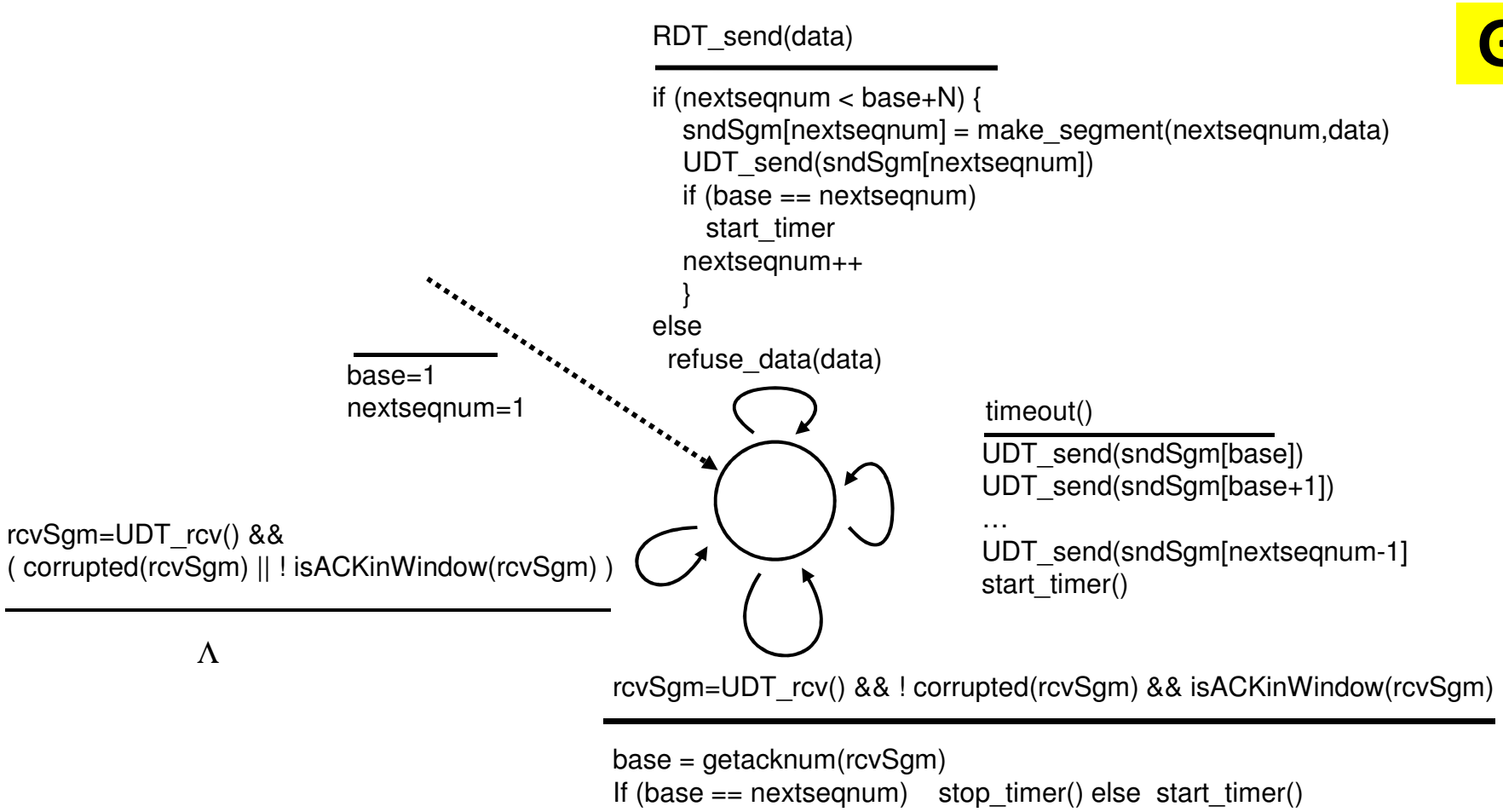
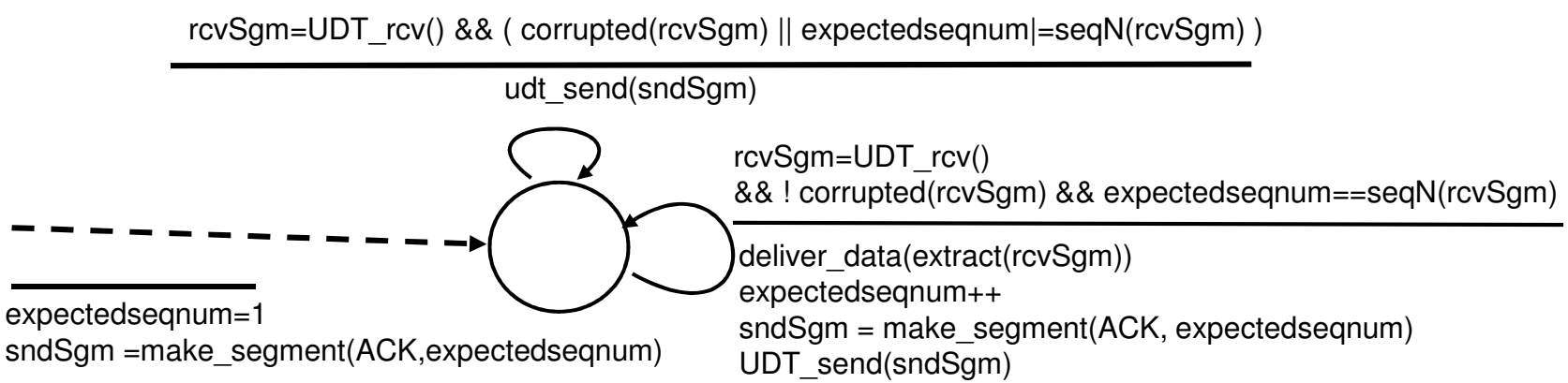
GBN


RDT_send(data)
if (nextseqnum < base+N) {
sndSgm[nextseqnum] = make_segment(nextseqnum,data)
UDT_send(sndSgm[nextseqnum])
if (base == nextseqnum)
start_timer
nextseqnum++
}
else
refuse_data(data)
base=1
nextseqnum=1
timeout()
UDT_send(sndSgm[base])
UDT_send(sndSgm[base+1])
…
UDT_send(sndSgm[nextseqnum-1]
start_timer()
rcvSgm=UDT_rcv() &&
(corrupted(rcvSgm) || ! isACKinWindow(rcvSgm))
Λ
rcvSgm=UDT_rcv() && ! corrupted(rcvSgm) && isACKinWindow(rcvSgm)
base = getacknum(rcvSgm)
If (base == nextseqnum) stop_timer() else start_timer()


rcvSgm=UDT_rcv() && (corrupted(rcvSgm) || expectedseqnum|=seqN(rcvSgm))
udt_send(sndSgm)
rcvSgm=UDT_rcv()
&& ! corrupted(rcvSgm) && expectedseqnum==seqN(rcvSgm)
deliver_data(extract(rcvSgm))
expectedseqnum++
sndSgm = make_segment(ACK, expectedseqnum)
UDT_send(sndSgm)
expectedseqnum=1
sndSgm =make_segment(ACK,expectedseqnum)