

Part 1

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 spec.

TABLE OF CONTENTS.

I Transmission features

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

II Functional software specifications.

1. User program - DEL language
2. Network program
3. Transmission Handler

III Link establishment procedure.

1. General procedure
2. Example

I Transmission features

I-1 Transmission checking

There exists two kind of transmission checking:

* IMP to IMP

It is a cyclic checksum computed and checked by the BBN hardware

* HOST to HOST

It is a special 16 bits checksum computed and checked by the HOST programs.

For this purpose a HOST memory is broken down into 1152 bit pieces A, B, C ... ($1152 = 2 \cdot 2^4 \cdot 3^2$ # packets)

For each of pieces, we calculate an end-around checksum and form the checksum as follows:

$$\text{Checksum} = \text{Sum of A} + 2 \times \text{Sum of B} + 4 \times \text{Sum of C etc.}$$

This 16 bits checksum is located just after the marking of the HOST heading, that is as the beginning of a message text (See fig 1)

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 inversion.

Remark: Example of an end-around carry sum:

$$\begin{array}{r} 101 \\ + 101 \\ \hline 1010 \\ \curvearrowright \end{array}$$

Checksum = 011

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

32 links are possible between two HOSTS. Each of those links are viewed as full duplex.

Link 0 is considered as a control link (request connection, status of any kind, ...).

The 31 others are used either for "teletype like" connections or for file transmission connections.

A "TTY like" connection is one where:

- ASCII characters are sent or received.
- Echoes are generated by the remote HOST.
- The remote HOST looks for specific characters (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 parted 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 exerts 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 an UCLA user, for example, will use its terminal exactly like the SRI user uses its own. Also only the program data are transmitted over the link (versus the user terminal dialogue) - See f

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 initialization procedure : see below
 - The HOST message Heading
 - The "HOST-HOST" checksum computation/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 purpose is to control the channel hardware unit.

This 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-log-in state. Standard TTY conventions 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 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 execution.

We start up a program we previously wrote on our TTY and the transmission with SRI -

* O2 select the standard UCLA communication program - This is the standard option for simple control of a remote HOST.

b) Connection to SRI

* Indicate 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 over link 0 upon connection on link 25.
- Waits for an acceptance from the SRI network program. This acceptance is in the form of another message over link 0.
- If it should happen that both SRI and UCLA try to initiate a connection over 25, the one with the higher priority will prevail. (This is extremely rare). We suggest that the priority be exactly the HOST identification number.
- This connection is teletype-like connection only a standard subset of ASCII characters is expected or accepted.
- The connection is a "pre-log-in" connection. The remote HOST expects its standard log-in sequence.

* Log-in at SRI.

This may be done either by the UCLA user program if it knows how, or by the man at UCLA by typing the required sequence. We are now talking at the command level of the SRI operating system -

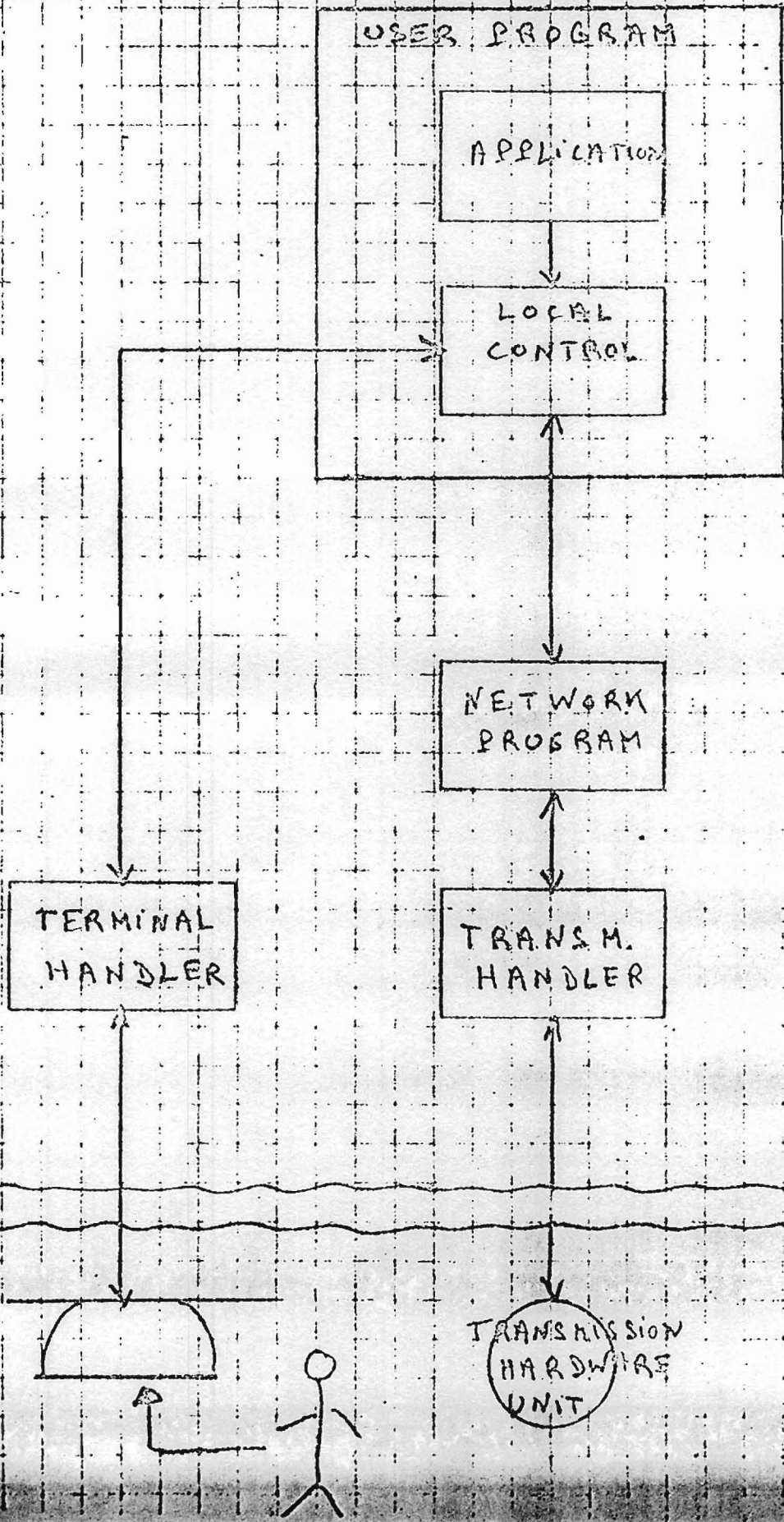
Get user program into SRI

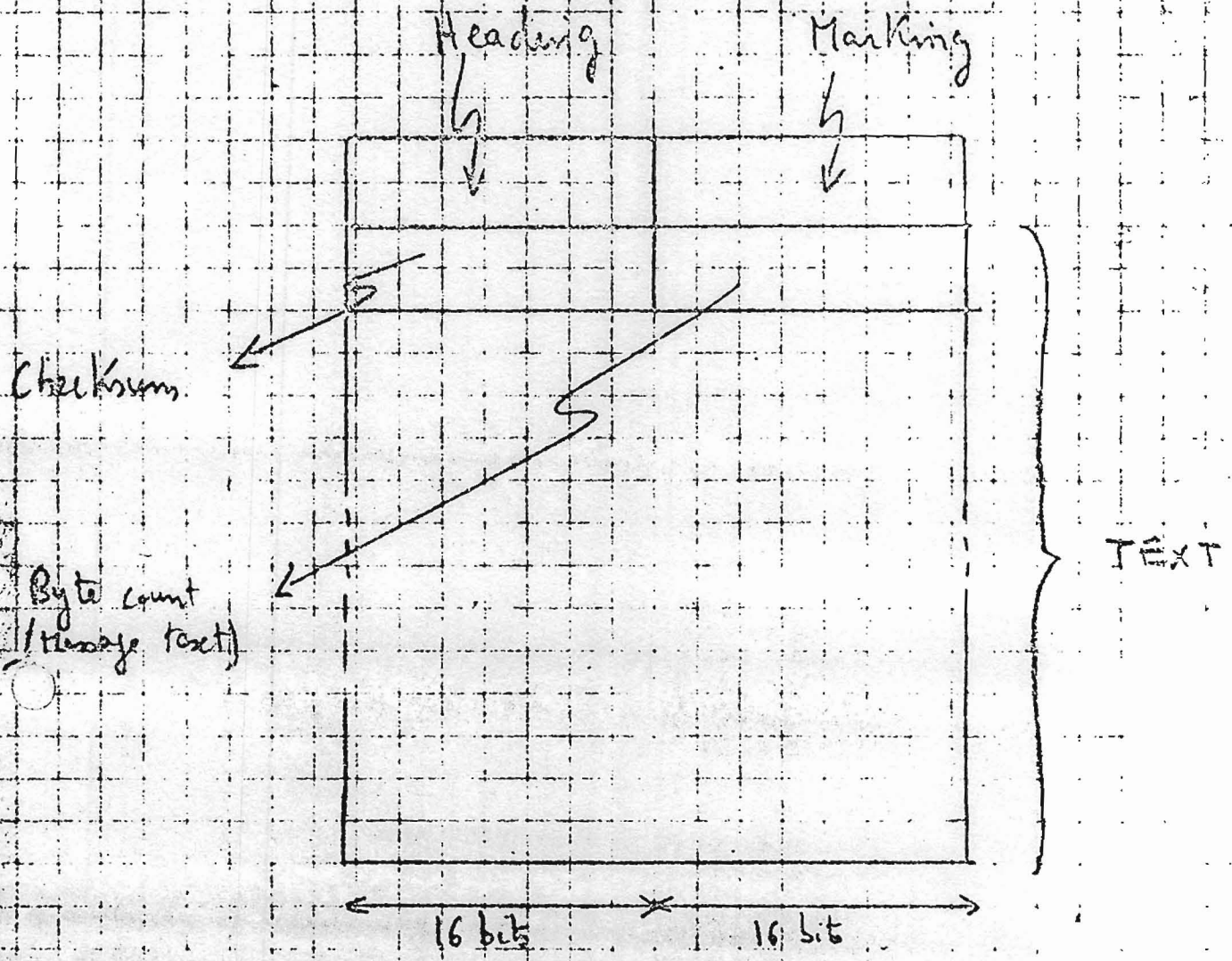
C) Request 'local control' program from SRI.

* The UCLA selected program sends message over the link to the SRI user program. The message 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.





(Fig. 1) : UCLA Host message