TENEX NCP Finite State Machine for connections
TENEX Memo Number 155

The attached figure describes the finite state machine used in TENEX version 1.33 to implement the ARPANET Host to Host protocol. This memo updates that of 27 August 1971, TENEX memo number 113, with regard to the finite state machine. Other parts of that memo have been incorporated into the JSYS manual and other documents.

The components of a finite state machine (FSM) are States, Events, and Actions. These are listed below:

States Events Actions

01 CLZD 00 RRFC 00 ANOP
02 PNDO 01 CLSR 01 AFNY
03 LSNG 02 CLSS 02 ACLS
04 RFCR 03 CLZR 03 ARFC
05 CLW1 04 CLZS 04 AOPB
06 RFCS 05 ACPT 05 AOPL
07 OPND 06 CONN 06 ACLL
10 CLSW 07 LISN 07 ACLO
11 DATW 10 RRFN 10 AEOR
12 RFN1 11 TIME 11 AEOS
13 CLZW 12 RRFB 12 AES1
16 FREE 13 AABT
14 ACKA

Note that there are two kinds of "close" events and actions: a close at the JSYS level (CLOSF) and one at the host-to-host protocol level (CLS). The names in the above list contain "CLS" if they are concerned with host-to-host CLS, and CLZ if the are concerned with CLOSF.

Each state will be briefly described below, along with events which may occur while a connection is in that state, and actions which are taken as the state is advanced.

A few overall notes: Actions are shown on the state diagram following a "/''. Any transition without an action shown generates "ANOP", a null action. The action "AFNY" means a "funny" event (i.e., one not expected in this state -- probably a bug) has occurred. The result of this action is an IMPBUG error printout. Any event NOT shown on the state diagram causes the state to loop back to itself and an AFNY to be generated. These are not shown explicitly on the diagram. Another "funny" event is the execution of an "accept" MTOPR by the user program when the state is not RFCR. However, an errant user program can do this, and no IMPBUG should be generated as a result, so most states show a loop to self with no action (i.e., an "ANOP") as a result of the ACPT event. The event "TIME" (also called "HUNG") simply means that the socket has not changed state for a specified time interval and may need to be prodded along. This time is currently two minutes, except for connections in error states where the FSM is stepped faster to clear the connection out.

State 16 - FREE - Free connection
A connection in this state has never existed, or is almost completely deleted. No events are expected in this state except a program CLOSF (events CLZR or CLZS), or an erroneous Accept. The Accept causes an error status bit to be set (AABT - Action: abort). A connection leaves the FREE state by the creation of a socket. This can be the result of a user OPENF or an incoming RFC. This causes the connection

to move into the CLZD state.

State 01 - CLZD - Closed
This state is very transitory. Events which cause a socket to be created also immediately cause a further state change to either PNDG (if the event was the receipt of an RFC), LSNG if the event was a Listen JSYS - i.e., an OPENF with a null extension), or RFCS (if the event was an OPENF with a foreign socket specified). In the latter case, an RFC is sent to the foreign socket (ARFC).

State 02 - PNDG - Pending
In this state, an RFC has been received, but no local program has indicated any interest in it. Events which may occur here are:
LISN - a program decides to listen on the socket. Since an RFC has arrived, the FSM is stepped to the same place it would have been if the LISN and RRFC had occurred in the other order; namely RFCR.
CONN - a program decides to connect to the foreign socket. This steps the FSM to OPND and causes the AOPB action (open the link and send the matching RFC).
TIME - The RFC (which was unsolicited) has sat around for two minutes without any local program deigning to act upon it. Therefore, to clear out the tables, it is refused (ACLS - a close is sent) and the next state is CLSW to await the matching CLS.
CLSR or CLSS - The site who sent the RFC has changed its mind and sent a CLS, so we send the matching CLS and return the socket to the FREE state.

State 10 - CLSW - Wait for CLS
This state is entered when no further activity is required on a connection except the receipt of a CLS from the far end. The CLS has already been sent to the far end. In some cases, the file system CLOSF has not been done, so events CLZR and CLZS cause loops into CLSW. The expected events are CLSR or CLSS, the receipt of the CLS from the far end of the connection, and these step the connection to state FREE. Also, if two minutes go by without the CLS, it is assumed that the required CLS response was lost by the foreign site so TIME causes the same action.

State 03 - LSNG - Listening for RFC
In this state, no network activity has occurred. The local program is, however, waiting for some attempt at connection. This state does not time out (event TIME loops into LSNG). The program may decide to stop listening (e.g., it may simply log out of TENEX) so events CLZR and CLZS move the connection to state FREE. The desired event is the receipt of an RFC for this socket. When an RFC arrives for a socket in a state other than CLZD, it is checked to see that the byte size matches that declared by the user program. It then becomes event RRFC if the byte size matches, or RRFB if it does not match. RRFB causes a LSNG socket to go to state CLSW and a CLS to be sent, refusing the connection. In the normal case, event RRFC steps the connection to state RFCR.

State 06 - RFCS - An RFC has been sent.
This state is entered when a program does a Connect (an OPENF with a specific foreign socket). The RFC has been sent, and the matching RFC is awaited. If the foreign site refuses the request for connection, sending a CLS, events CLSR and CLSS cause the matching CLS to be sent (ACLS) and the socket is stepped to FREE state. If the user program gives up waiting for the acceptance of the connection (events CLZS or CLZR), a timeout happens (TIME) or an RFC arrives but has the wrong byte size (RRFB), the socket is stepped to CLSW and a CLS is sent (ACLS). In the case of a properly accepted connection (event RRFC), the link tables are opened (AOPL) and the connection state goes to OPND.

State 04 - RFCR - An RFC has been received.
This state is reached when a Listen has been done by the user, and an RFC (with matching byte size) has been received. This state and the PNDG state make up the queueing mechanism for received RFC's. Queued RFC's in PNDG state are for sockets which are not listening (although another connection with the same socket number may have been listening -- this is just not the first one), so they time out after two minutes. In RFCR, though, a program has done a listen. Therefore, timeouts are suppressed to allow the accept to be done (event TIME loops to RFCR).
If the foreign site tires of waiting for the accept, it may send a CLS (events CLSR or CLSS) in which case the matching CLS is sent (ACLS) and the socket moves to state FREE.
Also, while in this state, the program may examine the foreign host and socket and decide to refuse the connection by doing a CLOSF (events CLZR and CLZS). This causes a CLS to be sent (action ACLS) and the socket steps to CLSW to await the matching CLS.
If the program likes the request for connection, it will accept it with an MTOPR JSYS (event ACPT), causing action AOPB (sending the matching RFC and opening the link tables), and the socket steps to state OPND.

State 07 - OPND - opened.
This is the state during which data may be transferred. Both RFC's have been sent. Allocation and RFNM activity are not considered in this state diagram, but until one end or the other tries to close the connection that activity proceeds at another level of the code. The exception is event TIME. After two minutes of inactivity on the socket, action ACKA (Check allocation) occurs. This action causes allocate resynchronizing to occur if the foreign host is known to understand that extension to the host-host protocol.
The remaining events and states are all associated with properly getting a connection closed and free, from the OPND state. This gets complicated. There are four initial events: closes done locally and closes from the remote end of the connection, each of which may be for a sending or a receiving connection. These are CLZR, CLZS, CLSR and CLSS.
CLZR - The local program closes a connection which has been receiving. A CLS is sent (ACLS) and the state is stepped to CLW1 which is similar to CLSW except that the link tables must be closed when a CLS comes

back.
CLZS - The local program closes a connection which has been sending. The state is stepped to DATW and action AEOS is performed. AEOS (action end of send) sets a done flag which lower level routines use to signal the FSM when the last data has been sent and acknowledged by RFNM.
CLSR - a CLS is received on a receive connection. This is the normal "end of file" case. Action AEOR (end of receive) occurs and the state moves to CLZW. AEOR causes the same flags to be set as AEOS, and in addition (if this connection is hooked to an NVT) causes an NVT DETACH to be performed.
CLSS - a CLS is received on a send connection, before we (the sending end) had closed it. Action AESl occurs, which is the same as AEOS except that messages queued to go out are discarded. The state moves to RFN2 to await the final RFNM.

State 05 - CLW1 - Close wait sub 1.
This state is the same as CLSW except that a link table entry remains to be cleared out. A CLS has been sent and we await the matching CLS. When it arrives, or two minutes go by, the state is stepped to FREE and action ACLL (close link) is performed.

State 11 - DATW - Final data wait.
This state is NOT the normal waiting for data on an open connection. It is only during the sending of the last of the data on a send connection which has been locally closed but which has not yet had all the data accepted by the far end.
RRFN - Received final RFNM - event is the desired one. When this occurs, all the data has been sent so we send a CLS (ACLS) and go to state CLW1 to await the matching CLS.
TIME may also occur. If it does, we pretend we had seen the final RFNM and act as for RRFN. This timeout may occur either because an RFNM has been lost by the IMP or the subnet, or because of unresposiveness on the part of the foreign host. The latter could occur if the amount of data to be sent when the CLOSF is done exceeds the available allocation at that time. If the foreign host does not send allocation, or disagrees with us and thinks allocation is outstanding, the timeout will free the socket.
CLSS may occur. If so, the far end has not accepted all the data, but wants to abort the connection. This is treated about as it was when it occurred from the OPND state, namely AES1 action, but must go to a different next state, RFN1, to distinguish the fact that a local CLOSF has already occurred.

State 14 - RFN2 - Final RFNM wait sub 2.
This state means that an unexpected close arrived when we were happily sending data. We now await the final RFNM for any outstanding data messages. When this occurs, or if there were no outstanding messages, the state moves to CLZW. ACLO is performed, sending a CLS (matching the unexpected CLS which arrived during OPND) and closing the link. If two minutes go by and no final RFNM arrives, we also just go to CLZW to prevent being hung by an unresponsive foreign host or an IMP/subnet failure, also performing ACLO. This is analogous to the

timeout from state DATW above.
While waiting for the RFNM, the local program may try to CLOSF the connection (CLZR, CLZS). If so, we go to state RFN1 and continue to await the final RFNM.

State 12 - RFN1 - Final RFNM wait sub 1. In this state, it’s all over but the final RFNM. When it arrives (RRFN) or two minutes go by (TIME), we close the link tables and send a CLS to the foreign site (ACLO). Since the CLS has already been received, the connection is now gone and we step it to FREE.

State 13 - CLZW - Wait for program to close file.
This is the normal state after an end of file when receiving, before the CLOSF (CLZR) has occurred. It is also the state of a send connection which was aborted (CLS) by the receiver (foreign host) when all sent messages have been RFNM’ed but the local CLOSF (CLZS) has not been done. We wait for the local program to do the CLOSF. When it does, we close the link table if this is a receive connection, and in either case step the connection to the FREE state.