

# 인터넷에서 한글 메시지를 위한 전자 메일 브라우저

이 정 현<sup>†</sup> · 임 성 락<sup>††</sup>

## 요 약

인터넷의 가장 중요한 서비스 중의 하나인 전자 메일 시스템은 정보 교환의 수단으로 널리 이용되고 있다. 그러나, 대부분의 기존 전자 메일 시스템은 7비트 코드를 지원하기 위하여 설계되었기 때문에 한글 메시지와 같은 8비트 코드를 전송하는데 문제점이 있다. 이러한 문제점을 해결하기 위하여 MIME, ISO-2022-KR과 같은 기법이 발표 되었지만 아직도 한글 메시지의 송수신에는 어려움이 있다. 본 논문에서는 일반 사용자가 보다 편리하게 한글 메시지를 송수신할 수 있는 전자 메일 브라우저 모델을 제시한다. 제시한 모델의 기본 개념은 MIME만을 지원하는 POP3 클라이언트를 확장한 것으로써 8비트 한글 메시지를 ISO-2022-KR이나 Quoted-Printable 혹은 Base64로 인코딩하여 7비트로 전송한다. 제시한 모델의 타당성을 검토하기 위하여 전자 메일 브라우저를 한글 Win95 환경에서 구현하여 실험하였다.

## E-Mail Browser for Hangul Message on Internet

Jong Hyon Lee<sup>†</sup> · Seong Rak Rim<sup>††</sup>

## ABSTRACT

One of the most important services on Internet, the e-mail system is used as a common method for exchanging the information. But, since most of the traditional e-mail systems have been designed to support the only 7-bit code, there are some problems to transmit the 8-bit code such as Hangul message. To overcome these problems, even the schemes like MIME, ISO-2022-KR have been announced, there are still difficulties to send and receive a Hangul message. This paper presents a model of e-mail browser which allows a user to send and receive the Hangul message more conveniently. Basic concept of this model is, as an extension of POP3 client which supports only MIME, the 8-bit of Hangul message is encoded with the ISO-2022-KR or Quoted-Printable or Base64, then transmitted with the 7-bit. To evaluate the feasibility of this model, a prototype of e-mail browser has been implemented and tested on the Hangul Win95 environment.

### 1. 서 론

전자 메일 시스템은 일반적으로 MTA(Message Transfer Agent)와 UA(User Agent)로 구성되어 있다. MTA는 메시지의 송수신을 직접적으로 담당하는 서

버 프로그램으로서 "sendmail"이 널리 사용되고 있다. UA는 최종 사용자가 메시지 작성, 내용 보기 및 삭제 등을 쉽게 할 수 있도록 지원해주는 클라이언트 프로그램으로서 "mail", "mailx", "Eudora", "Netscape-mail" 등이 있다.

기존의 "sendmail"과 같은 MTA은 메시지를 송수신하기 위하여 SMTP 프로토콜을 사용하고 있다[2]. SMTP는 기본적으로 7비트 ASCII 코드만을 지원하

† 정 회 원:(주)한일시스템 금융소프트웨어개발연구소  
†† 중신회원:호서대학교 컴퓨터공학과  
논문접수:1996년 11월 26일, 심사완료:1997년 7월 22일

기 때문에 한글과 같은 8비트 코드 혹은 2진 파일의 송수신은 어렵다[1, 3]. 이러한 문제점을 해결하기 위하여 다국어 및 멀티미디어 자료와 다양한 형태의 자료가 포함된 복합 메시지를 송수신할 수 있는 MIME (Multipurpose Internet Mail Extensions)이 발표되었다[9, 10]. MIME은 현재 인터넷 표준으로 잡혀가고 있으며 많은 UA에서 사용되고 있다. 한편, 국내에서도 한글 메시지에 대한 관심이 높아져 8비트로 구성된 한글 코드를 7비트로 코딩하기 위한 방법으로서 ISO-2022-KR 기법이 발표되었다[11]. 상당수의 MTA에서 이 기법을 이용한 “한글 sendmail”이 사용되고 있지만 “한글 sendmail”이 설치되지 않은 메일 서버에서는 “한글 sendmail”로부터 전송된 한글 메시지를 볼 수 없다.

한글 메시지를 지원하기 위한 방법에는 MTA단계에서 지원하는 방법과 UA단계에서 지원하는 방법이 있을 수 있다. “한글 sendmail”과 같이 MTA에서 한글 메시지의 코딩이 처리된다면 사용자는 UA에 무관하게 한글 메시지를 주고받을 수 있지만, 인터넷상의 모든 메일 서버에 “한글 sendmail”을 설치해야 하는 현실적인 어려움이 있다. 반면 UA에서 한글 메시지의 코딩이 처리된다면 사용자는 MTA에 무관하게 한글 메시지를 주고받을 수 있지만, 모든 UA에서 동일한 코딩 기법을 사용하도록 해야하는 어려움이 있다. 그러나, 대부분의 기존 UA들이 다양한 형태의 코딩 기법을 수용할 수 있는 MIME을 지원하고 있기 때문에 한글 메시지 코딩 부분을 MIME에 추가함으로써 UA 단계에서 지원하는 것이 보다 쉽다.

본 논문에서는 메일 서버에 설치된 MTA에 무관하게 UA 단계에서 한글 메시지를 송수신할 수 있는 메일 브라우저 모델을 제시하고자 한다. 제시한 모델의 기본 개념은 MIME부분에 한글 메시지 코딩을 위한 ISO-2022-KR 기법을 추가하여 8비트 한글 메시지를 ISO-2022-KR로 인코딩하여 7비트로 전송한다. 또한, 사용자의 편의성을 제공하기 위해 메시지를 서버로부터 클라이언트의 브라우저로 받아오기 위하여 POP3 프로토콜을 사용하고 브라우저에서 작성된 메시지를 전송하기 위하여 SMTP 프로토콜을 사용한다.

본 논문의 구성은 2장에서 메일 브라우저에서 사용하고 있는 여러 가지 프로토콜과 메시지 코딩 기법을 기술하고, 3장, 4장에서 메일 브라우저의 설계와 구

현 및 실험에 관하여 각각 기술한다. 마지막으로 5장에서 결론을 맺는다.

## 2. 관련 연구

현재 메일 시스템에서 사용되고 있는 송수신 규약은 RFC 821/822에 정의된 SMTP 프로토콜을 따르고 있다. 또한, 다국어 및 멀티미디어 자료와 다양한 형태의 자료가 포함된 복합 메시지를 송수신하기 위하여 코드 확장 기법인 MIME이 사용되고 있다. 한편, 국내에서는 한글 메시지를 송수신하는 방법으로 ISO-2022-KR도 사용되고 있다.

### 2.1 SMTP(Simple Mail Transport Protocol)

SMTP는 전자 메일 메시지를 효율적으로 전송하기 위한 인터넷 표준 프로토콜로서 RFC 821에 정의되어 있다[2]. SMTP는 메시지가 전달되는 중간 메일 서버와 무관하게 단지 신뢰성 있는 데이터 스트림 채널만을 요구한다. 송신-SMTP는 사용자의 메일 전송 요구를 받게 되면 수신-SMTP와 양방향 전송 채널을 형성한다. 송신-SMTP와 수신-SMTP는 형성된 채널을 통하여 SMTP 명령어와 메시지, 이에 대한 응답을 주고받는다. 수신-SMTP는 최종 목적지이거나 중계자일 수 있는데 최종 목적지일 경우, 수신된 메시지를 파일 시스템에 저장하고 사용자가 이를 받아 볼 수 있도록 한다. 중계자일 경우, 다음 중계자나 다음 목적지와 다시 채널을 형성한다. 이런 과정을 반복적으로 거쳐 목적지까지 메시지가 도달하면 메시지 전송은 성공적으로 종료한다.

SMTP에서 사용하는 메시지 형식은 RFC 822에 정의되어 있으며 메시지는 헤더와 바디로 구성된다[3]. 헤더는 메시지의 발신자와 수신자, 날짜, 제목 그 밖에 추가 정보로 구성되며 바디는 메시지의 내용이다. 메시지의 헤더와 바디는 하나의 공백 라인을 삽입하여 구분한다. 한편, 메시지의 모든 문자는 7비트 ASCII 코드를 사용해야 한다. 따라서, 한글과 같이 8비트 코드를 사용하는 메시지와 그림이나 소리와 같은 2진 파일을 전송하는데 부적합하다. 8비트 메시지와 2진 파일을 RFC 822에 정의된 형태로 주고받기 위해서는 송신측에서 “uuencode”와 같은 유틸리티를 사용하여 7비트 ASCII 코드로 변환하여 보내야 하고,

수신측에서는 이를 받아 저장하고 “uudecode”와 같은 유틸리티를 사용하여 다시 8비트 코드로 변환해 주어야 한다.

## 2.2 MIME(Multipurpose Internet Mail Extensions)

RFC 821/822에 정의된 SMTP에 의한 메시지 송수신은 7비트 ASCII 코드만으로 제한되어 있다. MIME은 이러한 문제를 해결하기 위한 코드 확장 기법으로서 RFC 1521/1522에 정의되어 있다[9, 10]. RFC 1521에는 메시지 바디의 코드 변환을 위한 인코딩 기법과 인코딩시 추가되어야 하는 헤더 필드에 대하여 정의하고 있으며, RFC 1522에는 메시지 헤더에서 8비트 코드를 7비트로 인코딩하기 위한 방법이 정의되어 있다. MIME에 정의되어 있는 코딩 기법은 Quoted-Printable과 Base64로써 서로 장단점을 가지고 있다.

Quoted-Printable 인코딩 기법은 MSB(Most Significant Bit)가 1인 문자 1바이트를 입력하여 4비트씩 나누고, 등호(=)와 그 뒤에 각각의 4비트를 16진수 코드로 나타내는 기법이다. 인쇄 가능한 ASCII 코드(코드값이 33~60, 62~127을 포함하여)는 ASCII 코드 그대로 나타내고, 그 이외의 ASCII 코드는 MSB가 1인 문자와 같은 방식으로 인코딩한다. SPACE와 TAB은 인코딩된 라인 중간에 나올 수 있으나 인코딩된 라인 끝에 올 때는 특별히 “=”(soft line break)로 인코딩된다. 메시지가 이 기법으로 인코딩되었음을 나타내기 위하여 메시지 헤더의 Content-Transfer-Encoding 필드를 “quoted-printable”로 설정해야 한다. 이 기법으로 코딩한 경우 모든 문자의 MSB가 1이면 인코딩된 메시지의 크기가 원래 메시지 크기의 3배만큼 증가하기 때문에 한글 메시지를 인코딩하기에는 효율이 떨어진다.

Base64 인코딩 기법은 24비트(3바이트)를 입력하여 6비트 단위로 잘라 64(2<sup>6</sup>)개의 문자로 인코딩하는 기법이다. 출력된 6비트의 값에 따라 영어 알파벳 대문자, 소문자, 숫자, ‘+’, ‘/’의 순서로 구성된 64개의 코드 체계로 표기한다. 메시지가 이 기법으로 인코딩되었음을 나타내기 위하여 메시지 헤더의 Content-Transfer-Encoding 필드를 “base64”로 설정해야 한다. 이 기법으로 코딩한 경우 코드의 정보는 6비트로 표시할 수 있기 때문에 SMTP에 적용할 수 있다. 그러나, 인코딩된 메시지의 크기는 원래 메시지 크기의

33%만큼 더 증가하기 때문에 2진 파일과 한글 메시지와 같이 MSB가 1인 코드의 양이 전체 메시지의 1/3이상인 경우에 효율이 높지만 영문 메시지에는 효율이 떨어진다.

한편, 헤더의 8비트 코드는 “=?charset?encoding?encoded-text?”의 encoded-word 형태로 표시한다[10]. “charset”은 인코딩의 대상이 되는 데이터의 코드 체계를 의미하며 “encoded-text”는 인코딩된 7비트 코드이다. “encoding”은 인코딩 기법으로서 ‘Q’와 ‘B’를 가질 수 있으며 ‘Q’는 Quoted-Printable, ‘B’는 Base64를 각각 의미한다.

## 2.3 ISO-2022-KR

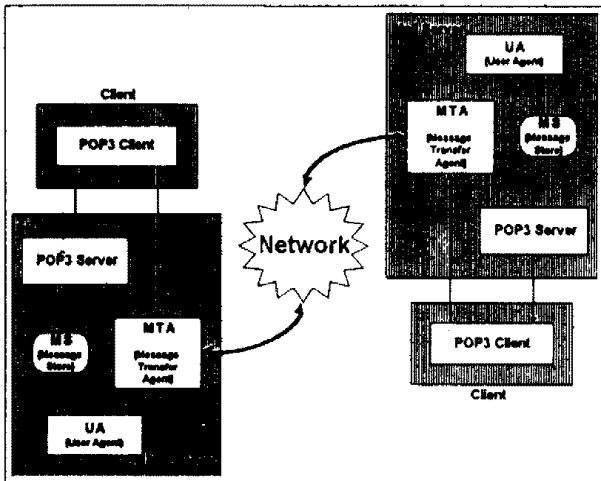
ISO-2022-KR은 ISO-2022를 기반으로 전자 메일 시스템에서 한글 메시지를 송수신하기 위하여 제안된 코딩 기법이다[11, 4]. 이 기법은 ASCII 영문자는 ASCII로 표시하고 한글은 MSB를 제거한 나머지 7비트를 해당 ASCII 코드로 표시한다.

메시지가 ISO-2022-KR로 인코딩되었음을 나타내기 위하여 메시지 헤더와 바디부분에 각각 표시한다. 헤더 부분은 헤더 필드에서 Content-Type 필드를 “text/plain; charset = iso-2022-kr”로, Content-Transfer-Encoding 필드를 “7bit”로 설정하여 나타낸다. 바디 부분은 한글이 나타나는 바디의 첫 행에 “[S)C”([, ASCII 코드값 27)를 삽입하여 메시지가 ISO-2022-KR로 인코딩되었음을 나타낸다. 인코딩된 한글 코드의 시작과 끝은 ‘N’(ASCII 코드값 14, SO)과 ‘O’(ASCII 코드값 15, SI)로 표시한다. 즉, ISO-2022-KR의 메시지 바디 정규 표현은 “ASC||\*/SO[encoded KSC-5601]\* SI|\*” 형태이다. 메시지 헤더의 한글 코드는 RFC 1522에서 정의한 방법에 따라 Quoted-Printable나 Base64를 이용하여 encoded-word 형태로 인코딩되며 encoded-word의 “charset”은 “EUC-KR”로 설정한다[8].

## 2.4 POP3(Post Office Protocol-Version 3)

POP3는 “Eudora”, “Netscape-mail”과 같은 클라이언트 프로그램에서 메일 서버에 도착된 메시지를 받아보기 위한 프로토콜로서 RFC 1081에 정의되어 있다[5]. 이 프로토콜을 이용하기 위해서는 메일 서버에 POP3 서버를 설치하여야 한다. POP3 서버는 클라이언트의 접속 요청이 있을 경우 클라이언트와 양방향

전송 채널을 형성하여 메시지를 주고받는다. (그림 1)과 같이 POP3 클라이언트에서 메시지 수신은 메일 서버에 설치된 POP3 서버를 통하여 이루어지지만 메시지 전송은 MTA에 직접 한다.



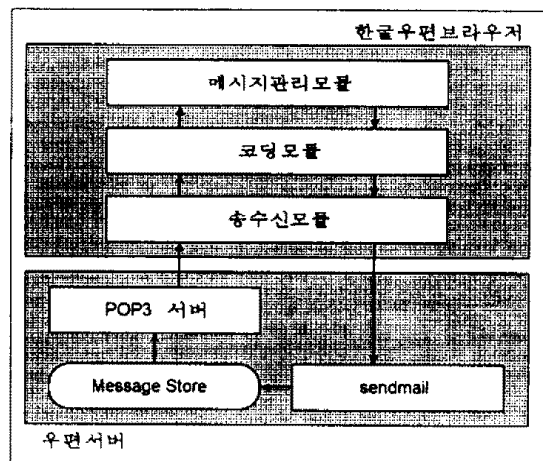
(그림 1) POP3 서버를 이용한 전자 우편 시스템  
(Fig. 1) E-mail system with pop3 server

### 3. 브라우저 설계

Eudora, Internet Mail, Netscape Mail 등 기존의 UA를 사용하여 한글 메시지를 수신 할 경우에는 송신 UA에 따라 부분적으로 디코딩된다. Eudora는 MIME에서 정의한 Quoted-Printable과 Base64 코딩 기법으로 인코딩된 메시지만을 디코딩하고, Internet Mail은 ISO-2022-KR, Quoted-Printable, Base64로 인코딩된 메시지를 모두 디코딩하지만 Content-Type 필드와 Content-Transfer-Encoding 필드의 내용에 따라 디코딩하기 때문에 “한글 sendmail”이 설치된 서버에서 작성되거나 중간 단계에서 ISO-2022-KR로 인코딩된 메시지는 디코딩하지 못한다. 메일 서버의 MTA가 한글 sendmail일 경우에는 ISO-2022-KR만을 지원하기 때문에 Quoted-Printable과 Base64로 인코딩된 메시지는 디코딩하지 못한다. 또한, MTA가 sendmail일 경우에는 어떠한 코딩 기법도 지원하지 않기 때문에 인코딩된 모든 메시지를 디코딩하지 못한다.

따라서, 본 논문에서 제시한 전자 메일 브라우저는

기존의 MTA에 상관없이 한글 메시지를 송수신할 수 있도록 설계한다. 이를 위하여 메시지 코딩은 기본적으로 ISO-2022-KR 기법을 사용하며, MIME의 Quoted-Printable과 Base64 코딩 기법은 사용자의 요구에 따라 선택적으로 지정함으로써 다양한 형태로 코딩된 메시지를 지원하도록 한다. 또한 사용자의 편의성을 제공하기 위하여 메시지를 서버로부터 클라이언트의 브라우저로 받아오기 위하여 POP3 프로토콜을 사용하고 브라우저에서 작성된 메시지를 전송하기 위하여 SMTP 프로토콜을 사용한다. 제시한 한글 메일 브라우저의 구조는 (그림 2)와 같이 메시지 관리, 코딩, 송수신 모듈로 구성된다.



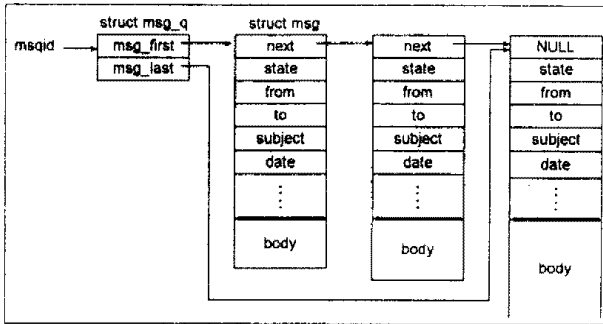
(그림 2) 한글 전자 우편 브라우저 모델  
(Fig. 2) Hangul e-mail browser model

#### 3.1 메시지 관리 모듈

2메시지 관리 모듈은 메시지 작성 및 송신과 수신된 메시지 관리 기능을 제공한다. 송수신된 메시지는 “송신 메시지함”과 “수신 메시지함”이라는 이름의 메시지함에 각각 저장하여 사용자로 하여금 삭제, 수정, 재전송 등을 보다 편리하게 할 수 있도록 한다. 또한, 사용자 부주의로 인하여 삭제된 메시지를 복구하기 위하여 “메시지 보관함”을 만든다. “메시지 보관함”은 “송신 메시지함”이나 “수신 메시지함”에서 삭제된 메시지를 저장하고 있다가 완전히 삭제하거나 다시 복구하는데 사용한다.

송수신한 메시지 관리의 효율을 높이고 저장 공간을 줄이기 위하여 브라우저에서 송신한 모든 메시지

는 “송신 메시지함”에 저장하고 송신시에 인코딩한다. 그러나, 브라우저에 수신된 모든 메시지는 디코딩하여 “수신 메시지함”에 저장한다. 모든 메시지함 및 메시지 형식은 (그림 3)과 같은 구조체로 설계한다.



(그림 3) 메시지 데이터 구조  
(Fig. 3) Data structure of message

### 3.2 코딩 모듈

기존의 MTA는 7비트 ASCII 코드만을 송수신하기 때문에 8비트 한글 메시지를 송수신하기 위해서 송신측에서는 8비트를 7비트로의 인코딩이 필요하며 수신측에서는 7비트를 8비트로의 디코딩이 필요하다. 코딩 모듈은 이와 같은 코드 변환을 담당하는 모듈로써 인코딩과 디코딩 부분으로 구성된다.

인코딩 부분은 메시지의 헤더와 바디를 7비트 ASCII 코드로 변환하는 부분이다. 메시지 바디의 한글은 기본적으로 ISO-2022-KR로 인코딩하며 사용자의 요구에 따라 Quoted-Printable이나 Base64로도 인코딩할 수 있도록 한다. 메시지 헤더는 RFC1522에 따라 encoded-word 형태로 인코딩한다. 그러나, “한글 sendmail”과의 호환을 위하여 바디가 ISO-2022-KR로 인코딩되었을 경우에만 헤더를 Base64로 인코딩하며, 그 외의 경우에는 바디와 같은 기법을 사용한다. (그림 5)은 메시지 바디의 ISO-2022-KR 인코딩 알고리즘이다.

또한, 인코딩 기법에 따라 헤더의 Content-Type 필드와 Content-Transfer-Encoding 필드를 다음과 같이 설정한다.

ISO-2022-KR 경우:

Content-Type: text/plain; charset = iso-2022-kr

```

while((t = 한 바이트 읽음) != EOF)
    if (t > 0x80) flag = 1;
rewind input file;
if (flag) "\t$\tC\n" 출력;
while((t = 한 바이트 읽음) != EOF) (
    if (t > 0x80) (
        if (lang == ASCII) SO 출력;
        lang = KSC5601;
        t - 0x80 출력;
    ) else if (t == 0x20) (
        t 출력;
    ) else (
        if (lang == KSC5601) SI 출력;
        lang = ASCII;
        t 출력;
    )
)
    
```

(그림 4) ISO-2022-KR 인코딩 알고리즘  
(Fig. 4) ISO-2022-KR encoding algorithm

Content-Transfer-Encoding: 7bit

Quoted-Printable 경우:

Content-Type: text/plain; charset = eu-kr

Content-Transfer-Encoding: quoted-printable

Base64 경우:

Content-Type: text/plain; charset = eu-kr

Content-Transfer-Encoding: base64

디코딩부분은 수신된 메시지의 헤더 필드중 Content-type 필드와 Content-Transfer-Encoding 필드의 내용에 따라 해당 기법으로 디코딩해야 한다. 그러나, ISO-2022-KR은 Content-type 필드나 Content-Transfer-Encoding 필드에 상관없이 “한글 sendmail”을 지나는 과정에서 인코딩되는 경우가 많다. 다시 말해, “한글 sendmail”은 헤더의 Content-Type 필드가 “text/plain; charset = eu-kr”인 경우에만 이것을 “text/plain; charset = iso-2022-kr”로, Content-Transfer-Encoding 필드에 “8bit”라고 되어 있는 것을 “7bit”로 바꾸며, 헤더의 8비트 코드는 Base64로 인코딩하고 바디의 8비트 코드는 KSC 5601로 간주하고 ISO-2022-KR로 인코딩한다[6]. 따라서, 본 논문에서는 수신된 메시지

가 ISO-2022-KR로 인코딩이 되었는지 먼저 확인하도록 설계한다. 바디에서 “[S]C” 문자열을 찾았을 경우에만 ISO-2022-KR로 디코딩하고 해당 문자열을 찾지 못하였을 경우에는 Content-Transfer-Encoding 필드의 내용에 따라 디코딩한다. 디코딩 기법을 알아 내지 못하였을 경우에는 수신된 메시지를 화면에 나타내도록 하여 사용자가 직접 디코딩 기법을 지정하도록 한다. (그림 5)는 메시지 바디의 ISO-2022-KR 디코딩 알고리즘이다.

```

while( t = 한 바이트 읽음) != EOF ) (
    if ( t == SO ) lang = KSC5601;
    else if ( t == SI ) lang = ASCII;
    else if ( lang == KSC5601 ) (
        if ( t == ' )
            t 출력;
        else (
            t + 0x80 출력;
            t = 한 바이트 읽음;
            t + 0x80 출력;
        )
    ) else if ( lang == ASCII )
        t 출력;
    else t 출력;
)
    
```

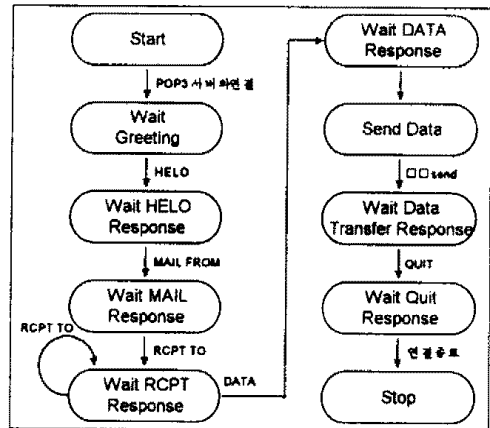
(그림 5) ISO-2022-KR 디코딩 알고리즘  
(Fig. 5) ISO-2022-KR decoding algorithm

### 3.3 송수신 모듈

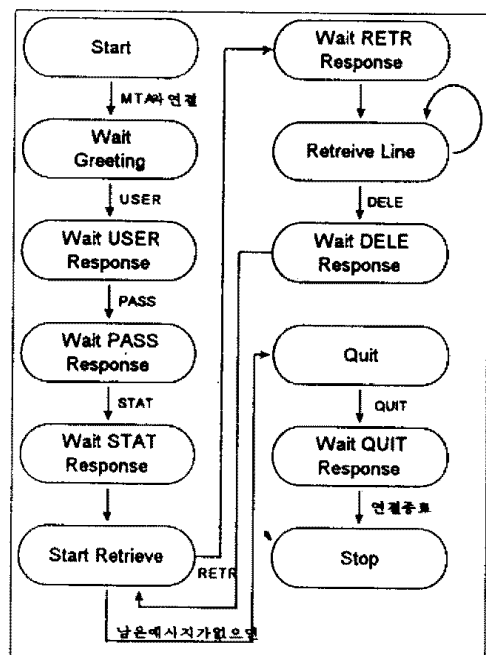
송수신 모듈은 메일 서버와 브라우저간의 통신을 담당하는 모듈로서 소켓을 이용한다. 소켓은 인터넷 통신을 위한 TCP/IP 프로그래밍의 기본으로서 UDP 데이터그램과 TCP 스트림 소켓으로 나누어진다[7]. UDP는 전송 속도가 빠르지만 전송되는 데이터의 신뢰성이 없으며 TCP 스트림 소켓은 연결이 설정된 상태에서 데이터를 주고받기 때문에 UDP에 비해 상대적으로 속도는 느리지만 신뢰성 있는 데이터 전송을 해준다. 이러한 이유로 전자 우편 서버의 MTA(SMTP)는 TCP 25번 포트를, POP3는 TCP 110번 포트를 이용한다.

본 논문에서 설계한 브라우저는 메시지 송수신시 메시지 전송을 위한 프로토콜인 SMTP에 따라 TCP

25번 포트를 통하여 MTA와 통신하며(그림 6), 메시지 수신시에는 메일 서버에 도착한 메시지를 클라이언트로 가져오기 위한 프로토콜인 POP3에 따라 TCP 110번 포트를 통하여 POP3 서버와 통신하도록 한다(그림 7).



(그림 6) 송수신시 브라우저 상태 전이도  
(Fig. 6) State transition diagram of browser on send



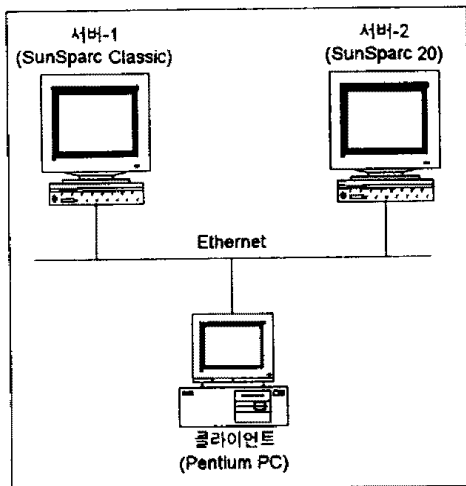
(그림 7) 수신시 브라우저 상태 전이도  
(Fig. 7) State transition diagram of browser on receive

#### 4. 구현 및 실험

본 논문에서 제시한 메일 브라우저 모델을 한글 Win95 환경에서 Microsoft Visual C++(Version 4.0)를 사용하여 구현하고, 제시한 모델의 타당성을 검증하기 위하여 (그림 8)과 같은 환경에서 실험하였다.

##### 4.1 실험 환경

(그림 8)에서 서버-1과 서버-2는 서로 다른 MTA를 지원하도록 “한글 sendmail”과 “sendmail”을 각각 설치한다. 메시지의 전송 및 수신을 위한 클라이언트의 UA로서는 “Eudora”, “Internet Mail”, 서버-1,2의 “Mail” 그리고, 본 논문에서 구현한 “브라우저”를 사용하였다.



(그림 8) 실험 환경  
(Fig. 8) Test environment

서버-1:

- SunSparc Classic(SunOS 4.1.3)
- MTA(한글 sendmail 설치)
- POP3 서버 설치(QPOP 2.2-b5)

서버-2:

- SunSparc 20(SunOS 5.5)
- MTA(sendmail 설치)
- POP3 서버 설치(QPOP 2.2-b5)

클라이언트:

- Pentium 60 IBM PC(한글 Win95)
- UA(브라우저, Eudora, Internet Mail 설치)

##### 4.2 실험 내용 및 결과

제시한 모델의 타당성을 검증하기 위하여 다음과 같은 방법으로 “ABCDEF가나다라마바”라는 메시지를 전송하여 보았다. 모든 메시지의 전달 과정은 서버-1을 거쳐 서버-2로 전달되도록 하였다.

[방법-1] “Eudora”를 사용한 경우:

- ① 메시지를 Quoted-Printable로 인코딩하여 전송한다.
- ② 메시지를 Quoted-Printable로 인코딩하지 않고 전송한다.

[방법-2] “Internet Mail”을 사용한 경우:

- ① 메시지를 ISO-2022-KR로 인코딩하여 전송한다.
- ② 메시지를 Quoted-Printable로 인코딩하여 전송한다.
- ③ 메시지를 Base64로 인코딩하여 전송한다.

[방법-3] “Mail”을 사용한 경우

- ① 서버-1의 “Mail”을 사용하여 전송한다.
- ② 서버-2의 “Mail”을 사용하여 전송한다.

[방법-4] 구현된 “브라우저”를 사용한 경우

- ① 메시지를 ISO-2022-KR로 인코딩하여 전송한다.
- ② 메시지를 Quoted-Printable로 인코딩하여 전송한다.
- ③ 메시지를 Base64로 인코딩하여 전송한다.

각각의 방법으로 전송된 메시지를 여러 UA로 디코딩한 결과는 <표 1>과 같다.

##### 4.3 실험 결과 분석

<표 1>과 같이 본 논문에서 제시한 브라우저를 사용하여 메시지를 수신할 경우 기존의 UA에 관계없이 한글 메시지가 정상적으로 디코딩됨을 확인할 수 있었다. 그러나, 기존의 UA를 사용하여 메시지를 수신할 경우에는 송신 UA의 인코딩 기법에 따라 부분적으로 디코딩된다. 서버-2의 Mail을 사용한 경우 MTA 자체는 인코딩을 안하지만 메시지가 전달되는 중간 단계에서 ISO-2022-KR로 인코딩되었기 때문에

〈표 1〉 메시지 바디의 디코딩 결과  
 〈Table 1〉 Decoding result of message body

송신 UA	인코딩	수신 UA				
		Eudora	Internet Mail	서버-1의 Mail	서버-2의 Mail	브라우저
Eudora	Quoted-Printable	O	O	X	X	O
	Non Encoding	X	X	O	X	O
Interent Mail	ISO-2022-KR	X	O	O	X	O
	Quoted-Printable	O	O	X	X	O
	Base64	O	O	X	X	O
Mail	ISO-2022-KR	X	X	O	X	O
	Non Encoding	X	X	O	X	O
구현한 브라우저	ISO-2022-KR	X	O	O	X	O
	Quoted-Printable	O	O	X	X	O
	Base64	O	O	X	X	O

서버-1의 Mail을 사용한 경우와 똑같은 결과를 보여 주었다. 이렇게 수신 UA에 따라 그 결과가 다르게 나타나는 이유는 다음과 같다.

(1) Eudora를 사용하여 수신할 경우:

Eudora는 MIME에서 정의한 Quoted-Printable과 Base64 코딩 기법만을 제공하기 때문에 이 기법으로 인코딩된 메시지만을 디코딩한다.

(2) Internet Mail을 사용하여 수신할 경우:

Internet Mail은 ISO-2022-KR, Quoted-Printable, Base64로 인코딩된 메시지를 모두 디코딩하였으나 Content-Type 필드와 Content-Transfer-Encoding 필드의 내용에 따라 디코딩하기 때문에 “한글 sendmail”이 설치된 서버에서 작성되거나 중간 단계에서 ISO-2022-KR로 인코딩된 메시지는 디코딩하지 못한다.

(3) Mail을 사용하여 수신할 경우:

메일 서버의 MTA가 한글 sendmail일 경우에는 ISO-2022-KR만을 지원하기 때문에 Quoted-Printable과 Base64로 인코딩된 메시지는 디코딩하지 못한다. 또한, MTA가 sendmail일 경우에는 어떠한 코딩 기법도 지원하지 않기 때문에 인코딩된 모든 메시지를 디코딩하지 못한다.

이상과 같이 본 논문에서 제시한 모델은 기존 UA가 가지고 있는 문제점을 해결하여 MTA와 UA에 상관없이 한글 메시지를 주고받을 수 있었다.

5. 결 론

국내에서 사용되고 있는 메일 시스템은 한글 메시지를 지원하기 위하여 ISO-2022-KR과 MIME이 혼용되고 있다. ISO-2022-KR의 경우 메일 서버의 MTA가 “한글 sendmail”이 아니면 ISO-2022-KR로 인코딩된 한글 메시지를 확인할 수 없다. 또한, “한글 sendmail”이 설치되어 있더라도 클라이언트의 UA가 MIME을 지원하지 않으면 Quoted-Printable이나 Base64로 인코딩된 한글 메시지를 확인할 수 없다. 특히, “한글 sendmail”은 “Eudora”와 같은 기존의 UA에서 인코딩하지 않고 8비트로 보내진 메시지를 ISO-2022-KR로 인코딩하기 때문에, 기존의 UA에서는 한글 메시지를 확인할 수 없다.

그러나, 본 논문에서 제시한 메일 브라우저는 MTA와 UA에 무관하게 한글 메시지를 확인할 수 있다. 또한, 메일 서버로부터 클라이언트의 브라우저로 메시지를 받아오기 위하여 POP3 프로토콜을 사용하고 브라우저에서 작성된 메시지를 전송하기 위하여 SMTP 프로토콜을 사용하여 사용자에게 편의성을 제



공한다.

### 참 고 문 헌

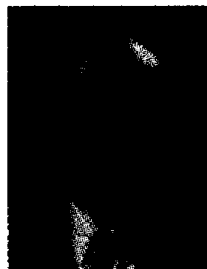
- [1] American National Standard Institute, "Coded character set-7-bit American national standard code for information interchange," ANSI X3.4-1968.
- [2] J. Postel, "Simple Mail Transfer Protocol," RFC 821, 1982.
- [3] D. Crocker, "Standard for the format of ARPA Internet Text Messages," RFC 882, 1982.
- [4] International Organization for Standardization, "Information Processing-ISO 7-bit and 8-bit coded character sets-Code extension techniques," International Standard, 1986, Ref. No. ISO 2022-1986 (E).
- [5] M. Rose, "Post Office Protocol-Version 3," RFC 1081, 1988.
- [6] Korea Industrial Standards Association, "Code for Information Interchange (Hangul and Hanja)," Korean Industrial Standard, 1987, Ref. No. KS C 5601-1989.
- [7] W. Richard Stevens, "UNIX Network Programming," Prentice-Hall, 1990.
- [8] Korea Industrial Standard Association, "Hangul Unix Environment," Korean Industrial Standard, 1992, Ref. No. KS C 5861-1992.
- [9] N. Borenstein, N. Freed, "MIME(Multipurpose Internet Mail Extensions) Part One:Mechanisms for Specifying and Describing the format of Internet Message Bodies," RFC 1521, 1993.
- [10] K. Moore, "MIME(Multipurpose Internet Mail Extensions) Part Two:Message Header Extensions for Non-ASCII Text," RFC 1522, 1993.
- [11] U. Choi, K. Chon, H. Park, "Korean Character Encoding for Internet Messages," RFC 1557, 1993.



### 이 정 현

1995년 호서대학교 컴퓨터공학과 졸업  
 1997년 호서대학교 컴퓨터공학과 석사  
 1997년~현재 (주)한일은시스템 금융소프트웨어개발연구소 연구원

관심분야: 운영체제, 전자우편시스템, 컴퓨터네트워크, 금융전산시스템



### 임 성 락

1979년 서강대학교 전자공학과 졸업  
 1983년 서울대학교 컴퓨터공학과 석사  
 1983년~1990년 금성 반도체(주) 연구소  
 1992년 서울대학교 컴퓨터공학과 박사

1992년~현재 호서대학교 컴퓨터공학과 조교수  
 관심분야: 운영체제, 실시간처리시스템, 분산처리시스템