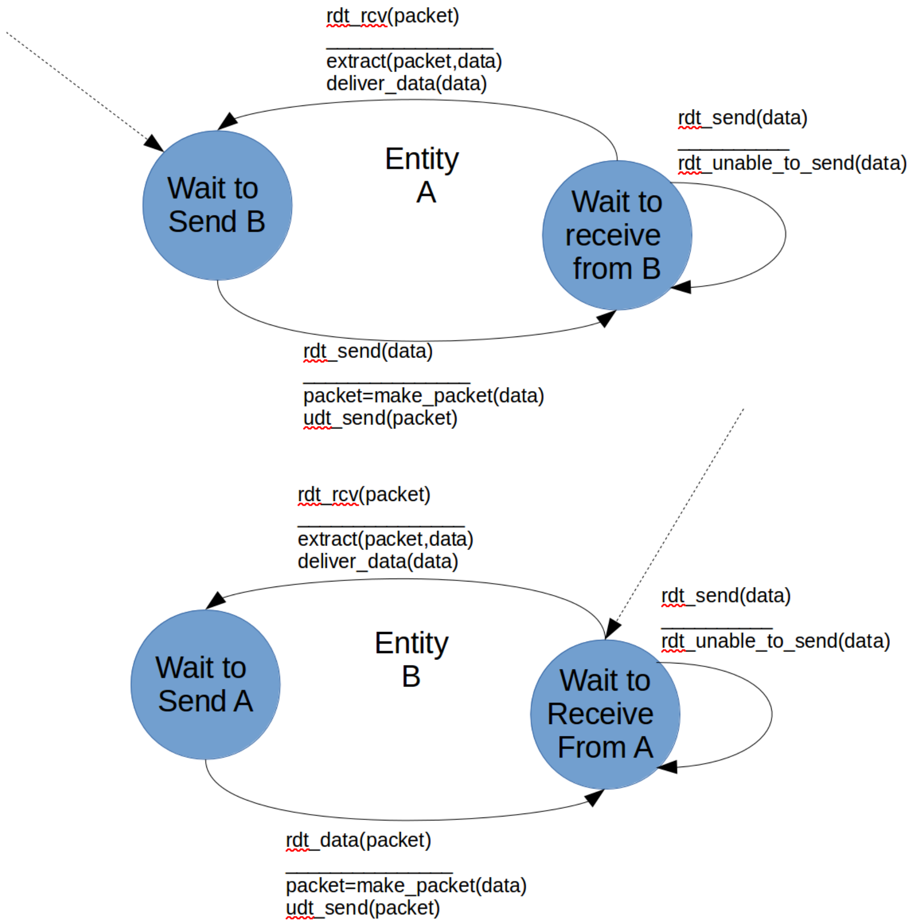
Lab Problems № 1

Steven Seppala & Seth Decker 30 October 2014

Lab Problems

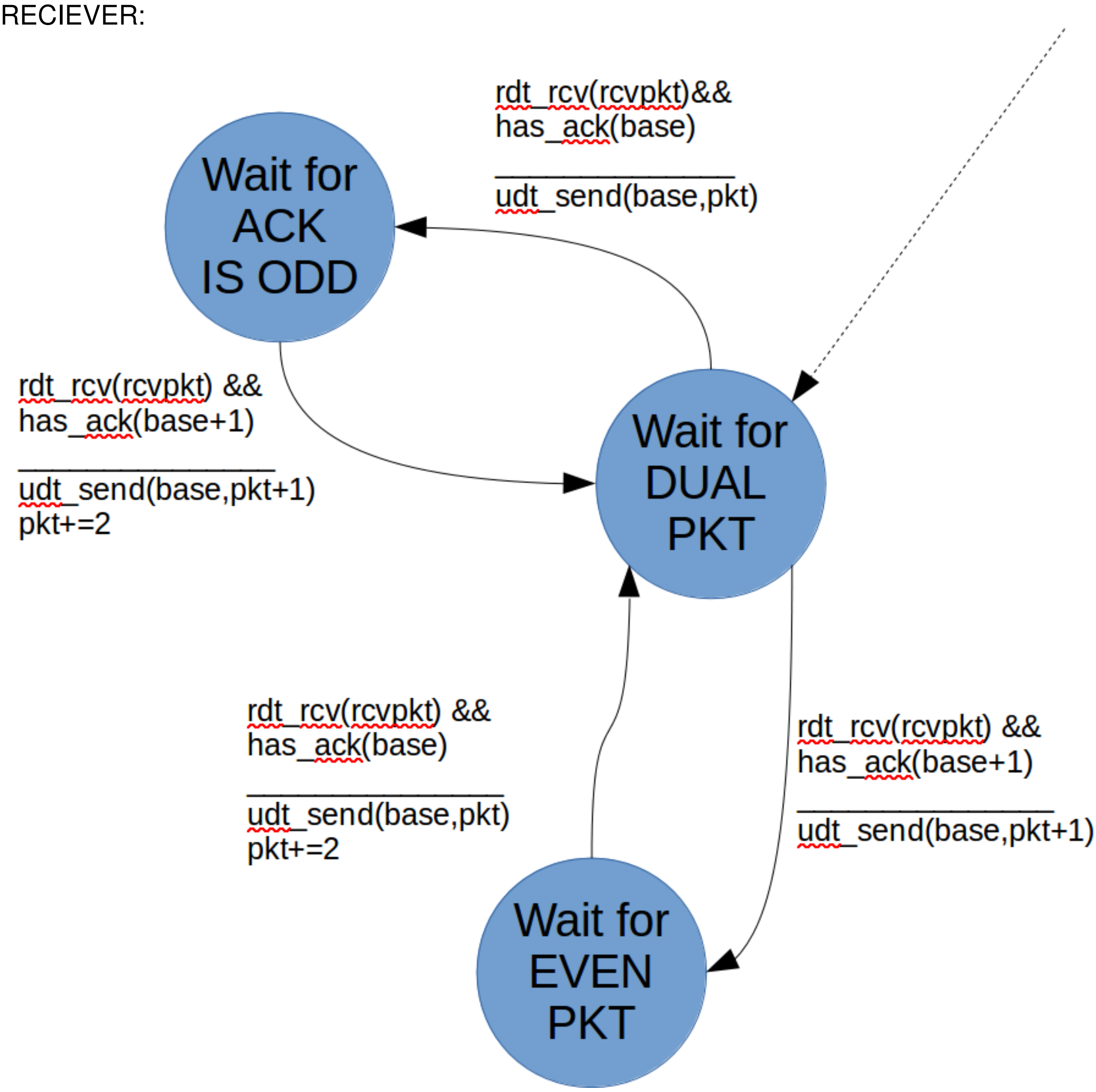
1. Problem 3.17



Wait to recieve a packet and then extracts data and delivers it. And then waits for a signal to make and send a packet tot the next state.

2. Problem 3.18

The sender waits for ACK before it can send the next pair of messages; the ACK messages received will have an ID that dictates which packet was received.
The timer and timeout function use a timer to wait for a packet or for a delta T period.
udt send function sends a packet with an acknowledgement ; base implies the even acknowledgement and base+1 implies odd acknowledgement.; has ack function implies that the received packet has the attached acknowledgement. ; udt send has a acknowledgement and a packet argument.
RECIEVER:



SENDER

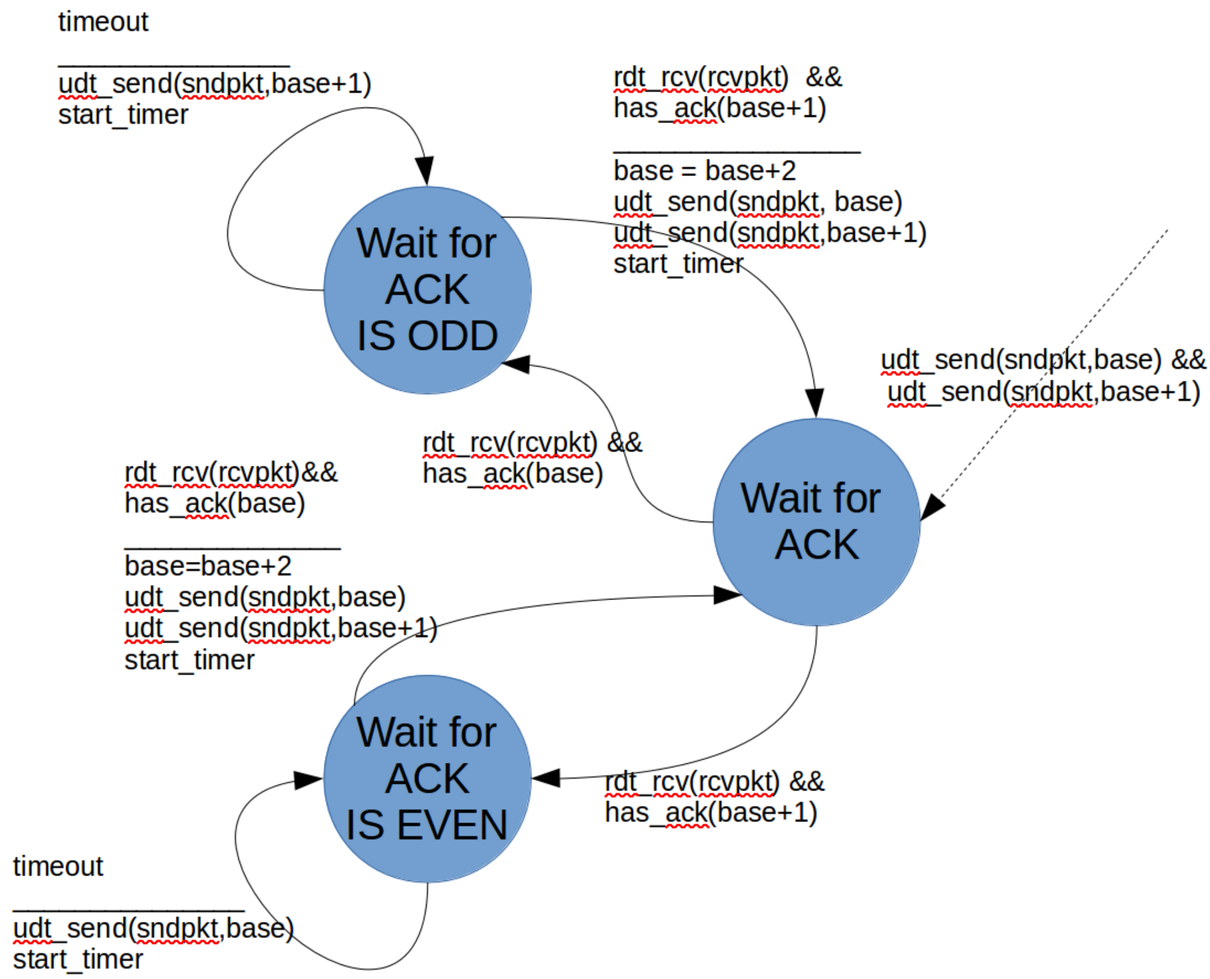

timeout
____
udt_send(sndpkt,base+1)
start_timer
rdt_rcv(rcvpkt) &&
has_ack(base+1)
____
base = base+2
udt_send(sndpkt, base)
udt_send(sndpkt,base+1)
start_timer
Wait for
ACK
IS ODD
udt_send(sndpkt,base) &&
udt_send(sndpkt,base+1)
rdt_rcv(rcvpkt) &&
has_ack(base)
rdt_rcv(rcvpkt)&&
has_ack(base)
____
base=base+2
udt_send(sndpkt,base)
udt_send(sndpkt,base+1)
start_timer
Wait for
ACK
Wait for
ACK
IS EVEN
rdt_rcv(rcvpkt) &&
has_ack(base+1)
timeout
____
udt_send(sndpkt,base)
start_timer

DIAGRAM:

```
make pkt pair (0,1)
send packet 0
                    loss
send packet 1
                              receive packet 1
                               buffer packet 1
                                  send ACK 1
receive ACK 1
(timeout)
resend packet 0
                             receive packet 0
                            deliver pair (0,1)
                              send ACK 0
receive ACK 0
```

3. Problem 3.22

   A.The receiver has ACKED k-1 and all the proceeding packets .If all the ACK has been received by the sender ,then the set is k,k+1, k+2. If all the ACK are lost, then the set is k-3, k-2,k-1
   B.The receiver is expecting the packet k, so the ACK of the proceeding packets must be sent to the receiver. If all the former 3 ACK hasn't arrived at the sender ,then values of ACK includes k-1,k-2,k-3; since packet k-1 has been sent , the ACK of packet k-4 is no doubt that has got the sender so there are no ACK less than k-3 in the currently propagating back to the sender, neither ACK more than or equal to k.
   Thus the possible values are k-3,k-2,k-1

4. Problem 3.26

   A)
   Given that the TCP sequence numer field is 4 bytes;
   $4 * 8bits = 32bits$
   Thus the possible sequence is $2^{32} \rightarrow 4294967296$
   The MSS is irrelevant because the sequene is an incriment of the number of bytes sent. So then we can say that the file size between Host A and Host B is the number of bytes gien from the equation of;
   $2^{32} \rightarrow 2^2 * 2^{30}$ (all in bytes of course)
   The result of this equation is 4 Gigabytes.
   B)
   The number of segments is $\frac{2^{32}}{536}$ where 536 is the max segment size given to us and the function is found in the book; the result of which is 8012999
   The total number of bytes for different layer headers are added from each and every segment sent over the link; and so we are given $8012999 * 66 = 528857934$bytes in the header. Added to the number field, we get $2^{32} + 528857934 = 4823825230 \rightarrow 4.824 * 10^9$bytes

   And using the transmit equation, we now get
   $\frac{4.824*10^9*8bits}{155*10^6 bps \rightarrow}$
   $\frac{38592}{155} \rightarrow$
   248.98 seconds