Ambrose

Network Working Group
Request for Comments #165
NIC #6779
Categories: C.3, D.1, D.3
Updates: None
Obsoletes: 123, 143, 145

J. Postel
UCLA - NMC
Computer Science
25 May 71

A Proferred Official Initial Connection Protocol*

This document specifies the third level protocol used to connect a user process at one site with a server process at another site. In one instance, the user process will be a Telnet and the server process will be a Logger, but there will be other cases.

This document describes a family of Initial Connection Protocols (ICP's) suitable for establishing one pair of connections between any user process and any server process, and further to describe the parameter values for connecting Telnets and Loggers. The description will be at two levels, the third or user level, and the second or NCP level.

Third Level Description

Notation

There is no standard notation for describing system calls which initiate and close connections or cause data to be sent, so the following *ad hoc* notation will be used.

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.

*This document is based on RFC 123 by S. Crocker and discussions by the ICP Committee.

Network Working Group
Request for Comments #165
NIC #6779
Categories: C.3, D.1, D.3
Updates: None
Obsoletes: 123, 143, 145

J. Postel
UCLA - NMC
Computer Science
25 May 71

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 local socket ℓ is closed.

A Family of ICP's

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 numbers the server will use. The final steps are for sockets S and S+1 at the server site to be connected to sockets U+3 and U+2 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+3, size = B_u)
Init (local = S+1, foreign = U+2, size = B_s)

The user executes the following:

Init (local = U, foreign = L, size = 32)
Receive (socket = U, data = S)
Optional Close (socket = U)
Listen or Init (local = U+3, foreign = S, size = B_u)
Listen or Init (local = U+2, foreign = S+1, size = B_s)

J. Postel
UCLA - NMC
Computer Science
25 May 71

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 B_s and B_u are the byte sizes to be sent by the server and user, respectively. If the user side declines to close socket U, then it must be handled automatically by the second level (see page 4).

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

(a) L is a send socket,

(b) B_s and B_u are legal byte sizes, and

(c) for each L there is only one 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.

Second Level Description

Notation

The following notation will be used for the NCP Control Command used in ICP.

STR(*ls*, *fs*, *s*)

ls = local send socket
fs = foreign receive socket
s = byte size

RTS(*ls*, *fs*, *l*)

ls = local receive socket
fs = foreign send socket
l = link

Network Working Group
Request for Comments #165
NIC #6799
Categories: C.3, D.1, D.3
Updates: None
Obsoletes: 123, 143, 145

J. Postel
UCLA - NMC
Computer Science
25 May 71

ALL($\underline{\ell}$, $\underline{m}$, $\underline{b}$)

$\underline{\ell}$ = link
$\underline{m}$ = message allocation
$\underline{b}$ = bit allocation

CLS($\underline{\ell s}$, $\underline{f s}$)

$\underline{\ell s}$ = local socket
$\underline{f s}$ = foreign socket

The same family of ICP's is now described again.

Server		User	
S1:	listening on socket L.	U1:	RTS(U, L, $\underline{\ell}_1$)
S2:	Wait for a match.	U2:	Wait for match.
S3:	STR(L, U, $\underline{s}_1$)		
S4:	Wait for allocation.	U3:	All($\underline{\ell}_1$, $\underline{m}_1$, $\underline{b}_1$)
S5:	Send data S in $\underline{s}_1$ bit bytes as allowed by allocation $\underline{m}_1$, $\underline{b}_1$.	U4:	Receive data S in $\underline{s}_1$ bit bytes.
S6:	CLS(L, U)	U5:	CLS(U, L)
S7:	RTS(S, U+3, $\underline{\ell}_2$)	U6:	STR(U+3, S, $\underline{s}_2$)
S8:	STR(S+1, U+2, $\underline{s}_3$)	U7:	RTS(U+2, S+1, $\underline{\ell}_3$)

The labels here imply no ordering except that ordering required by the Host-Host Protocol. Note that steps S7 and S8 can be reversed as can U6 and U7. Also, notice that at any time after S2 the server could initiate steps S7 and S8 in parallel with steps S3 through S6, and that at any time after U4 the user could initiate steps U6 and U7 in parallell with step U5.

J. Postel
UCLA - NMC
Computer Science
25 May 71

Following the above exchanges ALL commands would be exchanged and data transfers could begin.

At this level the parameters of the above ICP family are L, $\underline{m}_1$, $\underline{b}_1$, $\underline{s}_1$, $\underline{s}_2$, $\underline{s}_3$, $\underline{\ell}_1$, $\underline{\ell}_2$, $\underline{\ell}_3$.

L is a well known socket number and will be specified for each type of service.

m_1 and b_1 are allocation quantities for the transfer of a socket number.

$\underline{m}_1$ is specified to be 1.

$\underline{b}_1$ is specified to be 32.

$\underline{s}_1$, $\underline{s}_2$ and $\underline{s}_3$ are byte sizes. Only $\underline{s}_1$ is to be specified as $\underline{s}_2$ and $\underline{s}_3$ are to be left to the process involved.

$\underline{s}_1$ is specified to be 32.

$\underline{\ell}_1$, $\underline{\ell}_2$, and $\underline{\ell}_3$ are links and are not specified.

Note: Some hosts currently have difficulty sending 32 bit bytes. Thus, it is *temporarily* allowed to send the socket number S as four 8 bit bytes in one message.

It is legal for the NCP to receive RTS or STR before the corresponding local Init or Listen is issued. Therefore it is suggested that requests for connection to idle sockets be queued as allowed by time and space limitations.

Telnet - Logger ICP

For connecting Telnet and Logger processes, the ICP parameters are L=1, $B_u = \underline{s}_2 = 8$, and $B_s = \underline{s}_3 = 8$. (To clarify the socket number required, L = X'00000001').

Network Working Group
Request for Comments #165
NIC #6779
Categories: C.3, D.1, D.3
Updates: None
Obsoletes: 123, 143, 145

J. Postel
UCLA - NMC
Computer Science
25 May 71

Formalities

This proposed official protocol will become official if no serious objections are raised before 2 June 71. A telephone survey of Network Liaisons will be conducted by Jon Postel before that date. If no objections are raised this protocol will be declared official by the Working Group chairman.