유비쿼터스 컴퓨팅을 위한 이동 호스트의 서버 선택 알고리즘

김 선 호[†] · 윤 미 연[††] · 신 용 태[†††]

요 약

컴퓨터 하드웨어의 크기가 작아지고 성능이 향상되어 노트북, PDA 등 개인용 휴대 컴퓨터의 사용이 늘어남에 따라 물리적 위치에 관계없이 어느 곳에서나 원하는 정보를 손쉽게 접근하고자 하는 사용자들의 요구는 유비쿼터스 컴퓨팅 기술을 탄생시켰다. 본 연구에서는 기존의 유선 환경에서 빠르고 안정적인 서비스를 위하여 사용한 콘텐츠 분산 네트워크 기술을 이동 환경에 적용하여 이동 호스트가 적절한 콘텐츠 서버를 선택하여 서비스 받음으로써 이동 중에도 단절 없이 콘텐츠를 서비스 받을 수 있는 방안을 제시하였다. 이동 환경에 적합한 서버 선택 알고리즘을 제안함으로써 보다 효율적인 전송이 가능함을 실험을 통해 보였으며 이러한 연구는 유비쿼터스 컴퓨팅의 실현에 기여할 수 있을 것이다.

Server Selection Algorithm of Mobile Host for Ubiquitous Computing

Seonho Kim[†] · Miyoun Yoon[††] · Yongtae Shin[†††]

ABSTRACT

According as computer hardware became smaller and its performing power became improved due to the developed computer technology, clients demand more powerful accessibility to various information through getting the information at every time and every place. These needs led to the Ubiquitous computing technology which makes it possible to have access to a specific information regardless of clients' physical location. In this paper, we propose an approach that mobile host can get data from proper contents server by applying Contents Distributing Network to wireless network. By simulation, this paper proves that this approach can improve performance by using contents server selection in wireless network. This research will contribute to realizing the Ubiquitous computing technology.

키워드 : Mobile Host, Server Selection, CDN

1. 서 론

유비쿼터스란 언제 어디에서나 존재한다는 의미로써 유비쿼터스 컴퓨팅은 주변의 모든 사물에 칩을 넣어 언제 어디서나 사용이 가능한 컴퓨팅 환경이라는 개념으로 정의할 수 있다. 이것은 1999년 일본에서 유비쿼터스 네트워크로 개념이 확장이 되어 언제 어디서나 컴퓨터에 연결될 수 있는 네트워킹 환경을 의미하게 되었다. 이것은 휴대폰, PDA와 같은 휴대용 컴퓨터를 가지고 다니면서 멀리 떨어져 있는 각종 사물과 연결하여 그 사물을 사용한다는 개념으로까지 확장되었다[1]. 또한, 컴퓨터 하드웨어의 크기가 작아지고 성능이 크게 향상되어 노트북, PDA 등 개인용 휴대 컴퓨터의 사용이 늘어남에 따라 클라이언트의 물리적 위치에 관계없이 어느 곳에서나 원하는 정보를 손쉽게 접근하고자 하는 요구는 인터넷과의 연동을 통해 기존 유선 네트워크에서 제공해 왔던 다양한 서비스를 이동 환경에서도 가능케 하도록 추구하고 있으며 차세대 인터넷은 유선과 무선이 혼합된 초고속 네트워크 기반 하에 대용량 멀티미디어 데이터 전송을 가능하게 할 것이다.

반면, 인터넷상에 클라이언트와 웹 사이트의 급속한 증가에 따른 서버의 부하 가중과 네트워크 트래픽의 문제를 해결하기 위해 서버의 용량 증대 및 네트워크 자원의 증설 작업이 진행되어 왔다. 그러나 저장 용량, 네트워크 자원 등의 물리적인 용량의 확보는 인터넷 클라이언트와 웹 사이트의 증가 속도를 따라 잡지 못하기 때문에 클라이언트의 요구와 네트워크 트래픽에 효과적으로 대처할 수 없다. 이러한 문제를 해결하기 위한 대안의 하나가 콘텐츠 분산 네트워크(CDN)[2]이다. CDN은 콘텐츠 서버들을 다수 지역에 설치하고 정책적으로 접속 횟수가 잦은 대용량의 콘텐츠를 콘텐츠 서버에 미리 복제해 놓고 클라이언트의 요청에 대해 클라이언트의 위치와 가까운 콘텐츠 서버로 하여금 콘텐츠를 최단 경로를 통해 신속하게 서비스하도록 하는 기술이다.

그러므로 CDN 기반 기술을 이동 환경에 적용한다면 보다 심리스한 이동 서비스를 가능하게 하여 유비쿼터스 컴퓨팅 실현에 기여할 수 있을 것이다. CDN이 그 효과를 발휘하기

† 정 회 원 : 동덕여자대학교 정보과학대학 교수
†† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
††† 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2004년 7월 20일, 심사완료 : 2004년 9월 30일

위한 가장 중요한 이슈는 다수의 콘텐츠 서버 중에서 클라이언트의 요청을 가장 빠르고 안전하게 처리할 수 있는 서버를 선택하는 것이라고 볼 수 있다. 그러나 CDN은 유선 환경에 기반하여 설계되었으므로 클라이언트의 이동성을 고려하지 않아 유선 환경에서의 서버 선택 메커니즘을 이동 환경에 그대로 적용할 수는 없는 실정이다. 그러므로 이동 환경에서도 여전히 좋은 성능을 기대할 수 있는 새로운 서버 선택 메커니즘이 필요하다.

이에 본 논문에서는 유선 네트워크 상의 콘텐츠 서버 선택 메커니즘을 무선 네트워크 상에 적용 시킬 경우 고려 사항을 검토하고 이동 환경에 적합한 서버 선택 메커니즘을 이용한 이동 호스트의 서비스 개선 방안을 제안하고자 한다.

2. 유선 환경에서의 서버 선택

다수의 콘텐츠 서버를 사용하는 환경에서 가장 중요한 이슈는 클라이언트의 콘텐츠 요청에 가장 적절한 콘텐츠 서버가 응답하도록 하는 것이다. 서버의 선택은 클라이언트의 요청을 리다이렉트 해주는 서버 측에서 할 수도 있고 클라이언트가 콘텐츠 서버와의 인접도와 네트워크 상태 등을 고려하여 선택할 수도 있다.

[3]에서는 클라이언트의 요청을 웹 서버 들 간에 돌아가며 분배하기 위하여 Round Robin DNS를 사용한다. 클라이언트의 요청에 대해 여러 대의 서버에 동일한 비율로 서비스 요구가 전송 될 수 있도록 서버 IP 주소를 반환함으로써 서버의 부하를 분산하도록 하는 것이다. 구현이 간단하고 가장 일반적인 방법이지만, 이것은 DNS가 웹 서버의 상태는 알 수 없기 때문에 현재 웹 서버의 상태는 고려하지 않고 단순히 하나씩 차례로 선택을 해 나감으로써 각 서버에 접속하는 것을 공평하게 하고자 하는 것이다. Round Robin DNS 방식은 웹 서비스를 제공하는 여러 대의 서버가 모두 같은 성능을 갖고 있을 때 효과적인 방법으로 특정 서버에 장애가 발생할 경우 전파지연(Propagation Delay) 동안 일정한 비율의 클라이언트들은 서비스를 제공받지 못하는 문제가 있다.

[4]에서는 홉 수, RTT(Round Trip Time) 등 서버 선택 메트릭 간에 비교 연구를 하여 특정 서버의 과거 지연에 대한 통계 데이터가 현재의 요청 지연을 예측 할 수 있게 함으로써 가장 지연이 적은 서버를 선택할 수 있다는 것을 발견하여 응답시간이 아닌 비용 함수로 HTTP 지연 시간을 측정하여 분석하였다.

[5]에서는 사용 가능한 대역폭과 RTT를 측정하는 툴을 사용하여 파일 크기가 작은 경우는 RTT를 이용하여 서버를 선택하고 파일 크기가 큰 경우는 RTT와 사용가능한 대역폭을 함께 고려한 동적인 서버 선택 알고리즘을 택하여 과거 정적인 접근 메트릭과 HTTP 전송 시간을 비교한 결과 성능이 50% 이상 향상됨을 보였다. 그러나 대역폭을 고려하는 경우 시간과 네트워크 트래픽 면에서 오버헤드가 큰 단점이 있다.

[6]에서는 클라이언트 측면에서 2단계 서버 선택 메트릭을 제안하고 실험하였다.

190여개의 대상 서버에 클라이언트가 ping을 보내어 RTT가 작은 서버 집합을 먼저 선정한 후 해당 서버들을 대상으로 random 하게 서버를 선택한 2단계 ping-random 방법이 홉 수, AS 수, RTT, 크기가 작은 파일을 먼저 전송하게 하여 다운로드가 가장 빠른 쪽으로 전체 파일 전송을 요청하는 서버 선택 방법 등에 비해 파일 크기가 클수록 좋은 성능을 보였으며, 이것은 네트워크 상태를 반영할 수 있으며 서버 간의 부하 균형을 이룰 수 있는 장점이 제공하였다. 1단계 ping 서버 집합은 10일간 유효한 것으로 보고 지속하는데 이것은 선택된 집합의 순위는 일정기간 유지된다는 기존 연구 결과를[7] 기반으로 하는 것이다. 홉 수나 AS 수만을 고려한 것은 단지 임의로 서버를 선택한 것과 성능에 있어 큰 차이를 보이지 않았다.

[8]에서는 클라이언트가 대상 서버들에게 ping을 하여 일정시간 기다린 후 응답이 온 서버들 중 클라이언트와의 대역폭이 가장 좋은 서버를 선택하도록 하여 기존의 과거 전송 데이터에 의한 정적인 대역폭만을 고려한 것에 비해 좋은 성능을 보임을 나타내었다.

3. 무선 환경에 적용하기 위한 네트워크 모델

CDN을 이용한 유선 네트워크 상에서의 서비스를 이동 환경에 적용하기 위한 네트워크 모델을 정의한다. 적절하지 못한 서버의 선택은 CDN을 이용하는 목적을 달성할 수 없게 한다. 그러므로 이동 호스트의 서비스 요청에 대해 가장 적절한 서버를 선택하여 응답하도록 하는 것은 이동 호스트의 서비스 질을 높일 수 있는 가장 중요한 부분이라고 할 수 있다. 이동 컴퓨팅 환경은 유선 네트워크 환경과는 달리 전송 매체의 최대 전송양이 고정되어 있지 않고 기후, 간섭, 지형 등에 따라 유동적이며 이동 호스트는 전력 사용 등 여러 가지 면에서 매우 제한적인 능력을 갖고 있다. 그러므로 이동 환경에서의 서버 선택은 통신 제어를 위한 오버헤드 메시지를 줄이고 시스템 부하가 많은 부분들은 되도록 고정 호스트에서 처리 할 수 있도록 고려하여 설계해야 한다.

3.1 무선망의 제약 조건 및 설계 목표

3.1.1 제약 조건

무선망은 유선망에 비해 상대적으로 신뢰성이 떨어지고 낮은 대여폭을 지원한다.

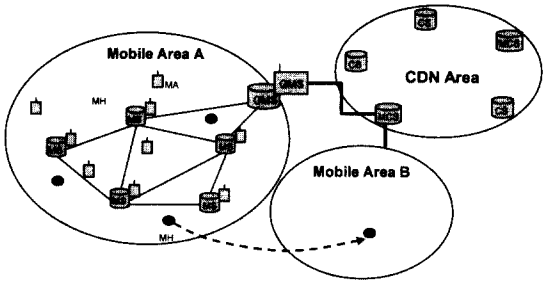
이동 호스트는 제한된 자원을 갖고 있으므로 복잡한 연산을 수행하기가 곤란하다.

3.1.2 설계 목표

무선 인터페이스 상의 통화량과 지연을 감소시켜야 한다. 고정 호스트에서 대부분의 일을 수행함으로써 이동 호스트의 작업을 줄여야 한다.

3.2 네트워크 모델 및 각 노드의 역할

본 네트워크 모델은 (그림 1)과 같은 구조를 가지며 다음을 가정한다.



(그림 1) 네트워크 모델

- 콘텐츠 서버 집합의 부분 집합으로 이동 클라이언트를 위한 콘텐츠를 보관하고 있는 서버 집합이 존재한다.
  콘텐츠 서버 집합을 S_{CDN}라 하고, 유선 호스트를 위한 콘텐츠를 저장하고 있는 서버를 CS, 이동 호스트를 위한 콘텐츠를 저장하고 있는 서버를 MCS라 할 때,
  다음 조건을 만족한다.

$$S_{CDN} = \{ CS_1, CS_2, \cdots, CS_{n-1}, CS_n, MCS_1, MCS_2, \cdots, MCS_{n-1}, MCS_n \}$$

- 각 MCS 서버는 자신의 부분 집합으로 게이트웨이 이동 서버(GMS)를 연결하고 있으며 콘텐츠 동기화, 라우팅 등 CDN 시스템의 모든 동작에 대해 자신과 연결된 집합에 그대로 반영한다.

$$MCS_i = \{GMS_1, \cdots, GMS_{k-1}, GMS_k, 1 \le k < n \}$$

- CDN 네트워크 상에서는 콘텐츠의 동기화, 라우팅, 요청 전달이 적절히 이루어지고 있으며 클라이언트의 빠른 접근 및 서비스 질의 향상을 보장하고 있다.

네트워크 모델의 각 노드들의 기능은 다음과 같다.

- 이동 콘텐츠 서버(MCS : Mobile Contents Server)
  콘텐츠 서버 중 Mobile 클라이언트를 위한 콘텐츠를 포함하고 있는 서버
- 게이트웨이 이동 서버(GMS : Gateway Mobile Server)
  이동 콘텐츠 서버와 일관된 콘텐츠를 보관하고 있는 고정된 데이터베이스 서버로 상대적으로 용량이 적은 MS를 지원한다.
- 이동 서버(MS : Mobile Server)
  GMS에서 프리패치 한 콘텐츠와 MS에서 캐싱한 콘텐츠를 보관하고 있는 고정된 데이터베이스 서버
- 게이트웨이 이동 에이전트(GMA : Gateway Mobile Agent)
  유선 네트워크에 유선으로 연결되어 있으면서 무선을 지원하는 고정된 지역 라우터로 지역 내의 이동 에이전트와 무선 통신 채널을 통해 통신하고 관리한다.
- 이동 에이전트(MA : Mobile Agent)
  이동 호스트와 통신할 수 있는 인터페이스를 가지고 있는 고정 라우터로 호스트의 이동성을 지원한다.
- 이동 호스트(MH : Mobile Host)
  무선 통신 채널을 통해 하나의 MA와 통신하며 이동성을 갖고 있다.

4. 제안하는 서버 선택 알고리즘

4.1 개 요

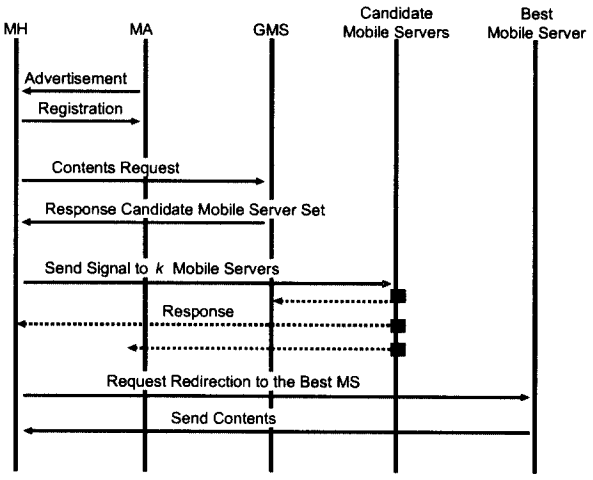
가장 적절한 서버라는 것은 서버의 부하가 적으며 이동 호스트와 지리적으로 가깝고 이동 호스트와의 사이에 트래픽이 적은 서버라고 할 수 있다.

본 연구에서는 게이트웨이 이동 서버가 관할하고 있는 영역 내 이동 서버 등의 부하 정보를 주기적으로 갱신하고 있다가 이동 호스트의 요청에 대해 전체 서버 중 부하가 적은 순으로 k개의 서버를 선택하여 이동 호스트에게 보내면 이동 호스트가 k개의 서버로 신호 메시지를 보내어 가장 빠른 응답을 한 서버로 요청하는 메커니즘을 제안한다. 이것은 각 이동 서버가 그들의 부하 정보를 주기적으로 게이트웨이 이동 에이전트에게 보냄으로써 구현한다. 여기에서도 부하 정보 수집을 위한 트래픽이나 이동 호스트의 신호 메시지 전송을 위한 오버헤드가 발생하지만 게이트웨이 이동 서버를 중심으로 하여 일정한 지역 내에서의 전송이므로 과다한 오버헤드는 발생하지 않는다고 할 수 있다.

이러한 메커니즘은 서버 측면과 클라이언트 측면에서의 선택을 모두 고려한 것으로 부하가 적고 트래픽이 한가하며 거리가 가까운 서버를 선택하여 서비스 요청을 함으로써 빠른 응답 시간을 기대할 수 있다.

4.2 서버 선택 시나리오

이동 호스트의 요청 전달 과정은 (그림 2)에서와 같이 다음 단계를 따른다.



(그림 2) 서버 선택 과정

① MA는 에이전트 광고 메시지를 주기적으로 방송한다.
② MH는 MA에 등록하고 GMA, GMS의 주소를 응답 메시지로 받는다.
③ MH가 GMS로 서비스 요청을 한다.
④ GMS가 영역 내에서 부하가 적은 이동 서버 k개를 후보 서버 집합으로 하여 응답한다.
⑤ 이동 호스트가 다시 k개의 서버로 신호를 전달한다.
⑥ 가장 먼저 응답이 온 서버에게로 콘텐츠 요청을 전달

하여 서비스를 받는다.

4.3 MH의 이동에 따른 서버 선택의 변화

이동 호스트가 이동한 경우 다음 2가지 경우를 고려할 수 있다.

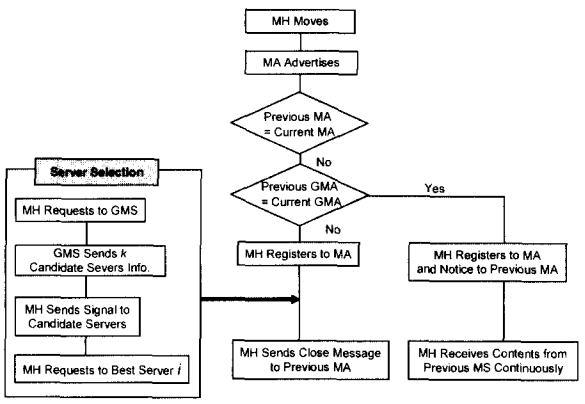
4.3.1 MH가 다른 도메인으로 이동한 경우

MH가 현재 도메인에 처음 진입한 경우이거나 다른 도메인에서 이동해 온 경우 MH는 주기적인 MA의 광고 메시지를 통해 자신이 가지고 있는 정보 중 GMA 주소가 NULL이거나 광고 메시지의 GMA와 다를 경우 이동 또는 처음 접속 시도 상태임을 알고 MA에게 자신의 등록 메시지를 보낸다. 등록 메시지를 받은 MA는 새로운 MH가 진입한 것으로 보고 GMA와 GMS의 주소를 응답 메시지로 보낸다. MH는 4.2의 서버 선택 시나리오에 의해 새로운 서버를 선택하여 최근 수신 번호 이후의 패킷을 요청하고 이동전의 MA에게 이동 사실을 알려 전송을 중단한다.

4.3.2 MH가 같은 도메인 내에서 이동한 경우

MH는 MA의 광고 메시지를 통해 이동을 인지하지만 자신에게 기록된 GMA 주소가 광고 메시지의 GMA 주소와 같은 것으로 보아 같은 도메인 내의 이동임을 안다. 새로운 MA에 등록하고 종전 MA에게 이동 사실을 알리고 새로운 서버 선택의 절차 없이 이전의 MS로부터 지속적으로 서비스를 받는다.

이와 같은 절차를 (그림 3)에서 나타내었다.



(그림 3) 이동 후의 변화

4.4 서버 선택 알고리즘

기존 유선 네트워크에서의 요청 전달을 위한 서버 선택은 주로 클라이언트 측에서 서버와의 거리와 서버의 부하를 고려한 알고리즘을 사용한다.

그러나 이동 환경에서는 이동 호스트의 자원이 충분하지 못하므로 이동 호스트에서의 지연을 줄이기 위해 부하 측정 작업은 서버에게 맡기고 최대한 이동 호스트의 작업을 줄여 효율적인 요청 전달을 할 수 있는 알고리즘을 제안한다. 이동 서버들은 그들의 부하 정보를 주기적으로 게이트웨이 이동 에이전트에게 보내고 게이트웨이 이동 에이전트는 이동 서버의 부하 정보 테이블을 유지하여 부하가 적은 서버를 선택하는데 사용한다.

4.4.1 이동 게이트웨이 서버의 서버 선택 알고리즘

게이트웨이 이동 에이전트가 이동 서버들로부터 부하 정보에 관한 피드백을 주기적으로 받아, 사용 가능하며(active) 부하가 적은 서버 k개를 선택한다. 이때 서버의 부하 메트릭으로는 CPU 점유율, 대기 중인 작업 수, 연결된 서비스 수, 기존 부하 정보가 고려된다.

Server-side Server Selection Algorithm

Select Candidate Servers

candidate_servers(Sc, S) {

for each server s_i in server set S,
$Load_i$ = MeasureLoad(i) ;
sort Load ;

$S_{Candidate}$ = {$Sc_1, \cdots, Sc_k$} //least loaded server of top k servers ;
Candidate Server Set ← $S_{candidate}$;

MeasureLoad(i)
$Load_{current} = 1/(CPU_{load} + Buffer_{job} + Connection_{cnt})$
$Load_{rate} = k \cdot Load_{previous} + (1-k) \cdot Load_{current}$ for $k < 1$
$Load_{result} = Load_{current} \times Load_{rate}$
$Load_{previous} = Load_{result}$
return $Load_{result}$

}

4.4.2 이동 호스트의 서버 선택 알고리즘

이동 호스트는 게이트웨이 이동 에이전트가 부하에 기반하여 선택한 k개의 후보 서버에 신호 메시지를 보내어 응답시간이 가장 빠른 서버를 선택하여 요청을 전달한다. 이때, 모든 서버들의 응답을 기다리지 않는다. 응답시간은 거리, 대역폭, 트래픽에 의해 영향을 받는다.

Client-side Server Selection Algorithm

Select Best Server

Best_server (s, Sc) {

for each candidate server i ∈ Sc //Candidate Servers
ping to i ;
if response i arrived
forward request to server i ;

5. 성능분석

본 연구에서는 이동 호스트에서의 처리를 위한 제어 메시지와 저장 크기를 유선 환경에서의 기존 ping-random[8], RR-DNS[3] 기법과 비교하여, 본 연구가 이동 환경에 적합한 가벼운 서버 선택 기법을 사용하고 있음을 보이고, 서버 선택 시 사용되는 메트릭으로 대역폭과 전송지연을 선택함으로써 파일 전송시간이 기존 연구와 비교하여 개선됨을 보인다.

5.1 실험환경

100개의 콘텐츠 서버를 대상으로 특정 이동 호스트에서 콘텐츠 서버를 선택하고자 한다. 이때 콘텐츠 서버부터 이동 호스트까지의 종단간의 전송지연은 실제 ping 테스트로 100여개 대학 사이트를 상대로 측정한 지연시간인 0.92[10]를 $\frac{1}{\lambda}$로 하는 지수분포에 의해 임의로 생성하였다. 지연시간 생성 식은 다음 식 (5.1)과 같다.

$$\frac{-lnR}{\lambda}, \quad 0 < R < 1 \tag{5.1}$$

5.2 제어 메시지와 저장 크기 비교

전체 콘텐츠 서버 집합(S)의 크기는 $n(S)$이고, 이들 중 후보 서버로 선택된 서버의 집합(Sc)의 크기는 $n(Sc)$이다. 이때 $n(S) = N$, $n(Sc) = K$라고 하면, 집합 Sc는 전체 집합 S의 부분 집합이므로 $N > K$이고, 후보로 선택된 서버 집합이므로 $N \gg K$이다. <표 1>은 기존 연구와[3, 8] 제안한 서버선택 기법을 서버선택을 위해 교환되는 제어 메시지의 크기와 이동 호스트에게 필요한 저장 공간을 비교한 것이다.

서버선택을 위해서는 ping-random[8]의 경우는 전체 서버 집합 S에게 모두 ping 메시지를 보내고 받아야 하므로, 메시지 교환을 위한 시간 복잡도는 O(N)이고, RR-DNS와 제안한 서버 선택 기법의 경우 GMA와 같은 역할을 하는 에이전트에게 한 개의 콘텐츠 서버를 할당받거나, 후보 집합 Sc를 받아서 해당 집합의 콘텐츠 서버들과만 ping메시지를 교환하기 때문에 시간 복잡도는 각각 $O(1)$과 O(K)이다.

이동 호스트에게 요구되는 저장소의 크기는 기존연구 ping-random[8]은 전체 콘텐츠 서버 집합 S에서 후보 집합 Sc를 5개를 저장하고 있어야 하므로 $\Theta(5K)$이며, RR-DNS[3]는 에이전트부터 받은 한 개의 콘텐츠 서버의 정보만을 갖고 있으면 된다. 제안한 서버선택 기법의 경우는 에이전트로부터 전송받은 후보서버 k개만 저장하고 있으면 되므로 $\Theta(K)$이다.

〈표 1〉 제어 메시지와 저장 크기 비교

	서버선택을 위해 교환되는 메시지의 크기	이동 호스트에게 필요한 저장크기	비 고
ping-random[8]	$O(N)$	$\Theta(5K)$	단말에서 콘텐츠서버까지의 전송지연시간을 고려
RR-DNS[3]	$O(1)$	$\Theta(1)$	GMA에서 무작위로 단말에게 콘텐츠서버를 할당
Proposed	$O(K)$	$\Theta(K)$	단말에서 콘텐츠서버까지의 전송지연 및 수신률을 고려

5.3 전송시간 비교

실험환경에서 생성된 임의의 종단 간 전송지연과 수신대역폭을 가지고 서버를 선택하고 10M 파일을 모두 전송받기까지의 시간을 측정하였다. 사용한 실험변수는 <표 2>와 같으며, 식 (5.2)는 ping-random[8]에서의 데이터 전송이 모두 완료될 때까지 걸리는 시간(T_{pr})으로 전체 콘텐츠 서버에게 ping 메시지를 보내고 전체 $5K$ 만큼의 집합이 생성될 때까지 응답 메시지를 수신 받는 시간($\max[T_{ps_i}]_{i \in U}$)과 5개의 Sc 집합에서 임의로 선택된 콘텐츠 서버에게 데이터 서비스를 요구하고 응답을 받는 시간($T_{selServer}$), 그리고 콘텐츠를 모두 전송받기까지의 시간($T_{download}$)으로 이루어진다. 이때, ping-random기법은 콘텐츠 서버의 CPU load를 고려하지 않기 때문에 서비스 거부가 발생할 수 있다.

식 (5.3)은 제안한 기법에서 콘텐츠를 모두 전송받기까지의 시간으로 식 (5.2)와 유사하나, ping 메시지 교환시간의 경우, 한 개의 후보 콘텐츠 집합 Sc로 ping메시지를 보내고 가장 빠른 응답 메시지가 올 때까지만 기다리는 시간($\min[T_{ps_i}]_{i \in U}$)으로 이루어져 있다. 마지막으로 식 (5.4)는 RR-DNS[3] 기법을 적용한 경우로, 선택된 서버로의 서비스를 요구하는 시간과 콘텐츠 전송이 완료되는 시간과 함께 이동 호스트의 해당 에이전트에게 서비스 받을 서버 선택을 요구하고 응답받는 시간($T_{reqServer}$)이 필요하다.

실험 결과, (그림 4)에서 보인 것과 같이 제안한 방법에 의한 전송시간은 평균은 40ms로써 RR-DNS와 ping-random에 비해 빠른 것으로 나타났다.

〈표 2〉 실험변수

변 수	설 명
T_{PS_i}	이동 호스트에서 콘텐츠 서버 PS_i까지의 ping 메시지 교환시간
$T_{selServer}$	선택된 콘텐츠 서버에게 서비스를 요구하는 메시지 교환시간
$T_{download}$	콘텐츠 서버가 선택된 후, 이동 호스트가 파일을 모두 전송받는데 걸리는 시간
T_{cs}	선택된 콘텐츠 서버에서 이동 호스트까지의 기본 RTT 시간
$T_{reqServer}$	RR-DNS에서 에이전트에게 이동 호스트가 콘텐츠 서버 지정을 위한 요구 메시지를 교환하는 시간

$$T_{pr} = \max[T_{ps_i}]_{i \in U} + (1 + p(1-p)^{l-1})T_{selServer} + T_{download} \tag{5.2}$$

$$T_{proposal} = \min[T_{ps_i}]_{i \in PS} + T_{selServer} + T_{download} \tag{5.3}$$

$$T_{RR-DNS} = T_{reqServer} + T_{selServer} + T_{download} \tag{5.4}$$

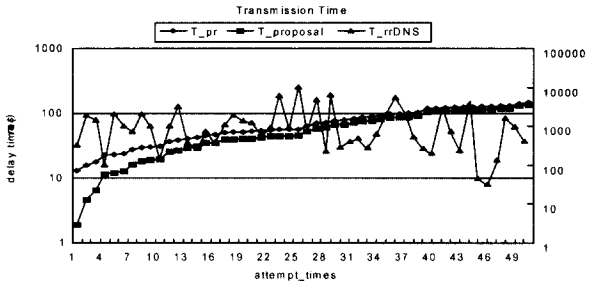
$$T_{download} = d\left(T_{cs} \sum_{i=1}^{n} \frac{a_i k - m(\frac{k}{m})^{a_i}}{a_i - 1} + \sum_{i=1}^{n-1} \frac{-ln(1-r_i)}{\lambda} \right),$$

$$a_i > 1,\ 0 < r_i < 1,\ d > 0 \tag{5.5}$$

무선에서의 패킷 트래픽의 특성을 고려한 데이터 전송시간은 식 (5.5)와 같다. 데이터 전송에 걸리는 총 전송시간은 패킷콜 사이즈에 따른 총 전송시간과 그 사이의 읽는 시간으로 나눌 수 있다. 이때, 패킷콜 사이즈는 Pareto with cutoff 프로세스를 따르며, 읽는 시간은 포아송 프로세스의 대기시간[9]을 적용할 수 있다. 이때 n은 전체 패킷콜 사이

즈의 크기가 전체 보내야할 데이터의 크기와 각 패킷의 오버헤드의 합과 같아질 만큼의 개수이다. m은 하나의 패킷콜 사이즈의 최대크기를 의미한다. d는 네트워크 상황의 유동성을 반영하기 위한 변수이다.

패킷콜 사이즈는 웹 트래픽의 특성을 반영하여 [4.5KB, 2MB] 사이의 값들로서 shape 파라미터(α_i)의 값에 따라 유동적으로 반영하며, 읽는 시간은 일반적으로 알려진 웹 트래픽의 읽는 시간 5초를 평균으로 하여 임의로 읽는 시간을 생성하였다[9]. k는 [1, 2]의 범위 값을 가지나 일반적인 값 1.1[9]로 산정한다. m은 패킷콜 사이즈의 최대 크기로 여기에서는 2MB으로 볼 수 있다.



(그림 4) 전송시간 비교

6. 결 론

본 연구는 유선 환경에서 그 성능을 인정받고 있는 콘텐츠 분산 네트워크를 무선 환경에 적용하여 이동 호스트를 위한 빠른 콘텐츠 전송을 위하여 다수의 콘텐츠 서버 중 적절한 하나의 서버를 선택하는 문제를 다루었다. 유선 환경에서는 ping-random에 의한 서버 선택이 좋은 성능을 나타내었지만 무선 환경에서는 메시지 교환과 데이터 저장 면에서 오버헤드가 크기 때문에 그대로 적용하기에 부적합하다.

그러므로 무선 환경에 적합한 게이트웨이 이동 에이전트와 이동 호스트가 협력하는 새로운 서버 선택 기법을 제안하여 제어 메시지와 저장 크기에 있어서 효율성을 확보하고 전송 속도도 더욱 개선하였다.

이러한 연구는 이동 환경에서의 빠른 전송을 가능하게 함으로써 유비쿼터스 컴퓨팅 라이프 실현에 기여할 수 있을 것이다.

참 고 문 헌

[1] http://www.ukoreaforum.or.kr.

[2] G. Peng, "CDN : Content Distribution Network," in Stony Brook University Tech. Reports, TR-125, Jan., 2003.

[3] T. Brisco, "RFC1794 : DNS support for load balancing," Apr., 1995.

[4] M. Sayal, Y. Breitbart, P. Scheuermann and R. Vingralek, "Selection Algorithms for Replicated Web Servers," In Proceedings of the Workshop on Internet Server Performance, 1998.

[5] RL Carter and ME Crovella, "Server Selection Using Dynamic Path Characterization in Wide-Area Networks," In Proceedings of INFOCOM'97, Apr., 1997.

[6] K. Hanna, N. Natarajan and B. Levine, "Evaluation of a Novel Two-Step Server Selection Metric," Network Protocols Ninth International Conference on ICNP'01, pp. 290-300, 2001.

[7] K. Hanna, N. Natarajan and B. Levine, "A Performance Study of Top Mirror Servers," UMass Tech Report 01-10, Jan., 2001.

[8] S. Dykes, K Robbins and C. Jeffery, "An Empirical Evaluation of Client-side Server Selection Algorithms," In Proceedings of INFOCOM'00, pp.1361-1370, Mar., 2000.

[9] Toni Janevski, "Traffic alnalysis and design of wireless IP networks. Artech House Publishers," pp.183-186, 2004.

[10] 김선호, 윤미연, 신용태 "시간 복잡도를 개선한 웹 서버 배치 알고리즘", 정보처리학회논문지C, 제11-C권 제3호, pp.345-352, 2004.

김 선 호

e-mail : shkim98@dongduk.ac.kr

1987년 이화여자대학교 사범대학 수학교육전공 학사

1992년 이화여자대학교 교육대학원 전자계산교육전공 석사

2004년 숭실대학교 컴퓨터학과 컴퓨터통신 전공 박사

1987년~1989년 대우전자부품(주) 전산실

1990년~1993년 한국생산성본부 정보화사업부

1998년~현재 동덕여자대학교 정보과학대학 강의전임교수

관심분야 : Internet Protocol, Mobile IP, CDN, DRM

윤 미 연

e-mail : myyoon@cherry.ssu.ac.kr

2000년 가톨릭대학교 수학과/컴퓨터학과 학사

2002년 숭실대학교 컴퓨터학과 공학석사

2002년~현재 숭실대학교 컴퓨터학과 박사과정

관심분야 : 멀티캐스트, 실시간프로토콜, Wireless Communication, 정보보호, CDN

신 용 태

e-mail : shin@comp.ssu.ac.kr

1985년 한양대학교 산업공학과 학사

1990년 Univ. of Iowa 컴퓨터학과 석사

1994년 Univ. of Iowa 컴퓨터학과 박사

1994년~1995년 Michigan State Univ. 전산학과 객원교수

1995년~현재 숭실대학교 컴퓨터학부 부교수

관심분야 : 멀티캐스트, 그룹통신, 인터넷 보안, 이동 인터넷 통신