

다중 무선 방송채널에서 상호 관련 데이터 할당 방법

(Interdependent Data Allocation a Scheme over Multiple Wireless Broadcast Channels)

박 성 욱 [†] 정 성 원 ^{**}
(Sungwook Park) (Sungwon Jung)

요 약 무선 환경에서의 방송은 클라이언트의 수에 상관없이 다수의 클라이언트들에게 데이터를 보낼 수 있기 때문에 많은 관심을 이끌어왔다. 다수의 이전 연구들에서는 한 개의 독립적인 데이터를 최단시간에 획득하는 것을 목표로 했고, 한 질의어 내에 포함된 상호 관련된 데이터들을 동시에 획득하는 것에 대해서는 연구되지 않았다. 또한, 이러한 논문들에서는 질의어 내의 데이터들이 동일시간에 다른 채널 상에서 나타나지 않도록 할당하는 문제에 대해서만 연구되었고 질의어 요청 확률을 기반으로 한 각 데이터의 접근확률은 반영되지 않았다. 본 논문에서는 데이터를 할당하고 동시에 한 질의어 속에 포함된 모든 독립된 데이터들을 획득하는데 요구되는 평균응답시간을 최소화하는 새로운 기법을 제시한다. 성능 분석에서는 우리가 제시한 기법이 현존하는 다른 기법들 보다 좋은 평균응답시간을 보여준다.

키워드 : 모바일 데이터베이스, 다중 무선 방송 채널, 브로드캐스트 디스크, 상호 관련 데이터 질의어

Abstract Broadcast in the wireless environment has drawn much attention because it is capable of sending data to clients regardless of the number of clients. Most previous researches have aimed at obtaining an independent data item in a minimum time. But, they have not been researched on simultaneously receiving dependent data items in a query. In addition, these papers have only researched allocation problem that have not been came out data items in a query in same time and different channels. The access probability of each data item based on query request probability have not been reflected. This paper proposes a new method of allocating data items and simultaneously minimizing average response time required in receiving all the dependent data items in a query. Our performance analysis shows that our proposed method gives better average response time over the existing methods.

Key words : Mobile Databases, Multiple Wireless Broadcast Channels, Broadcast Disk, Dependent Data Query

1. 서 론

최근 휴대폰이나 PDA와 같은 컴퓨터 하드웨어와 블

루투스과 같은 무선 네트워크 기술의 급격한 발달은 모바일 컴퓨팅을 점차 발전시키고 있다. 이러한 모바일 컴퓨팅 환경에서 모바일 사용자들은 언제 어디서나 다양한 종류의 서비스에 접근할 수 있게 되었다. 그러나 현존하는 기술들은 물리적 제약 조건으로 인하여 상향 대역폭(Upstream)과 하향 대역폭(Downstream)의 비대칭성과 모바일 기기의 배터리 부족 등에 문제가 있다.

Pull와 Hybrid 기반의 시스템에서는 모바일 사용자가 직접 서버에게 필요한 데이터를 요청하는데 모바일 사용자가 수가 증가함에 따라 동시에 데이터를 요청하는 빈도 또한 비례하여 증가한다. 결국 동시에 데이터를 요청하는 모바일 사용자가 많아질수록 모바일 사용자의 수에 비해 훨씬 적은 수의 서버는 과부하가 발생하고

[†] 학생회원 : 서강대학교 컴퓨터공학과 석사과정
psw0405@sogang.ac.kr

^{**} 종신회원 : 서강대학교 컴퓨터공학과 부교수
jungsung@sogang.ac.kr

논문접수 : 2008년 4월 10일

심사완료 : 2008년 11월 10일

Copyright©2009 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 데이터베이스 제36권 제1호(2009.2)

서비스 공급의 시간적 지연이 발생한다. 통신 대역폭에서는 하향 대역폭이 상향 대역폭에 비해서 훨씬 큰 비대칭성을 보인다. 동시 다발적인 많은 모바일 사용자의 데이터 요청은 상향 대역폭에서의 병목 현상을 발생시킬 수 있고 이에 반해 하향 대역폭의 사용률을 미비해진다. 더욱이 모바일 기기가 직접 서버에게 데이터를 요청하는 것은 모바일 기기의 큰 전력소모를 유발시킨다 [1-3]. 즉, 단일 서버로 모바일 사용자의 수에 제한 없이 높은 하향 대역폭을 최대한 활용하면서 데이터를 전달할 수 있는 Push 기반의 데이터 방송(data broadcast)이 보다 효율적이다. 따라서 본 논문에서는 데이터 전달에 있어서 Push 기반의 무선 데이터방송 환경을 기본으로 한다.

실제 서비스되고 있는 무선 데이터 방송 기술들로는 MSN의 Direct Service나 HughesNet을 예로 들 수 있다. 무선 방송 기술을 이용하는 MSN Direct Technology를 모바일 사용자의 내비게이션 장치에 탑재하여 다수의 무선 방송 채널을 통해 교통 정보, 주식 정보, 날씨 정보 등을 전송 받아 활용한다. 내비게이션 기능은 무선 채널을 통해서 전송받는 현 교통 상태와 사전에 기기에 저장된 완전한 지도 정보를 조합하여 최적의 경로를 선택할 수 있도록 해준다. 북미와 캐나다 지역에서는 위치 기반서비스를 제공해주기 위해 무선 방송 기법으로 MSN Direct 서비스가 활발하게 사용되고 있다[4]. HughesNet의 경우는 위성망을 활용하여 on-demand broadcast (Hybrid)를 제공하는 것으로 보다 빠른 속도로 인터넷을 활용할 수 있는 기술이다[5].

데이터 방송 기법은 데이터들의 주기에 따라 반복적으로 방송함으로써 대역폭의 낭비를 줄이고, 불특정 다수의 클라이언트들에게 데이터를 효율적으로 전달한다. 모바일 무선 환경에서 방송 대역폭은 한정되어 있으므로 클라이언트가 원하는 데이터들을 획득하는데 걸리는 시간을 줄이기 위해서는 무선방송채널에 데이터들을 효율적으로 할당하는 방법이 요구된다. 데이터 할당방법에 따른 성능을 측정하기 위해서 주로 사용되어지는 응답 시간(response time)은 클라이언트가 원하는 데이터 집합(질의어)을 요청한 순간부터 질의어에 속한 모든 데이터를 획득할 때까지 걸리는 총 시간을 의미한다. 또한 무선방송환경에서 또 다른 측정기준인 튜닝시간(tuning time)은 클라이언트가 원하는 데이터들을 모두 획득하기 위하여 실제로 방송채널을 듣고 있는 총 시간을 의미한다. 이러한 두 가지 측정기준인 응답시간과 튜닝시간을 최소화 할 수 있도록 데이터를 방송채널에 할당하는 많은 기법들이 연구되어져 왔다[1,2,6-19].

방송기법에서의 특정 중에 하나로 방송채널은 모든 클라이언트가 공유를 한다. 이는 클라이언트가 원하는 데

이타에 접근하기 위해서는 원하지 않는 데이터가 방송되어 지나가기를 기다려야하는 문제점을 발생시킬 수 있다. 이를 위한 해결 방법으로 데이터의 요청확률(access probability)에 따르거나 데이터들의 시멘틱(semantic)한 관계에 따라 서버에서 데이터들을 효율적으로 방송하는 기법에 대한 연구가 진행되었다[9-12].

다중 채널에서의 각 데이터의 요청확률에 따라 방송 프로그램을 스케줄링하는 방법들은 FLAT[1], Bin-Packing [2], VF^k[6], Greedy[7] 등으로 연구되었다. FLAT기법은 가장 단순하면서도 가장 수행속도가 빠른 방법으로 데이터를 요청확률(access probability)에 상관없이 모든 데이터를 채널에 균일한 수만큼 할당한다. 따라서 데이터의 요청확률분포가 한쪽으로 치우칠수록 평균응답시간이 커지게 된다. Bin-Packing 기법은 데이터 채널을 상자(Bin)로 간주하고 데이터의 요청확률을 기초로 상자(Bin)를 채워 나간다. 하지만 데이터의 요청확률을 어느 정도 반영을 될 뿐 최적의 분할지점을 선택하는 것이 아니기 때문에 응답시간의 뛰어난 향상을 이루지는 못한다. VF^k 알고리즘은 Greedy방법을 기초로 탑-다운(top-down) 방식으로 데이터를 노드로 하는 할당 트리를 생성한다. 트리의 레벨을 채널로 간주하여 트리에서 리프노드의 비용을 최소로 하는 할당 트리의 구성을 통하여 방송 프로그램을 스케줄링 한다. 여기서 언급한 리프 노드의 비용은 노드가 되는 데이터의 요청확률을 말한다. VF^k 알고리즘은 최적 알고리즘인 동적 프로그래밍(Dynamic Programming) 알고리즘 성능에 거의 가까우나 수행시간이 지나치게 오래 걸린다는 문제점이 있다. Greedy 기법은 수행시간이 빠르면서도 데이터의 요청확률(access probability)을 잘 반영한 기법으로 최적의 결과를 도출하는 동적 프로그래밍(Dynamic Programming)의 수행시간과 공간 복잡도(space complexity)에 대한 단점을 보완하였고 최적에 근접한 결과를 만들어 내는 방법이다. Greedy 알고리즘은 최초로 데이터들을 요청확률에 따라 내림차순으로 정렬하고, 다중 채널 평균응답시간이 최소가 되는 최적의 분할 지점을 선정하는 과정을 주어진 채널의 개수 만큼 반복 수행한다. 이 방식은 DP 알고리즘과 VF^k 알고리즘에 비하여 빠른 수행 시간을 가질 뿐만 아니라 데이터의 요청확률을 잘 반영하여 좋은 응답시간을 가질 수 있었다.

각 데이터의 요청확률(access probability)에 따라 데이터를 방송채널에 할당하는 방법들 이외에도 데이터들의 시멘틱(semantic)한 관계에 따라 방송채널에 데이터들을 할당하는 문제를 다루는 연구들도 또한 많이 있었다. 실제 무선방송환경에서는 개개의 데이터들이 독립적인 것이 아니라 서로간의 관련성을 가진 경우가 많다. 따라서 클라이언트들은 서로 시멘틱하게 관련된 여러

데이터들을 동시에 요청하는 경우가 더 많다. 예를 들어, 주식정보를 원하는 사용자가 모토로라와 같은 핸드폰 회사의 주식을 원한다면, 노키아나 삼성과 같은 관련된 회사의 주식정보 또한 요청할 확률이 높다. 우리는 이렇게 시멘틱하게 관련된 데이터들을 상호관련 데이터(dependent data)라고 부를 것이고 모바일 클라이언트가 요구하는 상호관련 데이터들의 집합을 질의어라고 할 것이다. 서버는 일정한 기간 동안에 클라이언트가 요청했던 질의어의 요구패턴에 따라 통계를 산출하여 그림 1과 같이 질의어 프로파일(query profile)을 생성한다. 질의어 프로파일 내에 각 질의어들은 전체 질의어 요구패턴에 따른 요청확률과 상호관련 데이터로 구성된다. 만약, 클라이언트들의 질의어 요구패턴이 동적으로 변화는 경우에는 일정한 주기로 클라이언트들의 새로운 질의어 요구패턴에 따라 질의어 프로파일을 수정한다. 그리고 수정된 질의어 프로파일을 기반으로 새롭게 데이터를 방송채널에 할당하여 방송한다.

단일 채널에서의 이러한 질의어의 평균응답시간을 최소화시키기 위한 할당 방법은 QEM[9,10]에서 제시되었다. Y.D.Chung이 제시한 QEM(Query Expansion Method)은 상대적으로 요청될 확률이 높은 질의어에 속한 데이터들에게 먼저 채널에 할당될 수 있는 기회를 주었다[9]. G.Lee는 [9]에서 제시한 방법의 엄격한 제한을 완화함으로써 개선된 방안을 제시하였다[10]. 그러나 [9,10]이 단일 채널환경만을 고려한 반면 M. S. Chen과 Hung은 다중 채널에서 적용될 수 있는 새로운 방법인 GA와 PBA(Placement-Based Allocation)를 각각 제시하였다[11,12]. GA[11]에서는 기존의 최적화 문제에 많이 쓰였던 genetic algorithm 기법을 다중 채널에 상호관련 데이터들을 할당하는 문제에 적용시켰다. 그러나 GA는 최종 방송 프로그램을 구성하기 까지 실행시간이 매우 크다는 단점이 있다. PBA[12]는 질의어에 포함된 데이터들이 동시에 같은 시간대에 방송됨으로써 질의어 평균응답시간이 커지는 문제점을 줄이기 위하여 데이터를 다중채널에 할당하는 순서에 관하여 다루었다. PBA에서는 더 높은 요청확률을 가지는 질의어에 속한 데이터들을 먼저 데이터 충돌(data conflict)이 발생하지 않도록 우선 할당하여 해당 질의어에서의 데이터충돌 가능성을 줄임으로서 질의어의 평균응답시간을 감소시킬 수 있었다.

그러나 이러한 방식들은 질의어의 요청확률만을 반영한 것이지 각 질의어 내에 포함된 데이터들의 요청확률을 반영한 것이 아니다. 또한 모든 데이터가 한 방송 사이클 내에서 오직 한번만 할당되기 때문에 그림 1과 같은 문제점이 생길 수 있다. 그림 1과 같은 질의어 프로파일(query profile) 내에서 데이터 B는 Q1에 속한 I,

Q1	0.4	I, L, M
Q2	0.3	B, R
Q3	0.2	B, L
Q4	0.1	B, C, F

그림 1 질의어 프로파일 내에서 데이터의 중요도

L, M 데이터 보다 더 자주 요청되는 데이터이다. 그러나 Q1의 요청확률이 가장 높기 때문에 할당 시에 I, L, M 데이터보다 우선순위에서도 밀리고 데이터의 높은 요청확률 또한 반영되지 못했다.

우리는 본 논문에서 이러한 문제점을 해결하기 위해 기존 방법에서처럼 할당 시에 질의어 할당 우선순위를 유지하여 데이터충돌(conflict data items)의 영향[12]을 줄이면서, 질의어 프로파일 내에서 상대적으로 많이 요청되는 데이터는 한 방송 사이클 동안 다중 채널 상에 여러 번 할당하여 해당 데이터를 획득하는데 요구되는 응답시간을 줄일 것이다. 그러나 이 논문에서 제안하는 방법은 각각의 데이터들이 다양한 반복횟수로 다중 채널 상에 나타나기 때문에 새로운 데이터 탐색기법이 없다면 매우 큰 튜닝시간(tuning time)이 요구될 것이다. 따라서, 우리가 제시하는 할당 방법에 따라 데이터를 채널에 할당한 이후에 원하는 데이터를 획득하는데 있어 튜닝시간을 최소화 하면서 작은 응답시간 안에 획득할 수 있는 적절한 인덱싱 기법 또한 같이 제시할 것이다.

이 논문의 남은 구성은 다음과 같다. 2장에서는 기존에 연구된 다중 채널에서의 질의어 할당 방법들과 문제점에 대해서 살펴본다. 3장에서는 상호관련 데이터를 다중 무선방송 채널 상에 효율적으로 할당하는 새로운 방법에 대하여 살펴보고 4장에서는 다중 채널에 할당된 데이터들을 효과적으로 찾을 수 있는 인덱싱 기법에 대하여 설명하겠다. 5장에서는 제시하는 데이터 할당과 인덱싱 방법을 검증하기 위한 성능을 평가하며 마지막으로 6장에서는 본 논문의 결론과 향후 연구 방향을 서술할 것이다.

2. 관련 연구

우리는 이전 장에서 시멘틱한 관계를 가지고 있는 데이터들을 작은 응답시간(response time)안에 획득할 수 있도록 데이터들을 방송 채널에 할당하는 여러 가지 연구들에 대하여 알아보았다. 본 2장에서는 다중 채널에서의 이러한 문제를 해결하기 위하여 제시된 두 가지 방법인 GA[11]와 PBA[12]에 대하여 좀 더 구체적으로 알아보겠다.

2.1 PBA

PBA에서는 클라이언트가 한 순간에 하나의 채널에만

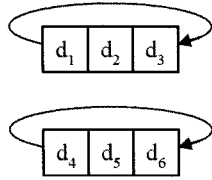


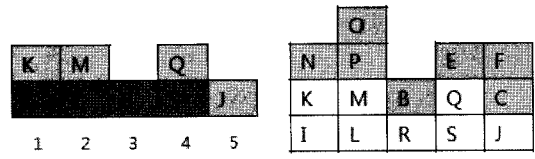
그림 2 데이터 충돌의 예제

질의어	확률	데이터
Q1	0.310	I, L, R, S
Q2	0.178	J, K, M, Q, R
Q3	0.128	C, N, P, Q, R
Q4	0.102	B, E, F, K, N, O
Q5	0.085	B, M, Q, T
Q6	0.074	A, K, Q, T
Q7	0.065	C, D, O, S
Q8	0.058	G, H, N, R

그림 3 질의어 프로파일 예제

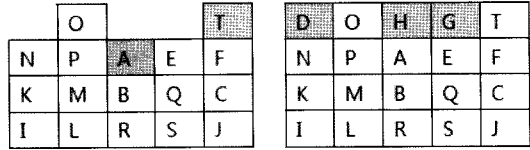
PBA에서는 질의어 응답시간을 줄이기 위하여 더 높은 요청확률을 가지는 질의어에 속한 데이터들을 먼저 데이터 충돌의 발생을 피하도록 할당하여 질의어 프로파일 전체로 볼 때 데이터 충돌의 발생 가능성을 줄일 수 있었다. 즉, 요청확률이 높은 질의어에서는 데이터 충돌의 발생 확률이 작지만 요청확률이 작은 질의어일수록 데이터 충돌의 발생확률이 커지게 된다. 그림 3과 같은 질의어 프로파일이 있다고 하자. 질의어 프로파일에서 여덟 개의 질의어가 요청확률에 따라 정렬되어져 있고, 전체 데이터는 20개로 4개의 데이터 채널에 할당되어질 것이다.

먼저 질의어 요청확률이 가장 높은 Q1에 속한 데이터들부터 데이터 채널에 할당한다. 그림 4의 (a)와 같이 타임 오프셋 처음에는 빈 데이터 채널에 Q1의 데이터 I, L, R 그리고 S가 순차적으로 1부터 4까지의 오프셋에 할당되어진다. 그리고 Q2에 속한 데이터들이 Q1의 할당 이후 할당되어진다. 이때 Q2에 속한 데이터들은 이미 할당되어진 데이터 집합 {R}과 아직 할당되어지지 않은 데이터 집합 {J, K, M, Q} 으로 나누어진다. 여기서 이미 할당되어진 집합에 속한 데이터들은 할당이 생략되어진다. 그리고 아직 할당되어지지 않은 데이터 집합의 J가 첫 번째 채널의 5번째 오프셋에 할당되어진다. 그리고 데이터 K, M 그리고 Q가 오프셋 1, 2 그리고 4에 차례로 할당되어진다. 이때 오프셋 3은 데이터 R과 Q의 데이터 충돌의 발생을 피하기 위하여 건너뛰어 할당되어진다. 그림 4의 (b),(c),(d)는 질의어 Q3~Q8이 할당된 모습을 보여준다. 여기서 마지막 질의어 Q8의 데이터 H가 할당될 때에는 데이터 채널 상에서 남아있



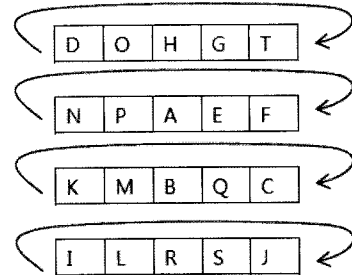
(a) Q1과 Q2의 할당

(b) Q3과 Q4의 할당



(a) Q5와 Q6의 할당

(a) Q7과 Q8의 할당



(a) PBA에 의하여 만들어진 방송 프로그램

그림 4 PBA의 실행 예제

는 빈 오프셋에 할당되어야하기 때문에 데이터 H와 데이터 R의 데이터 충돌은 피할 수 없다. 즉, PBA에서는 요청확률이 높은 질의어에서의 데이터 충돌에 비하여 요청확률이 작은 질의어에서의 데이터 충돌의 발생확률이 높다. 질의어 프로파일의 모든 데이터를 할당하면 그림 4(e)와 같이 방송될 데이터 채널이 모두 구성되어진다. 그러나 PBA에서는 질의어의 요청확률을 반영하여 데이터 충돌을 줄이기는 했지만 각 데이터의 요청확률을 반영하지는 못했다. 즉, 높은 요청확률을 가지고 있는 질의어에 포함되어 있진 않지만 많은 요청을 받을 데이터들에 대한 고려도 추가해야 할 것이다.

2.2 GA

GA[9]는 다중 방송 채널에 상호관련 데이터들을 최적적으로 할당하는 문제에 해결법을 제시하기 위해 기존에 많이 사용되었던 genetic algorithm을 이용한다. GA에서는 chromosome이라는 임식 해결책을 가지고 반복적인 fitness function의 평가를 통하여 최종적인 해를 찾는다. GA의 알고리즘의 최종 해의 성능은 fitness function의 구성에 따라 결정되어진다. 초기에 여러 chromosome이 임의적으로 또는 휴리스틱하게 생성되어진다. 예를 들어, 그림 5(a)의 질의어 프로파일에서 chromosome인 방송 프로그램은 임의적으로 그림 5(b)

Q1 = {D1, D2, D3, D4}	0.4
Q2 = {D1, D3, D4}	0.2
Q3 = {D4, D6, D1}	0.2
Q4 = {D1, D2, D5}	0.2

(a) 질의어 프로파일

D1	D4	D2
D6	D3	D5

(b) 임의의 방송 프로그램

그림 5. GA에 의하여 만들어진 방송 프로그램

와 같이 만들어질 수 있다.

모든 chromosome은 fitness function의 평가에 따라 fitness 값을 가지게 되고 가장 큰 fitness값을 가진 chromosome이 최고의 chromosome으로 결정된다. 이 과정을 반복함으로써 더 좋은 fitness 값을 가진 chromosome이 선택되어지고 방송 프로그램으로 변환된다. 그림 5의 질의어 프로파일을 예로서 GA의 알고리즘의 구체적인 방법을 살펴보겠다. 그림 5와 같이 네 개의 질의어로 구성된 질의어 프로파일의 평균응답시간은 식(1)과 같이 각 질의어의 응답시간과 요청확률의 곱을 모두 더한 값이 된다. 여기서 $T_{Access}(Q_i)$ 는 i 번째 질의어 Q_i 의 응답시간을 의미하고 $Pr(Q_i)$ 는 질의어 Q_i 의 요청확률이다. 그리고 $|Q|$ 는 질의어 프로파일에서 질의어의 개수이다.

$$T_{Access}(Q) = \sum_{i=1}^{|Q|} [T_{Access}(Q_i) * Pr(Q_i)] \quad (1)$$

질의어 프로파일의 평균응답시간을 구하기 위해서 각 질의어의 응답시간인 $T_{Access}(Q_i)$ 를 구해야 한다. 하지만 한 질의어에 포함된 여러 데이터들이 같은 시간대에 동시에 방송되어 데이터 충돌이 발생할 수 있으므로 직접적으로 계산해낼 수가 없어 질의어의 개조 함수를 통하여 Q_i' 와 cycleNO를 구해야 한다. 데이터 방송 채널의 각 시간 오프셋을 해쉬 버킷으로 구성하여 Q_i 의 데이터들을 저장한다. 이 때 가장 많은 데이터들을 포함하고 있는 해쉬 버킷에 포함된 임의의 데이터를 Q_i' 라고 하고 그 해쉬 버킷에 포함되어 있는 데이터의 수를 max라고 한다. cycleNo는 max-1로 구할 수 있다. Q_i' 와 cycleNo는 해당 질의어에 대해서 응답시간이 같은 시간대에 동시에 방송되는 데이터들 중에서 임의의 데이터 Q_i' 를 획득하는데 걸리는 시간과 나머지 데이터들을 획득하기 위하여 cycleNo 만큼의 다음 방송 사이클을 기다리는데 걸리는 시간의 합으로 구성된다는 것을 의미한다. 즉, 질의어 Q_i 의 응답시간 $T_{Access}(Q_i)$ 는 다음의 식

(2)와 같다. 여기서 $cycle_length$ 는 방송 사이클의 크기를 나타낸다.

$$T_{Access}(Q_i) = T_{Access}(Q_i') + cycleNo * cycleLength \quad (2)$$

GA에서 각 질의어에 대하여 구해진 $T_{Access}(Q_i)$ 를 식 (1)에 대입하여 질의어 프로파일의 평균응답시간 $T_{Access}(Q)$ 를 구하게 된다. 이렇게 구하여진 질의어 프로파일의 평균응답시간의 fitness는 $fitness(P) = 1 / T_{Access}(Q)$ 에 의하여 계산되어진다. 최종적으로 각 질의어의 fitness값들을 통하여 GA는 fitness가 가장 큰 chromosome으로 최선의 방송 프로그램을 구성할 수 있게 된다. 그러나 GA는 최종 방송프로그램을 구성하기까지 많은 실행 시간을 요구하고, 생성된 방송 프로그램이 항상 최적의 평균응답시간을 가진다는 것을 보장할 수 없는 단점이 있다. 또한, 초기에 임의로 구한 chromosome에 의해서 성능 상에 많은 영향을 받게 된다는 것도 문제점이다.

3. MA2D 데이터 할당 알고리즘

상호관련 데이터들을 무선다중채널에 할당하는 기존의 방법인 PBA와 GA와 같이 모든 데이터들이 다중채널 상에 오직 한 번씩만 할당되었다. 이는 데이터 충돌 시 다음 방송 사이클의 넘김이 증가하고 확률이 높은 질의어에 우선순위를 줄 수 없을 가능성이 높다. 그러나 데이터들 중에는 여러 질의어에 포함되어 많이 요청되는 데이터도 있을 것이고, 상대적으로 적게 요청되는 데이터도 있을 것이다. MA2D는 클라이언트들에게 상대적으로 자주 요청되는 데이터들을 하나의 방송 사이클 동안에 여러 번 할당하여, 데이터 충돌 시 다음 사이클로 넘어가지 않고 현 사이클 내에서 충돌된 데이터를 빠르게 획득할 수 있다. 이번 장에서는 각 데이터의 요청빈도에 따른 적절한 데이터의 할당횟수를 결정하고 다중채널 상에 효율적으로 할당하는 방법에 대하여 설명하겠다. 앞으로 본 논문에서 제안하는 방법을 MA2D라고 부를 것이다.

3.1 MA2D의 데이터 할당횟수 결정 방법

서버는 클라이언트들이 요청하는 질의어의 집합인 질의어 프로파일을 만들고 관리한다. 질의어 프로파일에 속해 있는 질의어들은 한 개 또는 다수의 데이터들로 구성되어져 있다. 그리고 각각의 질의어들은 그들의 요청빈도에 따라 확률로 나타내어진다. 즉, 모든 질의어의 요청확률의 총 합은 1이다. 서버는 이러한 질의어 프로파일을 사용하여 클라이언트로부터 요청되는 데이터의 요청빈도에 따라 각 데이터 마다 다중무선채널 상에 할당되는 횟수를 결정할 것이다.

그림 6은 클라이언트들로부터 받은 질의어 프로파일의 예이다. 질의어 Q1의 데이터 I, L, R, S는 각각 Q1

질의어	확률	데이터
Q1	0.310	I, L, R, S
Q2	0.178	J, K, M, Q, R
Q3	0.128	C, N, P, Q, R
Q4	0.102	B, E, F, K, N, O
Q5	0.085	B, M, Q, T
Q6	0.074	A, K, Q, T
Q7	0.065	C, D, O, S
Q8	0.058	G, H, N, R

그림 6 질의어 프로파일 예제

Data request frequency	R	Q	S	K	I	L	N	M	C	B
Data request frequency	674	465	375	354	310	310	288	263	193	187

Data request frequency	J	O	T	P	E	F	A	D	G	H
Data request frequency	178	167	159	128	102	102	74	65	58	58

그림 7 질의어 요청에 따른 각 데이터에 대한 상대적인 요청빈도수

Data request frequency	R	Q	S	K	I	L	N	M	C	B
Data request frequency	9	6	5	5	4	4	4	4	3	3

Data request frequency	J	O	T	P	E	F	A	D	G	H
Data request frequency	2	2	2	2	1	1	1	1	1	1

그림 8 각 데이터의 할당횟수

에 의해 0.310의 확률만큼 요청된다고 볼 수 있다. 질의어 Q1의 0.310이라는 확률의 의미는 질의어가 1000번 요구 되어질 때 Q1이 310번 요청된다는 것을 의미한다. 이와 같은 방법으로 Q2과 Q3은 각각 약 178번과 128번이 요청되고 각 데이터는 자신을 포함하고 있는 질의어의 요청횟수를 모두 더한 값만큼 요청될 것이다. 즉, 데이터 R은 Q1, Q2, Q3 그리고 Q8에 포함되어 있기 때문에 이들의 요청횟수의 합인 310+178+128+58=674번이 요청될 것이다. 이렇게 만들어진 요청횟수를 가지고 우리는 각 데이터에 대한 적절한 할당횟수를 결정한다.

그림 7을 보면 데이터 R이 674번 요청될 때 데이터 H는 58번 요청되는 것을 알 수 있다. 따라서 데이터 H의 요청횟수인 58로 데이터 R의 요청횟수 674를 나누면 상대적인 할당횟수를 구할 수 있다. 그러나 이러한 방식은 가장 적은 요청횟수가 가장 큰 요청횟수에 비하여 상대적으로 매우 작다면 전체 할당횟수의 크기가 커져 전체 방송 사이클이 너무 길어진다는 단점이 있다. 이 문제를 해결하기 위해 각 데이터의 요청횟수를 정규화하여 타당한 할당횟수를 구하는 방법이 필요하다. 그림 8의 예제에서는 A, D, G 그리고 H 데이터는 비교적 비슷한 요청횟수를 갖는다. 따라서 데이터 A의 요청횟수 74보다 작은 요청횟수에는 할당횟수를 한번으로 하여 모든 데이터의 상대적인 할당횟수를 구하였다. 이러한 방식을 통하여 데이터 R은 그림 8과 같이 할당횟수가 9가 될 것이고, 데이터 Q는 6회 할당횟수를 가질 것이다.

3.2 MA2D의 데이터 할당 방법

3.1절에서 질의어 프로파일에 속한 질의어의 평균응답 시간을 최소화하기 위해 각 데이터의 최선에 할당횟수를 정하였다. 이번 절에서는 각각의 데이터의 할당횟수를 이용하여 MA2D가 다중 채널 상에 어떻게 데이터를 할당할지를 설명하겠다. 먼저 모든 데이터의 할당횟수의 총합을 구한다. 이때, 가장 작은 할당횟수는 1이다. 즉, 모든 데이터가 한 방송 사이클 동안 적어도 한번은 할당되어진다.

데이터를 채널에 할당하는 순서는 PBA에서 제시하는 방법과 같이 질의어 프로파일 상에서 질의어 요청이 높은 첫 번째 질의어의 데이터들부터 할당된다. 더 높은 요청확률을 가지는 질의어에 속한 데이터들을 데이터 충돌이 발생하지 않도록 먼저 할당함으로써 전체 질의어 프로파일에 대해서 데이터 충돌의 발생 가능성을 줄일 수 있다. 데이터 충돌의 발생 가능성의 감소는 평균 응답시간의 감소를 의미한다. 그리고 요청확률이 높은 질의어에서는 데이터 충돌이 일어날 확률이 작지만 요청확률이 작은 질의어일수록 데이터 충돌의 가능성이 높아진다. 다음으로 할당 방법을 설명하기에 앞서 우선 다음과 같은 정의가 요구된다.

정의 1. T를 모든 데이터들의 할당횟수의 총 합이라고 하고 K를 데이터 방송 채널의 수라고 하자. 이때, Cycle_length는 $\lceil T/K \rceil$ 이다. 즉, Cycle_length는 각 데이터의 할당횟수의 총합을 채널수로 나눈 값으로 방송 사이클의 크기이다.

정의 2. Diff(X)는 방송될 데이터 집합에서 데이터 X가 다중 채널 상에서 주기적으로 나타나는 일정한 간격이다. Diff(X)는 $\lceil \text{Cycle_length} / tx \rceil$ 이다. tx 는 데이터 X의 할당횟수이다.

데이터 X를 지정된 위치에 처음 할당하고 직전에 할당된 위치로부터 Diff(X)의 크기 만큼 떨어진 위치의 다음채널에 다시 할당하는 것을 할당횟수 tx 만큼 반복한다. 즉, Diff(X) 만큼 떨어진 위치마다 채널을 순환하면서 데이터를 할당한다. 예를 들어, X라는 데이터의 할당횟수가 3이고, Diff(X)가 4 라고 하자. 데이터 X가 처음 할당된 위치가 1번 데이터 채널의 2번 오프셋이라면 데이터 X는 2번 데이터 채널의 6번 오프셋과 3번 데이터 채널의 10번 오프셋에 할당되어진다. 다수의 할당횟수를 가진 데이터는 일정한 간격을 유지하면서 채널들에 할당됨으로써 획득하는데 요구되는 시간을 최소화시킬 수 있다. 이때, 다음과 같은 세 가지의 특수한 경우가 발생할 수 있다.

1. 다수의 할당횟수를 가진 데이터의 다음 할당 오프셋이 현재 방송 사이클을 넘는 경우
2. 동일한 데이터가 다른 방송채널의 같은 오프셋에 할당되어

진 경우
3. 데이터가 할당되어야 할 위치에 이미 다른 데이터가 할당되어 있는 경우

1번째 경우와 같이 여러 할당횟수를 가진 데이터가 할당되어야 할 오프셋이 현재 방송 사이클의 범위를 넘는다면 다음 방송 사이클에서 할당되어야 한다.

그림 9와 같이 데이터 X의 할당횟수가 4이고 데이터 방송 채널이 4개라고 하면 Diff(X)는 4가 된다. 데이터 X의 초기 할당 위치가 1번 채널의 오프셋 9라면 다음 할당은 2번 채널의 오프셋 13에서 이루어진다. 그러나 13번 오프셋으로부터 Diff(X) 만큼 떨어진 오프셋은 현재 방송 사이클이 아닌 다음 방송 사이클이 될 것이다. 우리는 다음 방송 사이클이 아닌 현 방송 사이클 안에서 데이터 X가 방송될 오프셋을 결정하기 위하여 식 (3)과 같은 모듈로 연산을 사용한다.

$$\begin{aligned} \text{next offset} \\ = (\text{current offset} + \text{Diff}(X)) \% \text{Cycle_length} \end{aligned} \quad (3)$$

식 (3)을 통하여 데이터 X가 3번째로 할당될 오프셋은 $(13+4)\%16=1$ 로서 3번 채널의 1번 오프셋에 할당되어진다. 그리고 마지막 3번째 데이터 X는 4번 채널의 5번 오프셋에 할당되어지면서 할당을 완료하게 된다.

할당 시에 발생할 수 있는 2번째 경우는 한 데이터가 반복 할당되면서 다른 채널상의 동일한 오프셋에 할당되어질 수 있다는 것이다. 동일한 데이터가 같은 시간 오프셋에 방송된다면 데이터를 반복 방송하는 의미가 감소하게 된다.

그림 10에서와 같이 데이터 X의 할당횟수가 9이고 Diff(X)가 2라고 하자. 데이터 X의 초기할당이 1번 채널의 3번 오프셋에서 이루어 졌다면 데이터 X의 9번째 할당 위치는 2번 채널의 3번 오프셋이 될 것이다. 그러나 1번 채널의 3번 오프셋에 이미 데이터 X가 할당되어져 있으므로 1 이전 오프셋인 2번 채널의 2번 오프셋에 할당함으로써 동일 데이터가 같은 오프셋에 할당되는 문제를 피할 수 있다.

데이터를 방송 채널에 할당할 때 직면할 수 있는 3번

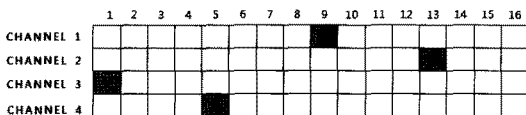


그림 9 할당과정에서의 발생 가능한 문제(1)

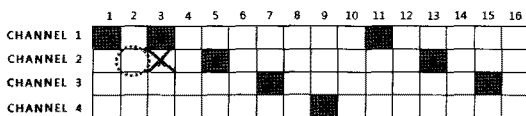


그림 10 할당과정에서의 발생 가능한 문제(2)

```

Input : query profile Q, the number of channels K
Output : multi broadcast cycle with allocated data items

Begin
diff[X] : data X is allocated at interval of diff[X].
current_offset : allocated position for current data item.
allocation frequency[X] : frequency of data item X allocated in channels during
a broadcast cycle

for (every data in query profile) {
  current_offset = the first empty position on multi channels
  for (allocation frequency[X] of data X) {
    if (current_offset is empty slot) {
      allocate data X to current_offset.
      move current_offset to next channel,
      shift offset to right as much as diff[X]
    } else {
      if (there is empty slot in the same offset as current_offset
      on different channel)
        allocate data X to current_offset.
        move current_offset to next channel,
        shift offset to right as much as diff[X]
      else
        if (there is empty slot one right from current_offset)
          allocate data X to current_offset.
          move current_offset to next channel,
          shift offset to right as much as diff[X]
        else if (there is empty slot one left from current_offset)
          allocate data X to current_offset.
          move current_offset to next channel,
          shift offset to right as much as diff[X]
    }
  }
}
end
    
```

그림 11 MA2D의 데이터 할당 알고리즘

제의 특수한 경우에서 지정된 할당 위치에 이미 다른 데이터가 할당되어져 있을 수 있다. 데이터가 할당되어야 할 채널의 해당 오프셋에 이미 다른 데이터가 할당되어져 있다면 그 데이터는 같은 오프셋의 아래 채널에 할당되어진다. 만약 할당되어야 할 위치의 모든 채널에 이미 다른 데이터가 할당되어 있다면 오른쪽으로 1만큼 이동한 오프셋에 할당한다. 그림 11은 데이터를 할당 시에 앞에서 언급한 특수한 3가지 경우에 대한 대처 방법을 모두 포함한 MA2D의 데이터 할당 알고리즘을 보여준다.

다음은 그림 8의 각 데이터 할당횟수를 이용해 MA2D가 다중채널에 데이터를 어떻게 할당하는 지에 대한 예를 보이겠다. 그림 8에서 모든 데이터의 할당횟수의 총합은 61이고 데이터 채널의 수가 4라고 하면 방송 사이클의 크기, 즉 cycle_length는 $\lceil 61/4 \rceil = 16$ 이 된다. 모든 데이터를 할당한 이후 남은 세 개의 버킷에는 임의의 데이터를 한 번 더 할당한다. 이 예제에서는 데이터 할당 횟수가 1인 데이터들을 한 번 더 할당했다.

먼저 그림 6의 질의어 프로파일에서 가장 높은 요청 확률을 가진 질의어 Q1에 속한 데이터 I, L, R, S를 먼저 채널에 할당한다. 데이터 I는 할당횟수가 4이므로 $\lceil 16/4 \rceil = 4$ 의 간격마다 4번을 주기적으로 채널을 순환하면서 할당되어진다. 여기서 16은 방송 사이클의 크기이다. 데이터 L도 역시 $\lceil 16/4 \rceil = 4$ 의 간격마다 할당횟

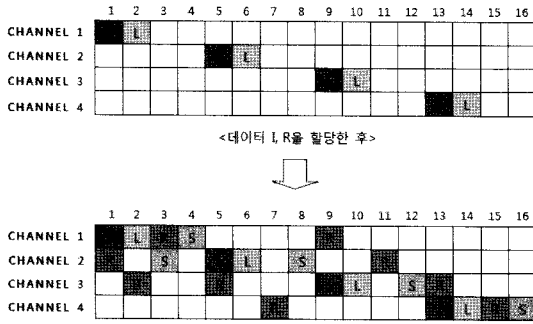


그림 12 질의어 Q1을 데이터채널에 할당한 이후 채널의 상태

수 4회 만큼을 채널을 순환하면서 할당된다. 데이터 R 은 $\lceil 16/9 \rceil = 2$ 인 간격마다 할당되어진다.

이때, 차례로 할당되어지면서 오프셋 15에 할당되어진 이후에는 $\text{Diff}(R) = 2$ 만큼 떨어진 오프셋에 할당해야 하지만 현재 방송 사이클이 끝나기 때문에 방송 사이클을 순환하여 현 방송 사이클의 $(15 + 2) \% 16 = 1$ 번 오프셋에 할당되어진다. R의 할당횟수가 9로 2의 간격마다 채널에 할당되어지면 하나의 방송 사이클의 크기인 16을 넘으면서 동일한 데이터가 다른 방송 채널 상의 같은 오프셋에 할당되어질 수도 있다. 만약, 같은 오프셋에 이미 동일한 데이터가 할당되어져 있다면 1만큼 이전 오프셋에 할당되어진다. 즉, 그림 12와 같이 3번 오프셋에 데이터 R의 9번째 반복 할당이 이루어져야 하지만 이미 1번째 채널의 3번째 오프셋에 데이터 R이 할당되어져 있으므로 1만큼 이전 오프셋인 2번 오프셋에 할당되어진다. 또한, 모든 방송 채널 상에서 할당되어야 할 오프셋에 동일한 데이터가 이미 할당되어져 있다면 지정된 할당 오프셋으로부터 1만큼 이후의 오프셋에 할당되어진다. 질의어 Q1의 마지막 데이터 S는 $\lceil 16/5 \rceil = 4$ 의 간격마다 데이터채널들에 할당되어진다. 질의어 Q1의 데이터 할당 과정은 그림 12와 같다.

질의어 Q1의 모든 데이터들이 할당되어진 이후에는 질의어 프로파일 상에서 질의어 요청확률의 순서에 따라 Q2에 속하여진 데이터들 J, K, M, Q, R이 할당되어진다. 데이터 J는 그림 13과 같이 1번 데이터 채널부터 시작하여 첫 번째 빈 버킷인 1번 데이터 채널의 5번 오프셋에 할당되어진다. 그리고 $\text{Diff}(J)$ 인 8 만큼 떨어진 위치마다 할당횟수 2회 만큼 반복 할당된다. 나머지 데이터 K, M, Q도 같은 방식으로 다중 채널 상에 할당되어진다. 그러나 질의어 Q2의 데이터 R은 질의어 Q1에도 포함되어 있어 이미 이전에 할당되었기 때문에 할당을 생략한다.

위와 같은 방법으로 질의어 프로파일상의 나머지 질의어 Q3부터 Q8까지에 속한 모든 데이터들을 방송 채널

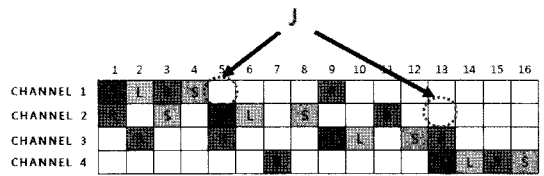


그림 13 질의어 Q2의 데이터 J가 할당될 위치

CHANNEL 1	I	L	R	S	J	K	M	Q	R	C	N	P	B	Q	M	E
CHANNEL 2	R	F	S	P	I	L	Q	S	O	K	R	T	J	A	N	C
CHANNEL 3	Q	R	M	B	R	C	N	D	I	L	M	S	R	K	G	H
CHANNEL 4	O	K	N	Q	K	T	R	E	F	B	Q	A	I	L	R	S

그림 14 데이터들이 채널에 할당된 모습

널에 할당하면 그림 14와 같이 최종 방송 사이클이 완성된다.

본 논문에서 제시한 MA2D에서 데이터의 할당횟수를 구하는 것은 단순한 연산만을 요구하기 때문에 $O(N)$ 의 시간이 소요된다. 여기서 N은 전체 방송되는 데이터의 개수이다. 또한 구해진 할당횟수를 이용하여 다중채널에 데이터를 할당하는데 $O(N+L)$ 가 소요된다. 여기서 L은 모든 데이터에서 추가적으로 할당되는 데이터들의 할당횟수의 총합이다. 결국 MA2D의 데이터 할당 알고리즘의 시간 복잡도는 $O(2N+L)$ 이 된다.

4. MA2D를 위한 인덱싱 기법

3장에서 우리는 질의어 기반의 방송기법에서 각 개별 데이터의 요청빈도를 반영하여 다중채널 상에 할당하는 MA2D 알고리즘에 대하여 알아보았다. 하지만 본 논문에서 제안한 MA2D는 기존의 논문과는 달리 각 데이터들이 서로 다른 수의 할당횟수를 가지게 되고 다양한 채널에 할당될 수 있기 때문에, GA나 PBA와 마찬가지로 MA2D를 위한 인덱싱 기법이 없다면 튜닝시간이 매우 커질 것이다. 이번 장에서는 이 문제를 해결하기 위해 N개의 데이터로 구성된 질의어를 $O(\text{인덱스 버킷의 크기} + N)$ 의 시간에 채널을 탐색하여 원하는 데이터들을 모두 획득할 수 있는 기준에 트리 형태의 인덱싱 기법들과는 달리 간결한 리스트 형태의 MA2D를 위한 인덱싱 기법을 제시한다.

4.1 인덱스 기술 적용을 위한 시스템 환경과 인덱스의 구조

본 논문에서의 방송 채널의 구조는 K개의 데이터 채널과 더불어 한 개의 인덱스 채널로 구성된다. 인덱스의 크기는 데이터의 크기에 비하여 상당히 작으므로 한 개의 인덱스 채널로도 충분히 인덱스 정보를 나타낼 수 있다. 따라서 인덱스 채널의 개수를 최소화 하고 데이터 채널에 더 많은 대역폭(bandwidth)을 할당하는 것이 더

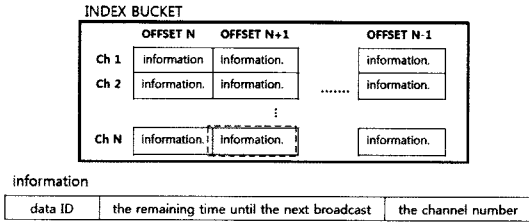


그림 15 인덱스 버킷의 구조

효율적인 방법이다. 클라이언트는 원하는 데이터들을 획득하기 위하여 방송 채널에 접속할 때, 제일 먼저 인덱스 채널에 접속한다. 인덱스 채널을 처음 탐색할 때, 어떤 인덱스 버킷이 방송 중에 있다면 새로운 인덱스 버킷이 방송될 때까지 대기 모드로 보낸 후 새로운 인덱스 버킷의 시작점부터 한 인덱스 버킷을 수신함으로써 클라이언트는 요청된 질의어에 속한 데이터들이 어떤 채널과 시점에 나타나는지를 알 수 있게 된다.

한 인덱스 버킷의 크기는 모든 데이터 채널에서의 방송되는 데이터들의 관한 정보(데이터의 ID, 해당 데이터가 다음 번 방송될 때까지 남은 시간 그리고 방송될 채널번호)의 총합이다. 첫 번째 변수인 데이터 ID와 세 번째 변수인 방송될 채널번호는 1byte로 표현할 수 있다. 그리고 두 번째 변수인 다음 번 방송될 때까지 남은 시간은 2bytes로 표현이 가능할 것이다. 따라서 임의의 오프셋에 특정 데이터 채널에서 방송되는 데이터에 대한 인덱스 정보는 4bytes로 표현된다. 결국 하나의 인덱스 버킷의 크기는 4bytes * Cycle_length * K이다. 각 데이터의 할당횟수의 총합은 이미 앞에서 정해졌기 때문에 데이터 방송채널의 개수가 증가하면 Cycle_length의 크기가 감소하고, 데이터 방송채널의 개수가 감소하면 Cycle_length의 크기가 증가한다는 것을 알 수 있다. 그림 15는 하나의 인덱스 버킷이 데이터들의 인덱스 정보를 어떻게 표현하는지 보여준다.

한 인덱스 버킷 안에는 현 방송 사이클에 다음 인덱스 버킷의 시작 시점부터 시작하여 다음 방송 사이클의 현재 인덱스 버킷이 끝나는 시점까지 순환하면서 모든 데이터 채널 상에 나타나는 데이터에 대한 정보가 나타난다. 즉, 그림 16 에서 인덱스 버킷 4는 현재 방송 사이클의 9번 오프셋부터 다음 방송 사이클의 8번 오프셋까지의 모든 데이터 채널 상에 나타나는 데이터에 대한 위치 정보를 가지고 있다. 그리고 인덱스 버킷 5는 현재 방송 사이클의 11번 오프셋부터 다음 방송 사이클의 10번 오프셋까지의 모든 데이터 채널 상에 나타나는 데이터에 대한 위치정보를 가지고 있다.

그림 16에서 OFFSET = 9의 (R,2,2)(O,8,4)(I,4,4)(F,9,2)가 의미하는 것은 9 라는 타임오프셋에 네 개의

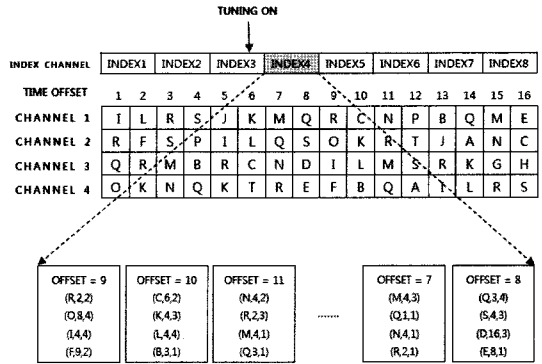


그림 16 인덱싱 기법

데이터 채널에 나타나는 데이터 정보가 각각 (R,2,2), (O,8,4), (I,4,4), (F,9,2) 라는 것이다. 괄호안의 첫 번째 인자는 데이터ID를 의미하고, 두 번째 인자는 해당 데이터가 앞으로 다시 나오기까지 남은 타임오프셋을 의미한다. 마지막 세 번째 인자는 다음번에 해당 데이터가 나오는 채널 번호이다. 두 번째와 세 번째 인자는 질의어에 속한 하나 이상의 데이터들이 다른 채널 상에 동시에 나왔을 경우 어떤 데이터를 먼저 획득할 것인지를 결정하기 위한 정보이다. 예를 들어, 한 질의어에 포함된 두 개 이상의 데이터가 동시에 방송된다면, 클라이언트는 두 번째 인자 값이 더 큰 앞으로 더 오랜 시간 후에 다시 방송될 데이터를 먼저 획득한다. 이는 모바일 클라이언트가 그 데이터를 놓친다면 다음 방송될 때까지 가장 많은 시간을 기다려야 하기 때문이다. 그리고 세 번째 인자를 통하여 다음번에 방송될 데이터가 나타날 채널번호를 알 수 있게 된다. 클라이언트는 대기모드로 있다가 해당 데이터가 방송될 시점에 해당 채널에 다시 접속하여 데이터를 획득하게 된다. 이러한 인덱스 정보를 이용하여 클라이언트는 자신이 요청한 질의어에 속한 데이터가 나오는 시점과 채널번호에 대한 정보를 유지하면서 튜닝시간을 최소화하여 데이터를 획득할 수 있게 된다.

MA2D의 인덱스 구축은 데이터 채널의 개수가 K개 일 때 $O(K * cycle_length * \text{인덱스 버킷의 개수})$ 의 시간에 이루어진다. 즉, $K * cycle_length$ 만큼의 작업이 인덱스 버킷의 개수 만큼 수행되어진다. 인덱스 정보를 구성하기 위한 컨트롤 정보들은 별도의 연산을 필요로 하지 않고 정해진 값들과 단순 계산들로 구할 수 있기 때문에 비교적 작은 시간에 인덱스를 구성할 수 있다.

4.2 인덱싱 기법의 예

이번 절에서는 우리가 제시한 인덱싱 기법이 앞장에서 제안한 MA2D 데이터 할당 방법에 대해서 어떻게 적용되는지 알아보겠다. 그림 6의 질의어 프로파일과 4

개의 데이터 방송채널이 있다고 하자. MA2D에 의하여 데이터들은 그림 17과 같이 데이터 방송 채널에 할당되어진다. 클라이언트가 질의어={ I, L, R, S }를 요청했다면 해당 데이터들을 획득하기 위하여 제일 먼저 인덱스 채널에 접속한다. 그림 17에서 클라이언트는 인덱스 버킷 3의 중간에 접속했으므로 새로운 인덱스 버킷이 방송될 때까지 대기모드로 기다린다. 인덱스 버킷 4가 방송될 때 클라이언트는 다시 채널을 수신하기 시작하여 하나의 인덱스 버킷을 탐색함으로써 해당 데이터들에 대한 위치정보와 시간정보를 알아낸다.

그림 17에서 오프셋(OFFSET) 9의 정보로부터 클라이언트는 질의어에 포함된 데이터 R과 데이터 I가 각각 데이터 채널 1과 데이터 채널 3에 동시에 나타난다는 것을 알게 된다. 그러나 한순간에 오직 한 개의 채널에만 접근할 수 있는 무선 환경의 특징으로 인하여 클라이언트는 데이터 R과 데이터 I중 한 개의 데이터를 선택하여 먼저 획득해야 한다. 이렇게 데이터를 선택해야 하는 상황이 왔을 때 두 번째 인자와 세 번째 인자를 통하여 클라이언트는 선택의 문제를 해결할 수 있다. 이 예제에서 클라이언트는 두 번째 인자 정보를 통하여 다음에 다시 방송되기까지 남은 시간이 큰 데이터 I를 3번 데이터 채널에서 획득해야 한다는 것을 알게 된다. 그림 17에서 음영으로 표시된 버킷은 데이터 R과 데이터 I가 다시 방송되는 위치와 시간을 보여준다. 만약 데이터 I를 이번에 획득하지 못한다면 다음에 다시 획득하기까지 데이터 R보다 더 많은 시간을 기다려야 하기 때문이다. 이어서 클라이언트는 오프셋 10에서 3번 데이터 채널에서 데이터 L을 획득하고, 이전의 오프셋 9에서 획득하지 못했던 데이터 R을 오프셋 11의 2번 채널에서 획득한다. 마지막으로 데이터 S는 오프셋 12의 3번 채널에서 획득하게 된다. 이 예제에서와 같이 클라이언트는 질의어에 속한 데이터들을 모두 획득하기 위하여 하나의 인덱스 버킷과 원하는 데이터들이 방송되는 시간만큼만 채널을 수신하면 된다.

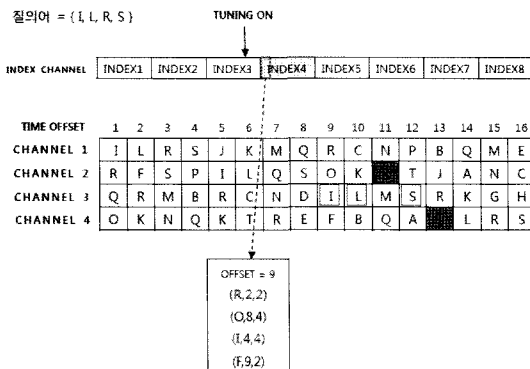


그림 17 인덱싱 기법의 예제

5. 성능 분석

본 장에서는 몇 가지 실험을 통해 본 논문에서 제시하는 MA2D가 효과적임을 보인다. 실험을 진행하는 동안 우리는 무선 다중 채널에 상호관련 데이터들을 할당하는 방법들인 PBA와 GA를 비교대상으로 삼았다.

본 논문의 모든 실험은 Intel Pentium(R) D CPU 3.40GHz 프로세서와 2GB 메모리상에서 시뮬레이션 되었다. 표 1은 우리의 실험에서 사용되는 실험 변수를 나타낸다. 실험 변수 중에 하나인 매개 변수 θ 는 모바일 클라이언트들의 질의어에 대한 비정규 분포의 접근 패턴을 모델링하기 위해서 사용되는 Zipf 분포이다. Zipf 분포는 일반적으로 모바일 방송 환경에서 클라이언트의 편향된 분포의 접근 패턴을 모델링 하는데 널리 사용된다. Zipf 분포는 다음의 식 (4)과 같이 표현된다. 식 (4)에서 N은 질의어 프로파일에서 질의어의 개수를 의미한다.

$$\Pr(Q_i) = \frac{(1/i)^\theta}{\sum_{i=1}^N (1/i)^\theta}, 1 \leq i \leq N \quad (4)$$

Zipf 분포를 가리키는 매개 변수 θ 는 편향된 질의어 접근 패턴을 나타낸다. $\theta=0$ 일 때, 모든 질의어는 같은 접근 확률을 가지게 되고 Zipf 분포는 유니폼 분포(uniform distribution)가 된다. 이 θ 값이 커질수록 Zipf 분포는 더욱 치우쳐진 분포(skewer distribution)가 되는데, 이것은 클라이언트들이 특정 질의어에 대한 선호도가 높고 남은 다수의 질의어에 대해서는 접근 확률이 매우 낮아짐을 의미한다. 본 논문의 실험에서의 기본적인 모델은 모바일 클라이언트, 서버, 그리고 인덱스 정보를 내보내기 위한 인덱스 채널과 데이터를 전송하기 위한 데이터 채널의 방송 디스크로 구성된다.

본 논문에서는 질의어의 응답 시간과 튜닝시간을 성능

표 1 실험 변수

Parameter	Values
전 체	
size of a data	128 bytes
Number of the items(N)	250~5000
Number of the queries	50
Number of data channels(K)	1~20
Average length of a query	1%~5%
Zipf parameter(θ)	0.2~1.4
기 본 환 경	
Number of the items	2000
Number of the queries	50
Number of data channels	4
Average length of a query	2%
Zipf parameter(θ)	0.8

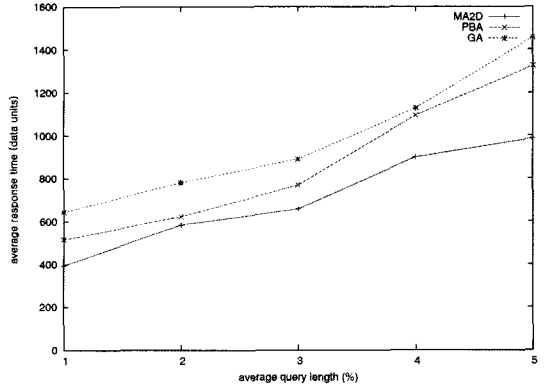
지표로 삼았다. 여기서의 응답시간은 접근시간(Access time)을 말하는 것으로서 클라이언트가 원하는 질의어에 속한 모든 데이터를 얻기 위하여 방송 채널에 처음 탐색(initial probe)하기 시작한 순간부터 최종적으로 원하는 데이터를 모두 획득할 때까지의 총 시간을 말한다. 결국 클라이언트가 데이터를 획득하기 위해서는 접근시간(response time) 만큼을 소비해야 한다. 따라서 접근시간은 방송 프로그램의 성능을 측정하는 위해서 사용된다. 또한 튜닝시간은 데이터를 얻기 위하여 실제로 채널에 접속하고 있는 시간을 의미한다. 즉 모바일 기기의 에너지 소비량을 측정하는 기준이 된다.

본 실험에서는 표 1과 같이 방송될 데이터의 개수, Zipf 분포, 데이터 채널의 개수, 그리고 질의어의 평균 길이와 같은 다양한 변수들에 따른 응답시간에 대한 변화를 측정할 것이고 튜닝시간 또한 측정할 것이다. 여기서 질의어의 평균 길이는 한 질의어 안에 포함되어 있는 데이터들의 평균 개수를 의미한다. 표 1에서 보여주는 질의어의 평균길이를 나타내는 단위 %는 전체 데이터의 개수에서의 비율을 뜻한다. 예를 들어, 전체 방송될 데이터의 개수가 1000개이고 질의어의 평균길이가 2%라고 한다면 질의어의 평균길이는 20개가 된다. 위 지표에 대한 성능 분석은 100번의 시뮬레이션을 통해서 측정되었다. 본 논문의 실험에서의 데이터와 인덱스에 대한 접근시간과 튜닝시간의 측정 단위는 논리적인 데이터 1 unit 을 기준으로 하였다.

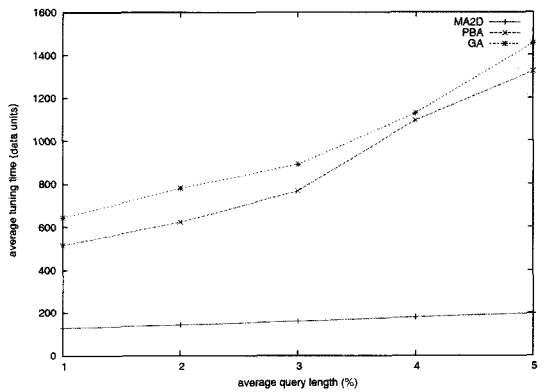
5.1 질의어의 평균 크기에 따른 성능

본 실험에서는 하나의 질의어 안에 포함되어 있는 데이터의 평균 개수를 의미하는 질의어의 평균 길이를 다양하게 변화시켰을 때 질의어 응답시간과 튜닝시간에 어떠한 영향을 미치는지 확인하였다. 실험 환경은 표1에서 제시한 기본 환경과 같고 평균 질의어 크기를 1%에서 5%로 변화시키면서 실험하였다. 평균 질의어 크기가 1%라는 것은 한 질의어 안에 평균적인 데이터의 개수가 전체 데이터의 개수 2000개 중의 1%인 20개라는 것을 뜻한다. 그림 18(a)을 보면 모든 데이터 할당 방법에서 질의어의 평균 크기가 커짐에 따라 질의어 평균 응답시간이 커진다는 것을 알 수 있다. 이는 한 질의어에 많은 데이터가 포함되어 질수록 데이터 충돌(conflict data items)의 발생 가능성이 커지기 때문이다. 그리고 데이터 충돌의 발생은 다음 방송 사이클까지의 대기 시간을 요구하므로 질의어 평균 응답시간의 증가를 야기시킨다.

하지만 MA2D에서는 빈번하게 요청되는 데이터들은 한 방송 사이클 안에 여러 번 할당되었고 방송 프로그램의 길이가 길어져 각 질의어의 평균응답시간을 다소 증가시킬 수 있지만 데이터 충돌이 발생 시에 다음 방



(a) average query length vs. average response time



(b) average query length vs. average tuning time

그림 18 질의어 크기의 변화에 따른 실험

송 사이클이 아닌 해당 데이터 방송 채널 상에 다시 데이터가 나타날 때까지만 대기하면 되므로 평균 질의어 응답시간의 향상을 이룰 수 있다. 그림 18(b)는 평균 질의어 크기가 1%에서 5%까지 변화함에 따라 평균 튜닝시간의 변화를 측정한 것이다. PBA와 GA는 인덱싱 기법을 활용하지 않기 때문에 질의어에 포함된 데이터들을 획득하기 위해선 모든 채널들을 순차적으로 검색해야 함으로 평균응답시간과 거의 같은 튜닝시간을 갖는다. 하지만 MA2D는 인덱스 정보를 활용하여 실제 원하는 데이터를 받는데 걸리는 시간과 인덱스 정보를 획득하는데 걸리는 시간만으로 질의어를 획득할 수 있다. 그림을 통하여 알 수 있듯이 우리가 제안한 MA2D가 기존의 PBA와 GA에 비하여 훨씬 우수한 튜닝시간을 갖는 것을 볼 수 있다.

5.2 서버 데이터베이스의 데이터 개수에 따른 성능

이번 절에서는 서버에서 유지하고 있는 데이터베이스의 데이터 개수의 변화에 따른 우리의 제안 방법과 다른 방법들과의 성능을 비교해 보도록 하겠다. 실험 환경

은 표 1에서 제시한 기본 환경과 같고, 전체 데이터의 개수를 250개에서 5000개로 변화시키면서 실험해 보았다. 서버에서 방송해야 하는 데이터의 양이 커진다는 것은 모바일 클라이언트 입장에서 필요로 하는 데이터를 얻는데 많은 시간이 걸린다는 것을 의미한다. 왜냐하면 필요로 하는 모든 데이터를 얻기 위해서는 필요로 하지 않은 다른 데이터들이 방송되는 시간 또한 기다려야 하고 데이터의 개수가 많으면 필요하지 않은 데이터의 개수 또한 증가할 것이기 때문이다.

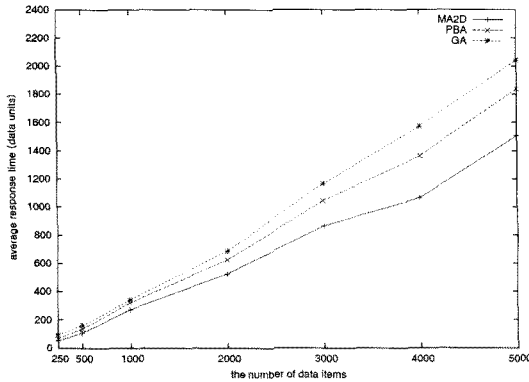
그림 19에서는 데이터베이스의 데이터의 개수를 증가시킬 때 MA2D와 기존의 다른 방법들과의 평균응답 시간과 평균튜닝시간을 비교하였다. 그림 19(a)을 통해 알 수 있듯이 제안한 MA2D가 기존의 다른 방식들보다 더 작은 평균응답시간을 가지는 것을 알 수 있다. 또한 데이터베이스의 데이터의 개수가 증가함에 따라 세 가지 방식의 평균 응답시간이 모두 증가 하는 것도 확인할 수 있다. 즉, 방송할 데이터의 개수가 많다는 것은 데이터들을 할당하기 위해 필요한 각 데이터채널에서의

방송 사이클과 질의어의 평균응답시간의 증가를 의미하게 된다. MA2D가 다른 기법들에 비해서 감소된 평균 응답시간을 보여주는 것은 데이터 충돌이 발생하였을 경우 충돌된 데이터를 다시 획득하는데 다른 기법들 보다 빠른 시간 안에 획득할 수 있기 때문이다. 그림 19(b)는 전체 데이터의 개수가 증가함에 따른 평균튜닝 시간의 변화를 보여준다. 평균 질의어의 크기는 기본 환경과 같이 2%로 설정하였으므로 전체 방송되는 데이터의 개수가 증가함에 따라 질의어의 크기 또한 증가하여 그림 18(b)에서의 질의어 크기의 변화 실험과 마찬가지로 평균튜닝시간이 증가하는 것을 알 수 있다. 그리고 질의어 크기의 변화에 따른 평균튜닝시간의 실험과는 달리 MA2D는 전체 데이터의 양이 많아지면 그로 인해 증가된 인덱스 정보를 보다 많이 수신해야 함으로 평균 튜닝시간이 질의어 크기의 변화 실험에서 보다 크게 증가하는 것을 알 수 있다.

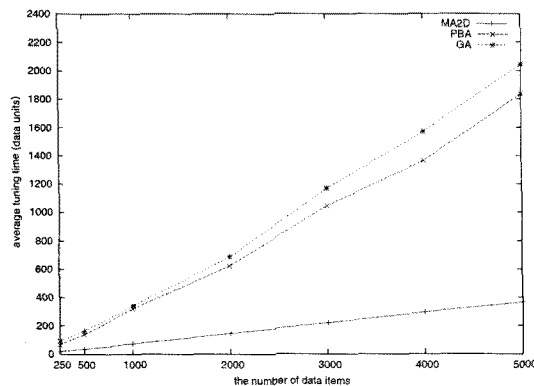
5.3 질의어 접근 편향성에 따른 성능

이번 절에서는 모바일 클라이언트들이 서버에 요청하는 질의어의 접근 편향성에 따른 질의어의 평균응답 시간과 평균튜닝시간을 측정한다. 모든 질의어의 요청확률의 합은 1이다. 우리는 앞에서 설명한 무선방송환경에서 모바일 클라이언트의 편향된 분포에 접근 패턴을 모델링 하는데 널리 사용되는 Zipf 분포를 사용할 것이다. 그림 20에서는 Zipf 변수 θ 를 0.2부터 1.4까지 0.2의 간격마다 측정된 결과 값을 보여준다. Zipf 변수 θ 는 숫자가 커질수록 특정 질의어에 더 많은 요청이 편향되어 있다는 것을 의미한다.

그림 20을 통하여 MA2D가 다른 기존의 방식들보다 Zipf 변수의 다양한 변화에 더 효율적이라는 것을 보여준다. 그림 20(a)의 두드러진 점은 Zipf 변수가 증가하여 질의어 선호도에 대한 편향성이 증가함에 따라 우리가 제시하는 MA2D 기법의 질의어에 대한 평균응답시간이 다른 기법들의 평균응답시간 보다 더 현저하게 작아진다는 것이다. 이것은 높은 요청확률을 가지고 있는 질의어에 속한 데이터들이 더 높은 요청빈도를 가지게 될 것이고 결국 데이터 방송채널에 더 많이 할당되어 전체적으로 요청확률이 높은 데이터의 충돌이 증가할 경우 평균적으로 MA2D의 응답시간이 다른 기법에 비해서 빠르기 때문이다. 이러한 특징으로 인하여 우리가 제안한 MA2D의 질의어 응답시간이 다른 기존의 방식에 비하여 좋은 성능을 가지게 된다. 그림 20(b)는 Zipf 변화에 따른 평균튜닝시간의 결과이며 MA2D의 평균튜닝시간이 Zipf 변화에는 큰 영향을 받지 않는다는 것이다. 이유는 MA2D의 평균튜닝시간은 질의어 안에 포함된 데이터의 개수와 전체 데이터의 총 개수에 의해서만 영향을 받을 뿐 Zipf 변화에 따라 영향을 받지 않

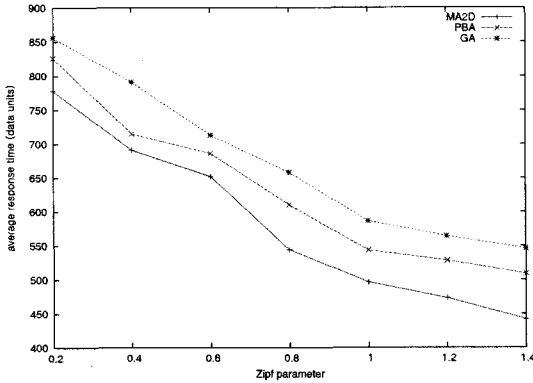


(a) the number of data vs. average response time

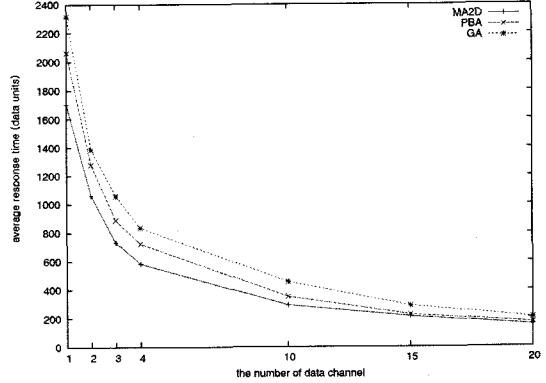


(b) the number of data vs. average tuning time

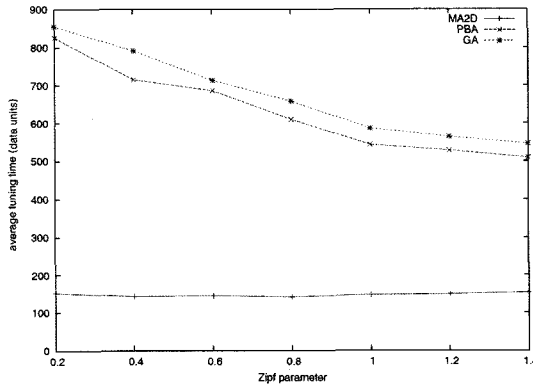
그림 19 전체 데이터 개수의 변화에 따른 실험



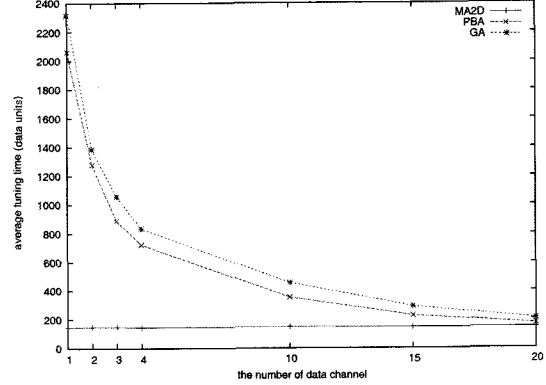
(a) Zipf parameter vs. average response time



(a) the number of data channel vs. average response time



(b) Zipf parameter vs. average tuning time



(b) the number of data channel vs. average tuning time

그림 20 Zipf parameter의 변화에 따른 실험

그림 21 데이터 방송 채널 개수의 변화에 따른 실험

기 때문이다. 그리고 Zipf 변화는 MA2D의 평균튜닝시간에 영향을 주는 요소인 인덱싱 정보의 크기에 대해서도 영향을 주지 않기 때문이기도 하다.

5.4 데이터 방송 채널의 개수에 따른 성능

이번 절에서는 데이터 방송채널의 개수를 1개에서 20개까지로 변화 시켰을 때의 질의어의 평균응답시간을 측정해 보았다. 나머지 실험 환경은 표 1에서 제시한 기본 환경과 같다. 그림 21(a)를 보면 모든 데이터 할당방법에서 데이터 방송채널의 개수가 증가함에 따라 질의어 응답시간이 감소하는 것을 확인할 수 있다.

이는 데이터 방송 채널의 개수가 증가할수록 하나의 데이터 채널에 할당되는 데이터의 개수가 적어져서 방송 사이클이 작아지기 때문이다. 작아진 방송 사이클은 질의어의 평균 응답시간을 줄일 수 있게 한다. 그림 21(b)은 본 실험에서 전체 데이터의 개수와 평균질의어의 크기가 고정되어 있고 우리의 제안 방법인 MA2D가 데이터 방송채널의 개수의 변화에 영향 받지 않기 때문에 일정한 평균튜닝시간을 가진다는 것을 보여 준다. 그

림 21을 통해서 MA2D가 다양한 데이터 방송 채널수에서 기존의 다른 방식들보다 더 우수한 질의어 평균응답시간과 평균튜닝시간을 가진다는 것을 알 수 있다.

6. 결론

무선방송환경에서의 기존 논문들은 클라이언트가 한 개의 독립적인 데이터만을 요청하고 획득하는데 걸리는 시간(response time)을 최소화로 만드는 것을 목표로 하였다. 그러나 실제 환경에서는 이렇게 개개의 데이터만을 요구하기 보다는 관련된 여러 데이터들을 동시에 요구하는 경우가 더 많다. 본 논문에서는 이러한 상호관련 데이터들을 더 작은 응답시간에 획득할 수 있는 효율적인 데이터 할당 방법인 MA2D를 제시했다. MA2D에서는 기존 방법에서처럼 데이터 할당 시에 질의어 할당 우선순위를 유지하여 데이터 충돌(conflict data items)의 영향을 줄이면서 질의어 프로파일 전체를 볼 때 상대적으로 많이 요청되는 데이터를 한 방송 사이클 동안 다중 채널 상에 여러 번 할당하였다. 또한 할당횟수를

기반으로 각 데이터를 일정한 간격을 유지하면서 다중 채널 상에 할당하여 질의어에 포함된 모든 데이터를 획득하는데 요구되는 평균응답시간을 최소화 하는 방법을 제안하였다. 그리고 본 논문에서는 서로 다른 수의 할당 횟수를 가지고 채널에 독립적으로 할당되는 데이터들을 작은 튜닝시간에 획득할 수 있는 MA2D를 위한 인덱싱 기법을 제시하였다. 하지만 할당횟수를 구하는 방식에서 가장 적은 요청횟수가 가장 큰 요청횟수에 비하여 상대적으로 상당히 작다면 전체 할당회수의 크기가 커져 전체 방송 사이클이 커지는 단점이 있었다. 향후, 우리는 데이터의 할당횟수를 효율적으로 정규화는 방법에 대한 추가적인 연구를 할 계획으로 있다.

참 고 문 헌

- [1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management For Asymmetric Communications Environments", Proc. ACM SIGMOD Conf. Management of Data, pp. 199-210, May 1995.
- [2] K. Prabhakara, K.A. Hua, and J. Oh, "Multi-Level Multi-Channel Air Cache Designs for Broadcasting in a Mobile Environment", Proc. Int'l Conf. Data Eng. (ICDE), 2000.
- [3] T. Imielinski, S. Viswanathan, and B.R. Badrinath. Power efficient filtering of data on air. In Proc. of EDBT Conference, pages 245 -258, March 1994.
- [4] MSN Direct Service. available at <http://www.msndirect.com>
- [5] <http://www.satellitelife.com/hughesnet.asp>
- [6] W.C. Peng and M.S. Chen, "Dynamic Generation of Data Broadcasting Programs for Broadcast Disk Arrays in a Mobile Computing Environmet", Proc. ACM Conf. Information and Knowledge Management (CIKM), pp. 35-45, Nov. 2000.
- [7] Wai Gen Yee, Student Member, IEEE, Shamkant B. Navathe, Edward Omiecinski, and Christopher Jermaine, "Efficient Data Allocation over Multiple Channels at Broadcast Servers", IEEE Transaction on computers, Vol. 51, No. 10, October 2002.
- [8] Song-Yi Yi, Seunghoon Nam, S. Jung, "Effective Generation of Data Broadcast Schedules with Different Allocation Numbers for Multiple Wireless Channels", IEEE Transaction on Knowledge and Data Engineering, vol.20, no. 5, pp. 668-677, 2008.
- [9] Y. D. Chung and M.H. Kim. "Effective Data Placement for Wireless Broadcast". Distributed and Parallel Databases, 9(2), 2001.
- [10] G. Lee and S. C. Lo. "Broadcast Data Allocation for Efficient Access of Multiple Data Items in Mobile Environments". Mobile Networks and Applications, 8, 2003.
- [11] J. -L. Huang and M. -S. Chen. "Dependent Data Broadcasting for Unordered Queries in Multiple Channel Mobile Environment." IEEE Trans. on Knowledge and Data Engineering, 16(6), Jun. 2004.
- [12] Hao-Ping Hung, Jen-Wei Huang, Jung-Long Huang, Ming-Syan Chen, "Scheduling Dependent Items in Data Broadcasting Environments", ACM SAC 2006.
- [13] T. Imielinski, S. Viswanathan, and B. Badrinath, "Data on Air: Organization and Access", IEEE Trans. Knowledge and Data Eng., vol. 9, no. 3, pp. 353-372, May 1997.
- [14] Narayanan Shivakumar, and Suresh Venkatasubramanian, "Efficient indexing for broadcast based wireless systems", Mobile Networks and Applications, vol. 1, no. 4, pp. 433-446, December 1996
- [15] A. Nanopoulos, D. Katsaros, and Y. Manolopoulos, "Effective Prediction of Web-User Accesses: A Data Mining Approach", Proc. WEBKDD Workshop, Aug. 2001
- [16] Jiun-Long Huang and Ming-Syan Chen, "Broadcast Program Generation for Unordered Queries with Data Replication", ACM SAC 2003.
- [17] S. Jung and B. Lee, S. Pramanik, "A Tree-Structured Index Allocation Method with Replication over Multiple Broadcast Channels in Wireless Environments", IEEE Transaction on Knowledge and Data Engineering, vol. 17, no. 3, pp. 311-325, Mar. 2005.
- [18] Kun-Feng Lin and Chuan-Ming Liu, "Broadcasting Dependent Data with Minimized Access Latency in a Multi-channel Environment", ACM IWCMC 2006.
- [19] Shuoi Wang, Hsing-Lung Chen, "TMBT: An Efficient Index Allocation Method for Multi-Channel Data Broadcast," AINAW'07, pp. 236-242, 2007.



박 성 욱

2006년 2월 서강대학교 컴퓨터학과 학사. 2008년 2월 서강대학교 컴퓨터공학과 석사. 2008년 2월 ~ 현재 삼성전자 정보통신총괄 연구원. 관심분야는 Mobile Computing Systems, WCDMA, GSM, WiBro, Mobile Database



정 성 원

1988년 서강대학교 전자계산학 학사. 1990년 M.S. in Computer Science at Michigan State University. 1995년 Ph.D. in Computer Science at Michigan State University. 1997년 ~ 2000년 한국전산원 선임연구원. 2000년 ~ 현재 서강대학교 컴퓨터공학과 부교수. 관심분야는 Mobile Databases, Mobile Computing Systems, Spatial Databases, Telematics, LBS, GIS