韓國컴퓨터情報學會 論文誌
第14卷 第2號, 2009. 2.

2009-14-2-3-4

하이브리드 브로드캐스트 환경에서 효과적인 동적 브로드캐스팅 기법

최 재 훈*, 이 진 승*, 강 재 우**

Effective Dynamic Broadcast Method in Hybrid Broadcast Environment

Jaehoon Choi *, Jinseung Lee *, Jaewoo Kang **

요 약

무선 환경이 점점 발전함에 따라 모바일 기기들의 사용이 증가하고 있다. 핸드폰, PDA등 모바일 기기를 이용하여 Cellular망 및 Wibro망 등을 통해 데이터를 주고받고 있다. 그러나 무선통신은 유선통신보다 훨씬 열악한 대역폭을 가지므로, 이러한 한계를 해결하기 위한 방법으로 브로드캐스트 시스템이 대두되고 있다. 보다 효율적인 브로드캐스트 시스템을 구축하기 위해 많은 연구들이 진행되어 왔는데, 그 연구의 한 축은 어떻게 하면 효율적인 브로드캐스트 프로그램을 작성할 것이냐에 대한 문제이다. 이 논문에서는 기존의 요청 확률(Request Probability)을 기반으로 작성된 정적 브로드캐스트 프로그램의 한계를 극복하는 동적 브로드캐스트 프로그램을 제안하고자 한다. 동적 브로드캐스트 프로그램은 실시간으로 전달되는 클라이언트 요청을 바탕으로 클라이언트의 요청이 반영된 다른 방송을 지속적으로 브로드캐스트하는 시스템을 말한다. 이를 통해 수시로 달라지는 클라이언트의 요구를 지속적으로 반영할 수 있는 브로드캐스트 프로그램을 만들어 보고자 한다. 본 논문에서는 [1]에 발표된 선행연구결과를 확장해 보다 자세한 방법론의 기술과 다양한 각도에서의 성능분석모델을 포함했다.

Abstract

We are witnessing rapid increase of the number of wireless devices available today such as cell phones, PDAs, Wibro enabled devices. Because of the inherent limitation of the bandwidth available for wireless channels, broadcast systems have attracted the attention of the research community. The main problem in this area is to develop an efficient broadcast program. In this paper, we propose a dynamic broadcast method that overcomes the limitations of static broadcast programs. It optimizes the scheduling based on the probabilistic model of user requests. We show that dynamic broadcast system can indeed improve the quality of service using user requests. This paper extends our previous work in [1] to include more thorough explanation of the proposed methodology and diverse performance evaluation models.

▸ Keyword : 브로드캐스트 스케줄(Broadcast Schedule), 동적 스케줄(Dynamic Schedule), 하이브리드 브로드캐스트(Hybrid Broadcast)

• 제1저자 : 최재훈 교신저자 : 강재우
• 투고일 : 2008. 12. 17, 심사일 : 2009. 1. 7, 게재확정일 : 2009. 2. 12.
* 고려대 정보통신대학 컴퓨터학과 ** 고려대 정보통신대학 컴퓨터학과 교수
※이 연구에 참여한 연구자(의 일부)는 '2단계 BK21사업'의 지원비를 받았음
※ 본 논문의 선행 연구결과는 2008년 차세대컴퓨팅 추계 학술대회에서 발표된바 있음

Ⅰ. 서 론

비대칭 통신 환경(Asymmetric Communication Environment)[2][3]이 제안된 이 후, 브로드캐스트 시스템은 지속적으로 발전해 왔다. 기존 대칭 통신 환경에서는 서버에서 클라이언트가 요청한 데이터를 보내주는 것이 기본방식이었다. 이런 방식을 풀 기반 통신(Pull-based Communication)[4]이라 하는데 클라이언트와 서버가 1:1로 통신하기 때문에 양방향 통신을 기본으로 한다. 양방향 통신을 하기 위해서는 Up-Down 스트림이 모두 있어야 하기 때문에 실제로 데이터가 지나다닐 수 있는 통로는 전체 대역폭의 반이라고 할 수 있다. 반면 비대칭 통신은 양방향 통신에 사용되는 대역폭을 한쪽으로 몰아서 넓은 대역폭을 확보할 수 있다[5]. 이 확보된 대역폭을 통해 서버에서 대량의 데이터를 한꺼번에 발송하고 클라이언트는 단지 보내주는 데이터 속에서 자신이 원하는 데이터만 찾아 다운 받게 되는데 이 같은 방식을 푸쉬 기반(Push Based)방식[6][7] 이라고 한다.

푸쉬 기반의 브로드캐스트 기법이 도입되면서 무선 통신에서의 대역폭 한계를 어느 정도 극복할 수 있게 되었지만, 그 이후로도 해결되어야 할 많은 문제들이 존재한다. 그 중 핵심적인 문제 중에 하나는 "어떤 데이터를 브로드캐스트할 것이냐"이다. 아무리 넓은 대역폭을 확보하고 데이터를 전송해 준다고 해도 클라이언트가 원하는 데이터가 아니라면 소용이 없기 때문이다. 초기의 브로드캐스트방식은 서버에서 지정된 데이터를 주기적으로 반복 전송하는 형태의 방송이었다. 이는 브로드캐스트 프로그램을 만드는 시점에서 최적화된 데이터일 뿐이지 시간의 흐름에 따라 변하는 사용자의 요구는 반영하지 못하는 한계점이 있었다. 하지만 그 후 하이브리드 브로드캐스트[8] 방식이 등장하면서 이 같은 문제를 어느 정도는 해결할 수 있게 되었다. 하이브리드 브로드캐스트 시스템은 Private Channel을 통해 클라이언트 요청을 받을 수 있게 된 시스템을 말한다. 이렇게 수집된 클라이언트 요청을 바탕으로 데이터의 평판을 조사하고 자주 요청된 인기 있는 데이터만을 추려 브로드캐스트할 수 있게 되었다. 이 같은 방식을 통해 이제 우리의 관심사는 "어떤 방식으로 브로드캐스트 프로그램을 구성할 것인가?"로 넘어 오게 된다.

브로드캐스트 프로그램에 다른 데이터보다 인기 있는 데이터를 상대적으로 자주 방송을 해주면 이를 원하는 클라이언트들이 보다 효과적으로 데이터를 받아갈 수 있을 것이다. 이 같은 아이디어를 바탕으로 제안된 이론이 브로드캐스트 디스크 방식[9][10]이다. 이는 데이터를 요청 빈도별로 나눈 후 각각을 디스크라 칭하고, 각 디스크의 상대적 빈도를 바탕으로 가중치를 주어 브로드캐스트 프로그램을 작성한다. 가중치 빈도가 높은 디스크를 'Fast Disk'라 하여 프로그램에 자주 실어주고, 낮은 디스크는 'Slow Disk'라 하여 Fast Disk와의 상대적인 차이만큼 적게 실어준다. 하지만 이런 방식은 통계적 분석에 중점을 둔 상태로 "브로드캐스트 프로그램을 어떻게 구성할 것인가?"만을 고려하고 있고, 실시간 애플리케이션 적용에 대해서는 전혀 고려하지 않은 상태이다. 이 논문에서 제안할 동적 브로드캐스트 프로그램은 실시간 애플리케이션에서 어떻게 하면 보다 효과적으로 브로드캐스트 디스크를 활용 할 수 있는지를 보여주고자 한다.

Ⅱ. 동적 브로드캐스트

동적 브로드캐스트 시스템은 실시간 애플리케이션이 운영되면서 사용자의 요청을 즉각적으로 브로드캐스트 프로그램에 반영해 주는 시스템을 말한다. 브로드캐스트 프로그램을 단 한 번씩만 방송하고, 해당 주기가 방송되는 동안 수집된 클라이언트 요청을 바탕으로 다음 주기를 제작한다. 이와 같은 작업은 반복적으로 수행하여 클라이언트 요청에 최적화 되어 있는 브로드캐스트 프로그램을 만들 수 있다.

2.1 가정

동적 브로드캐스트 프로그램을 만들기 위한 가정은 다음과 같다.

1. 서버와 클라이언트는 방송되는 모든 데이터를 알고 있다.
2. 브로드캐스트 디스크 크기는 이전 주기에서의 사용자 요청에 따라 달라질 수 있다.
3. 모든 데이터의 크기는 일정하다.
4. 이전 주기에서 사용자 요청이 한 건도 없었을 경우, 현재 주기를 한 번 더 발송한다.

이와 같은 가정을 통해 얻을 수 있는 이익은 다음과 같다. 첫째, 인기 있는 데이터는 다음 주기에서도 인기 있을 것이라는 가정이 생기게 된다. 인기 있는 데이터는 이전 주기에 해당 데이터를 요청했던 요청들에 의해 해당 주기에도 그 데이터가 포함될 확률이 높기 때문이다. 이로 인해 인기 있는 데이터를 요청하는 클라이언트는 요청 즉시 자기가 원하는 데이터를 받을 수 있는 기회를 확보하게 된다. 이는 인기 있는 데이터를 요구하는 대다수의 클라이언트의 액세스 시간을 효과적으로 줄일 수 있을 것으로 기대된다[11].

둘째, 인기 없는 데이터를 요청하는 사람은 해당 데이터가

현재 주기에 존재하지 않을 확률이 높음으로 데이터를 바로 받기 쉽지 않다. 하지만 현재 요청한 데이터가 다음 주기를 생성하는데 사용되므로 다음 주기에 클라이언트는 반드시 받을 수 있다. 인기 없는 데이터를 요청한 클라이언트는 평균적으로 1주기(현재주기의 1/2 + 다음주기의 1/2)의 액세스 시간을 기다리면 원하는 데이터를 받을 수 있다[12].

이런 특징은 기존 브로드캐스트 시스템에서 얻을 수 있는 인기 있는 데이터의 빠른 액세스 시간을 보장할 뿐 아니라, 인기 없는 데이터에 대한 응답도 보장 받게 한다는 점에서 강점을 가지고 있다.

2.2 브로드캐스트 프로그램 생성

이 챕터에서는 브로드캐스트 프로그램의 구조 모델과 이전 주기에서 발생한 요청을 기반으로 브로드캐스트 프로그램 생성 방법에 대해 알아보도록 한다. 브로드캐스트 알고리즘은 클라이언트의 요청 인기도를 반영한다.

브로드캐스트 프로그램 생성 방법은 선행연구[13] 알고리즘의 순서로 진행한다. 알고리즘 내 데이터는 같은 크기를 가지고 있다고 가정한다.

단계 1. 인기 있는 데이터부터 인기 없는 데이터 순으로 데이터를 정렬한다.

단계 2. 같은 인기도를 가진 몇 개의 영역으로 분할한다. 이 영역을 디스크라고 칭한다.

단계 3. 각 디스크의 상대적 브로드캐스트 빈도를 설정한다. 이 상대적 빈도는 정수로 표현한다. 예로 두 개의 디스크가 있다면, 디스크 A가 3번 브로드캐스트 할 때, 디스크 B가 2번 브로드캐스트 했다면 rel freq(A) = 3, rel freq(B) = 2로 설정한다.

단계 4. 각 디스크의 상대적빈도 최소공배수를 각 디스크의 상대적 빈도수로 나누어 chunk값을 구한다. chunk는 디스크를 분할한 작은 단위를 의미한다. 이전 단계의 예를 이용하면, chunk(A) = 2, chunk(B) = 3 값을 가진다.

단계 5. 브로드캐스트 프로그램을 아래와 같은 방법으로 각 디스크의 chunk를 교차하여 작성한다.

```
for j:= 0 to LCM of the relative frequencies -1
  for i:= 1 to number of disks
    Broadcast chunk Cj(i mod number of chunks(j))
  endfor
endfor.
```

2.3 동적 브로드캐스트 프로그램 생성

이번 섹션에서는 기존 브로드캐스트 디스크[14] 방식을 개선하여 우리가 제안한 동적 브로드캐스트 프로그램을 어떻게 만들 것인지에 대해 알아본다.

제안한 알고리즘은 다음과 같은 단계를 거친다.

단계 1. 모든 데이터를 균일하게 한 번씩 방송한다.

단계 2. 제일 첫 주기의 내용이 브로드캐스트 되는 동안 클라이언트 요청을 수집한다.

단계 3. 브로드캐스트 주기가 끝나고 나면 단계2에서 수집된 통계적 데이터를 바탕으로 인기 있는 데이터를 취합하여, 다음 브로드캐스트 프로그램을 제작한다.

단계 4. 3번에서 발생된 브로드캐스트를 방송하면서 다음 브로드캐스트 프로그램을 만들기 위한 클라이언트 요청을 모은다.

단계 5. 단계3과 단계4는 반복 수행한다. 만약 브로드캐스트 프로그램 1주기 동안 클라이언트의 요청이 단 한 건도 없으면, 해당 프로그램을 반복 방송한다.

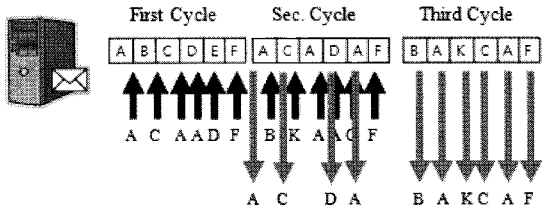

그림 1 실시간 동적 브로드캐스트 프로그램 예시
Fig. 1. Real Time Application of Dynamic Broadcast Program Example

첫 번째 주기는 데이터 요소들의 인기도가 아직 없기 때문에 공평하게 한번 씩 방송한다. 그리고 두 번째부터는 이전 주기 동안 모은 클라이언트 요청 기록을 바탕으로 만들어 진다. 데이터 인기도를 기반으로 만들어진 현재 주기를 이용하여 클라이언트들은 보다 효과적으로 데이터를 받을 수 있게 된다. 예를 들면 그림 1에서 보듯, 첫 번째 주기에서 데이터 A를 요청했던 클라이언트들은 두 번째 주기에서 첫 번째 데이터에 모두 요청했던 데이터 A를 받을 수 있다. 또한 첫 번째 주기에서의 클라이언트 요청에 의해 인기 있는 데이터로 판단된 데이터 A는 두 번째 주기에서 다른 데이터 요소보다 자주 발송되기 때문에, 두 번째 주기에서 데이터 A를 요청했던 클라이언트들은 확률적으로 세 번째 주기까지 기다리지 않고 두 번째 주기에서 데이터 A를 바로 받아 갈 수 있는 가능

성이 높다(두 번째 주기에서 요청된 세, 네번째 요청-검정 화살표). 반대로 인기 없는 데이터인 B를 요청한 클라이언트(두 번째 주기의 첫 번째 요청-검정 화살표)의 경우는 두 번째 주기에 존재하지 않기 때문에 바로 받을 수는 없지만 통계적으로 만들어지는 세 번째 주기에는 반드시 포함되기 때문에 아무리 오래 기다려도 평균 한 주기만 기다리면 원하는 데이터를 받을 수 있게 된다.

이처럼 제안한 동적 브로드캐스트 프로그램의 강점은 클라이언트들이 인기 있는 데이터를 요청했을 때는 최대한 빨리 받을 수 있고, 또한 동시에 인기 없는 데이터를 요청한 클라이언트들도 보장된 시간 안에 받을 수 있게 함으로써 대다수의 사용자를 만족시킬 수 있는 시스템이라 할 수 있다.

III. 실험

제안한 브로드캐스트 알고리즘을 실험하기 위하여 데이터의 수는 100개로 설정한다. 클라이언트가 요청하는 데이터는 100,000번으로 설정한다. 평가방법은 제안한 알고리즘으로 생성된 브로드캐스트 스케쥴의 총 액세스 시간을 분석한다. 총 액세스 시간은 클라이언트가 모든 요청한 데이터를 받을 때까지 걸리는 시간의 합을 의미한다.

3.1 데이터 확률 분포

관련 선행연구[15]를 참고하여 클라이언트의 요청을 Zipf 분포를 따른다고 가정한다. Zipf분포는 아래와 같은 식으로 표현할 수 있다.

$$p_i = \frac{(1/i)^\theta}{\sum_{i=1}^{M}(1/i)^\theta}, \quad 1 \leqslant i \leqslant M,$$

i는 데이터 인기순위, M은 데이터수, Theta(Θ)는 분포의 왜곡정도를 표시한다. Theta가 0이면 Pi 는 1이 된다. Theta가 증가할수록 분포는 상위 인기 데이터에 치우친 왜곡된 분포를 나타나게 된다.

3.2 성능평가 - Theta, 인기지속도 상관관계

먼저 Zipf 분포에서 Theta 값에 대한 민감도와 데이터의 인기 지속도는 총 액세스 시간에 어떤 영향을 미치는지 알아보자. Theta는 인기 있는 데이터가 다른 데이터에 비해 얼마나 더 인기 있는가를 측정하는 척도이고, 인기 지속도는 인기를 얻은 파일이 얼마나 오랫동안 인기가 지속된 것이냐에 대한 척도이다[16]. 실험 결과를 표로 나타내면 다음과 같다.

표 1. 총 액세스 시간과 Theta, 인기지속도 상관관계
Table 1. Total access time with temporality and popularity

인기 지속도 \ Theta	0.8	1	1.2	1.49
5000	8,776,320	6,316,437	3,332,297	1,379,920
10000	9,160,621	6,382,819	3,469,246	1,467,965
20000	9,094,176	6,322,514	3,448,246	1,443,495

표 1. 에서 봐도 쉽게 알 수 있듯이 총 액세스 시간은 인기 지속도 보다는 상대적 인기도의 영향을 많이 받는다. 인기 지속도는 10만번의 데이터 요청중에서 얼마의 주기로 인기 있는 데이터가 바뀌는 정도를 나타낸다. 예를 들어 5000이라고 하면 1~5000번의 요청안에서는 데이터 A가 인기를 얻다가 5001~10000번의 요청 사이에서는 데이터 B가 인기를 얻는 식이다. 반면 Theta 값은 0에 가까울수록 요청확률분포가 균일 분포를 따르게 되고, 커지면 커질수록 인기 있는 파일이 다른 파일들에 비해 요청되는 횟수가 훨씬 많다는 뜻이 된다[17].

그림 2에서 보듯이 인기지속도가 바뀌어도 동적 브로드캐스트 프로그램의 총 액세스 시간에는 큰 영향을 주지 못한다. 클라이언트의 요청 빈도가 바뀌지 않기 때문이다. 하지만 그림 3에서 보듯이 Theta 값의 변화는 동적 브로드캐스트 프로그램에 많은 영향을 준다. 가장 인기 있는 데이터의 출현 빈도수를 바꾸고, 그로 인해 한 브로드캐스트 프로그램에 실리는 데이터의 개수도 영향을 받는다. 상대적으로 다른 파일들이 적게 포함하게 되므로, 브로드캐스트 해야 하는 데이터개수 자체가 줄어들게 되어 보다 컴팩트한 브로드캐스트 프로그램을 생성할 수 있게 된다. 짧은 방송 시간은 다음 프로그램 생성까지의 요청수의 감소를 의미하며 이는 역시 다음 주기 프로그램도 컴팩트하게 만들어 질 수 있다는 뜻이 된다. 이렇게 컴팩트한 프로그램이 연속적으로 생성되어 방송이 되면 인기 없는 데이터를 요청해 다음 주기에서 데이터를 받아야 하는 클라이언트가 다음 주기에 접근할 수 있는 시간이 짧아진다. 이런 영향이 지속적으로 반복되어 가장 인기 있는 데이터의 빈도는 총 액세스 시간 큰 폭의 감소로 이어지게 된다[18].

위의 실험 결과로 동적 브로드캐스트 프로그램은 특정 데이터들이 다른 데이터들에 비해 인기도 차이가 분명할 때 유용한 시스템이다.

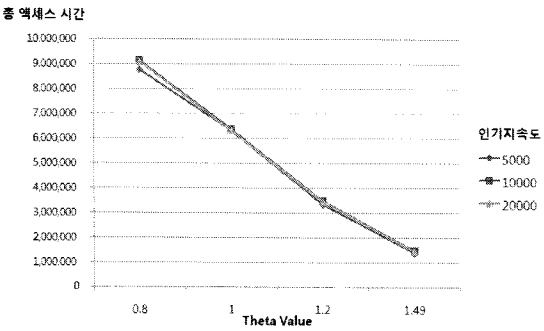

그림 2. Theta 값과 총 액세스 시간과의 상관 그래프
Fig. 2. Graph about total access time and Theta value

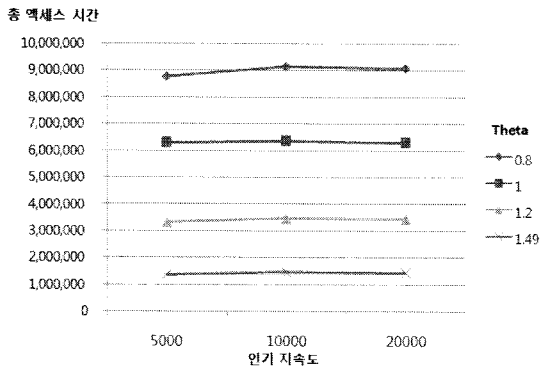

그림 3. 인기 지속도와 총 액세스 시간과의 상관 그래프
Fig. 3. Graph about total access time and Temporality

3.3 성능평가 - 동적 프로그램과 정적 프로그램 비교

3.2절에서 보듯이 Theta값이 작으면 프로그램 길이도 짧아지게 된다. 이렇게 동적 길이로 브로드캐스트하는 것이 정적 길이로 반복 방송하는 것에 비해 성능이 얼마나 향상 됐는지 알아보았다. 실험을 위해 사용된 데이터는 3.2절에서 실험했던 모델 중 인기 지속도 10000, theta 1.49를 이용하였다. 실험에 사용된 총 데이터의 수는 100개이고, 동적인 경우 평균적으로 약 60~70개 사이의 데이터를 방송하게 된다. 고정 시간 동안 방송 하는 것과 비교하기 위해 방송되는 데이터 개수를 정적으로 고정하고, 만들어진 프로그램을 반복 방송하는 식으로 실험을 진행하였다. 예를 들어 이전 주기의 요청 통계 자료를 통해 만들어진 다음 주기의 방송 데이터 개수가 60개 라고 하면 정적 120은 같은 방송을 정확하게 2번 방송하는 것이다. 정적 80이 경우 2번을 방송하기에 정해진 시간이 짧은 경우는 한번 방송 후 앞20개의 데이터만 반복 방송한다. 또 정해진 시간이 너무 짧아 생성된 프로그램의 데이터 수 보다 짧은 경우는 뒷부분에 방송되는 인기 없는 데이터는 방송이 되지 않는다. 이와 같은 이유로 응답을 받지 못하는 분실 데이터 개수도 따로 조사하였다[19]. 총 액세스 시간은 분실 데이터의 대기시간을 제외하고 계산하였다. 그림 4에서 보듯이 총 액세스 시간은 브로드캐스트 프로그램의 길이가 짧아질수록 좋아진다는 것을 알 수 있다. 인기 있는 데이터는 같은 주기에서 받을 확률이 높지만, 그렇지 않은 경우는 다음 주기까지 기다려야 하는데 프로그램의 길이가 길어지면 다음 주기까지 기다려야하는 시간이 길어지기 때문이다.

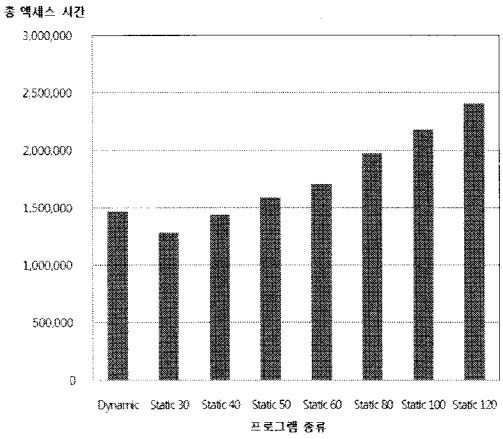

그림 4. 총 액세스 시간
Fig 4. Total Access Time

표 2. 성능평가 - 동적 프로그램과 정적 프로그램 비교
Table 2. Performance with Dynamic Program and Static Program

	동적	정적 30	정적 40	정적 50	정적 60	정적 80	정적 100	정적 120
총 액세스 시간	1,467,965	1,285,533	1,439,608	1,589,863	1,708,397	1,974,407	2,184,369	2,405,704
총 분실 데이터 개수	0	529	332	200	61	2	0	0
성능	100.00	147.63	126.34	121.93	119.84	134.58	148.80	163.88

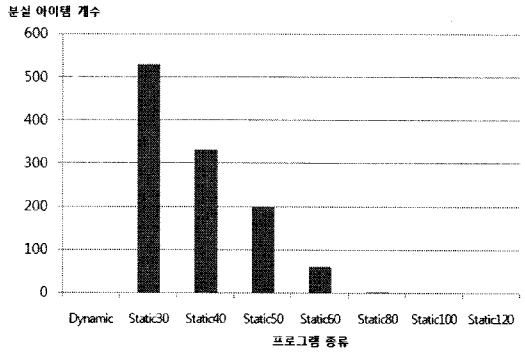

그림 5. 분실 데이터 개수
Fig 5. Total Missing data Count

그러나 무조건 짧을수록 좋은 것만은 아니다. 방송 시간을 한정하여 방송할 경우 1주기의 프로그램을 완전히 다 방송 할 수 없을 수도 있기 때문이다. 이런 경우 총 액세스 시간은 짧아지지만 요청한 데이터를 응답 받지 못하는 클라이언트가 생겨나게 된다. 그림 5.에서 보듯이 지정된 시간이 짧으면 짧아질수록 급격하게 많아지는 경향을 보인다.

위 2개의 결과를 토대로 아래의 식을 이용하여 표2.의 성능을 계산한다.

$$\text{성능}(\%) = \frac{\text{분실 데이타당 대기시간의 총합} + \text{정상 데이터의 총 액세스 시간}}{\text{동적 프로그램 총 액세스 시간}} \times 100$$

식을 통해 나온 성능 결과를 분석하면 동적 브로드캐스트 프로그램은 클라이언트에게 요청된 데이터는 반드시 받을 수 있다고 보장해 주면서도, 방송 시간을 상황에 맞춰 조절함으로 최적화된 브로드캐스트 프로그램을 생성할 수 있다.

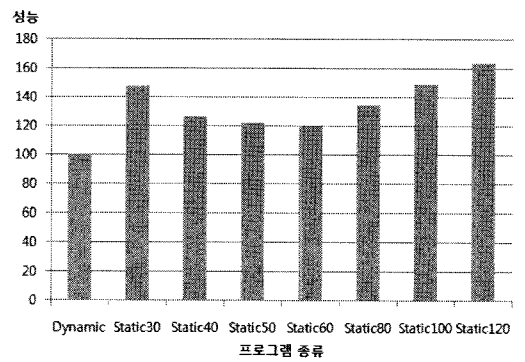

그림 6. 정적 프로그램과 성능비교
Fig 6. Performance improvement comparing with static Program

그리고 동적 브로드캐스트 프로그램의 평균데이터 개수가 60~70개인 점을 고려해 봐도 뛰어난 성능을 보이는 것을 알 수 있다. 비슷한 총 액세스 시간을 가지는 정적 40의 경우 동적 프로그램이 약 2% 성능 저하를 보이긴 하지만, 정적 40의 분실되는 데이터수가 330여개 달해 큰 성능차이가 발생한다. 반대로 비슷한 분실 데이터 개수를 가지는 정적 80의 경우와 비교해 봐도 성능은 약 25%정도가 좋은 것으로 나타났다.

Ⅳ. 요약 및 향후 연구

이번 연구에서는 클라이언트 요청을 확률적으로 분석하여 브로드캐스트 프로그램을 실시간으로 생성하여 전송하는 동적 브로드캐스트 프로그램을 제안하였다.

동적 브로드캐스트 프로그램의 강점은 데이터의인기도에 따라 프로그램을 제작함으로써 인기 있는 데이터를 원하는 대다수의 클라이언트에서 빠른 액세스 시간을 보장한다. 또한 낮은 인기도를 가지는 데이터를 요청하는 클라이언트도 반드시 데이터를 전송 받을 수 있게 보장해 줌으로서 브로드캐스트의 특성상 그 동안 희생을 강요당했던 낮은 인기도의 데이터를 요청하는 클라이언트도 고려할 수 있다.

제안한 동적 브로드캐스팅 방법은 특정 데이터에 요청이 집중되는 경향과 소수의 데이터 요청이 공존하는 어플리케이션에 활용된다면 좋은 성능을 보장 할 수 있을 것이다.

이번 연구를 통해 개선되어야 할 점은 다음과 같다.

1. 데이터의 Chunk값을 구하는 새로운 방법을 적용해 본다. 상대 빈도(rel freq)가 소수로 나오는 경우 최소공배수 과정에서 너무 큰 수가 생성되어 브로드캐스트 프로그램이 필요 이상 길어지는 경우가 발생한다. 예를 들어 데이터 A의 상대빈도가 37이고 데이터 B의 73인 경우, 최소공배수는 2701이 된다. Chunk(A)=73, chunk(B)=37로 설정되어 프로그램길이의 길이는 최소 110이 된다. 그러나 데이터 A와 B는 약 2배의 차이가 나므로 Chunk(A)=2, Chunk(B)=1로 설정하면 프로그램 길이가 줄어들어 성능의 향상을 기대할 수 있다.
2. 디스크 Chunking시 Chunk수가 데이터 개수보다 큰 상황이 발생하면 데이터가 자기 고유의 상대 빈도보다 자주 방송되는 경우가 발생한다. 이를 개선하면 보다 짧고 컴팩트한 브로드캐스트 프로그램을 만들 수 있어 성능 향상이 기대된다.
3. 데이터의 크기를 고려한 실험을 추가한다. 이번 실험은 단위 크기의 데이터를 가정하고 실험하였기 때문에 실

제 환경에서 발생할 수 있는 다양한 데이터 크기를 적용해서 실험한다면 보다 효과적인 알고리즘을 제안할 수 있을 것이라 생각된다.

참고문헌

[1] 최재훈, 이진승, 여운동, 강재우, "비대칭 통신 환경에서 효과적인 동적 브로드캐스팅 기법 연구," 한국 차세대 컴퓨팅 학회 추계학술대회논문집, 103-106쪽, 2008년 11월.

[2] N. H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," ACM/Baltzer Wireless Networks Vol. 5, No. 3, pp. 171-182, May 1999.

[3] W. Wang, and C.V. Ravishankar, "Adaptive data broadcasting in asymmetric communication environments," Proceedings of the International Database Engineering and Applications Symposium, IDEAS, pp. 27-36, August 2004.

[4] Q. Fang, S.V. Vrbsky, H.-C. Chen and Y. Dang, "Pull-based broadcasting with timing constraints," International Journal of Parallel, Emergent and Distributed Systems, Vol. 20, No.3&4, pp. 235-252, Sept. 2005.

[5] W. Ma, Y. Li and M. Zhou, "Analysis of dynamic broadcast scheduling algorithms in asymmetric communication environments," Jisuanji Gongcheng/Computer Engineering Vol. 32, No. 6, pp. 103-106, Mar. 2006.

[6] E. Kusmierek, Y. Lu and D.H.C. Du, "Periodic broadcast with dynamic server selection," Multimedia Tools and Applications, Vol. 34, No. 3, pp. 267-297, Sept. 2007.

[7] 이호선, 조익래, 이규하, "대규모 무선 센서 네트워크 환경을 위한 다중 Sink 브로드캐스팅 기법 설계," 한국컴퓨터정보학회논문지, 제 10권, 제 4호, 239-248쪽, 2005년 9월.

[8] Y. Guo, S.K. Das and C.M. Pinotti, "A new hybrid broadcast scheduling algorithm for asymmetric communication systems: Push and pull data based on optimal cut-off point," Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 123-130, July 2001.

[9] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: data management for asymmetric communication environments," In Proceedings of ACM SIGMOD Conference on Management of Data, pp. 199-210, May 1995.

[10] S. Acharya, M. Franklin and S. Zdonik, "Dissemination-based data delivery using broadcast disks," IEEE Personal Communications Vol. 2, No. 6, pp. 50-60, Dec. 1995.

[11] 박미화, 이용규, "사용자 프로파일을 활용한 모바일 방송에서의 동적 스케쥴링," 한국컴퓨터정보학회논문지, 제12권, 제2호, 111-121쪽, 2007년 5월.

[12] S. Jiang and N. H. Vaidya, "Scheduling data broadcast to impatient users," In Proceedings of the ACM international workshop on Data engineering for wireless and mobile access, pp. 52 - 59, Aug. 1999.

[13] V.C.S. Lee, X. Wu, and J.K.-Y. Ng, "Scheduling real-time requests in on-demand data broadcast environments," Real-Time Systems, Vol. 34, No. 2, pp. 83-99, Oct. 2006.

[14] S. Hameed and N.H. Vaidya, "Efficient algorithm for scheduling data broadcast," ACM-Baltzer Journal of Wireless Networks, Vol. 5, No. 3, pp.183-193, May 1999.

[15] J. P. Martin-Flatin, "Push vs. Pull in Web-Based Network Management," Proc. 6th IFIP/IEEE International Symposium on Integrated Network Management, pp. 3-18, May 1999.

[16] S. Aalto, J. Aaltonen and J. Karvo, "Quantitative performance comparison of different content distribution modes," Performance Evaluation, Vol. 63, No.4, pp. 395-422, May 2006.

[17] E. Ardizzoni, , A.A. Bertossi, M.C. Pinotti, S. Ramaprasad, R. Rizzi and M.V.S. Shashanka, "Optimal skewed data allocation on multiple channels with flat broadcast per channel," IEEE Transactions on Computers, Vol. 54, No. 5, pp. 558-572, May 2005.

[18] W.T. Chan, F.Y.L. Chin, Y. Zhang, H. Zhu, H. Shen and P.W.H. Wong, "A dynamic programming approach of finding an optimal broadcast schedule in minimizing total flow time," Journal of Combinatorial Optimization, Vol. 11, No. 2, pp. 177-187, Mar. 2006.

[19] D. Aksoy and M.S.-F. Leung, "Pull vs push: A quantitative comparison for data broadcast," GLOBECOM - IEEE Global Telecommunications Conference, Vol. 3, pp. 1464-1468, Nov. 2004.

저 자 소 개

최 재 훈

(E-mail : jaehoon@korea.ac.kr)

2007년 고려대 정보통신대학 컴퓨터학과 이학 학사

2007년~현재 고려대 정보통신대학 컴퓨터학과 석사과정

관심분야 : Online Stream Data Mining, Massive Log Data Mining

이 진 승

(E-mail : jjiinn@korea.ac.kr)

2008년 홍익대 컴퓨터공학과 학사

2007년~현재 고려대 정보통신대학 컴퓨터학과 석사과정

관심분야 : 데이터베이스 시스템, 소셜 네트워크

강 재 우

(E-mail : kangj@korea.ac.kr)

1994년 고려대 컴퓨터공학과 학사

1996년 University of Colorado at Boulder 컴퓨터공학과 석사

2003년 University of Wisconsin-Madison 컴퓨터학과 박사

2003년~2006년 North Carolina State University 컴퓨터학과 조교수

2006년~현재 고려대학교 정보통신대학 컴퓨터·통신공학부 조교수

관심분야 : 소셜 네트워크, 데이터 마이닝