Title: ARPA Network Functional Specifications

Author: G. Deloche

Installation: University of California at Los Angeles

Date: 5 May 1969

Network Working Group Request for Comment: 8
ARPA network: Functional area
TABLE OF CONTENTS

I  Transmission features
   1. Transmission checking
   2. HOST(A) to HOST(B) links

II  Functional software specifications
    1. User program - DEC language
    2. Network program
    3. Transmission Handler

III  Link establishment procedure
     1. General procedure
     2. Example
I. Transmission Features

I-1. Transmission Checking

There exist two kinds of transmission checking:

1. IMP to IMP
   It is a cyclic checksum computed and checked by
   the BBN hardware.

2. HOST to HOST
   It is a special 15 bit checksum computed and
   checked by the HOST programs.
   For this purpose a HOST message is broken down
   into 1152 bit pieces A, B, C, ... (1152 = 2.24 # pieces).
   For each such piece, we calculate an end-around carry
   sum and form the checksum as follows:

   Checksum = Sum of A + 2 * Sum of B + 4 * Sum of C, etc.

   This 15 bit checksum is located just after the
   marking of the HOST heading. That is at the beginning
   of the message text (See fig 4)

   This checking procedure allows the verification of
   the right IMP to IMP procedure. It also protects
   against HOST to IMP (or IMP to HOST) bad transmission,
   and against IMP packet number mismatches.
Remark: Example of an end-around carry sum:

\[
\begin{align*}
101 \\
+ 101 \\
\hline
1010 \\
\end{align*}
\]

Checksum = 011

I-2 HOST(A) to HOST(B) links.

32 links are possible between two HOSTs. Each of these links can be viewed as full duplex. Link 0 is considered as a control link (request connection status of any kind \ldots). The 31 others are used either for 'teletype-like' connections or for full transmission connections.

A "TTY-like" connection is one where:
- ASCII characters are sent or received.
- Echos are generated by the remote HOST.
- The remote HOST looks for specific ESC (break or interrupt control characters).
- The transmission is slow.
II Functional software specifications

- See fig 2 -

II-1 User program - DEL language

It's an application program that exists within a host. For example, the NLS program at SRI. For network purposes, this program should be viewed as partitioned in two: the local part and the hard part (the body).

- The hard part represents the user application.
- The local control part is the user interface. It executes immediate control of the terminal and provides specific responses to the man's inputs.

In order to facilitate and speed up remote interaction, the 'local control' program can be transmitted to another host. Thanks to that capability on UCLA, for example, we use its terminal exactly like the SRI user uses his own. Also only the program data are transmitted over the link (versus the raw terminal dialogue). See.

DEL language - (Decode Encode Language)

The 'local control' program should be written in the DEL language. When it is transmitted over to a remote

...
II-2 Network Program

This program should provide:

- The outgoing messages multiplexing (and incoming messages distribution)
- The link initiation procedure: see below
- The HOST message heading
- The "HOST HOST" checksum conjugation/checking
- The receiving of the RFNM control messages
- The supervisory control of the Handler program.

II-3 Transmission Handler Program

This program is initiated either by the network program, or by the I/O interrupt. Its function is to control the channel hardware unit.

The program is very short and closely related to the Network program.

Remark: As the communication is full duplex the Network and Handler programs can be viewed as divided into 2 parts: one is concerned with the outgoing messages, the other with the incoming messages.
III  Link establishment procedure

III-1 General procedure

- Establish link to Host (x).
  A "TTY like" connection is established to Host(x). The connection is in a pre-logon state. Standard TTY code and echo are expected. The remote Host provides the echo.

- Send/Receive characters over "TTY like" link.

- Establish file transmission link parallel to existing "TTY like" link. This must be executed by both Host and user programs.

- Send/Receive over "file like" link.

III-2 Example

Suppose that we, at UCLA, want to use NLS at SRI.

A) Local arrangements

* Log in on local TTY to Sigma 7. We are now talking to the command level of the Sigma operating system.
* Select an user program to put in execut.
We start up a program we previously wrote on
our TTY and the session with SRI...

As select the standard UCLA communication program. This is the standard option for single
control of a remote HOST.

b) Connection to SRI

Connect link to remote HOST

The previously selected program asks the UCLA network program to initiate a link to SRI. The Network
program:

- Selects an open link e.g. 25
- Sends a message to SRI and link opens
- Connects on link 25
- Waits for an acceptance from the
  SRI network program. This acceptance
  is in the form of another message over
  link 25.
- If it should happen that both SRI and
  UCLA try to initiate a connection over
  25, the one with the higher priority
  prevails. (This is extremely rare). We
  suggest that the priority be exactly
  the HOST identification number.
- This connection is teletypewriter connect.
  The network or only a standard subset of ASCII characters
  is expected or accepted.
- The connection is a "pre-log-in" conta.
  The remote HOST expects its standard
  log-in sequence.
x Log-in at SRI.
This may be done either by the UCLA "Set up" program if it knows how, or by the man at UCSC by typing the required sequence. We are now talking the command level of the SRI operating system.

2) Request 'local control' program from SRI.
* The UCLA selected program sends control over the link to the SRI user program. The SRI user requests that SRI transmit to UCLA the 'local control' program which is written in the DEL language.
* We compile this program through a compiler.
* We turn control of the TTY link a terminal over the just compiled DEL program.