IEN # 15
Supercedes: None
Replaces: None

William W. Plummer
BBN
15 September 1976

Section: 2.4.4.2

Thoughts about
TCP Retransmission Techniques

William W. Plummer
Bolt Beranek and Newman, inc.
Cambridge, MAss 02138
15 September 76

The current TCP retransmission scheme is the bare minimum which can be used. Even so, it consumes alot of machine time.

The retransmission scheme currently used by TENEX TCP works as follows: every packet (re)transmitted is marked with a desired next retransmission time and placed on a per-connection retransmission queue. The retransmission process is then scheduled to wakeup at the earliest retransmission time for all connections. When it is activated, the retransmitter scans all connections, retransmitting any packets which have elaspsed deadlines. The next retransmit time for the packet is formed by doubling the previous retransmit interval and adding it to the current time.

Thus, every time RX awakens, every un-ACKed packet on every connection is examined. It is likely that only a small fraction of these will actually be acted on. But it is also likely that the next wake up for the retransmitter will be very near in the future. This suggests:

(1) Make the retransmitter operate on a per-connection basis.

and,

(2) Establish a minimum time between activations of the retransmitter for any one connection.

The need for a retransmission scheme at all is due to packets being dropped or corrupted during transmission. In either case it is desireable to design the retransmitter so that it tends to avoid whatever disaster befell the first attempt at transmission. In the case of packets being garbled, the cause can be either internal (a failing gateway) or external (a noisy channel). To avoid internal garbling,

(3) If possible, route every transmission of a particular packet via a different gateway.

Using different gateways for each retransmission also will help minimize external interference in some cases. If the interference is continuous and situated in a bad place (right in front of the receiver), there is little that can be done to get packets through. However, if the interference is periodic, even if it is "off" just a small percentage of the time, there is hope. However, under such conditions it is unlikely that retransmissions every 10 to 60 seconds will work -- a 90% loss rate and 10 second retransmit interval yields a throughput of 0.01 packets per second.

Clearly, the retransmitter should "try harder" by retransmitting more often and by being more selective about what is transmitted under these circumstances. If a packet has been lost (in the sender-to-receiver direction), a hole will eventually appear at the left window edge in the receiver while the packet which will fill that hole will be at (or contain) the left window edge of the send window in the sender. This is the packet which is most critical and it should be retransmitted at a much higher rate than the others. The same argument applies in low-loss channels where getting a replacement through for an occasional dropped packet may mean that the next ACK received will acknowledge the entire send window.

(4) The first packet on the retransmit queue should be retransmitted N-times (N = 2?) as frequently as the second packet. The second packet should be retransmitted N-times as frequently as the third, ... etc.

The disadvantage of this suggestion is that it causes many more packets to be emitted than the simple scheme. Also, it uses much more of the machine. These effects can be minimized by running the retransmitter only when it is suspected that it is needed. This can be done by watching the rate at which packets are being ACKed (removed from the retransmission queue). If this rate drops some minimum, the retransmitter should be activated. Thus,

(5) Record with each packet the time at which it was last transmitted and a predicted time by which it should have been acknowledged (= transmit time plus two round-trip delays). Schedule the retransmitter to run at the latter time. When each ACK is processed, the round-trip delay estimate can be updated by computing a running average of "now" minus time of transmit) and the retransmitter can be rescheduled to a later time.

In accordance with (2) we will only schedule the retransmitter to run at times which are (say) at multiples of ten seconds. A simple way of rescheduling the retransmitter is to do it on the basis of the next packet in the queue if that means the retransmitter will be activated at a later time. If not, the packet after that is tried, etc. This consideration is necessary because the retransmit queue is not orderred by sequence number and recently transmitted packets are mixed with older packets. This fact also complicates the ACK processor since it must scan the entire retransmit queue every time an ACK is received.

(6) The retransmit queue for each connection should be ordered by sequence number.