

# TCP/IP アプリケーションプロトコルにおいて漢字コードを利用可能にする一方式

## 1 Q-8

串田高幸

日本アイ・ビー・エム株式会社  
東京基礎研究所

### はじめに

TCP/IP プロトコル (インターネットプロトコル) は、最近多くのところで利用されている。TCP/IP のレイヤーアーキテクチャには、複数のコード表現に関してサポートする適切なレイヤーが存在していない。そのため、もし異なるコード体系のホストが存在する環境において TCP/IP のアプリケーションを利用する場合、それぞれのアプリケーションプロトコルにおいて独自に拡張プロトコルを定義する必要がある。

本稿においては、TCP/IP における代表的な三つのアプリケーションプロトコルに対して、複数の種類の漢字をサポートするようにそれぞれプロトコルの拡張を定義し、それを利用する環境について述べる。

### アプリケーションプロトコルの拡張

TCP/IP プロトコル上にあるアプリケーションプロトコルはインターネットのコミュニティのなかから、あるいは研究プロジェクトより生まれてきた。そのなかで、もっともよく利用されているアプリケーションプロトコルであるSMTP, FTP及び Telnetに関して、複数の漢字コードのサポートを行なえるようにプロトコルの定義を拡張した。以下の節においてそれぞれのプロトコルの拡張について述べる。

#### SMTP(RFC821) [1]

SMTP(Simple Mail Transfer Protocol)は、メールをTCP上において転送するためのプロトコルであり、インターネットでは電子メールを交換するために広く利用されている。このプロトコルの記述は、RFC821にあり、これによると「トランスポートサービスが8-bit の送信チャンネルを用意しているとき、それぞれの7-bit 文字はハイオーダーのビットを0にクリアして、右づめに送られる。」と記述されている。そして、SMTPにおいては、7-bitのASCIIコード体系のみを送ることが考えられているので、TCPにおいて8-bitのデータ幅があるにもかかわらず、SMTPにおいては7-bitのデータしか送れないということが、現状の実装ではしばしばおこってしまう。

例えば漢字コードにおいてShift-JIS漢字コードは、2 バイトの8-bitコードを使用している。そのためこの漢字コードを内部コードとして利用しているホストの間において SMTPによってメールを交換する場合、SMTPの両端において内部コードとネットワーク上を流すコードの間でコード変換を行なわなければならない。これは、同じ内部漢字コードを持っているホストの間であると冗長となってしまふ。このような場合のコード変換の冗長をなくするため、SMTPにおいてクライアントの漢字コードとサーバの漢字コードを始めに決めて、それに従ってメッセージを転送するようにSMTPのなかに拡張コマンドを定義した。

この拡張コマンドは、DATA コマンドの拡張として定義したEDATAである。EDATA コマンドは、クライアントからサーバへ

送るときにアーギュメントを持っており、それによってデータの転送コード体系を決める。次に示すのが、EDATAのフォーマットである。

EDATA <SP> code <CRLF>

code:

ASCII	ASCIIコードセット
EBCDIC	EBCDICコードセット
JIS78	JIS1978 年版漢字コードセット
JIS83	JIS1983 年版漢字コードセット
SFTJIS	SHIFT JIS漢字コードセット
IBMKJ	IBM漢字コードセット

#### 拡張コマンドEDATA

このEDATAを使った SMTP の手続きは次のようになる。まずSMTPクライアントが、SMTPサーバにEDATAコマンドを送る。もしそのコマンドが受け取られれば、codeに従ってデータをSMTPサーバへ送る。もしSMTPサーバがEDATA コマンドをサポートしていない場合、DATA コマンドによって通常のSMTPの手続きによって7-bit ASCII コードでデータを転送する。メッセージの終了は、EDATAコマンドの場合もDATAコマンドと同様に<CRLF>.<CRLF> とする。

EDATAのcode パラメータは、また上記にあるようなコードセット以外も登録し利用することが可能である。

#### FTP(RFC959) [2]

FTP(File Transfer Protocol)は、RFC959に記述されており、インターネットのホスト間において、ファイルを転送するために作られたプロトコルである。このファイルの内容は、テキスト、イメージあるいはプログラム等である。このような種類のファイルをサポートするためFTPには、Data Representation Typeを制御する機能を持っている。FTPにおいては、ASCII, EBCDIC, IMAGE及びLOCAL TYPEが、Data Representation Typeとして定義されている。このうち、ASCII及びEBCDICは、テキストファイルを転送するために使用される。

このFTPにおいて、漢字コードが入っているようなテキストファイルを転送する場合、二つのホストにおいて内部漢字コードが同じであれば問題なく転送できるが、これらのホストにおいて内部漢字コードが異なる場合、ファイルの内容が適切なコードに変換されない。

ここでは、漢字テキストファイルをサポートするため、FTPに対してプロトコルの拡張を行なった。

FTPにおいてサポートされる漢字コードは、ASCIIとEBCDICがあり、これらのデータ表現に漢字コードを表現できるようにプロトコルを拡張した。これは、FTPのなかにコードを決めるためのコマンドとして、TYPEコマンドがあるので、それを次のように拡張した。

```

      A \ / N \ / 1
TYPE  --- T --- 2
      E / \ C / \ 3
      I
      L

```

現在あるパラメータ

```

A - ASCII
E - EBCDIC
I - Image
L <byte size> - Local Byte Size
N - Non-print
T - Telnet Format
C - Carriage Control (ASA)

```

新しく加えたパラメータ

第一フィールドがASCIIのとき

```

1 JIS 1978 年漢字
2 JIS 1983 年漢字
3 Shift JIS 漢字
* それ以外は、後から付け加えることが可能

```

第一フィールドがEBCDICのとき

```

1 IBM漢字
* それ以外は、後から付け加えることが可能

```

新しく付け加えた三番目のパラメータは、一番目のパラメータに依存しており、第一パラメータが ASCII の場合と EBCDIC の場合においては漢字コードセットの意味が違ってくる。

FTP において TYPE コマンドを拡張することにより、例えば JIS1978 年版であれば、「TYPE A N I」と送ることにより TCP 上の FTP ポートには、ASCII+JIS1978 年版漢字コードが送られることになる。

Telnet(RFC854) [3]

Telnet は、ネットワーク上にあるホストのプロセスをアクセスすることを行なうために通信リンクを用意しているプロトコルである。このプロトコルは、RFC854に定義されている。

多くの種類の端末があるため、Telnet は、端末のタイプをクライアントとサーバの間でセッションを始めるときに Negotiation によって決めている。この場合、一方が要求をだし、もう一方がその要求を受け取るか拒否するかを返す。TELNET のオプションは、TELNET 自身のプロトコルとして定義されており、それらのパラメータは、別にリストとして公開されている。

ここでは、TELNETにおいて複数の漢字コードをサポートするため、そのオプションに関して漢字コードセット行なう Negotiation option を定義する。この方法では、TELNET のプロトコルを拡張する必要がないため、実装することは比較的簡単である。TELNET のオプションとして次のような漢字オプションをもうける。

```

Command name and code
KANJI-CODE <option-ID>

```

漢字コードには、いくつかの種類があるので Subnegotiation option によって、それぞれ漢字コードを決めるように定義する。下の図が、Subnegotiation option の漢字コードである。

```

Kanji code set
JIS-KANJI-1978
JIS-KANJI-1983
SHIFT-JIS-KANJI
IBM-KANJI

```

このSubnegotiation option は、TELNETがどの漢字コードを使用するかを決めているものである。また、Negotiationの手続きは、RFC930と同様な方法によって行なわれる。[4]次の例は、Telnetにおける漢字コードを決めるときの手続きである。

```

1. IAC DO KANJI-CODE
2.                               IAC WILL KANJI-CODE
3. IAC SB KANJI-CODE SEND IAC SE
4.                               IAC SB KANJI-CODE
                               IS SHIFT-JIS-KANJI
                               IAC SE

```

この例では、「1. 送信者は、Subnegotiationにおいて漢字コードを受け取ることができる。」、「2. 送信者は、Subnegotiationにおいて漢字コードのタイプを送ることができる。」、「3. (Subnegotiation)サーバは、クライアントにたいしてクライアントの漢字コードを送ることを要求する。」、「4. (Subnegotiation)クライアントは、現在の漢字コードが Shift-JIS漢字であることをサーバに送る。」となっている。

おわりに

異なる漢字コードを持つオペレーティングシステムを持っているような環境においてTCP/IPを利用するため、SMTP、FTP及びTelnetに関して、その漢字コードのサポートをよようにプロトコルの拡張を定義した。

現在のところ、このアプリケーションの拡張プロトコルをVM/SP、OS/2及び4.3BSDのあるLAN環境において利用可能になるように実装を行なっている。

## 文献

- [1] J. Postel: Simple Mail Transfer Protocol(RFC821). In ARPA Network Working Group Requests for Comments, no.821. SRI International Menlo Park, Calif. (August 1982).
- [2] J. Postel and J. Reynolds: File Transfer Protocol(RFC959). In ARPA Network Working Group Requests for Comments, no.959. SRI International Menlo Park, Calif. (October 1985).
- [3] J. Postel and J. Reynolds: Telnet protocol specification(RFC854). In ARPA Network Working Group Requests for Comments, no.854. SRI International Menlo Park, Calif. (May 1983).
- [4] Marvin Solomon and Edward Wimmers: TELNET TERMINAL TYPE OPTION(RFC930). In ARPA Network Working Group Requests for Comments, no.930. SRI International Menlo Park, Calif. (January 1985).