Selective Repeat state machine

25. marraskuuta 2017 17:09

Rdt_rcv && ack && not corrupt && seqnum in window

_____

Mark as pkt ack'd[seqnum]
[canceltimer]
If seqnum == base {base = smalest unack'd seqnum}
Send buffered data{make pkts; send_pkts; start timers;nextseqnum+n_pkts;}

Default

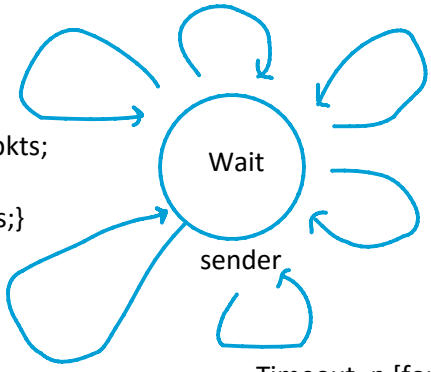
_____

Do nothing

Rdt_send && next_seqnum_ok [nextseqnum in window]

_____

Make pkt
Send pkt(nextseqnum,data,checksum)
Start timer (pkt)
Next_seqnum++



Rdt_send && !next_seqnum_ok

_____

Add data to buffer

Rdt_rcv && ack &&not corrupt && seqnum not in window

_____

Do nothing

Timeout_n [for packet n]

_____

Send_pkt(n)
Start timer(n)
--OR--
Single timer withsmall interval AND timestamp tracking for packets

Rdt_rcv & seqnum in [rcv_base, rcv_base+N] & not corrupted

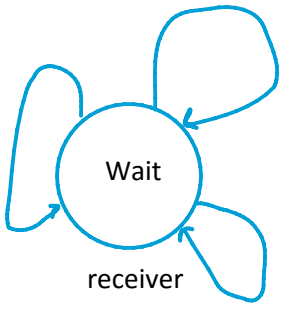
_____

Send_ack[seqnum];
Add to rcv_buffer;
If seqnum == rcv_base{for pkts in buffer {send data to upperlayer; remove from buffer;}do while there are pkts in consecutive;
Rcv_base = rcv_base+num_removed_pkts;

Rdt_rcv & seqnum in [rcv_base-N, rcv_base-1] & not corrupt

_____

Send_ack[seqnum];



Default

_____

Do nothing (except log)

Implmentation will use timeout as described in Kurose
Another way to do this would be resend-control messages sent by receiver when it detects skips in received seqnums. Implementation may make use of singel timer to simulate per packet timers…