

이메일 개인 프록시 서버

오서균^o 국대근 광추근 정규식
송실대학교 정보통신전자공학부
{skoh^o, dkkuk, gobarian, kchung}@q.ssu.ac.kr

An Email Personal Proxy Server

Seokyun Oh^o Daekeun Kuk Hukeun Kwak Kyusik Chung
School of Electronics Engineering, Soongsil University

요약

현재 이메일은 인터넷을 통한 통신수단으로서 대부분의 사람들이 사용하고 있다. 그러나 각 사용자가 여러개의 이메일을 소유하게 될 경우에는 개별적으로 관리해야하는 어려움이 따른다. 본 논문에서는 이메일과 관련된 개인 프록시 서버를 구축하여 이러한 관리를 해결하고자 한다. 제안된 이메일 개인 프록시 서버는 사용자가 어떠한 이메일을 선택하던 단일 뷰를 제공하고 나머지는 개인 프록시 서버와 외부 이메일 서버가 담당하게 된다. 실제 구현을 통해 본 논문에서 제안한 이메일 개인 프록시 서버의 유효성을 확인하였다.

1. 서론

2004년 2월 정보통신부의 '2003년 하반기 정보화 실태' 보고서에서 지난해 말 현재 우리나라의 인터넷 이용 인구가 전년(2002년) 같은 기간에 비해 2백95만명 늘어난 2천9백22만 명이라고 밝혔다. 인터넷을 이용하는 사용자들은 주로 자료 및 정보검색(72.8%)을 위해 인터넷을 가장 많이 사용하며 다음은 게임, 이메일, 뉴스 검색, 쇼핑 등을 하기 위한 것으로 나와 있고 이들은 대부분 일주일 평균 인터넷 이용시간은 12.5시간으로 나왔다. 그리고 그 이용자 대부분은 이메일 계정을 보유하고 있으며 평균 보유 개수는 1.7개로 조사되었다[1]. 이와 같이 이메일은 인터넷의 급성장과 더불어 많은 사람들이 사용하고 있고 앞으로도 그러한 증가추세는 계속 될 것으로 보인다.

그림 1은 이메일과 관련된 프로토콜을 정리한 그림이다. 메일을 수신하는 방식은 메일 서버에서 메일을 저장하고 관리하는 방식(IMAP4[2], HTTP[3])과 메일 서버로부터 메일을 수신하여 자신의 컴퓨터에 메일을 저장하고 관리하는 방식(POP3[4])이 있다. 이러한 다양한 수신 프로토콜로 인해 메일 계정을 여러개 사용할 경우 개별적으로 관리해야하는 어려움이 따른다.

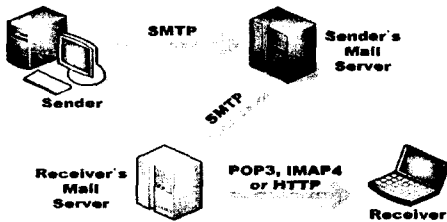


그림 1 이메일 관련 프로토콜

이러한 어려움을 해결하고자 본 논문에서는 하나의 개인 프록시 서버를 이용하여 이메일을 통합 관리하는 이메일 개인 프

록시 서버(EPPS: Email Personal Proxy Server)를 제안한다. 개인 프록시 서버는 '사용자(개인)'를 대신하여 요청된 작업을 수행하고 사용자에게 수행 결과를 보여주는 것'으로 정의할 수 있다. 이 개념은 Personal-level Routing[5], Information Retrieval[6]에 응용이 가능하고 본 논문에서는 이메일에 적용한다. 이를 그림으로 나타내면 그림 2와 같다.

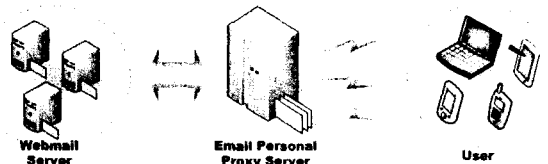


그림 2 이메일 개인 프록시 서버

본 논문의 구성은 다음과 같다. 2장에서는 기존 및 제안된 EPPS를 설명하고, 3장에서는 제안된 EPPS의 구현을 설명한다. 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 이메일 개인 프록시 서버

2.1 기존 EPPS

기존 EPPS로는 EMUmail[7]이 있고 그림 3은 이를 나타낸다. EMUmail은 네트워크상에 하나의 서버로써 그 서버에는 계정들이 저장되어 있다. 사용자가 EMUmail 서버에 접근하게 되면 저장된 각 메일 계정에서 메일을 가져와 목록을 작성한 후 그 결과를 HTML 형식으로 사용자 로컬 컴퓨터에 보여준다. 이 EMUmail의 경우에는 EMUmail 서버에 접근하여 한 번의 로그인만으로 다른 계정들의 메일들을 한 번에 볼 수 있고 별다른 설치 없이도 웹이 지원되는 모든 곳에서든 확인이 가능하다는 장점을 가진다. 하지만 그 메일들을 따로 저장하지 않기 때문에 매번 접속을 할 때마다 메일들을 가져오고 그 목록을 작성하는 등의 시간이 오래 걸려 상당한 지연을 가지고 온다. 그리고 로그인을 하지 않는 동안에는 각 계정의 메일 서버에 접근을 하지 않기 때문에 메일서버의 에러 등으로 인해 접속이 불가능할 때는 그 메일 계정으로 온 메일을 확인 할 수 없다는

단점이 존재한다.

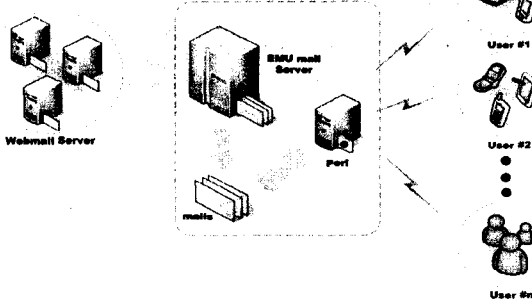


그림 3 EMUmail

2.2 제안된 EPPS

제안된 EPPS는 그림 4와 같다. 제안된 EPPS는 로그인을 하지 않아도 EPPS가 수시로 각각의 메일 계정에 접속하여 메일을 가져와 데이터베이스에 저장한 후 사용자가 로그인을 하면 데이터베이스에 저장된 메일을 보여준다.

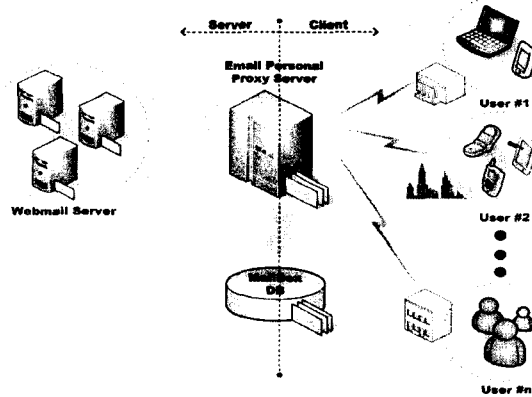


그림 4 제안된 EPPS

제안된 방식은 로그인 후 메일 확인 시에 지연시간이 크지 않으며 로그인 당시 실제 이메일 서버에 접속이 불가능하여도 접속할 수 없게 된 시간 전까지의 메일을 모두 볼 수 있다는 장점을 가진다.

3. 구현

3.1 EPPS 프로그램

이메일 개인 프록시 서버에 사용되는 프로그램은 Visual C++ 6.0과 MFC를 이용하여 구현하였다. 사용자 정보 및 이메일을 저장하기 위해 데이터베이스로 Microsoft Access 2002를 사용하였다.

그림 5는 이메일 개인 프록시 서버의 전체 구성도를 나타내고 관련 프로그램과 데이터베이스를 정리하면 표 1와 같다.

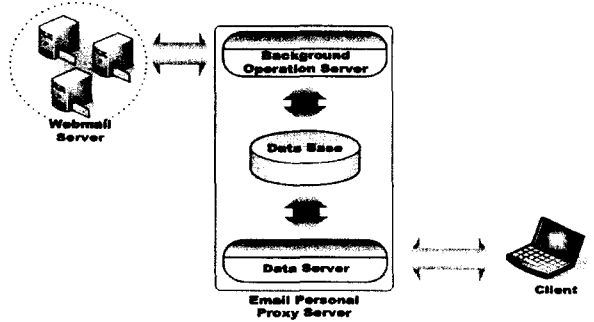


그림 5 EPPS의 전체 구성도

표 1 EPPS 프로그램과 데이터베이스

프로그램	기능
Background Server	자동 메일 수신 프로그램으로 EPPS 내에서 동작한다. 이메일 헤더는 데이터베이스에 저장하고, 본문은 텍스트 파일로 저장한다. 사용자가 저장한 계정의 이메일 서버와 직접 소켓 통신을 통해서 메일 수신 프로그램(POP3/IMAP4)에 따른 명령어를 주고 받으며 이메일을 가져오게 된다.
Data Server	클라이언트와 통신을 하여 백그라운드 서버(Background Server)에 의해서 데이터베이스에 저장된 이메일 헤더와 본문 내용을 클라이언트의 요구에 따라 보내주게 된다.
Client	데이터 서버(Data Server)와 통신을 하여 이메일 데이터를 가져오고 이를 사용자에게 보여준다. 메일 전송 프로토콜(SMTP)을 이용하여 메일을 전송한다.
LOGIN.mdb	클라이언트/백그라운드 서버에서 로그인을 하는 경우에 사용하기 위해 EPPS를 사용하는 모든 사용자의 ID/PW가 저장되어 있다. 각 필드는 ID, PASSWORD, USER NAME으로 되어 있다.
SERVER.mdb	사용자의 메일 계정이 저장되어 있다. 백그라운드 서버는 이 데이터베이스를 검색하여 사용자의 메일 계정과 직접 통신하여 데이터를 가져오게 된다.
HEADER.mdb	이메일 헤더가 저장되어 있다. 각 필드는 DATE, SERVER, SIZE, ADDRESS, SUBJECT, SENDER로 되어있다. 백그라운드 서버에서 이메일 서버를 통해 받은 이메일의 헤더를 각각 분리하여 저장하게 된다.

3.2 동작 과정

EPPS의 동작 과정을 정리하면 다음과 같다.

Step 1. 사용자가 EPPS에 접속하여 자신의 이메일 계정을 등록한다.

Step 2. 백그라운드 서버는 주기적으로 등록된 이메일 계정을 이용하여 이메일 서버에 직접 접속하여 메일을 자동으로 가지고 와서 헤더는 데이터베이스에 저장하고, 본문은 텍스트 파일로 저장한다.

Step 3. 사용자가 이메일을 확인하고자 할 때는 EPPS에 접속하여 로그인을 한다.

Step 4. 로그인 후 메일 가져오기를 선택하면 클라이언트 프로그램은 데이터 서버 프로그램과 통신을 하여 데이터베이스에 저장된 헤더 내용을 가져오게 된다.

Step 5. 사용자가 여러 개의 헤더 중 하나를 선택하면 클라이언트는 데이터 서버와 통신을 통해 텍스트 파일에 저장된 본문 내용을 가져온다.

그림 6은 백그라운드 서버 프로그램을 나타낸다. 이 프로그램은 EPPS 내에서 각 사용자의 메일을 전담하여 수신하는 기능을 함으로 모든 사용자에게 하나씩 부여되고 로그인 과정을 통해 독립적으로 동작한다. 클라이언트 프로그램의 초기 화면과

유사하고, 메일 송신 기능을 제외하고 클라이언트의 모든 기능을 할 수 있다. 이러한 화면을 통해 관리자 모드로 사용하여 백그라운드 서버 프로그램의 동작을 확인할 수 있다.

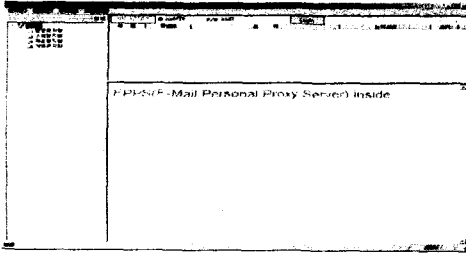
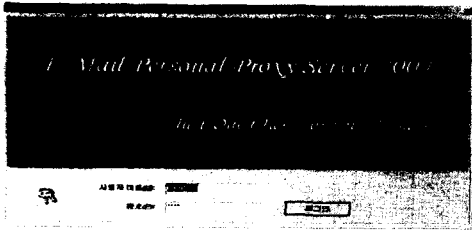
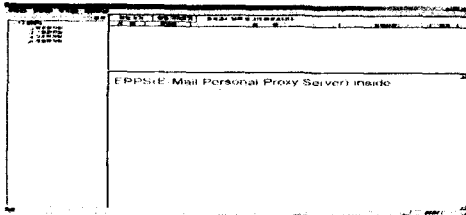


그림 6 백그라운드 서버 프로그램

그림 7(a)은 클라이언트의 로그인 화면을 나타낸다. 아이디와 패스워드가 틀렸을 경우, 틀린 횟수와 함께 경고 메시지를 보여주고 3회 이상 틀린 경우 프로그램을 종료한다. 로그인이 성공한 경우 그림 7(b)와 같이 나타나며, 데이터베이스에 저장되어 있던 사용자 이름을 상단에 표시한다.



(a) 로그인 화면



(b) 로그인 후 화면

그림 7 클라이언트 프로그램

그림 8은 클라이언트 프로그램에서 메일 가져오기를 실행한 후를 나타낸다. 사용자가 헤더 중 하나를 클릭하게 되면 클라이언트는 데이터 서버와의 통신을 통해 본문 내용을 보여주게 된다.

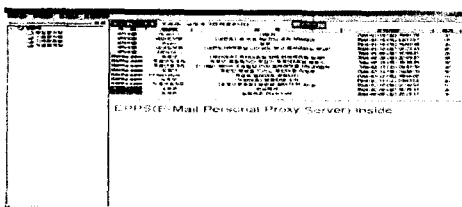


그림 8 클라이언트 프로그램 (메일 가져오기)

그림 9는 데이터 서버 프로그램을 나타낸다. 데이터 서버 프로그램은 클라이언트의 접속을 기다리다가 요청이 있을 때 포트를 할당해주고 EPPS에 저장된 내용을 클라이언트에 전송해 주는 기능을 담당한다.

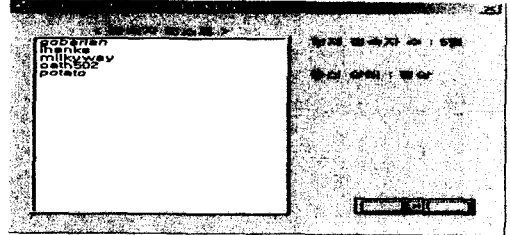


그림 9 데이터 서버 프로그램

4. 결론

본 논문에서는 현재 가장 많이 사용되고 있는 이메일과 관련하여 이들의 사용상 문제점을 지적하고 이를 해결할 수 있는 이메일 개인 프록시 서버(EPPS)를 제안하였다. 제안된 시스템은 사용자가 여러 메일을 사용할 경우 효과적으로 동작함을 확인하였다.

향후 연구 방향을 요약하면 다음과 같다.

(1) 에이전트를 이용한 이메일 가져오기 : 일반 포털 사이트에서 지원하는 이메일의 경우 대부분 서버에서 이메일을 확인 가능한 IMAP를 지원하나 이 메일을 원격에서 가지고 오기 위해서는 일반 포털 사이트의 주소가 아닌 실제 IMAP 서버의 주소를 알아야 하는 문제점이 존재한다. 에이전트를 이용하면 실제 IMAP 주소를 모르더라도 사용자 대신 포털 사이트에 접속 및 로그인 하여 메일 정보를 HTTP를 통해 전송하는 시스템의 구현이 가능하다.

(2) 클러스터링에 기반한 이메일 개인 프록시 서버 : 이메일 개인 프록시 서버의 경우 범용적으로 사용하게 될 경우 많은 사용자를 처리하기 위해 고성능의 서버를 사용하거나 저가의 서버들을 이용한 클러스터링 방법이 존재한다. 클러스터링을 이용할 때 주의할 점은 사용자에 따른 부하 분산시에 사용자와 EPPS를 1:1로 연결해 주기 위해 Hash 값을 이용해야 한다.

참고문헌

- [1] 정보통신부, "2003년 하반기 정보화 실태", <http://www.mic.go.kr/index.jsp>
- [2] Internet Message Access Protocol - Version 4rev1, <http://www.imap.org/papers/docs/rfc3501.html>
- [3] HTTP, <http://services.msn.com/svcs/hotmail/httpmail.asp>
- [4] Post Office Protocol, <http://andrew2.andrew.cmu.edu/rfc/rfc1725.html>
- [5] Mobile People Architecture, <http://mpa.stanford.edu>
- [6] Z. Lacroix, A. Sahuguet, R. Chandrasekar, and B. Srinivas, "A novel approach to querying the web: Integrating retrieval and browsing," ER97 Workshop on Conceptual Modeling for Multimedia Information Seeking, Los Angeles, 1997.
- [7] EMUmail, <http://www.emumail.com>