# Sender

rdt_send
_____
sndpkt=make_pkt(0, data, checksum)
udt_send(snd pkt)
timer_start

**wait for call 0 from above**

**wait for ACK 0**

rdt_rcv(rcvpkt)
&&(corrupt(rcvpkt)
|| is Ack(rcvpkt, 1))
_____
∧

rdt_rcv(rcvpkt)
_____
∧

timeout
_____
udt_send
(snd_pkt)
start-timer

rdt_rcv(rcvpkt)
&& not corrupt(rcvpkt)
&& isAck(rcvpkt, 0)
_____
stop timer

rdt_rcv(rcvpkt)
_____
∧

**wait for call 1 from above**
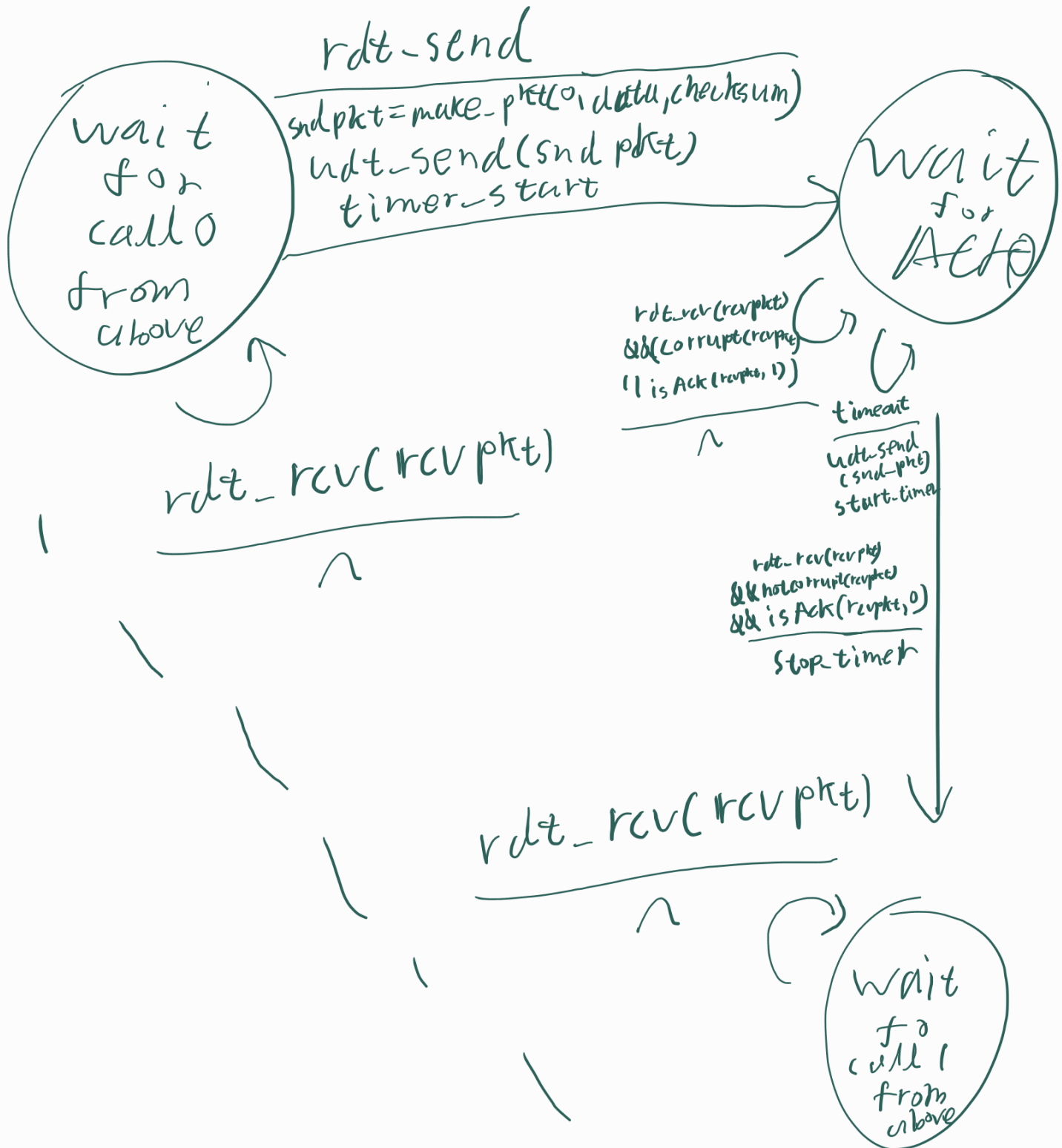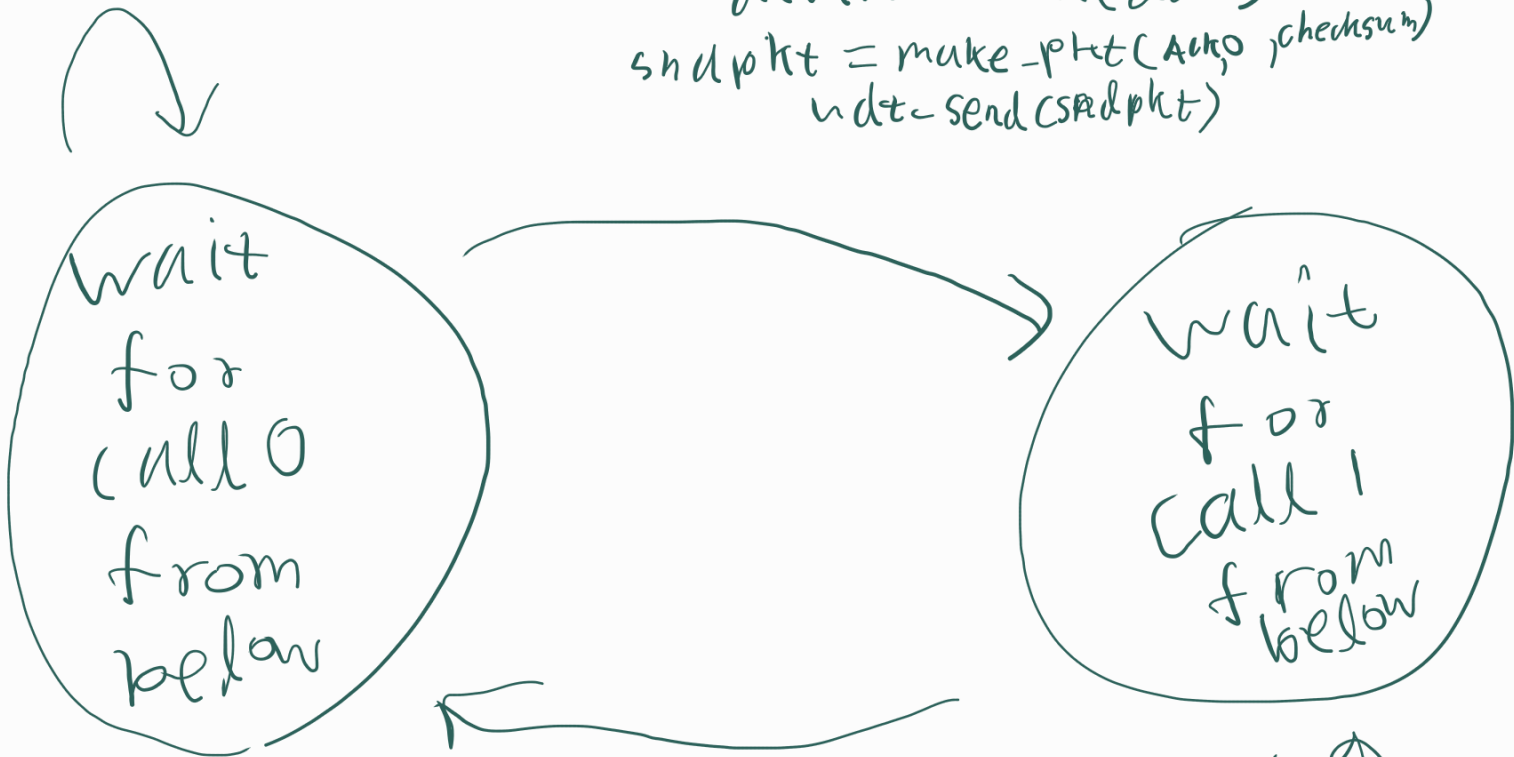
# reciever

rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& has_seq1(rcvpkt)
———————————
sndpkt = make_pkt(ACK,1, checksum)
udt_send(sndpkt)

rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& has_seq0(rcvpkt)
———————————
extract(rcvpkt, data)
deliver_data(data)
sndpkt = make_pkt(ACK,0, checksum)
udt_send(sndpkt)

**wait for call 0 from below**

**wait for call 1 from below**

rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& has_seq1(rcvpkt)
———————————
extract(rcvpkt, data)
deliver_data(data)
sndpkt = make_pkt(ACK,1, checksum)
udt_send(sndpkt)

rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& has_seq1(rcvpkt)
———————————
sndpkt = make_pkt(ACK,1, che
udt_send(sndpkt)