

## 주문형 데이터 방송을 위한 인덱스 구성

강선희<sup>o</sup>, 이상돈

국립 목포대학교 정보공학부

### Index Organization for On-Demand Data Broadcast

Sunny Kang<sup>o</sup>, Sangdon Lee

Division of Information Engineering, Mokpo National University

#### 요 약

무선 이동 컴퓨팅 환경에서 자원의 비대칭성을 효과적으로 극복하는 것이 매우 중요하다. 이 중에서도 배터리 용량은 클라이언트의 유용성을 결정하는 중요한 자원이다. 그러므로 무선 컴퓨팅 환경에서 매우 효과적인 데이터 방송 기법은 데이터를 빠르게 전달하는 것은 물론 이동 클라이언트의 배터리 소모를 감소시킬 수 있는 효율적인 데이터 전달 기법을 필요로 한다. 본 논문에서는 주문형 데이터 방송 환경을 위한 인덱스 구성 방안을 제안한다. 제안 기법은 방송 서버에서 대기중인 데이터 접근 요구 중 방송 우선순위가 높은 일부 데이터를 대상으로 인덱스를 구성하여 미리 방송함으로써 이를 참조하는 클라이언트의 적용 시간을 감소시켜 전력소모를 줄이도록 한다. 제안 기법이 효과적으로 이동 클라이언트의 적용 시간을 감소시키는 것을 실험을 통해 검증한다.

#### 1. 서론

무선 통신과 이동 컴퓨팅은 최근 몇 년동안 비약적인 성장을 이루고 있는 새로운 정보서비스의 환경이다. 휴대용 컴퓨터 기술의 발전과 무선 통신 기술의 진보에 힘입어 시간과 장소에 제한 받지 않고 컴퓨팅 서비스를 받을 수 있는 이동 컴퓨팅 환경에 대한 연구가 활발히 진행되고 있다. 이동 컴퓨팅 환경에서 물리적인 제약은 이동 클라이언트가 방송 서버와 비교하여 매우 제한된 자원을 갖는다는 것이다. CPU, 메모리, 배터리 용량, 통신대역폭 등과 같은 유용한 자원을 포함하여 매우 제한된 자원을 갖는다. 특히나 배터리 용량은 클라이언트의 유용성을 결정하는 중요한 자원 중의 하나이다. 이러한 자원의 비대칭성은 이동 컴퓨팅 환경에 적합한 데이터 전달 기법을 요구한다.

무선 이동 환경에서 데이터 방송은 정보를 가지고 있는 서버가 이동 클라이언트에게 데이터를 방송하면 클라이언트가 자신이 원하는 데이터가 방송채널에 나타날 때 이를 검색하는 방법이다. 이러한 이동환경에서 데이터 방송은 이동 클라이언트의 전력소모를 감소할 수 있는 매력적인 데이터 전달기법이다[1]. 데이터 방송의 효율성을 측정하는 기준은 접근 시간(access time)과 적용 시간(tuning time)이다. 접근 시간은 클라이언트가 질의를 요청한 순간에서부터 모든 요청한 데이터 항목을 수신할 때까지 경과된 시간을 말한다. 적용 시간은 클라이언트가 요청한 데이터를 얻을 때까지 방송채널을 수신하면서 소모하는 시간을 말한다. 효율적인 데이터 전달을 위해서는 이러한 두 가지 기준을 최소화시키는 것이 요구된다.[3, 4, 6]. 이동 클라이언트의 활동 상태(active mode)에서 전력소모는 수면 상태(doze mode)에서보다 더 크다. 그러므로 클라이언트의 전력소모를 감소시키기 위해서는 적용 시간을 줄이는 것이 주요 목표이다.

적용 시간의 감소를 위해서는 클라이언트에게 데이터를 받을 시점을 미리 알려주는 것이 필요하며, 이를 위해 방송 인덱스를 구성하는 것이 효과적이다[4]. 방송 인덱스는 데이터의 접근 패턴이 미리 알려져 클라이언트의 명시적인 요구가 없이 반복적으로 데이터를 방송하는 주기적 방송 환경에서 주로 연구되어왔다[2, 4, 5, 7]. 그러나 클라이언트의 가변적인 접근 요구에 따라 방송 스케줄이 동적으로 결정되어야 하는 주문형 방송 환경을 위한 인덱스 구성 기법은 상대적으로 많은 연구가 이루어지지 않았다. 한편 주기적 데이터 방송 환경에서 제안되었던 인덱스 기법들을 주문형 데이터 방송 환경에 직접 적용하는 것은 가능하지 않다[8].

본 논문에서는 방송 서버에서 대기중인 데이터 접근 요구 중 방송 우선순위가 높은 일부 데이터를 대상으로 인덱스를 구성하여 미리 방송함으로써 이를 참조하는 클라이언트의 적용시간

을 감소시켜 전력소모를 줄이도록 하는 주문형 데이터 방송 환경을 위한 인덱스 구성 및 데이터 방송 구성 방안을 제안한다. 인덱스에는 각 데이터의 방송 시간 정보를 포함하고 있고 이를 통해 클라이언트가 각 개별 데이터 항목의 수신 시점을 인지하여 적용 시간을 줄이게 된다. 2장에서는 관련 연구들을 설명하고, 3장에서는 방송을 위한 인덱스 구성과 데이터 구성 기법에 대하여 기술한다. 4장에서는 제안기법의 성능을 평가하기 위한 실험결과를 분석하고, 마지막으로 5장에서 결론을 맺는다.

#### 2. 관련 연구

데이터 방송은 통신 대역폭의 비대칭성을 이용하여 높은 대역폭의 서버가 하향 채널을 통해 데이터를 방송하고, 클라이언트는 원하는 데이터가 방송채널에 나타날 때 이를 접근하게 함으로써 클라이언트의 수에 무관하게 효과적으로 데이터를 전달할 수 있는 확장성이 큰 데이터 전달 방법이다[1]. 데이터 방송은 크게 두 종류로 구분이 가능하다. 주기적 데이터 방송은 방송 대상 데이터 항목의 순서와 방송의 빈도 등 클라이언트의 접근 요구가 방송 전에 미리 결정되어 전체적인 방송 스케줄이 결정되는 정적인 방송스케줄을 사용한다[3]. 반면에 주문형 데이터 방송은 방송의 전체 순서가 미리 결정되어지는 것이 아니라 클라이언트의 동적인 데이터 요청에 따라 방송 중에 결정되는 동적 방송 스케줄을 갖는다. 동적 방송 스케줄은 방송 서버에 대기중인 데이터 요청에 대한 방송순서를 결정하는 방송 스케줄링 정책에 따라 결정된다. 결국 주문형 데이터 방송 환경에서 데이터 방송 순서는 요청분포와 스케줄링 정책에 기반하여 적용되어진다.

주기적 방송 환경에서는 데이터가 방송되기 이전에 이미 데이터의 방송 시점이 결정된다. 그러므로 이를 사용하여 전체 방송 스케줄에 대한 인덱스를 구성하고, 실제 데이터의 방송에 앞서 인덱스를 먼저 방송하는 방식으로 클라이언트의 적용 시간을 크게 줄일 수 있다[4]. 주문형 데이터 방송을 위해 효율적인 인덱스의 구조들에 대한 연구가 이루어져 왔으며, 해싱, 트리 구조 또는 시그니처(signature)를 사용한 기법들이 제안된바 있다[2, 4, 5, 6, 7].

방송 인덱스의 적용을 위해서는 클라이언트가 요청한 데이터가 실제 방송되기 전에 방송 시점을 미리 정확히 아는 것이 필요하다. 그러나 주문형 데이터 방송은 방송 중에 방송스케줄이 동적으로 결정되어지기 때문에 전체 방송 스케줄에 대해서 데이터가 언제 방송되어질 것인지를 정확히 정의하는 것이 쉽지 않다. 그러므로 주기적 방송 환경을 위해 제안되었던 인덱스 기법들이 그대로 주문형 방송 환경에 적용될 수 없다. 주문형 데이터 방송 환경을 위해 최초로 제안된 인덱스 접근 방법

[8]은 주문형 데이터 방송 환경에서 방송 서버에 데이터의 요구가 쌓이게 되고, 이 데이터 요구들을 기반으로 방송 순서가 결정된다는 점에 착안하여 각 개별 데이터의 방송 시간을 예측하는 대신에, 일정 시간동안에 방송이 될 가능성이 큰 데이터 집합을 선정하고 이를 클라이언트에게 알려주는 인덱스 힌트 기법이다. 이 접근 방법에서는 개별 데이터 항목에 대한 방송 시간 정보를 포함하지 않기 때문에 클라이언트는 요구한 데이터가 일정 시간구간 동안에 방송될지 여부를 알 수 없고, 이로 인해 해당 구간동안 계속해서 방송 채널을 감시해야만 했다. 본 논문에서 제안하는 기법에서는 인덱스 정보에 개별 데이터의 방송 시간 정보까지도 포함하도록 함으로써 클라이언트가 보다 세밀하게 방송 채널을 감시 여부를 판단할 수 있도록 하여 낭비되는 적용시간을 줄이도록 하였다.

3. 인덱스 구성과 데이터 방송 구성 기법

3.1 인덱스 구성

방송 슬롯(broadcast slot)은 방송 데이터를 포함하고 있는 데이터 슬롯과 인덱스를 포함하고 있는 인덱스 슬롯으로 구분한다. 각 슬롯은 헤더정보를 가지며, 각 슬롯을 정의하는 슬롯 구분자를 통해 인덱스와 데이터를 구분한다.

인덱스는 [인덱스 집합, 인덱스 구간]으로 구성된다. 인덱스 집합은 <키, 시간>의 쌍으로 구성된다. 인덱스 집합에서 키는 데이터 항목의 키 부분을 의미한다. 인덱스 집합에서 시간은 키를 포함하고 있는 데이터 항목이 방송되는 시점을 나타낸다. 인덱스 구간은 인덱스 집합에 있는 모든 데이터 항목이 방송되는 동안의 시간 간격으로 나타내진다. 모든 시간은 단위 방송 시간(Broadcast tick)을 사용하는 것으로 가정한다.

데이터 슬롯은 키 부분과 데이터 부분으로 이루어진다. 데이터 슬롯의 마지막에는 다음 인덱스 슬롯의 위치를 지시하는 포인터(Next Index Pointer : NIP)를 가질 수 있다. 이 NIP를 클라이언트가 데이터를 요청하고 방송 채널을 감시하기 시작하는 시점에 어떻게 반응할 것인지를 결정하는 역할을 한다. 만일 클라이언트가 처음 받은 슬롯이 인덱스 슬롯이었다면 당연히 클라이언트는 인덱스 슬롯을 보고 해당 데이터의 수신 가능성을 확인할 수 있다. 그러나 처음 받은 슬롯이 데이터 슬롯인 경우에는 NIP의 여부에 따라 다음 두가지로 반응한다. 만일 NIP가 없는 경우(이하 NIPX라 함) 클라이언트는 다음 인덱스 슬롯을 만날때까지 계속해서 방송 채널을 감시해야 한다. 반면에 NIP가 있는 경우에는(이하 NIP0라 함) 다음 인덱스 슬롯의 방송 시점을 알 수 있으므로 클라이언트가 수면모드로 들어가게 된다.

그림 1은 데이터 방송에서 인덱스를 갖는 방송스케줄을 보여준다.

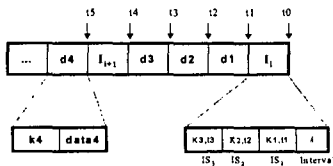


그림 1 인덱스 구성의 예

3.2 방송 데이터 구성

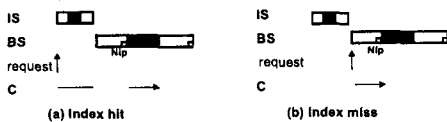


그림 2 방송 데이터 구성의 예

서버는 클라이언트에게 방송 채널을 통하여 동적으로 인덱스를 생성하고 인덱스와 데이터를 보낸다. 즉, 인덱스 집합(Index Set:IS)과 데이터 집합(Broadcast Set:BS)을 보낸다. IS는 서버의 현재 우선순위 큐로부터 상위 N개 데이터 항목을 선택함

으로써 인덱스 집합을 결정한다. IS는 데이터를 구분하는 키 정보와 그 키를 가진 데이터가 방송되어질 시간 정보가 포함된다. 방송 시간은 데이터의 크기와 우선순위의 곱으로 계산된다. BS에 포함된 데이터가 전부 방송되는 시간은 인덱스의 크기와 단위 방송시간의 곱으로 계산된 시간이 소요된다.

그림 2는 클라이언트 C가 데이터 항목 a를 요청하고 방송 채널을 수신하는 두 시점을 나타낸다. 이 예에서 (a)의 경우(Index hit)는 클라이언트가 방송 채널 수신을 개시한 시점에 필요한 인덱스 정보를 수신한 경우이다. (b)의 경우(Index miss)는 클라이언트가 방송 채널 수신을 개시한 시점에 이미 인덱스 정보가 방송되어버린 경우를 의미한다.

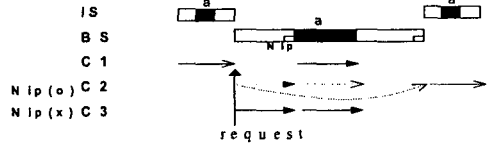


그림 3 데이터구성요소(NIP)에 따른 클라이언트 수신 방법

그림 3은 데이터 항목 a를 요청한 클라이언트 C1, C2, C3가 IS와 BS에 대해서 어떻게 동작하는지를 보여주는 예이다. 이 예에서 실선은 방송채널을 수신하는 활동상태를 의미하고 점선은 클라이언트가 수면 상태에 있음을 의미한다. Index miss가 발생한 경우 BS가 방송된 후 다음 인덱스가 방송된다. 만일 클라이언트가 이 시점을 알지 못하는 경우(즉 NIPX의 경우) 클라이언트는 다음 인덱스의 방송 시점까지 활동상태를 유지해야만 한다. BS의 크기는 IS의 크기에 비교하여 상당히 클 것이므로 이로 인한 전력소모가 커질 수도 있다. 그림 3에서 C3의 경우에 해당한다. 반면에 NIP0는 데이터 방송 구성요소에 NIP를 포함하게 하여 이 구간동안에 수면 모드에 들어갈 수 있으며 그림 3에서 C2에 해당한다.

3.3 방송 스케줄링

방송 서버는 현재 클럭과 i번째 인덱스를 생성할 때 클럭 값, clock<sub>i</sub>를 갖는다. 인덱스 구간(I<sub>i</sub>)을 위한 clock = clock<sub>i-1</sub> + I<sub>i-1</sub> 일 때 인덱스 I<sub>i</sub>를 생성하고 그것을 방송한다. 그렇지 않으면 인덱스 집합으로부터 우선 순위가 가장 높은 데이터 항목을 방송한다. 데이터 항목이 방송된 후 서버 큐에서 그 항목은 제거된다. 다음은 데이터의 방송을 위한 기본적인 스케줄링 알고리즘이다.

<방송 스케줄링>

```

1: while(방송 서버의 큐가 비어있지 않은 동안) {
2:   만일 (clock = clocki-1 + Ii-1)이면 {
3:     N개의 최상위 방송 우선순위 데이터 선정
4:     인덱스 생성
5:     인덱스 방송 }
6:   else {
7:     방송 대상 데이터 집합중 최우선 순위 데이터 선정
8:     해당 데이터 방송
9:     방송된 데이터를 서버 큐에서 제거
10:  } /* else */
11: } /* while */
    
```

위의 스케줄링 알고리즘은 NIPX를 위한 것으로서 NIP0 기법의 경우에는 약간의 수정이 필요하다. 그 이유는 NIPX의 경우 Index miss가 발생시 클라이언트가 다음 인덱스 정보를 수신할 때까지 수면 모드에 있게 되는데, 만일 이 동안에 그 클라이언트가 요청했던 데이터가 실제로 방송이 되어버리면 위의 스케줄 정책에 따라 해당 데이터가 서버 큐에서 제거되어버린다. 그렇게되면 수면 모드에 있던 클라이언트는 다시 깨어나더라도 해당 데이터를 수신할 수 없게 되거나 또는 다른 클라이언트들이 그 데이터에 대해서 다시 요청을 하는 시점까지 수신이 지연되므로 상대적인 불이익을 당하게 된다. 그러므로 이를 해결하기 위해서 NIP0의 경우에는 이 클라이언트에 대한 새로운 요청을 백업 큐에 일단 저장하였다가 새로운 인덱스가 생성되면 자기 전에 백업 큐에 있는 데이터가 요청을 현재 큐에 병합하는

방식을 사용한다.

4. 실험 및 결과

4.1 시뮬레이션 환경

시뮬레이션 언어인 CSIM을 사용하여 1개의 방송 서버와 많은 클라이언트가 존재하는 주문형 데이터 방송 환경을 구축하였다. 한 페이지가 방송되는 시간을 단위 방송시간(broadcast tick)이라 하고 이를 성능 측정기준으로 사용한다. 서버는 총 100개의 페이지를 갖는 데이터베이스를 보유하고 있다. 각 클라이언트는 서버에 페이지를 요청하며 각 요청의 간격은 포아송 분포를 따른다고 가정하였다. 요청되는 데이터가 편중되는 경우를 위하여 각 페이지에 대한 클라이언트들의 요청은 Zipf 분포를 따르도록 하였다.

성능 분석은 적응시간과 응답시간을 목표로하여 세가지 기법, 즉, 인덱스 정보에 개별 데이터의 방송 시간을 포함하지 않는 경우(참고문헌 [8] 중에서 HintsOnly 기법임-이하 HINTSONLY로 표기)), 그리고 본 논문에서 제안하는 인덱스 및 방송 기법 두 가지인 NIPX와 NIPO를 비교하였다.

4.2 적응 시간 분석

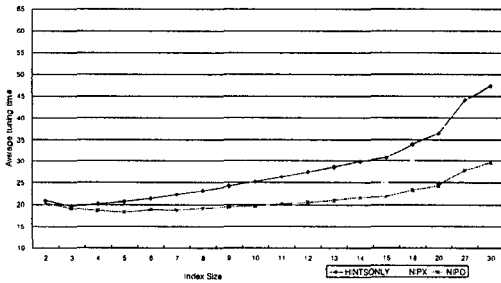


그림 4 평균 적응 시간

그림4는 제안된 기법들의 평균 적응 시간을 보여준다. 인덱스의 크기가 전체 데이터베이스의 20% 이하인 경우 기존 연구 결과인 HINTSONLY에 비해서 본 논문에서 제안한 두 기법의 적응 시간이 상대적으로 작았다. 즉 제안 기법이 효과적으로 클라이언트의 소비 전력을 줄일 수 있음을 의미한다. 작은 인덱스 크기 구간에서는 NIPX가 NIPO 보다 우수하였으나 인덱스 크기가 전체 데이터베이스의 약 10% 보다 커지면 성능이 역전되었고 특히 20%이상 크기가 되면 HINTSONLY 보다도 성능이 감소하였다. 이는 NIPX의 경우 데이터 집합을 방송하는 동안 지속적으로 데이터를 수신해야하며, 데이터 집합이 많아지는 경우, 즉 인덱스의 크기가 커지는 경우 그 효과가 더욱 증가하기 때문이다. 기존 기법에 대한 제안 기법의 적응시간 감소 효과를 일부 정리하면 표 1과 같다.

	인덱스 크기=9	인덱스 크기=18
NIPX	23.93 %	10.83 %
NIPO	21.18 %	36.42 %

표 1 인덱스 크기에 따른 적응 시간의 감소

4.3 응답 시간 분석

제안 기법은 인덱스 구성과 데이터 구성에 따라 방송 대역폭 이용에 있어서 약간의 오버헤드를 갖는다. 왜냐하면 인덱스 정보가 부가적인 대역폭 소비를 필요로 하기 때문이다. 또한 데이터 구성요소 NIP 정보가 부가적인 대역폭 소비를 필요로 하기 때문이다. 그러므로 제안 기법은 HINTSONLY 기법보다 평균 응답 시간이 길어지게 된다. 그림5는 각 기법들의 평균 응답시간을 비교하는 결과이다. NIPO가 가장 큰 응답 시간을 가지며, NIPX는 HINTSONLY와 큰 차이를 보이지 않았다. 그러나 표 2에서 정리하였듯이 응답시간의 증가는 8% 이내로서 40% 가까운

적응시간의 감소 효과에 비하면 매우 작은 수치라 할 수 있다.

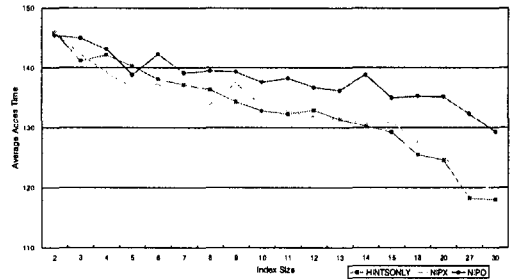


그림 5 평균 응답 시간

	인덱스 크기=9	인덱스 크기=18
NIPX	2.29 %	1.80 %
NIPO	3.72 %	7.81 %

표 2 인덱스 크기에 따른 응답 시간의 증가

5. 결론

본 논문에서는 주문형 방송을 위한 인덱스 구성과 이를 지원하기 위한 방송 데이터 구성 방안을 제안하였다. 인덱스 내에 개별 데이터에 대한 방송 시간 정보를 포함하고 인덱스 정보의 방송 시점에 대한 정보를 방송 데이터에 포함시키므로써 클라이언트가 보다 세밀하게 방송 채널의 수신 여부를 결정할 수 있고, 이를 통해 적응 시간을 효과적으로 감소시킬수 있음을 실험을 통해 검증하였다. 앞으로 변화하는 클라이언트의 접근 요구 및 데이터 편중도 등에 적용하여 적절한 인덱스의 크기를 결정할 수 있는 연구가 필요하다.

참고문헌

[1] Tomasz Imielinski, B. R. Badrinath, "Mobile Wireless Computing: Challenge in Data Management", Communications of the ACM, Vol37, No. 10, October 1994  
 [2] M. S. Chen, P. S. Yu, K. L. Wu. "Indexes Sequential Data Broadcasting in Wireless Mobile Computing", In Proceedings of the 17th International Conference on Distributed Computing Systems(ICDCS97), Baltimore, Maryland, USA, 1997  
 [3] S. Acharya, M. J. Franklin, S. Zdonik. "Broadcast disks : Data management for asymmetric communication environments", In Proc. of ACM SIGMOD Conf. on Management of Data, 1995  
 [4] T. Imielinski, S. Viswanathan, B. R. Badrinath. "Data on the Air - Organization and Access". IEEE transactions on Data and Knowledge Engineering, 1997.  
 [5] W. C. Lee, and D. L. Lee, "Using signature Techniques for information Filtering in Wireless and Mobile Environment". Special Issue on Database and Mobile Computing, Journal on Distributed and Parallel Databases, 1996  
 [6] T. Imielinski, S. Viswanathan, B. R. Badrinath. "Energy efficient filtering of data on air". In Proc. of ACM SIGMOD Intl. conference on Management of Data, 1994  
 [7] N. Shivakumar, and S. Venkatasubramanian. "Energy-efficient indexing for information dissemination in wireless system", ACM-Baltzer journal of Mobile Networks and Nomadic Applications(MONET), 1996  
 [8] Sangdon Lee, Donald Carney, Stanley B. Zdonik. "Index Hint for On-Demand Broadcasting", In Proc. of intl. conference Data Engineering, 2003