

# 무선통신망의 에러가 데이터방송의 인덱스 기법에 미치는 영향 분석

정희원 정의종\*, 김재훈\*

## Analysis of the effect of wireless link errors on multi-level indexing broadcast schemes

Eui-jong Jeong\*, Jai-Hoon Kim\* *Regular Members*

### 요 약

현재의 이동 컴퓨팅 환경은 더욱 보편화되고 있다. 그러나 전원장치의 제약과 이동통신망의 높은 장애율로 사용에 불편을 느낀다. 여러 이동 단말기는 공통의 관심이 있는 데이터를 서버로 부터 받는 방법으로 브로드캐스트를 많이 이용한다. 이때 다중 레벨(multi-level) 인덱스 기법을 이용해 클라이언트는 원하는 데이터를 필터링(filtering) 해서 수신함으로써 에너지의 효율적 사용을 기할 수 있고, 인덱스 중복을 이용하여 원하는 데이터의 액세스(access) 시간을 줄일 수 있다. 본 논문에서는 전체 복사 인덱스(full replication index) 기법과 분산 인덱스(distributed index) 기법에 대하여 데이터 수신시 액세스 시간과 튜닝(tuning) 시간을 구하고, 전체 복사 인덱스 기법에 대하여 무선 통신망의 장애에 따른 성능저하를 줄이기 위하여 장애율에 따른 최적의 중복회수를 구하였다.

### ABSTRACT

Mobile computing and wireless networks are fast-growing technologies. Data broadcast is effective in dissemination-based applications to transfer the data to a large number of users because the downstream communication capacity is relatively much greater than the upstream communication capacity. Index based organization for data broadcast is important to reduce power consumption. We model the indexing data broadcast on the error prone wireless link, using Marcov model. We analyze the average access time and tuning time for full replication and distributed index schemes. We also measure the performance of data broadcast schemes by simulation with various system parameters.

### 1. 서 론

서버가 데이터를 전송하는 방법은 데이터 방송(data broadcasting)과 요청받은 데이터의 전송(interactive on demand)이다<sup>[1]</sup>. 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<sup>[3,6]</sup>. 그때 클라

이언트는 전송 채널로 들어오는 데이터를 필터링함으로써 원하는 데이터만 읽어 들인다. 방송 채널(Broadcast channel)로 전송되는 데이터를 공중의 저장장치(storage on the air), 즉, 서버의 기억장치의 확장으로 생각할 수 있다. 그러나 요청받은 데이터의 전송은 클라이언트가 원하는 데이터를 uplink channel을 통해 데이터를 요청하고 서버는 클라이언트에게 요청했던 데이터를 전송하는 것이다. 소수의 클라이언트에게 사용되는 데이터를 전송할 때 이용

\* 아주대학교 정보통신 전문대학원 정보통신 공학과({avenue, jaikim}@madang.ajou.ac.kr).

논문번호: 00344-0828, 접수일자: 2000년 8월 28일

※ 이 논문은 2000년도 한국학술진흥재단의 지원에 의하여 연구되었습(KRF-2000-041-E00293).

된다. 이런 데이터를 방송(broadcast)하는 것은 대역 폭의 낭비이다. 실제상황에서는 위의 두가지 방법이 혼합되어 사용된다.

클라이언트는 도즈 모드(doze mode)와 액티브 모드(active mode)에서 동작된다<sup>[1]</sup>. 도즈 모드란 최소의 전력만을 사용하여 작업하는 상태이고 방송 채널로 어떤 데이터가 들어오는지 확인하지 않는다. 액티브 모드는 방송 채널로 들어오는 데이터가 무엇인지 알기위해 프로세서(processor)가 작업하여 배터리를 사용하는 상태를 말한다. 현재의 클라이언트는 원하는 데이터를 받을 때만 액티브 모드이고 그렇지 않을 때는 도즈 모드로 있는데 이런 능력을 선택적 데이터 수신(selective tuning)<sup>[11]</sup>이라고 하고, 에너지(energy)의 효율을 높일 수 있다. 선택적 데이터 수신이 가능하려면, 원하는 데이터의 위치를 알아야 한다. 데이터의 사이사이에 인덱스를 둬서 원하는 데이터를 쉽게 찾을 수 있다. 효율적인 데이터 방송을 위해 기존의 기법들<sup>[1,2,3,4,5,6,7,8,9,10]</sup>이 개발되었다. 본 논문에서는 데이터 접근(data access) 시간을 줄이기 위하여 전체 복사 인덱스(full replication index)와 분산 인덱스(distributed index)기법을 고려하였다. 무선통신망의 장애를 고려하여 데이터 액세스 시간과 튜닝 시간을 분석하였고 최적의 인덱스 중복회수를 구하였다.

본 논문에서 사용되는 용어는 첨부에 설명하였다.

## II. 인덱스 방송기법

### 1. 방송을 위한 데이터 구성

브로드캐스트에 의해 전송된 데이터는 서버에 파일(file)의 형태로 존재하고 파일은 중추 키(primary key)에 의해 구분되는 레코드(record)들로 구성된다. 파일의 크기와 내용은 자주 변한다. 서버는 다수의 클라이언트들에게 주기적으로 브로드캐스트하며, 클라이언트는 전송되는 데이터를 받아서 중추 키(primary key)값에 의해 구분되는 레코드를 저장한다. 그러나 파일에 대한 갱신(update)는 연속된 전송사이클에 반영된다<sup>[2]</sup>. 필터링은 중추 키 값의 간단한 패턴매칭(simple pattern matching)에 의해 이뤄지며, 대부분의 시간을 도즈 모드로 있게 된다. 데이터는 계속 바뀌므로 서버는 클라이언트에게 지속적으로 이것을 알려야하기 때문에, 디렉토리(directory) 및 데이터 항목(item) 이름(name)을 브로드캐스트한다. 다양한 채널에서 많은 데이터가 전송되므로 클라이언트는 원하는 데이터만 선택해 듣는 선택적

데이터 수신 기능이 필요하다. 서버는 데이터와 인덱스를 방송한다.

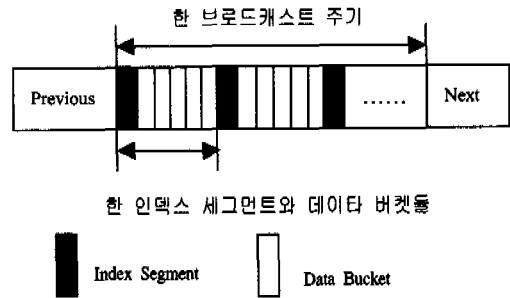


그림 1. 1 : m Index

### 2. 1 : m Index<sup>(1)</sup>

그림 1은 1 : m 인덱스를 나타내기 위한 설명이다. 인덱스 버킷과 데이터 버킷으로 구분되어 있으며 한 브로드캐스트 주기에 m번 인덱스를 중복시킨다. 모든 버킷은 다음 인덱스 세그먼트(정확하게 말하면 root 인덱스 버킷)의 위치(오프셋:offset)을 갖고 있다. 각 인덱스 세그먼트의 첫 버킷은 첫 필드(field)에 마지막에 브로드캐스트되었던 레코드의 중추 키가 들어가 있고, 두번째 필드는 다음번 브로드캐스트의 시작 오프셋이 들어가 있다. 이것은 클라이언트가 원하는 데이터의 읽기가 실패했을 때, 다음번 브로드캐스트의 시작까지 쉬었다가 읽기를 시작하게(tune) 한다.

다음의 프로토콜에 의하여 원하는 데이터를 수신한다.

- ① 현재의 버킷을 읽는다.
- ② 다음의 가장 가까운 인덱스 세그먼트의 주소를 알려주는 오프셋을 읽는다.
- ③ 도즈 모드로 들어가고 인덱스 세그먼트에서 읽기를 시작하게(tune)한다.
- ④ 인덱스 세그먼트에서 중추 키로 원하는 데이터의 위치를 결정한다. 다중 레벨 인덱스를 이용할 때에는 연속적인 인지(probe)가 일어난다.
- ⑤ 중추 키에 의해 구해진 데이터의 위치까지 쉬었다가 읽기를 시작해서(tune) 데이터를 다운(down) 받는다.

다음은 액세스 시간과 튜닝 시간을 분석한다. 사용되는 파라메타는

- n = 한 버킷이 인덱스할 수 있는 양
- 인덱스 트리의 레벨의 수로서, 인덱스 트리가 완전

평균(fully balanced)할 때이다.

$$K = \lceil \log_n \text{Data} \rceil$$

- Index = 인덱스 버킷의 크기
  - Data = 한 브로드캐스트 주기의 버킷의 수
- 위의 파라미터로 분석하면, 다음과 같다. m은 인덱스 세그먼트의 중복도이다.

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

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

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

$$\text{tuning time} = 2 + K$$

인덱스 단계를 거치면서 데이터를 읽는데, 처음과 마지막에 데이터를 읽기 때문이다. 액세스 시간을 최소화하는 최적의 인덱스 중복도(m)를 구하면,

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

### Ⅲ. 무선통신망의 장애를 고려한 Multi-level 인덱스 브로드캐스트

이동 컴퓨팅 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.

무선 통신망의 장애를 고려한 인덱스 브로드캐스트 프로토콜은 다음과 같다.

- ① 현재의 버킷을 읽는다. 여기서 데이터가 손상되면, 다음 버킷을 읽는다.
- ② 다음의 가장 가까운 인덱스 세그먼트의 주소를 알려주는 옵셋을 읽는다.
- ③ 도즈 모드로 들어가고 인덱스 세그먼트에서 읽기를 시작하게(tune)한다.
- ④ 인덱스 세그먼트에서 중추 키로 원하는 데이터의 위치를 결정한다. 다중레벨 인덱스이기때문에 연속적인 인지(probe)가 일어난다. 이때, 데이터 읽기가 실패하면, 처음부터 다시한다.
- ⑤ 중추키에 의해 구해진 데이터의 위치에서 튠(tune)해서 다운받는다. 실패하면, 처음부터 다시한다.

위의 프로토콜을 이용할 때 액세스 시간과 튠닝 시간을 구하면 다음과 같다. 무선통신망 장애는 장소나 현재 상황에 따라 변하기 쉽다. 장애가 발생할 확률은 지수분포간격(평균 1/p마다 발생)으로 발생한다고 가정하였다. 이때 한 버킷의 인덱스 또는 데이터가 손상된다고 가정하였다.

인덱스의 구조에는 전체 복사 인덱스(full replication index)와 분산 인덱스(distributed index) 기법이 있는데 전체 복사 인덱스는 인덱스 중복시 전체 인덱스 트리를 중복하는 기법이다. 이 기법은 인덱스 중복도가 높을 경우, 인덱스 오버헤드(index overhead)가 발생하는 반면 너무 낮으면, 인덱스 읽기의 실패시 다음 인덱스까지 기다리는 시간이 길어진다. 데이터 버킷의 위치를 알 수 없다. 이에 적당한 인덱스 중복도를 구하는 것이 중요하다. 전체 인덱스의 중복은 낭비가 심하기 때문에 인덱스 트리의 상위레벨 일부분만 중복하여 액세스 시간을 줄일 수 있다. 이 기법을 분산 인덱스(distributed index) 기법이라고 한다. 본 논문에서는 전체 복사 인덱스와 분산 인덱스 기법에서 액세스 시간과 튠닝 시간을 각각 분석한 후 비교하였다.

#### 1. 전체 복사 인덱스 기법

(full replication index)

다중 레벨의 인덱스를 고려하였기 때문에 이 모델(model)에서는 k 단계의 인덱스로 가정되었다.

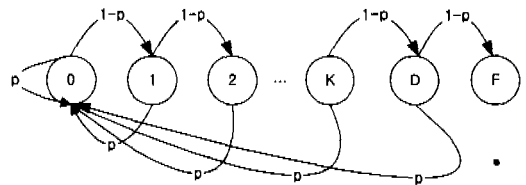


그림 2. 마코프 모델(Markov Model)

그림 2에서와 같이 마코프 모델(Markov model)로 나타낼때, 상태 0(state 0)은 첫번째 데이터를 읽는 상태를 나타내며, 상태 1은 최상위 인덱스(root index)를 읽는 상태를 나타내고, 상태 I(2-K)는 I 단계의 인덱스를 읽는 상태를 나타내며, 상태 D는 데이터를 읽는 상태를 나타내고, 상태 F는 종료상태를 나타낸다.

액세스 시간을 계산하기 위하여 상태전이에 따른 비용을 구하면(k index level) 표1과 같다.

표 1. 액세스 시간을 위한 상태전이 비용분석

상태 전이	전이 비용	설명
0 → 0	1	데이터나 인덱스 버킷을 읽다가 실패했을 경우 다음 bucket을 읽는다.
0 → 1	(D/m+1)/2	가장 가까운 최상위 인덱스를 읽는데 걸리는 평균시간.
1 → 0	1	최상위 인덱스를 읽기가 실패했을 경우 다음 버킷을 읽는다.
1 → 2	n / 2	최상위 인덱스를 읽고 다음 단계의 인덱스를 읽는 평균시간.
2 → 0	1	실패했을 경우 상태 0부터 다시 시작.
2 → 3	(n+n <sup>2</sup> )/2	다음 단계의 인덱스를 읽는데 걸리는 평균시간.
.....	.....	
k-1 → 0	1	
k-1 → k	(n <sup>k-2</sup> + n <sup>k-1</sup> )/2	다음 단계의 인덱스를 읽는데 걸리는 평균시간.
K → 0	1	
K → D	(D+m)/2	마지막 단계의 인덱스를 읽고 데이터 버킷을 읽는데 걸리는 평균시간.
K → 0	1	

위의 상태전이 비용을 이용하여 각각의 비용을 구하면,

$$C_{01} = c_{01} + c_{00} \cdot \frac{p}{1-p}$$

$$C_{02} = C_{01} \cdot \frac{1}{1-p} + c_{10} \cdot \frac{p}{1-p} + c_{12}$$

$$C_{03} = C_{02} \cdot \frac{1}{1-p} + c_{20} \cdot \frac{p}{1-p} + c_{23}$$

$$C_{04} = C_{03} \cdot \frac{1}{1-p} + c_{30} \cdot \frac{p}{1-p} + c_{34}$$

.....

$$C_{0k} = p \sum_{i=1}^{k-1} \frac{1}{(1-p)^i} + \sum_{i=0}^{k-1} \frac{c_{i(i+1)}}{(1-p)^{k-i-1}}$$

모든 버킷에는 인덱스에 대한 읍셋 정보를 포함하므로 어떤 상태에서 0의 상태로 전이되는 비용은 1이다. C<sub>0y</sub>는 상태 0에서 상태 y로 이동하기 위한 평균 비용이고(중간 경로의 모든 상태전이 비용 포함), c<sub>xy</sub>는 상태 x에서 상태 y로 직접 전이하기 위한 비용이다.

그리고 튜닝 시간을 계산하기 위하여 전이 비용을 구하면 표2와 같다. 각 비용은 한 버킷만 읽으면 된다.

이상의 상태전이에 따른 비용을 바탕으로 액세스 시간과 튜닝 시간을 구하면 다음과 같다.

표 2 튜닝 시간을 위한 상태전이 비용 분석

상태전이	전이비용	설명
0 → 0	1	각 상태전이에 따라 한 버킷만 읽으면 되므로 각각은 1이다.
0 → 1	1	
1 → 0	1	
1 → 2	1	
.....	.....	
K → 0	1	
K → D	1	

access time =

$$\frac{1}{2(1-p)^k} \left( 1 + \frac{1}{n} \sum_{i=2}^{k-1} \{n \cdot (1-p)\}^i \right) + \frac{1}{(1-p)^k} \left( \frac{D}{2m} + \frac{I}{2} + \frac{n}{2} - \frac{np}{2} + 1 \right) + \frac{1}{2}(D+mI)$$

tuning time =

$$(1+p) \sum_{i=0}^k \frac{1}{(1-p)^i} - p$$

액세스 시간을 최소화 하기위한 최적의 인덱스 중복도(m)를 구하기위해 액세스 시간을 미분하여 해를 구하면 다음과 같다(k level index).

$$m = \sqrt{\frac{D}{I(1-p)^k}}$$

## 2. 분산 인덱스 기법 (Distributed Index)

그림 3는 인덱스 트리의 예이다. 인덱스를 분산 배치하는 방법을 요약한 후 액세스 시간과 튜닝 시간을 분석한다.

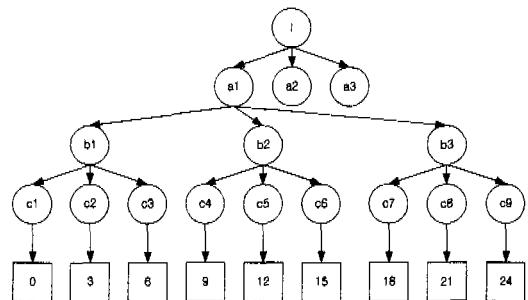


그림 3. 인덱스 트리의 예

분산 인덱스 기법은 전체 복사 인덱스 기법의 단점인 인덱스 중복 비용을 줄이기 위하여 인덱스 세

그먼트에서 불필요한 부분의 중복을 억제하는 기법이다. 분산 인덱스 기법을 세분하면 인덱스 트리에서 일정 레벨 이상의 내용만 중복시키는 부분 경로 중복(partial path replication), 인덱스 트리의 모든 상위 레벨을 중복시키는 전체 경로 중복(entire path replication), 그리고 중복이 전혀 없는 비중복(non replication distribution) 등이 있다.

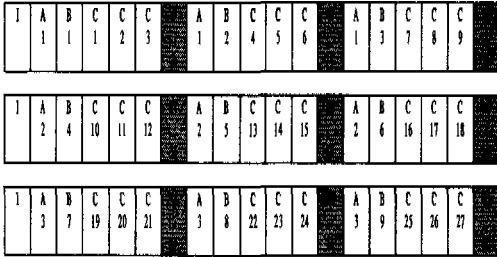


그림 4. Partial Path Replication.

본 논문에서는 부분 경로 중복 기법에 대하여만 비용 분석을 하였다. 그림 4는 부분 경로 기법의 예를 나타내는데 여기서는 첫번째 레벨(I)의 내용만 중복 시킴을 나타낸다.

무선 통신망의 장애를 고려한 부분 경로 중복 기법의 프로토콜은 III.1 장에서 설명한 전체복사 인덱스 기법의 프로토콜과 기본 내용이 동일하다.

그림 5 에서와 같이 마코프 모델(Marcov model)로 나타낼때, 상태 0(state 0)은 현재의 버킷을 읽는 상태를 나타내며, 각 상태에서 상태0으로 전이될 확률은 p인데 그림에서는 생략하였다. 상태 1은 최상위 인덱스(root index)를 읽는 상태를 나타내고, 상태 I는 I 단계의 인덱스를 읽는 상태를 나타내며, 상태 D는 데이터를 읽는 상태를 나타내고, 상태 F는 종료상태를 나타낸다. 그림 5에서 제일 위의 모

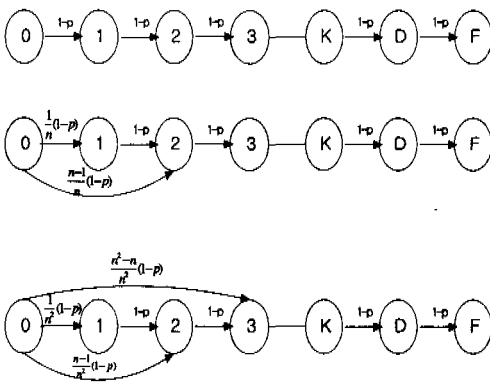


그림 5. 마코프 모델(Markov model)

델은 인덱스 중복 레벨이 1인 분산 인덱스 기법(부분경로 중복)을 나타내고, 가운데는 2, 세번째는 3인 분산 인덱스 기법을 나타낸다.

액세스 시간을 계산하기 위하여 상태전이에 따른 비용을 구하면(k index level) 표3과 같다.

표 3. 액세스 시간을 위한 상태전이 비용분석

상태전이	전이비용	설 명
0→0	1	데이터나 인덱스 버킷을 읽다가 실패 했을 경우 다음 bucket을 읽는다.
0→1	$\frac{1}{2}(\frac{D}{n^r} + \frac{n^{k-r}-1}{n-1})$	가장 가까운 인덱스를 읽는데 걸리는 평균시간.
1→0	1	인덱스를 읽기가 실패했을 경우 다음 버킷을 읽는다.
1→2	$\frac{1}{2}(\frac{D}{n^r}(n^r-1) + \frac{n^{k-r}-1}{n-1}(n^r-1) + \frac{n^{r+1}-1}{n-1})$	인덱스를 읽고 다음 단계의 인덱스를 읽는 평균시간.
2→0	1	
2→3	n/2	실패했을 경우 상태 0부터 다시 시작.
K-1→0	1	다음 단계의 인덱스를 읽는데 걸리는 평균시간
K-1→K	$n^{k-1}/2$	다음 단계의 인덱스를 읽는데 걸리는 평균시간
K→0	1	
K→D	$D/(2*n^k)$	마지막 단계의 인덱스를 읽고 데이터 버킷을 읽는데 걸리는 평균시간.

위의 상태전이 비용을 이용하여 액세스시간과 튜닝시간을 구하면 다음과 같다.

$$\begin{aligned}
 \text{access time} = & \frac{1}{(1-p)^k} - 1 + \frac{D}{2n^r} \\
 & + \frac{1}{2(1-p)^k n^2} \sum_{i=2}^{k-1} \{n^i \cdot (1-p)\}^i \\
 & + \frac{1}{2(1-p)^{k-1}} \left( D - \frac{Dp}{n^r} + n^r \right) \\
 & + \frac{1}{2(1-p)^{k-1}} \left( \frac{n^k + p - n^{k-r}p}{n-1} \right) \\
 \text{tuning time} = & (1+p) \sum_{i=0}^k \frac{1}{(1-p)^i} - p
 \end{aligned}$$

### IV. 파라미터 분석

다음은 파라미터 변화에 따른 성능을 측정하였다. 파라미터를 정리하면, D는 한 브로드캐스트 주기당 데이터 버킷의 크기(개수)를 나타내고, k는 인덱스 레벨을 나타내며, I는 인덱스 세그먼트의 크기(인덱스 버킷의 개수)를 나타내고, n은 한 인덱스버킷이 인덱싱할 수 있는 데이터 버킷의 크기를 나타내고, r은 분산 인덱스(부분 경로 중복) 기법에서 인덱스 중복 레벨을 나타낸다.

그림 6은 무선통신망의 장애율에 따른 액세스 시간의 변화를 나타낸다. "m=x"는 인덱스의 중복도가 x인 전체복사 인덱스 기법의 성능을 나타내고 "r=1"은 인덱스 중복 레벨이 1인 분산 인덱스 기법의 성능을 나타낸다. 무선통신망 장애율에 따른 액세스 시간의 변화를 보면, 장애율이 높아질수록 액세스 시간이 증가한다. 전체복사 인덱스 기법은 인덱스 중복도가 적정 수준에서 높으면 (이 실험에서는 3 또는 4) 장애에 따른 인덱스 재수신을 위한 대기 시간이 짧으므로 성능이 좋고 장애율에 덜 민감함을 알 수 있다. 또한 인덱스 중복의 부담이 적은 분산 인덱스 기법의 성능이 우수함을 알 수 있다.

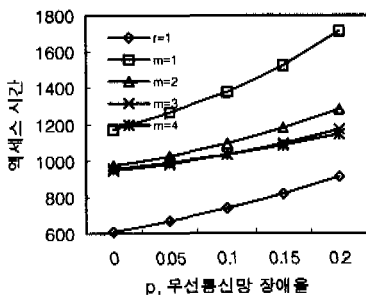


그림 6. 무선통신망의 장애율에 따른 액세스 시간의 변화 (D=1000, k=3, I=111, n=10)

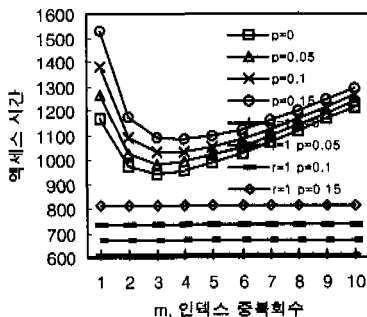


그림 7. 무선통신망의 장애율에 따른 액세스 시간의 변화 (D=1000, k=3, I=111, n=10)

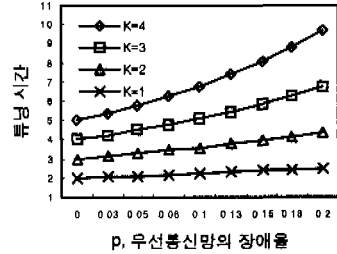


그림 8. 무선통신망의 장애율에 따른 튜닝 시간의 변화

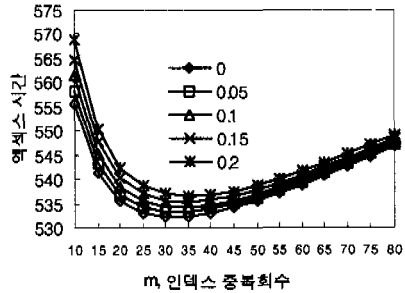


그림 9. 인덱스 중복회수에 따른 액세스시간의 변화. (D=1000, k=1, I=1, n=1000)

그림 7은 전체 복사 인덱스 기법에서 인덱스 중복회수에 따른 액세스 시간의 변화를 나타낸다. 이 그래프에서 인덱스 중복 레벨이 1인 분산 인덱스 기법("r=1이고 p가 0, 0.05, 0.1, 0.15인 경우")의 성능도 비교하였다. 액세스 시간은 데이터 크기, 인덱스 레벨, 무선통신망 장애율, 인덱스 중복회수에 따라 영향을 받음을 알 수 있다. 장애율에 따라 최적의 인덱스 중복회수가 존재함을 알 수 있다.

그러나, 분산 인덱스 기법은 인덱스 중복회수와는 해당 사항이 없고 중복레벨에만 영향을 받기 때문에 성능의 변화가 없다. 그림 6에서와 같이 인덱스 중복의 부담이 적은 분산 인덱스 기법의 성능이 우수함을 알 수 있다.

그림 8은 무선통신망 장애율에 따른 튜닝시간의 변화를 나타낸다. III장의 분석에서와 같이 전체 복사 인덱스 기법과 분산 인덱스 기법의 튜닝 시간은 동일하다. 튜닝 시간은 무선통신망의 장애율(p)과 인덱스 레벨(K)에만 영향을 받지 인덱스 중복회수에는 영향을 받지 않는다. 이 그래프에서 인덱스 레벨 (K)이 증가함에 따라 인덱스 액세스시 장애발생 가능성이 높기 때문에 장애율의 변화에 더욱 민감함을 알 수 있다.

그림 9는 시스템 파라미터를 변경하여 그림7과 같은 방법으로 성능을 비교하였다. 이 때 인덱스 레

벨이 1이므로 분산인덱스 기법의 의미는 모든 데이터 버킷 전에 인덱스 버킷을 위치시키는 것이므로 인덱스 중복도가 데이터 버킷의 크기(개수)와 동일 한, 즉  $m=D$  ( $m=1000$ )인, 전체복사 인덱스 기법과 동일한 성능을 보일 것이다. 이 때 인덱스의 중복이 과다하여 성능이 좋지 않다(그래프의 범위 밖으로 표시되어 있지 않음). 전체복사 인덱스 기법의 경우 최적의 인덱스 중복회수가 존재함을 알 수 있다.

### V. 결론

무선통신망의 장애를 고려하여 전체복사 인덱스 기법을 이용한 방송기법의 액세스 시간과 튜닝 시간을 분석하였고 액세스 시간을 최소화 시키기 위한 최적의 인덱스 중복도를 구하였다. 분석결과에 따르면, 장애도에 따라 인덱스 세그먼트의 중복회수를  $\sqrt{Data/(Index(1-p)^k)}$  만큼 중복해야함을 알 수 있다. 분산 인덱스 기법(중복레벨1)에 대하여 액세스 시간과 튜닝 시간을 분석하였고 전체복사 인덱스 기법과 성능을 비교하였다. 다중 레벨의 인덱스 버킷이 분산되어 있을 경우 어느 레벨까지 중복시켜야 최소의 액세스 시간을 구할 수 있을지 향후 연구로 남겨 놓았다.

#### 첨부: 용어설명

- 액세스 시간(Access time) : 클라이언트가 중추 키 값으로 원하는 데이터를 찾아 저장하는 데까지의 평균 걸린 시간을 말한다. 인지 대기시간(Probe Wait)과 방송 지연시간(Broadcast Wait)의 합으로 나타낼 수 있는데, 인지 대기시간은 지난번 데이터를 읽고 인덱스를 읽는데 걸린 평균 시간을 말하며 방송지연시간은 최초의 인덱스를 읽고 원하는 데이터를 읽는데 걸린 평균 시간을 말한다. 인지 대기시간과 방송 지연시간은 반비례한다. 방송지연시간을 최소로 할려면, 인덱스를 한번만 전송하면 된다. 이때는 필요한 데이터에 에러가 발생하면 항상 다음 인덱스를 기다려야 하므로 인지대기시간은 증가할 것이다. 인지대기시간을 최소로 할려면, 인덱스들 매 데이터 사이에 끼워넣으면 되나 브로드캐스트의 길이가 증가하므로 방송지연시간은 증가할 것이다.
- 튜닝 시간(Tuning time) : 채널을 결정하기 위해 걸린 시간, 즉 채널을 결정하기 위해 다운받은 인덱스 및 데이터를 읽는 시간을 의미한다. 이 시간동안

- 이동 클라이언트(mobile client)는 액티브 모드로 있다.
- 버킷(Bucket) : 브로드캐스트의 가장 작은 논리적인 단위이다. 버킷의 크기는 모두 같다. 액세스 시간과 튜닝 시간은 버킷의 수로 측정된다.
- 인덱스 버킷(Index bucket) : 필요한 데이터를 읽기위한 읍셋들의 버킷.
- 액세스 최적화(access optimal) : 액세스 시간이 최소로 될려면, 인덱스 버킷이 없을 때, 즉 브로드캐스트의 길이가 최소일 때이다. 하지만 이 경우는 튜닝 시간이 최대가 되어 항상 액티브 모드로 있어야만 한다. 이때 액세스 시간은  $Data / 2$ 로 튜닝 시간과 동일하다.
- 튜닝 최적화(tuning optimal) : 인덱스 버킷을 브로드캐스트의 시작부분에 한번만 넣는 것이다. 클라이언트는 최초로 인덱스를 읽고 중추 키로 원하는 데이터의 위치를 알아내어 그때까지 도즈 모드로 들어간다. 원하는 데이터의 위치가 되면, 액티브 모드로 전환되어 필요한 데이터를 읽어낸다. 튜닝 시간은 인덱스 단계의 수와 최초의 인덱스 위치를 알기위한 데이터를 읽는 것과 원하는 데이터를 읽는 것의 합으로 구할 수 있다.
- 인덱스 중복 레벨( r ) : 분산 인덱스 브로드캐스트에서 전체 인덱스 세그먼트를 중복하는 것은 인덱스 오버헤드(index overhead)가 생긴다. 따라서 인덱스 트리에서 위의 레벨만을 중복시키고 아래 레벨은 중복시키지 않는다. 이때, 중복시키는 인덱스 레벨을 의미한다.

#### 참고 문헌

- [1] T. Imielinski, S. Viswanathan, B. R. Badrinath "Energy efficient indexing on air," ACM SIGMOD 94-5/94 Minneapolis, Minnesota, USA, pp 25-36. 1994.
- [2] E. Pitoura, G. Samaras "Data management for mobile computing," Chapter 4. Information Management. Kluwer Academic Publishers, 1998.
- [3] David Cheriton, "Dissemination oriented communication systems," Stanford University, Tech. Rept. 1992.
- [4] G. Herrman et al., "The datacycle architecture for very large high throughput database systems," in Proc ACM SIGMOD Conf. pp 97-103, 1987.

[5] T. Imielinski, S. Viswanathan, B. R. Badrinath "Power efficient filtering of data on air," Proc of 4<sup>th</sup> International Conference on Extending Database Technology, Cambridge U.K, pp. 245-248. Mach 1994.

[6] T. Imielinski, S. Viswanathan, B. R. Badrinath "Data on Air : Organization and Access," IEEE Transactions on Knowledge and Data Engineering, Vol. 9. NO. 3, MAY/JUNE 1997, p.p. 353~372.

[7] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," International Conference On Mobile Computing and Networking, Seattle Washington, pp 151-162, 1999.

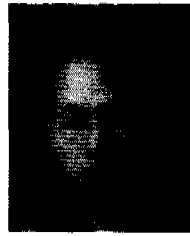
[8] Joe Chun-Hung Yuen, Edward Chan and Kam-yiu Lam, "Adaptive Data Broadcast Strategy for Transactions with Multiple Data Requests in Mobile Computing Environments," Proceedings of Sixth International Conference on Real-Time Computing Systems and Applications, Institute of Electrical and Electronics Engineers, 1998.

[9] Ping Xuan, Subhabrata Sen, Oscar Gonzalez, Jesus Fernandez and Krithi Ramamritham "Broadcast on Demand: Efficient and Timely Dissemination of Data in Mobile Environments," Proceedings of the 3<sup>rd</sup> IEEE Real-Time Technology and Applications Symposium 1997 (RTAS '97), p.p.38~48.

[10] Ming-Syan Chen, Philip S. Yu and Kun-Lung Wu "Indexed Sequential Data Broadcasting in Wireless Mobile Computing," Proceedings of the 17<sup>th</sup> International Conference on Distributed Computing Systems (ICDCS), pp. 124-131. 1997. IEEE.

정의종(Eui-Jong Jeong)

학생회원



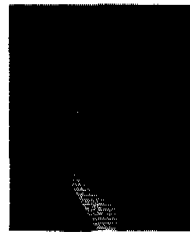
2000년 : 아주대학교 정보 및 컴퓨터 공학부 졸업 (학사)

2001년~현재 : 아주대학교 정보통신 전문대학원 석사과정

<주관심 분야> 이동컴퓨팅, 실시간 시스템, 결합허용 시스템,

김재훈(Jai-Hoon Kim)

정회원



1984년 : 서울대학교 제어계측 공학과 (학사)

1993년 : Indiana University, Computer Science (석사)

1997년 : Texas A&M University, Computer Science (공학박사)

1984년~1991년 : 대우통신(주) 컴퓨터연구실 대리

1995년~1997년 : Texas A&M University, Graduate Research Assistant

1997년~1998년 : 삼성전자(주) 컴퓨터시스템팀 수석연구원

1998년~현재 : 아주대학교 정보통신전문대학원 교수  
<주관심 분야> 분산시스템, 실시간시스템, 이동컴퓨팅