

JULY 22, 1987

THE NEVER ENDING STACK GARBAGE PROBLEM

Steve Bellenot

A collection of conditions which show that stack garbage is a real pain.

1. $SVT < GVT \leq LVT$.

In particular, we can't send messages for now. Also if we require Time Warp to always save the state, then there is no stack garbage.

2. Object A has sent two query messages of the same length to object B at time "now".

In particular, putting a limit of one (or zero) queries per object per virtual time of each length would eliminate the "problem" below.

3. The query messages A sends to B has holes.

For example, struct { char x; int y } maybe compiled to something like:

-x-		big hole
-y-		

There have been several claims that having the application clear the message buffer before using it would solve the stack garbage problem. Somewhat user unfriendly, but suppose he/she then does a "struct copy" (it could copy the hole). What if the user knows his struct has no holes, but forgets to fill in a field.

THE "PROBLEM":

Time Warp gets the first of the qrymsg calls, both query replies are waiting in the input queue, but how does Time Warp pick the correct one? The query replies are matched to the original query messages, both in the output queue, but the stack garbage prevents the current query text from matching either query output.