Network Working Group
Request for Comments: 42

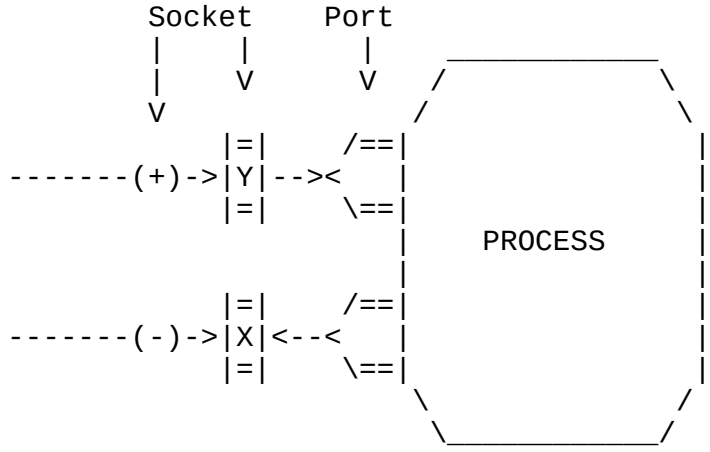
E.I. Ancona
M.I.T. Lincoln Laboratory
31 March 1970

# Message Data Types

Proposal:

We propose that the first eight bits of a normal message be reserved for a message data type. Adoption of this convention does not in any way signify agreement as to the actual data types to be used. It merely establishes the convention that the first eight bits of every normal message are not available for user data.

Discussion:




It is important that conventions regarding the contents of messages be set up early so that there will not be a large proliferation of such conventions between every pair of programs running on the network.

As network usage grows, network languages may develop for specifying both the syntax and semantics of messages. However, even before such conventions are developed, a simple way of describing such a specification is by means of a message type which both sender and receiver know how to interpret.

It is important that currently running programs still run with this convention; thus, we propose that two system programs be written which initially put in and test and remove the type information from the message. Let us call these two programs X and Y, for lack of

better names. In general, X and Y will perform transformations on the data, e.g., change character sets or number formats. As network usage grows, X and Y might become table driven with the table specified by the user.

Standard Types and Local Types:

We propose to distinguish between two kinds of message data types: standard and local.

Since our two transformation programs cannot be expected to perform a transformation between every possible data representation and the data representation of the machine they are running on, and also since the addition of a data representation should not necessarily involve a change to X or Y, we propose that only a fixed number of message types have meaning throughout the network. These are standard types.

There are two classes of local types: MYLOCAL and YOURLOCAL. A message type MYLOCAL n implies: this is type n of the set of types of the sending host. YOURLOCAL n implies: this is type n of the set of types of the receiving host.

Conventions:

A possible implementation of standard and local types is to define standard type 0 to be YOURLOCAL and standard type 1 to be MYLOCAL. In these cases, the second byte would be the local type number.

Local type 0 would mean user-specified, i.e., the message contents are unchanged and unchecked. Installations would define their own local type numbers and these would normally be available from the Network Information Center.

Thus initially, all messages sent to currently running programs will be type 0, n and all messages received from currently running programs will be type 1, n where n is the local type number of the character set of the installation.

Examples of Possible Standard Types:

0. YOURLOCAL
1. MYLOCAL
2. U.S. Ascii
3. EBCDIC
4. Mod 33 TTY Ascii

5. Load table driven translator table #n. If, in the future, the X and Y transformation boxes are table driven, this gives the table. The table number n is stored in the second byte of the message.
6. Use table driven translator table #n.
7. Network standard graphics message.

Examples of Local Types:

1. Local Character sets, e.g., Lincoln writer, DEC Ascii, etc.
2. Graphics local messages, e.g., TX-2 Apex display executive calls, GSAM.