

# 무선통신 장애를 고려한 indexing broadcast 기법의 성능분석

정의중, 김재훈  
이주대학교 정보통신전문대학원

## Analysis and Measurement of Data Broadcast Schemes on Error Prone Wireless Link for Resilient Mobile Computing

Eui-jong Jeong, Jai-Hoon Kim  
College of Information and Computer Engineering, Ajou University

### 요 약

현재의 이동 컴퓨팅 환경은 더욱 보편화되고 있다. 그러나 배터리의 제약과 이동통신망의 높은 장애율로 사용에 불편을 느낀다. 여러 mobile client 는 공통 관심이 있는 데이터를 서버로부터 받는 방법으로 broadcasting 을 많이 쓰는데, 이때 indexing 기법을 이용해 클라이언트는 원하는 데이터를 filtering 해서 수신함으로써 에너지의 효율적 사용을 기할 수 있다. index 를 중복시킴으로써 원하는 데이터의 access 시간을 줄이고 무선 통신망의 장애에 따른 성능 저하를 줄일 수 있다. 본 논문에서는 장애율에 따른 최적의 중복회수를 구하고 데이터 수신시 access 시간과 tuning 시간을 구한다.

### 1. 서론

서버가 데이터를 전송하는 방법은 데이터 방송(data broadcasting)과 요청받은 데이터의 전송(interactive on demand)이다[1]. 데이터 방송은 서버가 클라이언트에 의해 필요로 하는 데이터를 미리 스케줄해서 전송 채널 (communication channel)에 전송하는 것으로 클라이언트가 일정한 지역내에 있는 한 추가 비용없이 전송할 수 있다. 그때 클라이언트는 전송 채널로 들어오는 데이터를 filtering 함으로서 원하는 데이터만 읽어 들인다. 방송 채널 (Broadcast channel)로 전송되는 데이터를 공중의 저장장치 (storage on the air)로 생각할 수 있고, 따라서 서버의 기억장치의 확장으로 생각할 수 있다. 그러나, 요청받은 데이터의 전송은 클라이언트가 원하는 데이터를 uplink channel 을 통해 데이터를 요청하고 서버는 클라이언트에게 요청했던 데이터를 전송하는 것이다. 소수의 클라이언트에게 사용되는 데이터를 전송할 때 이용된다. 이런 데이터를 방송 (broadcast)하는 것은 대역폭의 낭비이다. 실제상황에서는 위의 두가지 방법이 혼합되어 사용된다.

클라이언트는 doze mode 와 active mode 에서 동작된다[1]. doze mode 란 최소의 전력만을 사용하여 작업하는 상태이고 방송 채널로 어떤 데이터가 들어오는지 확인하지 않는다. active mode 는 방송 채널로 들어오는 데이터가 무엇인지 알기 위해 processor 가 작업하여 배터리를 사용하는 상태를 말한다. 현재의 클라이언트는 원하는 데이터를 받을 때만 active mode 이고 그렇지 않을 때는 doze mode 로 있는데 이런 능력을 선택적 데이터 수신(selective tuning)이라고 하고, energy 의 효율을 높일 수 있다. 선택적 데이터 수신이 가능할려면, 원하는 데이터의 위치를 알아야 한다. 데이터의 수이 사이에 index 를 둠으로써 원하는 데이터를 쉽게 찾을 수 있다. 기존의 index 기법[1]에서는 데이터 접근(data access) 시간을 줄이기 위하여 index 를 효과적으로 배치하고 중복시키는 방법을 고려하였다. 본 논문에서는 무선통신망의 장애를 고려하여 데이터 접근 시간과 tuning 시간을 분석하였고 최적의 index 중복회수를 구하였다.

### 1. Indexing Broadcast

다음은 본 논문의 기반인 설명한 index broadcast 기법[1]을 간단히 요약한 것이다.

#### 1.1 Broadcast 를 위한 Data 구성 [1]

브로드캐스트에 의해 전송된 데이터는 서버에 file 의 형태로 존재하고 file 은 primary key 에 의해 구분되는 record 들

로 구성된다. file 의 크기와 내용은 자주 변한다. 서버는 다수의 클라이언트들에게 주기적으로 브로드캐스트하며, 클라이언트는 전송되는 데이터를 받아서 primary key 값에 의해 구분되는 record 를 저장한다. 그러나, file 에 대한 갱신(update)는 연속된 전송사이에 반영된다. filtering 은 primary key 값의 simple pattern matching 에 의해 이뤄지며, 대부분의 시간을 doze mode 로 있게 된다. 데이터 item 은 계속 바뀌므로 서버는 클라이언트에게 지속적으로 이것을 알려야 하기 때문에, directory 및 data item name 을 브로드캐스트한다. 다양한 channel 에서 많은 데이터가 전송되므로 클라이언트는 원하는 데이터만 선택해 듣는 선택적 데이터 수신 기능이 필요하다. 서버는 data 와 index 를 방송한다. 다음은 본 논문에서 사용되는 용어에 대해 간략히 설명한다.

#### ◆ Access time

클라이언트가 primary key 값으로 원하는 데이터를 찾아 저장하는 데까지의 평균 걸린 시간을 말한다. Probe Wait and Bcast Wait 의 합으로 나타낼 수 있는데, probe wait 은 지난번 데이터를 읽고 index 를 읽는데 걸린 평균 시간을 말하며 bcast wait 은 최초의 index 를 읽고 원하는 데이터를 읽는데 걸린 평균 시간을 말한다. probe wait 와 bcast wait 는 반비례한다. bcast wait 을 최소로 할려면, index 를 한번만 전송하면 된다. 이 때는 필요한 데이터가 깨지게 되면 항상 다음 index 를 기다려야 하므로 probe wait 는 증가할 것이다. probe wait 를 최소로 할려면, index 를 매 데이터 사이에 끼워넣으면 되나 broadcast 의 길이가 증가하므로 bcast wait 는 증가할 것이다.

#### ◆ Tuning time

채널을 결정하기 위해 걸린 시간, 즉 채널을 결정하기 위해 다운받은 인덱스 및 데이터를 읽는 시간을 의미한다. 이 시간동안 mobile client 는 active mode 로 있다.

#### ◆ Bucket

broadcast 의 가장 작은 논리적인 단위이다. bucket 의 크기는 모두 같다. access time 과 tuning time 은 bucket 의 수로 측정된다.

#### ◆ Index bucket

필요한 데이터를 읽기위한 offset 들의 bucket .

#### ◆ access optimal

Access time 이 최소로 될려면, index bucket 이 없을 때, 즉 브로드캐스트의 길이가 최소일 때이다. 하지만 이 경우는 tuning time 이 최대가 되어 항상 active mode 로 있어야만 한다. 이때 Access time 은 Data / 2 로 tuning 시간과 동일하다.

#### ◆ tuning optimal

Index bucket 을 브로드캐스트의 시작부분에 한번만 넣는

것이다. 클라이언트는 최초로 index 를 읽고 primary key 로 원하는 데이터의 위치를 알아내어 그때까지 doze mode 로 들어간다.

원하는 데이터의 위치가 되면, active mode 로 전환되어 필요한 데이터를 읽어낸다. tuning time 은 index level 의 수와 최초의 index 위치를 알기위한 데이터를 읽는 것과 원하는 데이터를 읽는 것의 합으로 구할 수 있다. probe wait 는  $(Data + Index) / 2$  이고 bcast wait 는  $(Data + Index) / 2$  이다. 따라서 access time 은  $Data + Index$  이다. 이것은 access time 이 최대가 된다.

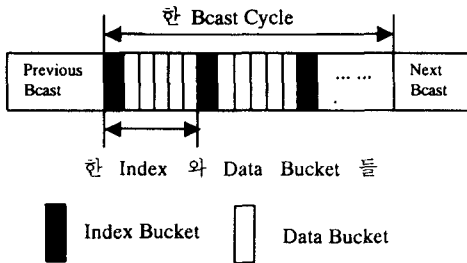


그림 1. 1 : m Index

1.2.1 : m index(1)

그림 1 은 1 : m index 를 나타내기 위한 설명이다. index bucket 과 data bucket 을 구분해야 한다. 한 broadcast cycle 에 m 번 index 를 넣는 방법으로, 모든 bucket 은 시작부분에 다음 index segment 에 대한 offset 을 갖고 있다. 각 index segment 의 첫 bucket 은 첫 field 에 마지막에 broadcast 되었던 record 의 primary key 가 들어가 있고, 두번째 field 는 다음번 브로드캐스트의 시작 offset 이 들어가 있다. 이것은 클라이언트가 원하는 데이터의 읽기가 실패했을 때, 다음번 브로드캐스트의 시작으로 tune 하게 한다. 다음의 protocol 에 의하여 원하는 데이터를 수신한다.

- ① 현재의 bucket 을 읽는다.
- ② 다음의 가장 가까운 index segment 의 주소를 알려주는 offset 을 읽는다.
- ③ doze mode 로 들어가고 index segment 에서 tune 한다.
- ④ Index segment 에서 primary key 로 원하는 데이터의 위치를 결정한다. multilevel index 를 이용할 때에는 연속적인 probe 가 일어난다.
- ⑤ primary key 에 의해 구해진 데이터의 위치까지 tune 해서 데이터를 down 받는다.

다음은 access 시간과 tuning 시간을 분석한다. 사용되는 parameter 는

- ◆ n = 한 bucket 이 index 할 수 있는 양
- ◆ index tree 의 level 의 수로서, index tree 가 fully balanced 할 때이다.

$$K = \lceil \log_n Data \rceil$$

- ◆ Index = Index bucket 의 크기
  - ◆ Data = 한 브로드캐스트 주기의 Bucket 의 수
- 위의 파라미터로 분석하면, 다음과 같다. m 은 index bucket 의 중복도이다.

$$probe\ wait = \frac{1}{2} \cdot \left( Index + \frac{Data}{m} \right)$$

$$bcast\ wait = \frac{1}{2} \cdot (m \cdot Index + Data)$$

$$access\ time = \frac{1}{2} \cdot Index \cdot (m + 1) + \frac{1}{2} \cdot Data \cdot \left( \frac{1}{m} + 1 \right)$$

$$tuning\ time = 2 + K$$

인덱스 단계를 거치면서 데이터를 읽는데, 처음과 마지막에 데이터를 읽기 때문이다. Access time 을 최소화하는 최적의 인덱스 중복도(m)를 구하면,

$$M = \sqrt{\frac{Data}{Index}}$$

2. 무선 통신망의 장애를 고려한 Indexing Broadcast

이동 computing 환경에서는 무선 통신망에 장애가 발생할 가능성이 높다. 이 때 어떻게 인덱스를 배치하고 어느정도 중복해야 하는 가를 고려하는 것이 중요하다. 기본적으로 원하는 데이터를 읽다가 데이터가 손상되었을 경우, 다음번 주기의 데이터를 읽는다. 본 논문에서는 index bucket 의 크기는 충분히 커서 multilevel 의 index 라 해도 한 bucket 안에 포함된다고 가정한다.

무선 통신망의 장애를 고려한 index broadcast protocol 은 다음과 같다.

- ① 현재의 bucket 을 읽는다. 여기서 데이터가 손상되면, 다음 bucket 을 읽는다.
- ② 다음의 가장 가까운 index segment 의 주소를 알려주는 offset 을 읽는다.
- ③ doze mode 로 들어가고 index segment 에서 tune 한다.
- ④ index segment 에서 primary key 로 원하는 데이터의 위치를 결정한다. multilevel index 이기때문에 연속적인 probe 가 일어난다. 이때, 데이터 읽기가 실패하면, ①부터 다시한다.
- ⑤ primary key 에 의해 구해진 데이터의 위치에서 tune 해서 down 받는다. 실패하면, ①부터 다시한다.

위의 protocol 을 이용할 때 access 시간과 tune 시간을 구하면 다음과 같다. 무선통신망 장애는 지수분포간격(평균 1/p 마다 발생)으로 발생한다고 가정하였으며 이 때 한 bucket 의 index 또는 데이터가 손상된다고 가정하였다.

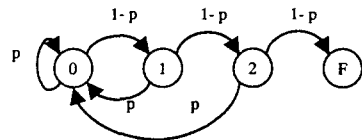


그림 2. Markov model

그림 2 에서와 같이 Markov model 로 나타낼때, state 0 은 첫번째 데이터를 읽는 상태를 나타내며, state 1 은 index 를 읽는 상태를 나타내고, state 2 는 데이터를 읽는 상태를 나타내며, state F 는 종료상태를 나타낸다. state F 에서 다시 state 0 부터 다시 시작한다.

access 시간을 계산하기 위하여 상태전이에 따른 cost 를 구하면,

- ◆ 0 → 0 : 1  
데이터나 index bucket 을 읽다가 실패 했을 경우 다음 bucket 을 읽는다.)
  - ◆ 0 → 1 :  $(D/m + 1) / 2$   
가장 가까운 index bucket 의 읽는데 걸리는 평균시간
  - ◆ 1 → 0 : 1  
index bucket 을 읽기가 실패했을 경우 다음 bucket 을 읽는다.
  - ◆ 1 → 2 :  $(D + m \cdot I) / 2$   
index bucket 을 읽고 data bucket 을 읽는 평균시간
  - ◆ 2 → 0 : 1  
실패했을 경우 0 부터 다시 시작.
  - ◆ 2 → F : 0  
한 주기가 끝나고 0 부터 다시한다.
- 그리고 tuning 시간을 계산하기 위하여 transition cost 를 구하면 다음과 같다. 각 cost 는 한 bucket 만 읽으면 된다.

- ◆ 0 → 0 : 1
- ◆ 0 → 1 : 1
- ◆ 1 → 0 : 1
- ◆ 1 → 2 : 1
- ◆ 2 → 0 : 1
- ◆ 2 → F : 1

access 시간을 최소화 하기 위한 최적의 인덱스 중복도 (m)를 구하기 위해 미분하면,

$$\begin{aligned}
 \text{access time} &= \frac{p}{1-p} \left( 1 + \frac{1}{1-p} \right) \\
 &+ \left( m + \frac{1}{1-p} \right) \cdot \frac{\text{Index}}{2} \\
 &+ \left( 1 + \frac{1}{m(1-p)} \right) \cdot \frac{\text{Data}}{2} \\
 \text{tuning time} &= \frac{2-p}{(1-p)^2} + 1 \\
 m &= \sqrt{\frac{\text{Data}}{\text{Index}} \cdot \frac{1}{1-p}}
 \end{aligned}$$

### 3. 파라미터 분석

한 Broadcast 의 주기당 데이터의 크기는 1000 으로 가정했으며, 파라미터를 변화하면서 분석한 결과를 표시하면 다음과 같다.

그림 3 은 인덱스 중복회수에 따른 access 시간의 변화를 나타낸다. 장애율에 따라 인덱스 중복회수가 최소가 되는 값이 존재함을 알 수 있다. 그림 4 는 무선통신망 장애율에 따른 access 시간을 나타낸다. 장애율이 높아질수록 access 시간이 증가함을 알 수 있다. 그림 5 은 무선통신 장애율에 따른 tuning 시간을 나타낸다. 무선통신망의 장애율에만 영향을 받지 인덱스 중복회수에는 영향을 받지 않음을 알 수 있다.

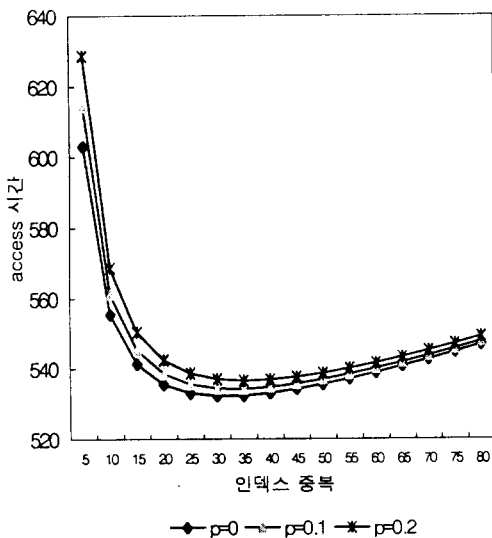


그림 3. 인덱스 중복회수에 따른

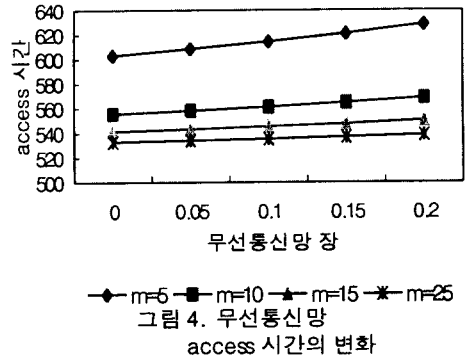


그림 4. 무선통신망

access 시간의 변화

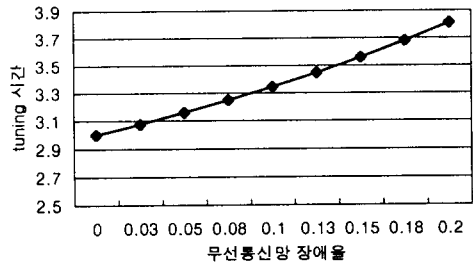


그림 5. 무선통신망 장애율에 tuning 시간의 변화

### 4. 결론 및 향후 연구

이 논문에서는 무선통신망의 장애가 빈번할 때 인덱스 기법을 이용한 방송기법에서 인덱스를 어느정도 중복해야 하는 가를 고려해 보았다. 분석결과에 따르면, 장애가 큰 지역일 경우 인덱스를 많이 넣어야 함을 알 수 있다. 현재는 여러 단계의 인덱스 버킷이 있을 경우 어떻게 배치하고 어떻게 중복시켜야 하는가에 관한 연구를 진행중이다.

### 5. 참고 자료

- [1] T. Imielinski, S. Viswanathan, B. R. Badrinath "Energy efficient indexing on air" SIGMOD 94-5/94 Minneapolis, Minnesota, USA, 1994 ACM.
- [2] E. Pitoura, G. Samaras "Data management for mobile computing".
- [3] David Cheriton, "Dissemination oriented communication systems", Stanford University, Tech. Rept. 1992.
- [4] David J. Goodman, "Trends in cellular and cordless communications," IEEE Communications Magazine, June 1991.
- [5] G. Herrman et al., "The datacycle architecture for very large high throughput database systems," in Proc ACM SIGMOD Conf., 1987, pp 97-103.
- [6] T. Imielinski, S. Viswanathan, B. R. Badrinath "Power efficient filtering of data on air" Proc of 4<sup>th</sup> Intl Conference on Extending Database Technology, Cambridge - Mach 94.
- [7] T. Imielinski, S. Viswanathan, B. R. Badrinath "Data on Air : Organization and Access," Submitted for publication.