



The following algorithm is used to decide whether the circuit between an IMP and a very distant Host is dead or alive. We first define what we call a special packet -- this is (logically) a one word packet consisting of only the control word and having the SPECIAL PACKET bit set to one. All packets which are not special packets (i.e., which are regular data packets or null packets) have the SPECIAL PACKET bit set to zero. In a special packet, none of the control word fields or bits have their usual meanings; consequently, a special packet cannot be used to acknowledge data packets or send data. In a special packet, only one bit other than the SPECIAL PACKET bit has any meaning, the HELLO/I-HEARD-YOU bit.

Every  $r$  seconds both IMP and Host (independently) send a HELLO packet, a special packet with the HELLO/I-HEARD-YOU bit set to zero. When either IMP or Hosts receives a HELLO packet, it must promptly (with highest priority) send the other an I-HEARD-YOU packet, a special packet with the HELLO/I-HEAR-YOU bit set to one. In other words, the I-HEARD-YOU packet is an acknowledgement of the periodic HELLO packet, and a I-HEARD-YOU packet must only be sent as acknowledgement for a HELLO packet. If either IMP or Host sends more than  $t$  HELLO packets without receiving an I-HEARD-YOU packet in acknowledgement, the IMP or Host declares the line dead. Once either IMP or Host declares the line dead, it must send or accept no packets (either special or regular) for  $2*t*r$  seconds to allow the other party also to declare the line dead. After waiting  $2*t*r$  seconds, an attempt is made to bring the line alive. This is done by sending HELLO packets (but no regular packets) every  $r$  seconds while noting received I-HEARD-YOU packets until  $k$  HELLO packets in a row are acknowledged with I-HEARD-YOU packets. While doing this, received HELLO packets must be acknowledged with I-HEARD-YOU packets. Once acknowledgement for  $k$  HELLO packets have been received in a row (i.e., one acknowledgement every  $r$  seconds for  $k$  intervals[1]), the line is declared alive, and regular packets again may be sent, received, and acknowledged along with the periodic (every  $r$  seconds) HELLO packets. If a regular data packet is received while a party is trying to bring the line up (due perhaps to slight timing differences between the parties at the ends of the line), the data packet must not be acknowledged.

The odd/even bits, the used/unused bits, and the channel filling and emptying sequences must be initialized at start up[2] and reinitialized every time the line is declared dead. If either the IMP or Host decides the line is dead, the same action is taken as the IMP or Host normally takes when the other's ready line is down. The line being up causes the same action as is normally taken when the ready line is up. The value of  $r$  is currently 1.25 seconds, the value of  $t$  is currently 4, and the value of  $k$  is currently also 4.

It is likely that the values of  $r$ ,  $t$ , and  $k$  will be adjusted in the future; very distant Host programmers are advised to make it easy to change these parameters.

#### Endnotes

[1] In particular, the IMP implementation requires the receipt of an acknowledgement within  $r$  seconds of the transmission of a HELLO packet in order to consider that the HELLO packet was successfully acknowledged.

[2] At start-up, the line must be assumed to be dead and the procedure of waiting  $2*t*r$  seconds before sending HELLO packets, etc. must be used to bring the line alive initially.

[ This RFC was put into machine readable form for entry ]  
[ into the online RFC archives by Jeff McClellan 1/98 ]

