

무선 이동 웹 서비스를 위한 분산 프록시 시스템의 설계[†]

김기섭^o, 이혁준, 정광수^{*}
광운대학교 컴퓨터공학과, 전자통신공학과^{*}
askstar@explore.kwangwoon.ac.kr, {hlee, kchung}@daisy.kwangwoon.ac.kr

Design of a Distributed Proxy System for Wireless Mobile Web Service

Kisup Kim^o, Hyukjoon Lee, and Kwangsue Chung^{*}
Dept. of Computer Engineering, Kwangwoon University
Dept. of Electronic Communications Engineering, Kwangwoon University^{*}

요 약

변환코딩 프록시 서버는 무선 이동 데이터 망을 통한 웹 검색 시의 응답지연을 줄이기 위하여 널리 채택되고 있는 방법 중 하나이다. 그러나, 프록시 서버로 네트워크 트래픽이 집중되는 병목현상과 프록시 서버의 동작 중단에 의한 전체 서비스의 중단이 이 방법의 최대 단점이다. 본 논문에서는 이러한 문제를 해결하기 위하여 분산 프록시 서버 시스템을 제안한다. 제안하는 시스템은 지역적으로 분산된 다수의 프록시 서버로 구성되어 있으며, 각 프록시 서버가 담당 지역 내 클라이언트들의 요청을 처리하도록 함으로써 작업량을 분담시킨다. 또한 클라이언트의 이동에 따른 프록시 서버간의 핸드오프를 지원하기 위한 새로운 프록시 서버 핸드오프 메시지 프로토콜을 제안한다.

1. 서론

변환코딩 프록시 서버(transcoding proxy server)는 월드 와이드 웹 검색 시의 응답속도를 향상시키는 방법으로 무선 이동 데이터 서비스에서 널리 사용되고 있다. 프록시 서버의 주요 기능은 캐싱과 변환코딩이다. 변환코딩은 무선 링크상에 전송될 데이터를 손실 압축하여 전송량을 줄임으로써 네트워크의 체감속도를 빠르게 하는 방법이다[1, 2, 3]. 프록시 서버 방식의 가장 큰 단점은 트래픽의 집중현상과 프록시 서버에 문제가 발생할 경우 전체 네트워크의 서비스가 중단되는 현상이다.

무선 이동 웹 서비스를 위한 분산 시스템을 적용한 연구로는 다음과 같은 것들이 있다. [4]는 단일 웹 서버로 집중되는 네트워크 트래픽을 분산시키기 위해 웹 서버상의 특정 파일에 대한 클라이언트의 접근(access)이 일정 한계치를 초과할 때, 웹 서버는 네트워크 대역폭을 최소로 사용하는 캐시 서버를 선택하여, 특정 파일을 캐시하는 복제(replication) 서비스를 제공한다. 선정되는 캐시서버는 사용자로부터 거리상 가까운 곳에 있는 캐시 서버이다. 계속적으로 특정 지역에서 다른 파일에 대한 접근이 한계치를 넘으면, 네트워크 대역폭을 최소로 사용할 수 있는 캐시 서버를 추가로 결정하여 파일을 복제한다. 무선 이동 환경에서 클라이언트의 이동성 지원에 관한 연구로서는 CORBA 기반 응용 프로그램을 위한 이동성 지원 게이트웨이를 구현한 사례가 있는데, 이동성 지원 게이트웨이 간의 핸드오프 과정에 발생하는 데이터 전송 중단되는 문제를 전송계층에서 메시지 기반의 프로토콜을 적용하여 방지하였다[5].

본 논문에서 제안하는 분산 프록시 서버 시스템은 프록시 서버의 분산화에 대한 최초의 연구로서, 지역적으로 분산된 다수의 프록시 서버로 구성되어 있으며, 각 프록시 서버가 담당 지역 내 클라이언트들의 요청을 처리하도록 함으로써 작업량을 분담시킨다. 또한 이동호스트의

이동으로 인한 프록시 서버 간 핸드오프 시 발생하는 웹 데이터의 전송 중단을 프록시 서버 핸드오프 프로토콜을 적용함으로써 해결하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 분산 프록시 서버 시스템 구성과 기능에 대해 살펴보고, 3 장에서는 분산 프록시 서버 시스템에 적용될 프록시 서버 핸드오프 프로토콜에 대해 기술한다. 4 장에서는 구현, 5 장에서 결론을 맺는다.

2. 분산 프록시 서버 시스템

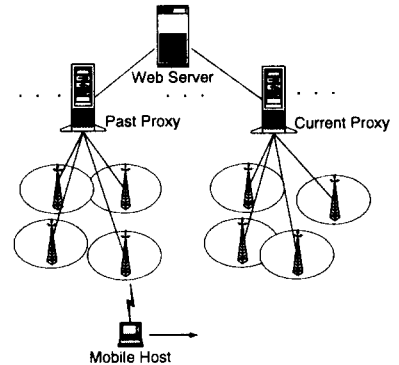


그림 1 분산 프록시 서버 시스템

이동 호스트의 이동으로 인해 서로 다른 프록시들에 의해 서비스 되고 있는 셀들간에 핸드오프가 발생하게 되면 이동 호스트는 기존의 프록시 서버와의 연결을 끊고 새로운 프록시 서버로 연결을 설정해야 한다. 웹 데이터 전송 중에 새로운 프록시로 연결을 설정하게 되면, 데이터 전송이 중단 되는 현상이 발생하게 된다. 분산 프록시 서버 시스템

[†] 본 연구는 2000년도 대학기초 연구지원사업의 연구 결과임.

은 웹 데이터 전송 중단 문제를 본 논문에서 제안하는 프록시 서버 간 핸드오프 처리하여 클라이언트에게 이음새 없는 웹 서비스를 제공할 수 있다. 프록시 서버 핸드오프 처리 과정은 이동 호스트와 현재 프록시 서버, 현재 프록시 서버와 과거 프록시 서버, 그리고 현재 프록시 서버와 웹 서버 간의 메시지 교환을 통한 상호작용으로 이루어진다. 논문에서 언급되는 과거 프록시 서버는 프록시 서버 사이의 핸드오프 이전에 이동 호스트에게 서비스하던 프록시 서버이며, 현재 프록시 서버는 프록시 서버 간의 핸드오프 이후 이동 호스트가 새롭게 연결을 설정한 프록시 서버이다.

2.1 분산 프록시 서버

분산 프록시 서버 시스템을 구성하는 기능 모듈은 크게 분산 프록시 서버와 클라이언트 상에 위치하는 클라이언트 에이전트로 나누어질 수 있다. 분산 프록시 서버는 일반적인 웹 프록시 서버에 변환코딩과 프록시 서버 핸드오프 처리 기능을 추가한 프록시 서버이다. 분산 프록시 서버는 HTTP[6] 처리기, 요청 메시지 처리기, 정제 데이터 처리기, 정제 데이터 전송기, 데이터 정제기, 캐쉬 관리기 그리고 파일 리스트 관리기로 구성되어 있다. 요청 메시지 처리기(request message handler)는 요청 메시지들을 HTTP 처리모듈과 핸드오프 처리 모듈로 분기 시킨다. 정제 데이터 처리기(distilled data handler)는 과거의 프록시 서버로부터 정제데이터를 전송 받아 캐시에 저장하고 정제 데이터 전송기(distilled data dispatcher)는 과거의 프록시 서버와 웹 서버에게 요청할 파일리스트를 구성하여 정제 데이터 처리기와 HTTP 처리기로 전달한다. HTTP 처리기(HTTP handler)는 HTTP 요청 메시지를 생성하여 웹 서버로 전송하고, HTTP 응답 메시지 즉 HTML 문서와 영상 데이터들을 수신하여 데이터 정제기로 전달한다. 데이터 정제기(distiller)는 웹 서버로부터 받은 객체들을 손실 압축하며, 정제 데이터 전송기는 클라이언트로 정제 데이터를 전송한다.

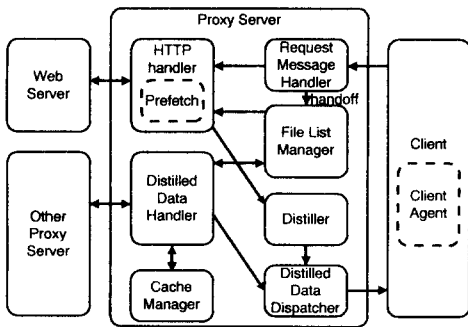


그림 2 프록시 서버 구조

2.2 클라이언트 에이전트

클라이언트 에이전트내 핸드 오프 처리 모듈은 위치 정보 관리기, 프록시 서버 연결 관리기, 핸드오프 요청 메시지 생성기, 그리고 파일 처리기로 구성되어 있다. 기지국 핸드오프가 발생되었을 때 무선망 운영 센터가 위치 정보 관리기에게 기지국 ID 정보를 보내준다고 가정한다. 프록시 서버 연결 관리기는 수신된 기지국 ID 정보를 통해 프록시 서버 핸드오프 발생여부를 확인 후, 새로운 프록시 서버로 TCP 연결을 설정한다. 핸드오프 요청 메시지 생성기는 프록시 서버에게 핸드오프 처리 개시를 알려주는 핸드오프 요청 메시지를 생성한다. 생성된 메시지는 TCP 연결을 통해 새로운 프록시 서버로 보내진다. 파일 처리기는

프록시 서버로부터 파일들을 받아 브라우저로 전달한다. 이상 언급된 프록시 서버와 클라이언트 에이전트의 구성 요소들이 이동 호스트에게 이음새 없는 웹 서비스를 제공하기 위해 프록시 서버 핸드오프 처리를 상호 협력하여 수행한다.

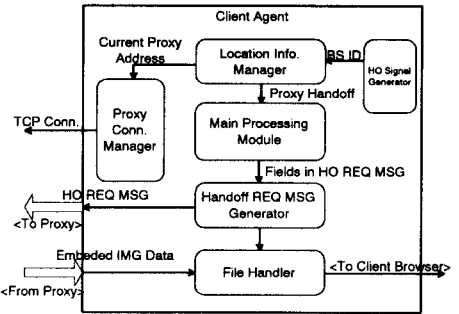


그림 3 클라이언트 에이전트

3. 프록시 서버 핸드오프 프로토콜

3.1 프록시 서버 핸드오프 처리

프록시 서버 핸드오프 처리는 크게 세 단계로 프록시 서버에게 핸드오프 처리를 요청하는 단계, 프록시 서버간 동기화 단계, 그리고 웹 서버와 프록시 서버간의 핸드오프 처리단계로 구성되어 있다. 프록시 서버 핸드오프가 발생되었을 때, 클라이언트는 새로운 프록시 서버로 연결을 설정한 후, 프록시 서버로 핸드오프 요청 메시지를 전송한다. 핸드오프 요청 메시지를 수신한 프록시 서버는 핸드 오프 처리를 시작한다. 프록시 서버는 웹 페이지 내에 포함된 파일들 중에서 받지 못한 파일들의 리스트를 구성하고, 이 파일 리스트는 과거 프록시에게 요청할 파일 목록으로 다음 단계에서 정제 데이터 메시지에 포함된다.

두 번째 과정은 프록시 서버간 동기화 과정으로서, 페이지 내의 영상 데이터를 웹 서버로부터 반입과 프록시 서버에서 데이터 정제 처리의 반복을 피하기 위해 프록시 서버는 핸드오프 이전에 정제 처리가 완료된 데이터를 과거의 프록시 서버에게 요청한다. 과거의 프록시 서버는 해당되는 정제 데이터를 가지고 있을 경우 정제 데이터 메시지를 만들어 현재 프록시 서버로 전송한다. 정제 데이터 메시지를 수신한 현재 프록시 서버는 정제 데이터를 추출하여 캐시에 저장하고 이동 호스트로 전송한다.

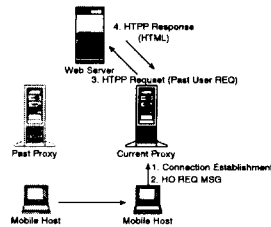


그림 4 초기화 단계

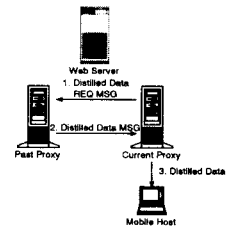


그림 5 프록시 서버간 동기화

마지막 단계로서, 현재 프록시 서버는 이동 호스트와 과거의 프록시 서버로부터 받지 못한 웹 페이지내의 나머지 파일들을 웹 서버로 요청한다. 이 요청에 따른 응답인 HTML 파일과 영상 파일은 정제되어 이

동 호스트로 전송한다. 이와 같은 과정을 통해 이동 호스트에게 이용세 없는 무선 이동 웹 서비스를 제공할 수 있다.

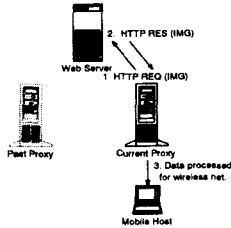


그림 6 웹 서버와 프록시 서버간의 핸드오프 처리

3.2 메시지 포맷

프록시 서버 핸드오프 프로토콜의 구성 메시지에는 핸드오프 요청 메시지, 정제 데이터 요청 메시지와 정제 데이터 메시지가 있다. 핸드오프 요청 메시지는 이동 호스트가 프록시 서버에게 핸드오프 처리를 요청할 때 사용된다. 핸드오프 요청 메시지를 이루는 필드들은 다음과 같다. "Total Size"는 전체 메시지의 바이트 수이다. "User Request"는 프록시 서버 핸드오프 이전에 사용자가 요구한 웹 페이지의 HTTP 요청 메시지이다. "List of Files Received at MH"는 "User Request"의 페이지를 구성하는 파일들 중 핸드오프 이전에 이동 호스트가 과거 프록시 서버로부터 수신한 객체들의 리스트이다. 파일 전송 중에 프록시 서버와의 연결이 끊어져 파일 일부분을 수신한 경우가 발생하기도 한다. 핸드오프 이전까지 수신된 바이트 정보는 "Received Bytes"에 기록된다. 이러한 경우에는 프록시 서버는 파일의 부분전송을 요청하는 partial GET 를 이용해서 수신 못한 파일의 일부분을 전송 받을 수 있다.

| | |
|------------------------------|-------------------------------|
| Total Size (2 byte) | Past Proxy IP Address (4byte) |
| User Request | |
| List of Files Received at MH | Received Bytes (4 byte) |

그림 7 핸드오프 요청 메시지

현재 프록시 서버와 과거 프록시 서버 사이에서 교환되는 메시지는 정제 데이터 요청 메시지와 정제 데이터 메시지가 있다. 정제 데이터 요청 메시지는 과거의 프록시 서버로 정제 데이터를 요구하는 메시지이다. 이 메시지는 과거 프록시 서버로 요청할 파일명들과 해당 파일들의 만료시간을 기록하는 "File Name"과 "Expired Time"의 필드들을 포함하고 있다. 정제 데이터 메시지는 정제 데이터 요청 메시지의 응답으로서 과거 프록시 서버에 의해 생성되어, 현재 프록시 서버로 전송된다. 정제 데이터 메시지는 "File Name"에는 정제된 파일의 이름, "Distilled Data"에는 정제된 파일의 데이터를 포함하고 있다.

| | |
|---------------------|-----------------------|
| Total Size (2 byte) | No. of Files (1byte) |
| File Name 1 | Expired Time 1(8byte) |
| . | . |
| File Name n | Expired Time n(8byte) |

그림 8 정제 데이터 요청 메시지

| Total Size (4byte) | File Name | Size of Distilled Data (4byte) |
|--------------------|-----------|--------------------------------|
| Distilled Data | | |

그림 9 정제 데이터 메시지

4. 구현 및 실험

분산 프록시 서버 시스템은 프록시 서버와 클라이언트 에이전트로 나뉘어 구현되었다. 프록시 서버는 Apache[7] 서버를 기반으로 UNIX 환경에서 GNU C 를 사용하여 구현하였다. 클라이언트 에이전트는 윈도우 기반의 Microsoft Visual C++에서 구현되었으며, 브라우저와 함께 동일 클라이언트에서 동작한다. 브라우저의 HTTP 요구 처리 부분은 다중 쓰레드 방식을 적용하여 구현하였다.

무선 이동 환경 하에서 분산 프록시 서버 시스템의 성능 측정을 위한 실험환경은 다음과 같다. 무선 인터페이스는 CDMA 단말기를 사용하며, 프록시 서버 핸드오프 상황은 핸드오프 트리거 모듈로 발생시킨다. 이동 호스트는 노트북 PC 를 사용하고, 프록시 서버는 솔라리스 기반의 UNIX 시스템에서 구동된다. 분산 프록시 서버 시스템과 단일 프록시 서버 시스템 간 성능 비교 척도는 웹 응답 시간을 선정하였다. 네트워크 상황의 변화가 응답 시간에 영향을 미치는 것을 최소화하기 위해 선정된 실험 페이지를 대상으로 20 회씩 웹 응답 시간을 반복 측정하여 그 평균치를 산출할 것이다.

5. 결론

기존 단일 프록시 서버 시스템은 다수의 클라이언트와 서버의 사이에 위치하여 사용자의 요청을 처리해 주고 있어, 클라이언트의 요청과 웹 서버의 응답이 프록시 서버로 집중된다. 본 논문에서 단일 프록시 서버로 집중되는 네트워크 트래픽을 나누어 처리하기 위한 분산 프록시 서버 시스템을 제안하였다. 클라이언트의 이동으로 웹 전송이 중단되는 문제점을 해결하기 위해 프록시 서버 핸드오프 프로토콜을 개발하였고, 프록시 서버 핸드오프 프로토콜 처리 모듈들은 프록시 서버와 클라이언트 에이전트에 추가 구현되어 있다.

향후 계획으로는 프록시 서버들이 지역적으로 배치되어 있으므로, 지능형 선반입을 통한 위치 종속 서비스(location-dependent service)를 추가 구현할 예정이다[8,9].

참고 문헌

- [1] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir, "Adapting to network and client variability via on-demand dynamic distillation," Operating Systems Review, vol. 30, no. 5, Oct. 1996.
- [2] B. Housel and D. Lindquist, "Webexpress: A system for optimizing web browsing in a wireless environment," in Proceeding of the Second ACM/IEEE International Conference on Mobile Computing and Networking, 1996.
- [3] K. Ham, S. Jung, S. Yang, H. Lee, K. Chung, "Wireless-adaptation of WWW Content over CDMA", Sixth IEEE International Workshop on Mobile Multimedia Communications (MOMUC '99), San Diego, Nov. 1999.
- [4] James Gwertzman, Margo Seltzer, Harvard University, "The Case for Geographical Push-Caching".
- [5] Mads Haahr, Raymond Cunningham and Vinny Cahill "Supporting CORBA Applications in a Mobile Environment", Mobicom '99, Seattle Washington USA, 1999.
- [6] HTTP (Hypertext Transfer Protocol), <http://www.w3.org/Protocols/>
- [7] Apache Web server, <http://www.apache.org/>
- [8] Arup Acharya, Tomasz Imielinski and B. R. Badrinath, Rutgers University, "DATAMAN project: Towards a Mosaic-like Location-Dependant Information Service for Mobile Clients".
- [9] Tao Ye, H. Arno Jacobsen, Randy Katz, "Mobile awareness in a wide area wireless network of infostations", Mobicom '98, Dallas Texas USA, 1998