

인덱싱을 이용한 일관성 있는 데이터의 에너지 효율적인 브로드캐스트

권원석,^o 김성수

아주대학교 정보통신대학 정보 및 컴퓨터공학부

An Energy Efficient Broadcast of Consistent Data Using Indexing

Wonseok Kwon, Sungsoo Kim

Division of Information and Computer Engineering, Ajou University

요 약

휴대형 컴퓨터와 무선통신기술의 급속한 발달은 사용자에게 언제, 어디서나 필요한 컴퓨팅을 수행할 수 있는 이동 컴퓨팅이라는 새로운 패러다임을 발생케 하였다. 이동 컴퓨팅 환경은 낮은 대역폭(bandwidth)과 저 수준의 전송 품질을 가지고 있기 때문에 서버가 어떠한 방법으로 많은 수의 클라이언트들에게 데이터를 전송할 것인가에 대한 문제점이 발생한다. 이러한 문제를 해결하기 위한 방법으로 데이터 dissemination을 이용하는 새로운 시스템 구조들이 제안되었다. 본 논문에서는 데이터 dissemination의 한 기법인 브로드캐스트 기법에서의 데이터 일관성 유지 문제와 이동 클라이언트의 효율적 전력 사용 문제를 동시에 고려한 브로드캐스트 기법을 제안하였다

1. 서론

휴대형 컴퓨터와 무선통신기술의 급속한 발달은 사용자에게 언제, 어디서나 필요한 컴퓨팅 환경을 제공할 수 있게 하였고, 현재 다양한 사용자의 요구를 수용하기 위하여 이동 컴퓨팅에 관한 연구가 세계적으로 활발히 진행되고 있다

이동 컴퓨팅 환경은 낮은 대역폭(bandwidth)과 저 수준의 전송 품질을 가지고 있기 때문에 비대칭(asymmetric) 통신을 하는 어플리케이션을 지원하는데 있어 서버가 어떠한 방법으로 많은 수의 클라이언트들에게 데이터를 전송할 것인가에 대한 문제점이 발생한다. 이러한 문제를 해결하기 위한 방법으로 데이터 dissemination을 이용하는 새로운 시스템 구조들이 제안되었다[3, 4, 5]

데이터 dissemination을 하는 방법은 크게 on-demand 방법과 데이터 브로드캐스트(data broadcast) 방법으로 나눌 수 있다. On-demand 방법은 클라이언트가 요구하는 데이터들을 서버에서 dissemination하는 접근 방법으로 클라이언트의 수가 증가하게 되면 시스템의 확장에 있어 큰 문제를 가지게 된다 반면 데이터 브로드캐스트 접근 방법은 데이터를 전달하는 서버가 주기적으로 그리고 연속적으로 클라이언트에게 데이터를 브로드캐스트 한다. 브로드캐스트를 기반으로 하는 데이터 dissemination의 비용은 클라이언트의 수와 상관없이 결정된다. 그러므로 대역폭의 효율성이 고려되어야 할 많은 수의 클라이언트에게는 데이터 브로드캐스트를 기반으로 한 dissemination이 적당하다고 할 수 있다

데이터 브로드캐스트에서 중요하게 고려되어야 할 점은 서버가 브로드캐스트하는 데이터를 가지고 클라이언트가 트랜잭션을 수행하게 되면 서버는 항상 클라이언트에게 일관성 있는 데이터를 제공해야 한다는 것이다. 데이터 브로드캐스트 방법은 서버가 클라이언트들에게 데이터를 일방적으로 전달하는 푸쉬(push) 방식이기 때문에 서버에 의해 브로드캐스트 되는 데이터들이 서버에서 갱신(update)될 때마다 클라이언트는 이러한 사실을 알아내야 한다 즉, 현재 클라이언트가 가지고 있는 데이터와 현재 서버가 브로드캐스트 하는 새롭게 갱신된 데이터의 일관성이 유지되어야 한다

이동 컴퓨팅 환경에서 연구되어야 할 또 다른 문제는 클라이언트의 전력 소모 문제이다. 배터리의 용량 증가는 향후 10년 간 약 20%만의 용량이 증가될 것으로 예상되기 때문에 클라이언트가 효율적으로

전력을 사용하는 방법이 필요하다[3] 일례로 AT&T에서는 이동 환경을 고려하여 팜탑(palmtop) 컴퓨터용 Hobbit 칩을 개발하였으며, 이 칩의 전력 소모량은 active 모드에서는 250mW를 사용하고, doze 모드에서는 50μW의 전력을 사용하고 있다

이에 본 논문에서는 위에서 언급한 브로드캐스트 되는 데이터의 일관성 유지 문제와 브로드캐스트 환경에서 이동 클라이언트의 효율적 전력 사용 문제를 동시에 고려한 데이터의 브로드캐스트 기법을 제안하고 성능을 분석한다. 2장에서는 이에 관련된 연구들을 살펴보고, 3장에서 본 논문의 시스템 모델을 정의한다. 4장에서는 데이터 브로드캐스트 모델을 정의하고, 5장에서 본 논문에서 제안된 기법인 갱신 플래그를 가진 반복 인덱싱 기법을 살펴본다. 6장에서는 성능을 분석하고, 마지막으로 7장에서 본 논문의 결론과 향후 연구 방향에 대해 논의한다

2. 관련 연구

무선 네트워크의 저 수준의 서비스 품질은 데이터의 일관성 보장과 일관성 충돌 탐지를 어렵게 하고 있다 이 문제를 해결하는 방법 중 일관성 수준을 좀 더 완화하는 방법이 있을 수 있다. [1]에서는 이 단계의 일관성 모델을 제안하여 의미상 비슷한 데이터들을 하나의 클러스터(cluster) 그룹으로 만들어 같은 클러스터 안에 속한 데이터들은 엄격하게 일관성을 유지하고 서로 다른 클러스터 사이에는 어느 정도의 수준을 두어 데이터의 비일관성을 허락하였다

[2]에서는 일관성 있는 데이터를 전송하는 방법으로 재-브로드캐스트(re-broadcast) 기법을 이용하였다. 여기에서는 데이터의 갱신이 일어났을 경우 이 데이터가 이미 브로드캐스트 되었는지 아닌지를 판단하여 만약 클라이언트 쪽으로 이미 데이터가 브로드캐스트 되었다면 서버에서 이 데이터를 갱신이 일어난 직후 바로 다시 브로드캐스트함으로써 데이터의 일관성을 유지하였다. 그러나 이 기법은 클라이언트가 항상 브로드캐스트 채널을 감시하여 갱신된 데이터를 받아야 하기 때문에 클라이언트는 항상 active 모드를 취해야 되며, 따라서 전력의 소모가 커지게 된다

[3]은 클라이언트가 브로드캐스트 채널을 항상 감시함으로써 인해 발생하는 전력의 소모 문제를 효과적으로 해결할 수 있는 방법으로 인덱싱 기법을 이용하였다. 이 기법에서는 브로드캐스트 된 데이터를 인덱싱하여 데이터와 함께 인덱스를 브로드캐스트 함으로써 클라이언트는 인덱스만을 확인하여 자신이 원하는 데이터가 언제 브로드캐스트 될 지 알 수 있다. 따라서 해당되는 데이터가 브로드캐스트 될 때만 active 모드

본 연구는 1999년도 정보통신부 정보통신 우수시범학교 지원사업에 의한 결과임.

로 전환함으로써 에너지를 효율적으로 사용하였다. 그러나 이 기법은 브로드캐스트 되는 데이터의 일관성 보장을 고려하지 않았다.

3 시스템 모델

본 논문에서 고려하는 이동 컴퓨팅 시스템은 정보를 소유하고 있는 서버와 셀룰러 시스템과 같은 무선 네트워크를 통해 서버와 연결되는 다수의 클라이언트로 구성된 시스템이다.

서버는 브로드캐스트 될 데이터들을 관리하는 데이터베이스(database)를 가지고 있다 여기서 데이터들은 주식 시세나 뉴스와 같이 매우 동적으로 변화하는 아이템들이며, 최신의 정보로 갱신하는 데이터들의 갱신 트랜잭션(update transaction)은 데이터베이스 내에서 two-phase locking이나 타임 스탬프(timestamp)와 같은 잘 알려진 concurrency control 프로토콜에 의해 갱신 트랜잭션들 사이의 일관성 충돌없이 수행된다.

서버는 데이터베이스로부터 추출된 데이터들을 주기적으로 그리고 연속적으로 브로드캐스트 한다 이때 브로드캐스트의 주기는 고정될 수도 있고 가변적일 수도 있다

이동 클라이언트는 서버로부터 데이터를 가져오기 위하여 이동 트랜잭션(mobile transaction)을 수행한다 이동 트랜잭션은 읽기 전용(read-only) 트랜잭션으로서 클라이언트의 트랜잭션에 의해 변경된 데이터들은 서버에게 전달되지 않는다. 또한 갱신 트랜잭션과 이동 트랜잭션에 관련된 데이터의 일관성 충돌은 상대적으로 낮다고 가정한다.

4 데이터 브로드캐스트 모델

4.1 이동 클라이언트가 일관적이지 않은 데이터를 받는 경우

데이터의 브로드캐스트 프로세스는 갱신 트랜잭션과 동시에 수행될 수 있기 때문에 브로드캐스트 프로세스와 갱신 트랜잭션이 적절하게 제어되지 않으면 이동 트랜잭션은 일관적이지 않은 데이터 값을 받을 경우가 발생하게 된다

다음의 예를 통해 데이터 아이템이 브로드캐스트 될 때 갱신 트랜잭션과 브로드캐스트 프로세스가 적절히 제어되지 않을 경우 발생하는 문제점을 설명한다

예) 이동 트랜잭션은 데이터 아이템 D_1 과 D_2 를 읽기 원하고 갱신 트랜잭션은 D_2 와 D_3 를 갱신한다고 가정하자. 만약 전체 스케줄이 다음과 같이 진행된다면 클라이언트는 일관성이 없는 데이터를 받게된다

- ① 첫 번째 브로드캐스트 주기에서 이동 트랜잭션은 D_2 를 읽는다
- ② 갱신 트랜잭션에 의해 D_2 와 D_3 이 차례대로 갱신된다

③ 두 번째 브로드캐스트 주기에서 이동 트랜잭션은 D_1 을 읽는다. 다음과 같은 상황에서 이동 트랜잭션은 D_2 에 대한 갱신 트랜잭션이 일어나기 전의 값을 읽어들었기 때문에 충돌이 발생하여 이동 클라이언트의 작업에 영향을 주게 된다

4.2 데이터의 일관성을 보장하는 데이터 브로드캐스트 모델

본 논문에서 사용된 인덱스는 전혀 새로운 개념은 아니다. [3, 4]에서 인덱싱을 이용한 기법이 제안되었으나 이 기법은 일관적인 데이터 브로드캐스트를 보장하지 않는다. 이에 본 논문은 이 기법을 이용하여 클라이언트 측면에서 에너지 효율적이면서 데이터를 일관성 있게 전달받을 수 있는 브로드캐스트 기법을 제안하였다.

본 논문에서 제안하는 데이터의 브로드캐스트 기법인 갱신 플래그를 가진 반복 인덱싱(replicated indexing with update flag)에서 사용되는 데이터와 인덱스 구조는 다음과 같이 구성된다

데이터들은 각각의 고유 키(primary key)를 가지고 있고 버킷(bucket)으로 나누어져 구성되며, 시간에 따라 서버에 의해 갱신되고 크기도 변해 될 수 있다. 서버에서는 이러한 데이터들을 주기적으로 브로드캐스트하며 클라이언트는 데이터 버킷의 고유키를 가지고 원하는 데이터들을 브로드캐스트 채널에 접근함으로써 얻을 수 있게 된다

인덱스들 사용하면 클라이언트는 브로드캐스트 채널의 인덱스 정보만을 감시함으로써 원하는 데이터들이 언제 브로드캐스트 되는지 알 수 있기 때문에 그 데이터가 실제 브로드캐스트 될 때까지 doze 모드로 작업을 수행할 수 있게 되어 에너지를 효율적으로 사용할 수 있게 된다

클라이언트들이 일관성 있는 데이터를 얻는 방법은 데이터와 함께 브로드캐스트 되는 인덱스에 데이터의 갱신 여부를 표시하는 플래그(flag)를 사용함으로써 해결할 수 있다. 클라이언트는 브로드캐스트 채널에서 인덱스의 갱신 플래그를 확인함으로써 클라이언트가 필요한 데이터에 대한 일관성을 보장받을 수 있다.

데이터를 구성하고 있는 버킷들은 그림 1에서와 같이 그 버킷의 키와 데이터, 그리고 현재 시점에서 가장 가까운 인덱스의 위치를 표시하는 포인터(pointer)로 구성된다

인덱스는 그림 2와 같이 하나의 인덱스가 가리키는 데이터 버킷의

키와 그 버킷의 위치를 가리키는 포인터, 그리고 그 데이터의 갱신 여부를 표시하는 플래그(flag)로 구성된다 [3, 4]에서 제안된 인덱싱 기법은 데이터를 얻기 위하여 사용되는 전력을 최소화하는 쪽으로 초점을 맞추었으므로 인덱스를 트리 형태로 만들어 인덱스의 탐색시간을 줄였으나 본 논문에서는 인덱스를 통하여 데이터의 갱신 여부를 확인해야 함으로 모든 인덱스의 내용을 탐색하여야 한다. 따라서 인덱스의 탐색 시간을 줄이기 위해 인덱스를 트리 형태로 만들 필요가 없다



그림 1. 데이터 버킷의 구조



그림 2 인덱스의 구조

5 갱신 플래그를 가진 반복 인덱싱

5.1 정의와 접근 프로토콜

그림 3에서 볼 수 있듯이 갱신 플래그를 가진 반복 인덱싱 기법은 하나의 주기로 브로드캐스트 되는 데이터들의 인덱스를 그 주기 안에서 m 번 반복하여 삽입하는 방법으로 정의한다. 앞 장에서 언급했듯이 모든 데이터 버킷들은 가장 가까운 인덱스의 포인터를 가지고 있다.

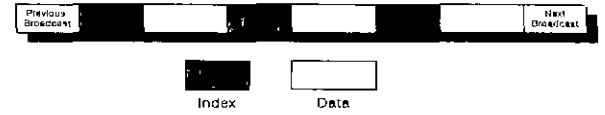


그림 3 갱신 플래그를 가진 반복 인덱싱

갱신 플래그를 가진 반복 인덱싱 기법에서 키 K 를 가지고 있는 데이터의 버킷으로의 접근과 갱신된 데이터를 확인하는 프로토콜은 다음과 같다

Step A
브로드캐스트 채널에서 현재의 버킷을 튜닝한다.

Step B
튜닝된 버킷에서 현 위치에서 가장 가까운 인덱스의 주소를 결정하기 위하여 포인터 값을 읽는다.

Step C
클라이언트는 doze 모드를 취하며 Step B에서 결정된 인덱스가 브로드캐스트 될 때 active 모드로 전환되어 인덱스 버킷들을 탐색한다.

Step D
모든 인덱스를 탐색하여 키가 K 인 데이터 버킷의 포인터를 확인한다. 이동 트랜잭션에 필요한 데이터 중 이미 데이터를 전달받았으나 갱신이 일어난 데이터에 대해 플래그를 통해 확인함으로써 갱신 데이터에 대한 포인터도 확보한다. 만약 인덱스에서 확인된 데이터 버킷이 이미 브로드캐스트 되었다면 클라이언트는 doze 모드에서 작업을 수행하며 다음 주기의 데이터 브로드캐스트가 실행되면 active 모드로 전환하여 첫 번째 인덱스를 튜닝하여 Step D를 반복한다.

Step E
클라이언트는 키 K 를 가지고 있는 데이터 버킷이 브로드캐스트 될 때 doze 모드에서 active 모드로 전환되어 해당하는 데이터 버킷을 다운로드 한다. 또한 클라이언트는 Step D에서 얻은 갱신 데이터에 대한 포인터를 가지고 갱신된 데이터가 브로드캐스트 될 때 active 모드로 전환되어 데이터를 다운로드 하여 데이터의 일관성을 유지한다

5.2 이동 트랜잭션이 일관성 있는 데이터를 보장받는 예

4.1절에서의 예를 들어 적용하면 다음과 같이 일관성 있는 데이터를 얻을 수 있게 된다

- ① 첫 번째 브로드캐스트 주기에서 프로토콜에 의해 D_2 를 읽는다

- ② 갱신 트랜잭션에 의해 D_1 와 D_2 가 차례대로 갱신된다.
- ③ 프로토콜에 의해 두 번째 브로드캐스트에서 클라이언트는 D_1 을 읽기 위해 인덱스를 탐색하게 된다. 인덱스를 탐색하게 되면 클라이언트는 D_1 가 갱신된 사실을 알게 됨으로 D_1 과 함께 D_2 를 다시 읽게 된다.

6. 에너지 사용 분석

6.1 분석 파라미터

본 논문에서 제안된 갱신 플레그를 가진 반복 인덱싱의 에너지를 분석하기 위해 다음의 파라미터를 고려한다[4]

● 접근 시간 (access time)

접근 시간은 클라이언트가 어떠한 고유키를 가지고 있는 데이터 버킷을 다운로드 받기 원하는 시간부터 그 데이터 버킷을 실제 다운로드한 시간까지 소요된 평균 시간으로 정의된다. 접근 시간의 단위는 데이터 버킷의 수로 결정된다

● 튜닝 시간 (tuning time)

튜닝 시간은 클라이언트가 브로드캐스트 채널을 감시하는데 소요되는 시간으로 정의된다 튜닝 시간 파라미터는 클라이언트가 데이터를 얻기 위해 사용한 전력의 양을 나타낸다.

클라이언트의 접근 시간을 결정하기 위하여 다음 두 가지의 파라미터를 고려한다

● 프로브 대기 (probe wait)

프로브 대기는 클라이언트가 원하는 데이터를 얻기 위하여 그 데이터에 대한 정보를 인덱스에서 얻기 위해 프로브가 가장 가까운 인덱스를 찾는데 걸리는 시간으로 정의한다.

● 브로드캐스트 대기 (broadcast wait)

브로드캐스트 대기는 인덱스로부터 클라이언트가 원하는 데이터의 정보를 확인한 후 그 데이터를 다운로드 받을 때 걸리는 시간으로 정의한다

접근 시간은 프로브 대기와 브로드캐스트 대기를 합한 것으로 정의된다

본 논문에서 제안된 갱신 플레그를 가진 반복 인덱싱 기법의 비교 대상으로 접근 시간을 최적화 하는 알고리즘을 설명한다.

● 접근 시간 최적화

접근 시간 최적화 알고리즘을 사용하면 최적의 접근 시간을 얻을 수 있다 최적의 접근 시간을 구하기 위해서는 인덱스를 사용하지 않고 데이터만 브로드캐스트 하면 되고, 클라이언트는 항상 active 모드에서 브로드캐스트 채널만을 감시하면 된다. 그러므로 접근 시간 최적화 알고리즘에서의 접근 시간은 식 (1)과 같이 결정할 수 있다. 접근 시간 최적화 기법은 데이터만을 브로드캐스트 하기 때문에 튜닝 시간을 고려할 필요가 없다.

$$AccessTime = DATA/2, DATA \cdot \text{데이터 크기} \quad (1)$$

6.2 갱신 플레그를 가진 반복 인덱싱의 접근 시간과 튜닝 시간

갱신 플레그를 가진 반복 인덱싱에서의 접근 시간은 클라이언트가 고유키 K 를 가지고 데이터를 다운로드 받을 과정만을 고려한다. 이 절에서 사용되는 기호들은 다음과 같다

$DATA$: 전체 데이터 버킷의 수, n : 하나의 버킷에서 인덱스가 차지하는 용량, $INDEX$: 인덱스가 차지하는 버킷의 수
 $INDEX$ 의 값은 전체 데이터 버킷의 수를 버킷의 용량으로 나눈 값이므로 다음과 같이 구할 수 있다

$$INDEX = DATA/n$$

접근 시간은 프로브 대기 시간과 브로드캐스트 대기시간의 합으로 표현됨으로 다음과 같이 구할 수 있다[4].

$$\text{접근 시간} = ((m+1) \times INDEX + ((1/m) + 1) \times DATA)/2 \quad (2)$$

여기서 m 은 인덱스의 반복 회수로서 최적화된 인덱스의 반복 회수 m 은 식 (2)를 m 에 관해 미분함으로써 구할 수 있다[4]

$$m = \sqrt{DATA/INDEX}$$

튜닝 시간은 첫 번째 프로브가 가장 가까운 인덱스 버킷을 얻기 위해 걸리는 시간과 인덱스를 탐색하는데 걸리는 시간, 그리고 마지막으로 인덱스를 통해 확인된 포인터로부터 하나의 데이터 버킷을 받는데 걸리는 시간의 합으로 구할 수 있다.

$$\text{튜닝 시간} = 2 + INDEX$$

6.3 에너지 사용량 비교

이 절에서는 갱신 플레그를 가진 반복 인덱싱의 에너지 사용량을 6.1 절에서 구한 접근 시간 최적화 기법의 에너지 사용량과 비교하였다 클라이언트가 소모하는 에너지의 양은 다음과 같이 구할 수 있다.

$$\text{에너지 사용량} = (\text{튜닝 시간} \times \text{Active 모드에서의 전력 사용량}) +$$

(접근 시간 \times Doze 모드에서의 전력 사용량)

에너지 사용량을 비교하기 위해 다음과 같은 브로드캐스팅 시스템을 가정하였다 브로드캐스팅 서버는 총 320,000 바이트의 정보를 주기적으로 Hobbit 칩을 사용하는 클라이언트에게 전달하며 브로드캐스트 채널은 10Kbps의 대역폭을 가지고 있고, 하나의 버킷의 길이는 128 바이트라고 가정하였다 하나의 버킷에서 인덱스가 차지하는 용량 n 은 10에서 125까지 5씩 증가하는 것으로 가정하고 버킷의 용량에 따라 에너지 사용량이 어떻게 되는지를 비교하였다

그림 4의 실선 부분은 클라이언트가 갱신플레그를 가진 반복 인덱싱을 이용할 때 n 을 10에서 125까지 변화시켰을 때의 에너지 사용량을 보여주고, 점선 부분은 접근 시간 최적화 기법을 사용했을 때 클라이언트가 사용하는 에너지의 양을 보여준다 여기서 접근 시간 최적화 기법은 인덱스를 사용하지 않기 때문에 n 의 값과는 무관하며 데이터 버킷의 수가 2500개이므로 에너지 사용량은 항상 3125 Joule로 일정하다. 그림에서 볼 수 있듯이 본 논문에서 제안된 갱신 플레그를 가진 반복 인덱싱 기법은 n 이 125일 때, 접근 최적화 기법보다 약 60배 정도의 에너지 절감 효과를 가져 올 수 있다 버킷의 용량 n 이 크기 때문에 상대적으로 인덱스가 차지하는 버킷의 수가 작아지므로 실제 에너지를 사용하는 시간인 튜닝 시간이 줄어들어 그림에서와 같이 에너지 사용량에 큰 차이를 보이고 있다

반면 n 이 10일 경우에는 본 논문에서 제안된 갱신 플레그를 가진 반복 인덱싱 기법은 접근 최적화 기법보다 약 6배 정도의 에너지 절감 효과를 가져 올 수 있다. 버킷의 용량 n 이 작기 때문에 인덱스 버킷의 수가 증가하게 되고, 따라서 튜닝 시간이 길어지기 때문에 n 이 125일 경우와 비교해 상대적으로 에너지 사용의 효율이 줄어드는 것을 볼 수 있다.

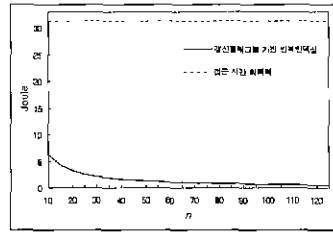


그림 4. 에너지 사용량

7. 결론

데이터 dissemination의 한 기법인 브로드캐스트는 클라이언트의 수와 상관없이 비용이 결정되기 때문에 상대적으로 대역폭이 낮은 이동 환경에서 많은 수의 클라이언트를 효율적으로 지원할 수 있다. 본 논문에서는 클라이언트 측면에서 에너지를 효율적으로 사용할 수 있고 일관적인 데이터를 얻을 수 있는 브로드캐스트 기법인 갱신 플레그를 가진 반복 인덱싱 기법을 제안하고 이 기법에서 사용된 데이터의 구조를 살펴보았다. 추가되어야 할 문제로는 인덱스의 데이터 수정 여부를 가리키는 플레그를 통해 확인된 수정 데이터를 얻는데 걸리는 시간이 접근 시간 계산에 함께 고려되어야 한다. 또한 제안된 기법을 시뮬레이션하여 본 논문에서 제안된 기법이 타당하다는 것을 수학적으로 분석한 결과와 비교 검증해야한다.

참고문헌

- [1] E Pitoura and B. Bhargava, "Maintaining Consistency of Data in Mobile Distributed Environment," In Proceeding of 15th International Conference on Distributed Computing Systems, pp. 404-413, May 1995
- [2] K. Lam, M. Au and E. Chan, "Broadcast of Consistent Data to Read-Only Transactions from Mobile Clients," In Proceeding of WMCSA99, Feb. 1999.
- [3] T. Imielinski, S. Vishwanath and B. Badrinath, "Energy Efficient Indexing on Air," In proceeding of the ACM SIGMOD Conference on Management of Data, pp 25-36, May 1994.
- [4] T. Imielinski, S. Vishwanath and B. Badrinath, "Power Efficient Filtering of Data on the Air," In Proceeding of the International Conference of EDBT, pp. 245-258, March 1994.
- [5] S. Acharya, et. al., "Broadcast Disks: Data Management for Asymmetric Communication Environments," In Proceeding of the ACM SIGMOD Conference on Management of Data, pp. 199-210, May 1995