

seq = ... (start sequence number)  
ack = 0

Client appl. initiates a tcp connection  
sendpkt = makepkt(seq,S)  
ip\_send(sendpkt)  
start\_timer

Server application creates a listen socket  
 $\epsilon$

timeout  
ip\_send(sendpkt)  
start\_timer

ip\_rcv(rcvpkt)  
not( \* )  
ip\_send(sendpkt)  
start\_timer

syn-sent

listen

timeout  
ip\_send(sendpkt)  
start\_timer

ip\_rcv(rcvpkt)  
has\_S(rcvpkt)  
ack=getseqnum(rcvpkt)+1  
sndpkt=make\_pkt(seq,ack,A,S)  
ip\_send(sendpkt)  
start\_timer

ip\_rcv(rcvpkt)  
not( \*\* )  
ip\_send(sendpkt)  
start\_timer

syn-rcvd

ip\_rcv(rcvpkt)  
notcorrupt(rcvpkt)  
has\_S(rcvpkt)  
has\_A(rcvpkt)  
isAck(rcvpkt,seq+1) } \*  
stop\_timer  
seq=seq+1  
ack=getseqnum(rcvpkt)+1  
sndpkt=make\_pkt(seq,ack,A)  
ip\_send(sendpkt)

ip\_rcv(rcvpkt)  
notcorrupt(rcvpkt)  
has\_A(rcvpkt)  
isAck(rcvpkt,seq+1) } \*\*  
stop\_timer

established

Variablen:  
seq: INT  
ack: INT  
sndpkt: TCP/IP PDU