Tuning Null Messages

In general, a null message is a "promise" that no future message will have a smaller time stamp.

Some simulations (depending on what is being simulated) can guarantee that every event scheduled, at any time T, will not be scheduled before T + L.

L is called the lookahead for the simulator.

Hence, a null message sent at Sim time = T can safely have a time stamp of T + L.

Why is this useful?

Having a larger timestamped event on Q_K allows other queues (Q_1, Q_2 ...) to be processed, using Invariant 2, without needing more messages from S_K

When to send a null event?

- After firing any event
  (suggestion of inventors of algorithm)
- Demand-driven (pull): whenever S_i is "stuck", request S_j to send a null message, where Q_j is empty. Better yet, be proactive: S_i can request a null message from S_j when the size of Q_j becomes small and Q_j has the least time stamp of any event.

Suppose all simulators S_1, S_2 .. S_n have looked ahead L (not unreasonable)

Let T_s be simtime of smallest clock of any of S_1 S_2 ... S_n

Claim it is safe for any simulator to fire any event with timestamp in $[T_s, T_s + L]$, regardless of Invariant 2.

Other ideas: do away with null messages, but use a "deadlock detector" algorithm. After deadlock is detected, the simulators can go ahead using the smallest event in the network (in any message or queue).