, 그렇지 않으면 cold 아이템이라 부르겠다. 즉, hot 아이템의 집합 A와 cold 아이템 집합 A^c은 아래와 같이 정의된다.

$$A=\{i | r_i \geq \theta\},\ A^c=\{i | r_i < \theta\} \tag{4}$$

각 집합의 원소 개수를 각각 N_A와 N_{A^c}로 표기하겠다.

먼저 집합 A에 있는 아이템을 리퀘스트 비 r_i가 큰 것부터 작은 순서로 나열하도록 하고 $j(j=1, 2, \cdots, N_A)$로 표기하겠다. 즉, 가장 높은 리퀘스트 비를 갖고 있는 아이템이 $j=1$이 되고 두 번째 아이템이 $j=2$가 된다.

또한, A^c에 있는 각각의 아이템에 대하여 가중치(weight)를 다음과 같이 정의하며, R_i, W_i, S_i는 r_i, w_i, s_i를 각각의 평균으로 나누어 표준화시킨(normalized) 값이다.

$$\mu_i = \frac{R_i W_i}{S_i} \tag{5}$$

A^c에 있는 아이템은 위에서 계산된 가중치 값이 큰 아이템부터 작은 순서로 나열하고 $j(j=N_A+1, N_{A+2}, \cdots, N)$로 표기한다. 즉, 가장 큰 가중치 μ를 가진 cold 아이템은 $j=N_A+1$가 된다.

이제 캐러셀을 구성하는데, 서버는 cold 아이템보다 hot 아이템에 우선권을 부여한다. 또한, 캐러셀

내에서 hot 아이템은 1 이상의 방송 빈도수를 갖게 되는데, 방송 빈도수에 대해서 상한값을 정해주도록 한다. 즉, hot 아이템의 방송 횟수 f는 $1 \leq f \leq U$의 범위에서 결정된다. 그리고 cold 아이템은 캐러셀에서 hot 아이템이 차지하는 부분의 나머지에 포함되며, cold 아이템의 방송횟수는 1로 제한되어 있다. hot 아이템 i의 방송횟수 f_i를 결정하기 위하여 다음과 같은 관계식을 살펴보도록 한다.

$$f_i' \cdot s_i = r_i \cdot B \tag{6}$$

위의 식 (5)는 캐러셀 내에서의 hot 아이템 하나가 차지하는 부분을 리퀘스트 비와 크기를 고려하여 결정하기 위함이다. 이것을 이용하여 f_i는 아래와 같이 결정되며, $[\cdot]$는 반올림 연산을 의미한다.

$$f_i = \min\{\max 1, [f_i']\}, U\} \tag{7}$$

이와 같이, hot 아이템에 대한 방송 횟수가 모두 결정되고 cold 아이템의 가중치 순으로 재배열이 끝나게 된다.

이제 캐러셀에 포함될 아이템을 결정해야 한다. 하나의 캐러셀을 구성하는 아이템 크기의 총합은 그 캐러셀 길이를 넘지 말아야 한다. 캐러셀에 포함되는 hot 아이템과 cold 아이템의 개수를 각각 M_h, M_c이라고 표기하자.

Case 1. 만약 $s_{N_A+1} + \sum_{j \in A} s_j \cdot f_j \leq B$이 성립한다면 $\sum_{i \in A} s_i f_i + \sum_{j=N_A+1}^{N_A+1+M} s_j \leq B$ 을 만족하는 최대값 M이 M_c가 된다. 즉

$$M_c = \arg\max_M \left\{ \sum_{i \in A} s_i f_i + \sum_{j=N_A+1}^{N_A+1+M} s_j \leq B \right\} \tag{8}$$

이다.

따라서 그 캐러셀은 hot 아이템 모두와 cold 아이템 $j = N_A+1, \cdots, N_A+1+M_c$으로 구성되며, $M_h = N_A$이다.

Case 2. 만약 $s_{N_A+1} + \sum_{j \in A} s_j \cdot f_j > B$이면

$\sum_{j=1}^{M} s_j f_j \leq B$을 만족하는 최대값 M이 M_h가 된다. 즉,

$$M_h = \arg\max_M \left\{ \sum_{j=1}^{M} s_j f_j \leq B \right\} \tag{9}$$

이다. 따라서 그 캐러셀은 hot 아이템 $j = 1, 2, \cdots, M_h$만으로 이루어지며, M_c는 정의에 의해 0이 된다.

다음 단계는 식 (8)과 (9)에 의해 결정된 $M_h + M_c$개의 아이템을 캐러셀에서 배치하기 위한 위치를 결정하는 것이다. 이를 위해, 캐러셀에서 총 L개의 슬롯을 고려하는데 이 때, L은

$$L = \sum_{j=1}^{M_h} f_j + M_c \tag{10}$$

에 의해 계산된다. 캐러셀 내에서의 hot 아이템 j의 편성 간격 d_j은

$$d_j \equiv [\frac{L}{f_j}] \tag{11}$$

에 의해 결정된다.

가장 큰 리퀘스트 비를 가진 hot 아이템부터 캐러셀 배치를 하게 되는데, hot 아이템 j는 캐러셀 내에서 식 (11)에 의해 계산된 d_j개의 슬롯만큼 간격을 갖고 f_j번 중복하여 배치된다. 이와 같은 방법으로 각각의 빈도수와 배치간격에 따라 M_h개의 hot 아이템을 캐러셀에 배치시킨다. 만약 M_c가 0이 아니라면, hot 아이템의 위치를 모두 결정한 뒤에 캐러셀의 가장 앞에 비어 있는 슬롯에 가장 높은 가중치를 가진 cold 아이템부터 차례로 포함시키도록 한다. 이때, 슬롯끼리의 충돌을 고려해야하는데, 만약 아이템의 위치를 결정하는데 있어서 이미 그 슬롯이 다른 아이템에 의해 차지되었다면, 위치를 정하고자 하는 그 아이템은 다음의 비어 있는 슬롯을 배정받도록 하여 슬롯의 충돌을 피하도록 한다.

2.2 캐러셀 구성의 예

본 논문에서 제안된 캐러셀 구성 메커니즘을 예를 들어 설명하도록 하겠다. 서버가 아이템 스토리지에 {1,2,3,4,5}의 아이템을 보유하고 있으며 각각의 크기는 $s_1 = 2$, $s_2 = 3$, $s_3 = 1$, $s_4 = 2$, $s_5 = 2$(tu)이고 캐러셀의 길이 B는 $B = 18$(tu)으로 주어져 있다고 하자. 또한, 위 식에 의해 hot 아이템의 집합이 $A = \{2, 4\}$이고 각각의 방송빈도수가 $f_2 = 3, f_4 = 3$으로 계산되었으며 가중치에 따른 순서에 의해 재배열된 cold 아이템 집합은 $A^c = \{5, 1, 3\}$이라고 하자. 이와 같은 가정하에 $s_2 \cdot f_2 + s_4 \cdot f_4 + s_5 \leq 18$이기 때문에, A에서의 Case 1에 해당되고 $M_h = 2$이고 $M_c = 1$이 되며 캐러셀에 포함되는 아이템은 두 개의 hot 아이템 {2, 4}와 하나의 cold 아이템 {5}이

그림 2. 완성된 캐러셀의 예

다. 또한, $L=7$, $d_2=2$, $d_4=2$가 된다. 결정된 아이템의 순서는 2, 4, 2, 4, 2, 4, 5가 되며 그림 2와 같다.

Ⅲ. 모의실험

본 논문에서 포아송 분포를 따르고 평균 λ requests/tu의 속도로 서버의 큐에 들어오는 리퀘스트와 Zipf 분포를 따르는 아이템 선호도를 가정한다. 표 1에서는 모의실험에서 사용한 파라미터를 나타냈으며, request의 큐 길이를 나타내는 T는 클라이언트의 patience time과 관련된 값으로 이 논문에서 30초, 즉 150tu 동안 받은 리퀘스트를 저장할 수 있는 값이다. 각 캐러셀마다 그 캐러셀에 포함된 아이템 정보를 알려주는 디렉토리가 캐러셀 앞에 포함되지만 본 논문에서의 모의실험에서는 이 디렉토리를 포함시키지 않았다. 100번의 실험결과에 대한 평균값을 계산하도록 하였고 이러한 환경에서 모의실험을 통해 Response time과 Success probability를 측정하였다.

이와 같은 가정으로 ICR/R, FCC[9]와 이 논문에서 제안한 FBS를 비교하였다. 먼저 ICR/R과 FCC 알고리듬을 살펴보자. 실험을 위해 캐러셀에 포함되는 아이템을 결정해야 하는데, [9]에서는 결정된 아이템으로 캐러셀을 업데이트 하는 방법만 설명되었을 뿐, 아이템을 결정하는 구체적인 방법을 기술하지 않았기 때문에 이 논문에서는 캐러셀에 포함될 아이템을 결정하기 위하여 클라이언트의 리퀘스트만을 고려하도록 하였다. 즉, 리퀘스트의 통계를 바탕으로 하여 캐러셀을 구성한다. 만약 현재 캐러셀과 비교하여 새로 포함된 아이템이 없으면 현재 캐러셀을 반복하여 방송한다. 한편, 새롭게 제안한 알고리듬 FBS는 이전 캐러셀과 똑같은 구성의 새로운 캐러셀이 계산될 때를 제외하고는 새롭게 계산된 캐러셀을 방송하도록 한다. 업데이트 시점이 현재 캐러셀을 방송하는 도중이라면 FCC는 그 캐러셀을 끝까지 방송한 뒤 새롭게 결정된 캐러셀을 내보내지만, ICR/R은 현재 캐러셀을 중단하고 새로운 캐러셀을 방송한다. 또한 ICR/R에서 아이템의 순서는 새로운 아이템이 캐러셀의 가장 앞에 놓이고 각 아이템 별로 가장 최근에 방송된 이후 경과된 시간에 의해 순서를 결정한다.

표 1. simulation setup

meaning	value
request arrival rate (λ)	0.1~1.0/tu
data size (s_i)	log-normal dist. with average 4tu
size of a carousel (B)	80~200tu
number of items in server's DB (N)	100
size of requeset queue (T)	100λ, 150λ, 200λ
run time	20,000 sec
number of runs	100
Zipf skew coefficient	0.1~1.0
bandwidth	100kbps (DMB data channel BW)
upper bound(U)	3

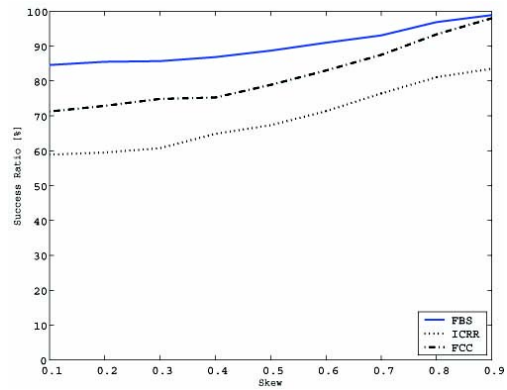

그림 3. $\lambda=0.6$일 때 Success Probability 비교(B=100tu, T=150λ)

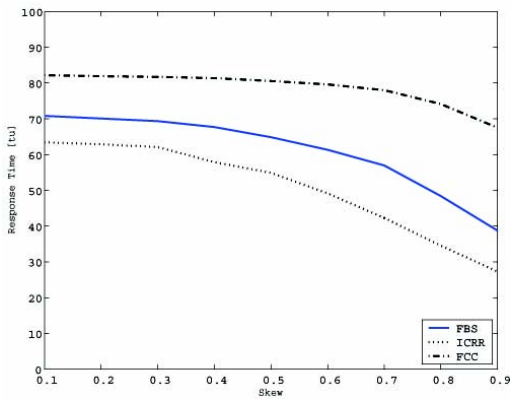

그림 4. $\lambda=0.6$일 때 Response time 비교(B=100tu, T=150λ)

그림 3과 4는 리퀘스트가 timeunit 당 0.6의 속도로 도착할 경우, 위에서 기술된 세가지 알고리듬의 Response Time과 Success Probability의 실험결

과를 나타내고 있다. λ값의 변화에 따른 결과도 이와 비슷한 양상을 띄고 있다. 세 알고리듬의 성공률 변화추이를 살펴보면, 공통적으로 Zipf 분포의 skew coefficient 값이 1에 가까울수록 응답성공률이 높아지는 것을 알 수 있다. skew coefficient 0이 의미하는 것은 모든 아이템의 선호도가 균일한 분포를 갖고 있음을 뜻하므로, 그림 4에서의 결과는 아이템에 대한 선호도가 특정 부분에 편중도를 갖고 있을 때 리퀘스트에 따라 아이템을 선정하는 것이 효율적이라는 결과를 의미하기도 한다. 특히, 이 논문에서 제안한 FBS가 가장 높은 Success Probability를 가져왔다. 이것은 FBS가 더 많은 종류의 아이템을 방송하는 FCC보다도 높은 응답성공률을 가져온 이유는 FBS에서는 리퀘스트가 도착한 이후 대기시간을 고려하여 프로그램을 편성하였기 때문이다. 반면에, ICR/R은 새로운 아이템이 다음 캐러셀에 포함되도록 계산이 되면 지금 방송되는 캐러셀을 중지시키고 새로운 캐러셀을 방송하게 되므로, 포함된 모든 아이템이 방송되지 못하는 캐러셀이 발생하게 된다. 따라서 ICR/R의 응답성공률이 상대적으로 낮게 나오는 결과를 초래하게 된다.

한편, Response Time의 결과를 살펴 보면 ICR/R의 결과가 가장 짧게 나타나는 것을 볼 수 있다. 이 또한 각 알고리듬의 캐러셀 구성 방법과 전송방식에 따른 차이에 기인한 것으로, 새로운 아이템을 위주로 하여 최종 방송 이후의 경과시간을 측정하여 방송하는 ICR/R의 response time이 가장 짧은 시간을 얻어냈다. 그러나, 본 실험에서 캐러셀의 디렉토리를 제외하고 전송하였으므로, 디렉토리를 추가했을 경우를 가정하면, ICR/R의 Response Time은 증가할 것으로 예측된다. 또한 클라이언트는 캐러셀 앞에 도착하는 디렉토리를 보고 캐러셀 정보를 알게 되지만, 실제 전송은 그 캐러셀의 아이템의 많은 부분이 방송되지 않고 다시 새로운 디렉토리를 받고 캐러셀을 수신하게 되는 것이다.

그림 5와 6에서는 캐러셀의 길이 B를 변화시킬 때의 Response Time과 Success Probability를 나타냈다. B를 각각 80tu부터 180tu 까지 증가시켰을때 B가 커질수록 Response Time은 길어지고 Success Probability는 낮아지는 양상을 보이고 있는데 이것은 캐러셀이 커지면 각 아이템의 방송 간격이 커지기 때문에 response time의 증가는 예상한 결과이다. 따라서 캐러셀 길이에 따라 클라이언트의 patience time을 고려한 방송횟수 f의 upper bound 값인 U를 변화시켜야 할 필요가 있다. 또한 캐러셀

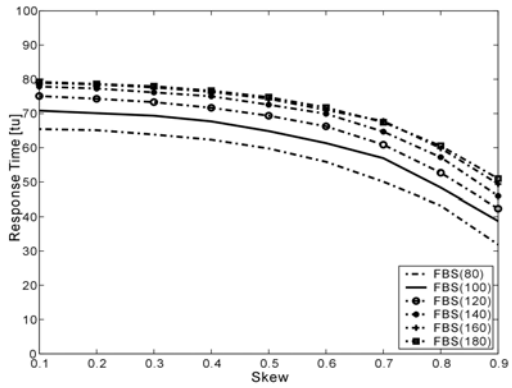

그림 5. B값에 따른 FBS의 Response Time 추이($\lambda = 0.6$)

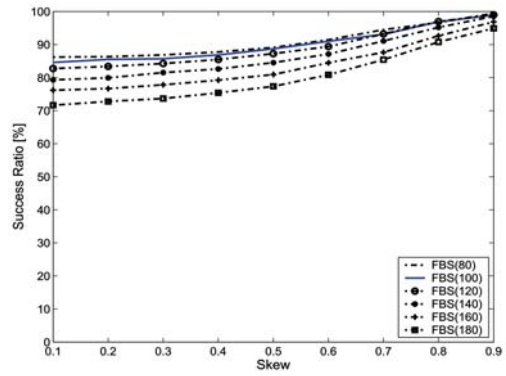

그림 6. B값에 따른 FBS의 Success Probability 추이($\lambda = 0.6$)

의 길이가 커지면서 캐러셀에 포함되는 cold item의 개수가 증가하게 되지만 한편으로는, 클라이언트가 기다리는 최대한의 대기시간을 넘어서면서 클라이언트의 포기확률을 높이게 되므로 success probability가 증가하게 된다.

그림 7과 8에서는 $\lambda = 0.6$으로 고정시켰을 때, 서버의 request queue 길이 T가 100λ, 150λ, 200λ일 때의 실험결과를 보여주고 있다. T가 길어지면 클라이언트의 request가 서비스를 받을 때까지 queue에서 머물러 있을 확률이 높아지므로 success probability가 높아지지만, 그에 따라 response time이 다소 증가하는 경향을 보이게 된다.

이 논문에서 제안한 알고리듬은 클라이언트의 patience time을 캐러셀의 길이에 따라 고려하였으며, 이것은 방송횟수의 upper bound 값이나, request queue size 등과 연관성을 갖고 Response Time이나 Success Probability에 영향을 끼치는 결과를 보이고 있다. 따라서, 캐러셀의 길이를 결정하게 되면 그에 맞춰 B나 T의 값을 적절하게 결정하여 알고리듬의 성능을 향상시켜 줄 수 있다.

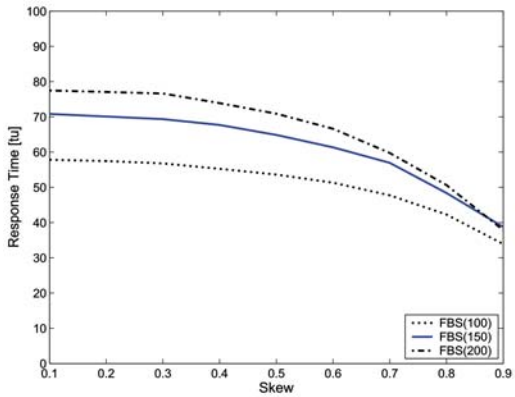

그림 7. T를 변화시켰을 때의 FBS Response Time 추이 ($\lambda = 0.6$)

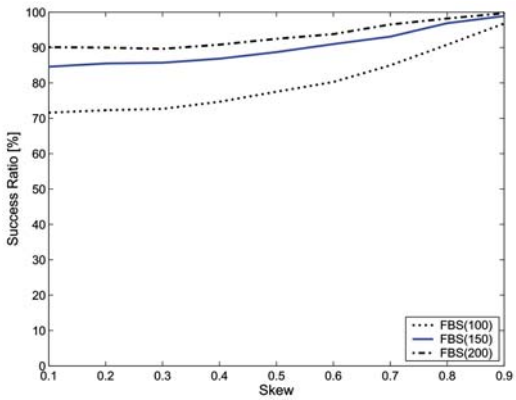

그림 8. T를 변화시켰을 때의 FBS Success Probability 추이 ($\lambda = 0.6$)

Ⅳ. 결 론

본 논문에서는 주문형 디지털 방송환경에서 캐러셀에 기반한 데이터 방송 스케줄링 알고리듬을 제안하였다. 이 알고리듬은 방송 아이템이나 그 아이템에 대한 리퀘스트 정보를 이용하여 적절한 프로그램을 구성하여 방송함으로써 클라이언트의 만족도를 높이고자 하였다. 리퀘스트가 많은 아이템은 하나의 캐러셀에서 자주 방송하도록 하여 클라이언트가 기다리는 시간을 줄이고자 하였으며, 아이템에 대한 리퀘스트의 대기시간이나 아이템 크기까지 고려하여 실제 모의 실험을 통해 높은 응답 성공률을 얻어냄과 동시에 상대적으로 짧은 응답시간을 가져올 수 있었다.

본 논문에서 제안한 방법은 시스템의 방송 정책에 따라 리퀘스트 비에 대한 경계값 θ을 변경할 수 있으며 hot 아이템의 방송빈도수에 대한 상한값 U 또한 캐러셀 길이 등을 고려하여 정할 수 있다. 앞으로, 본 논문의 결과는 멀티미디어 디지털 방송의 데이터 방송 부분에 효과적으로 이용될 수 있을 것으로 기대된다.

참 고 문 헌

〔1〕 http://www.worlddab.org

〔2〕 Ping Xuan, Subhabrata Sen, Oscar Gonzalez, Jesus Fernandez, Krithi Ramamritham, “Broadcast on demand : Efficient and Timely Dissemination of Data in Mobile Environments,” Proc. of the 3rd IEEE Real-Time Technology and Applications Symposium (RTAS ’97), p.38, June 1997.

〔3〕 Swarup Acharya, Rafael Alonso, Michael Franklin, Stanley Zdonik, “Broadcast Disks : Data Management for Asymmetric Communication Environments,” Proc. of the ACM SIGMOD Conference, San Jose, CA, May 1995.

〔4〕 Nitin H. Vaidya, Sohai Hameed, “Scheduling data broadcast in asymmetric communication environments,” Wireless Networks, vol. 5, p.171~182, May 1999.

〔5〕 Demet Aksoy, Michael Franklin, “R×W : A Scheduling Approach for Large-Scale On-Demand Data Broadcast,” IEEE trans. on Networking, vol.7, no.6, pp.846~860, Dec. 1999.

〔6〕 Jesus Fernandez-Conde, Krithi Ramamritham, “Adaptive Dissemination of Data in Time-Critical Asymmetric Communication Environments,” Proceedings of the 11th Euromicro Conference on Real-Time Systems, pp. 195~203, June 1999.

〔7〕 Yufei Guo, Sajar K. Das, Cristina M. Pinotti, “A New Hybrid Broadcast Scheduling Algorithm for Asymmetric Communication Systems : Push and Pull Data Based on Optimal Cut-off Point,” Proc. of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, pp.123~130, 2001.

〔8〕 ETSI EN 301 234-v.1.2.1-1999-02, “Digital Audio Broadcasting : Multimedia Object

Transfer protocol," ETSI, 1999.
〔9〕 Sven Buchholz, Steffen Gobel, Alexander Schill, Thomas Ziegert, "Dissemination of mutable sets of web object," Proceedings of the 13th IASTED International Conference on Pararllel and Distributed Computing and Systems, August, 2001.

최 수 정 (Sujeong Choi) 정회원

1996년 2월 서울시립대학교 수학과 졸업
1999년 8월 서울시립대학교 수학과 석사
2005년 8월 서울시립대학교 전자전기컴퓨터공학부 박사
현재 서울시립대학교 정보기술연구소 연구원
<관심분야> 휴대통신 서비스, 디지털방송 서비스

박 익 현 (Ik Hyun Park) 준회원

2004년 2월 서울시립대학교 전자전기공학부 졸업
2004년 3월~현재 서울시립대학교 전자전기컴퓨터공학부 석사과정
<관심분야> 디지털 방송 서비스

강 상 혁 (Sang Hyuk Kang) 정회원

1990년 2월 한국과학기술원 전기 및 전자공학과 졸업
1994년 2월 한국과학기술원 전기 및 전자공학과 석사
1997년 2월 한국과학기술원 전기 및 전자공학과 박사
2001년 1월~2002년 12월 University of California at Berkeley연구원
1997년 9월~현재 서울시립대학교 전자전기컴퓨터공학부 부교수
<관심분야> 초고속 통신망, 라우터 및 스위치, 멀티미디어전송, 무선네트워킹