A Proferred Official ICP

By Initial Connection Protocol (ICP), I mean a third level protocol
which is initiated by a user process at one site in order to contact a
server process at another site. Typically, the user process will be a
Telnet and the server process will be a logger, but there may be other
cases.

In this RFC, I wish to describe a family of ICPs suitable for
establishing one pair of connections (one in each direction) between any
user process and any server process, and to propose further a particular
subset of this family as the standard ICP for connecting user processes
to loggers on systems which accept teletype-like devices.

Notation

We have no standard notation for describing system calls which initiate
and close connections or cause data to be sent, so I will use the
following ad hoc notation.

Init (local = l, foreign = f, size = s)

    causes the local Host to attempt to establish a connection between
    socket l at the local Host and socket f, with a byte size of s for
    the connection.

    l is a 32 bit local socket number,
    f is a 40 bit foreign socket number, the high-order eight bits
    of which specify the foreign Host, and
    s is an eight bit non-zero byte size.
    The sum of l and f must be odd.

Listen (local = l, size = s)

    causes the local Host to wait for a request for connection to local
    socket l with byte size s. The process will be woken when a
    connection is established. The parameters l and s are the same as
    for Init.

Send (socket = l, data = d)

   The data named by d is sent over the connection attached to local
   socket l. l must be a send socket attached to a connection. d is the
   name of a data area.

Receive (socket = l, data = d)

   The receive side counterpart to send.

Close (socket = l)

   Any connection currently attached to a local socket l is closed.

A Family of ICPs

Briefly, a server process at a site attaches a well-advertised send
socket L and listens. A user process initiates connection to L from its
receive socket U. The byte size for this connection is 32. The server
process then transmits a 32-bit even number S and closes the connection.
The 32-bit number S and its successor, S+1, are the socket number the
server will use. The final steps are for sockets S and S+1 at the
server site to be connected to sockets U+1 and U respectively at the
user site.

Using the notation, the server executes the following sequence:

   Listen (socket = L, size = 32)
   [Wait until a user connects]
   Send (socket = L, data = S)
   Close (socket = L)
   Init (local = S, foreign = U+1, size = Bu)
   Init (local = S+1, foreign = U, size = Bs)

The user executes the following:

   Init (local = U, foreign = L, size = 32)
   Receive (socket = U, data = S)
   Close (socket = U)
   Init (local = U+1, foreign = S, size = Bu)
   Init (local = U, foreign = S+1, size = Bs)

Note that L is a send socket (odd), while S and U are receive sockets
(even). Where L, S or U are used as values of local, they are 32-bit
numbers; where they are values of foreign, they are 40-bit numbers. The
parameters Bs and Bu are the byte sizes to be sent by the server and
user, respectively.

Examination of the above sequences reveals that an ICP is characterized
by the three numbers L, Bs and Bu, and must meet the restrictions that

    (a) L is a send socket,
    (b) Bs and Bu are legal byte sizes, and
    (c) for each L there is only on pair of associated byte sizes.

This last restriction prevents two distinct services from being
available through the same socket and distinguished only by the byte
sizes.

Telnet ICP

For connecting teletype-like users, i.e. interactive and ASCII, to Hosts
serving such users, I propose an ICP of the form described above and
characterized by L = 1, Bs = Bu = 8. [There has been some confusion
about socket numbers. Here I specifically mean L = X00000001]

Formalities

I propose that the Telnet ICP be made official. Comments should be
published before the May NWG meeting, the subject will be discussed
there, and we will decide there to accept or reject this protocol.

        [This RFC was put into machine readable form for entry]
          [into the online RFC archives by Jeff Sorte 5/97]