ISO-2022에 근거한 한글 메일 Encoding 방법 제안과 Prototype 구현 (Draft)

최우형
한국과학기술원 전산학과
uhhyung@kaist.ac.kr

1992년 1월 28일

1 개요

현재 네트웍을 이용한 전자메일은 단순한 영문의 내용에서 멀티미디어를 다룰 수 있는 다국어를 지원하는 메일로 바뀌어 나가는 추세에 있다. 이에 따라서 IETF에서는 RFC-822[RFC-822]에 정의되어 있는 메시지 형식을 멀티미디어화 하기위한 822-Extension Working Group이 구성되었고, 이 Group에서는 최근 IESG에 MIME [MIME]이라는 제목의 Final Draft를 제출하였는데, 이는 곳 Internet의 Proposed Standard가 될 전망이다.

RFC-822 Entension Working Group에서 다국어 지원 Spec을 만들면서 국내에서 쓰이고 있는 방법에 대한 관심이 있었고, 또한 국내에서도 한글메일에 대한 관심이 고조되고 있던바, 한글을 7-bit 환경에서 인코딩하는 방법의 필요성이 제기되었다.

이 글은 ISO-2022에 근거한 한글 메시지의 인코딩 방법을 설명하고 그와 관련한 Prototype 구현, 구현의 이슈, 앞으로의 전망등을 고찰한다.

2 Encoding 방법

2.1 일반적인 배경설명(용어에 익숙하지 않은 사람들을 위해서)

다음에 나올 약간 techinical한 용어와 설명을 좀 더 직관적으로 쉽게 말하면 다음과 같다. 먼저 ISO-2022는 7-bit 환경과 8-bit 환경에서 여러개의 코드셋을 사용하기 위해 정의된 일종의 코드 확장법이다.

이 문서에서 7-bit 환경만을 이야기하는 이유는 이질적(heterogenious) 환경하에서의 상호동작성(interoperability)를 극대화 하기 위함이다. 다시 말하면, 이 문서가 기술하는 인코딩 방법은 단지 인터넷에서만 쓰이는 것을 범위로 하는 것이 아니라, 빗넷, 패킷교환망, 공공망(PSDN, PSTN) 등 모든 환경에서 서로 메시지를 교환하기 위한 일관된 방법을 제시하기 위한 것이기 때문이다.

예를 들어서 다음과 같은 상황을 생각해 보자. 8-bit 환경을 지원하는 네트웍에서 7-bit환경의 네트웍으로 전달된 메시지가 다시 8-bit 환경의 네트웍으로 전달되는 경우를 생각할 수 있다. 만약 8번째 비트를 전달할 수 있는 상황을 기본적으로 가정한다면 위와같은 상황에서는 모든 8번째 비트에 대한 정보가 없어진채로 상대방에게 전달될 것이다.

위와같은 문제의 해결에는 두가지의 방법이 있을 수 있을것이다. 하나는 중간의 게이트웨이에서 전송화일을 변환해 주는 것이고 또 하나는 처음부터 메시지를 7비트로 인코딩 하는 방법이다. 따라서 이 문서는 위의 두가지 방법에서 쓰일 수 있는 표준의 방식을 제시하는 것이다.

2.2 Encoding 방법

G0와 G1영역을 사용한다. US-ASCII가 G0에 KSC-5601이 G1에 할당(designate)된다. SO는 G1영역을 SI은 G0영역을 부르는데(invoke) 사용된다. 8비트 환경에서는 MSB는 의미를 지니지 않으며, 단지 G1영역을 구분하는데 쓰인다.[1]

가정된 Escape Sequence를은 다음과 같다.

- **ESC 2/0 4/4** – Announcer Escape Sequence
  "The G0 and G1 sets shall be used. In a 7-bit environment. SI invokes G0 and S0 invokes G1 into columns 2 to 7. In an 8 bit environment, the designating escape sequences also invoke the G0 and G1 sets into

[1] 용어에 익숙치 않은 사람은 [ISO-2022]를 참고하기 바란다.

columns 02 to 07 and 10 to 15, respectively; no locking shift functions shall be used. See 9.2"

- **ESC 2/8 4/2** – Designating Escape Sequence
  Designate ASCII into G0

실제 메시지의 구조를 Regular expression으로 나타내면 다음과 같다.

```
ESC $ ) C [ ASCII / SO KSC-5601* SI ]*
```

3 Prototype 구현

한글 메일의 구현방법에는 크게 두가지의 방법이 있을 수 있겠다. 그림 1에서 보는 바와 같이 Mail User Agent에 두개의 필터—하나는 유저를 위한, 또 하나는 시스템을 위한—를 두어서 기존의 응용프로그램을 그대로 두고 구현하는 방법이고 또하나는 그림 2에서 보는 바와 같이 MTA를 고치는 방법이다.

첫번째 방법은 기존의 응용프로그램에 아무런 추가의 노력 없이 사용자 레벨에서의 간단한 필터의 설치도 한글 메일을 사용할 수 있다는 장점이 있는데 반해 모든 사용자가 자신의 메일러를 다시 Configure해야 하는 불편함이 있고, 두번째 방법은 단순히 시스템의 MTA(Mail Transfer Agent)를 고치는 것으로 유저에게 완벽한 투명성(transparency)을 제공하면서 한글 메일로 전환할 수 있는 장점이 있는 반면, 모든 MTA를 고쳐야 한다는 현실적으로 어려운 점이 있다. 따라서 Prototype은 MUA(Mail User Agent)상의 필터를 두는 방식으로 구현되었다.

현재 구현된 필터는 KS 한글, N-Byte 한글 터미널 사용자를 위한 MUA용, MTA용 필터들인데, 설명을 위해 먼저 용어에 대한 정리를 하기로 한다. `iso`는 Encoded Code를 나타내고, `ks`는 KS Code, `n`은 N-Byte Code를 각각 나타내는 것으로 한다.

이 필터는 유저의 환경에 따라 다른것으로 대치될 수 있는데, 예를들어 유저가 N-Byte 한글 터미널을 사용한다면 유저측의 필터는 `iso2n`으로, KS 한글 터미널을 사용한다면 `iso2ks`등이 될 것이다. 결국 이런 필터들은 유저레벨의 투명성을 제공하는 모듈이고, MUA와 시스템 MTA상에 존재하는 `ks2iso, n2iso`등은 시스템레벨의 투명성을 제공하게 되는 것이다.

현재 유저레벨 필터를 내장할 수 있도록 patch되어 있는 MUA는 elm, smail 이고, 다른 MUA에서는 Configuration 옵션을 선택하여야 한다.

이 Prototype은 `CAIR.KAIST.AC.KR`의 `ftp/pub/hangul/hmail` 디렉토리에서 Anonymous ftp를 통해서 얻을 수 있다.

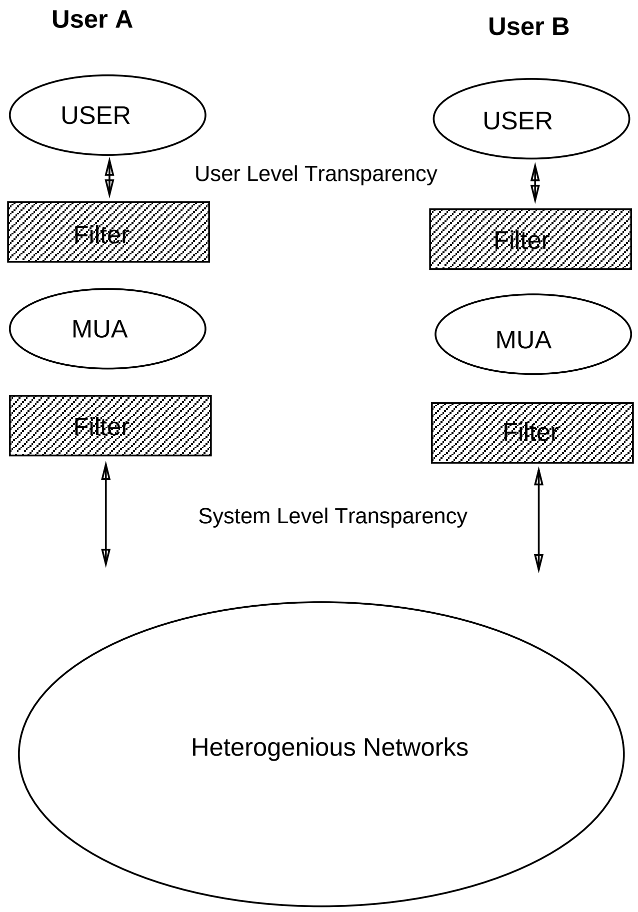

그림 1: Mail User Agent상에서의 필터에 의한 구현

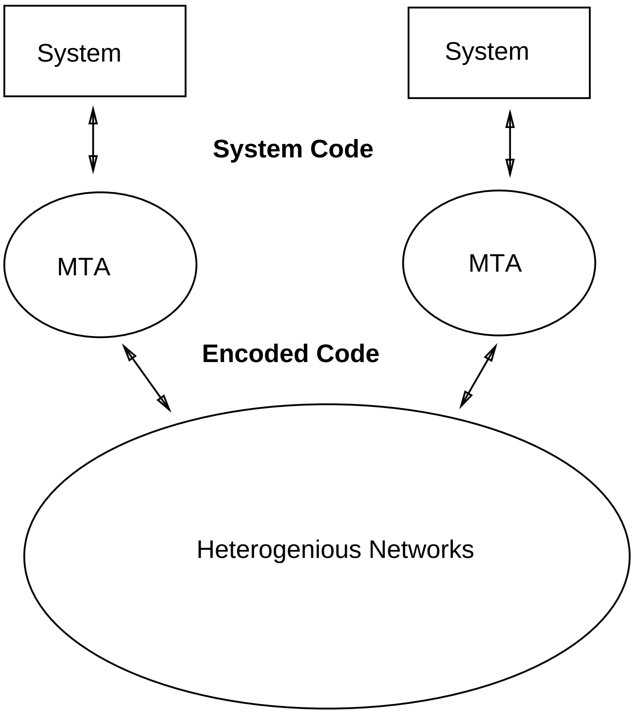

그림 2: Mail Transfer Agent상에서의 필터에 의한 구현

4 앞으로의 전망

현재의 Prototype은 KAIST를 중심으로 적지않은 사용자를 가지고 있으며 현재 SDN, KREONet의 사용자들과 해외의 교포들에게도 일부 보급되어 있다.

좀더 편리한 유저환경을 위해서는 결국 구현 이슈중의 하나였던 MTA를 고치는 방법으로 구현도 되야 한다. 이에 대한 구현은 현재 진행중이고, 또한 MIME 그룹에서 같이 나온 Draft인 Header Encoding 방법 [RFC-HDRS]에 따른 Header에서의 한글 사용의 구현도 현재 진행중이다.

또한 고려해야 할 점은 앞으로 나오게 될 KS 확장 코드를 포함하는 인코딩방법과 현재 제안되고 있는 3-Byte 한글 코드를 위한 인코딩방법들을 들 수 있겠다.

참고 문헌

[MIME] Borenstein, N.S. "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies," *Internet Draft*, Mar. 1992.

[RFC-HDRS] Moore,Keith, "Representation of Non-Ascii Text in Internet Message Headers", Internet Draft, draft-ietf-822ext-msghead-01.txt.

[ISO-DIS-10646] Draft International Standard–Information Processing–

[ISO-2022] International Standard–Information Processing–ISO 7-bit and 8-bit coded character sets-Code extension techniques, ISO 2022:1986.

[RFC-822] Cocker, D., "Standard for the format of ARPA Internet text messages," RFC-822, Network Information Center, August, 1982.