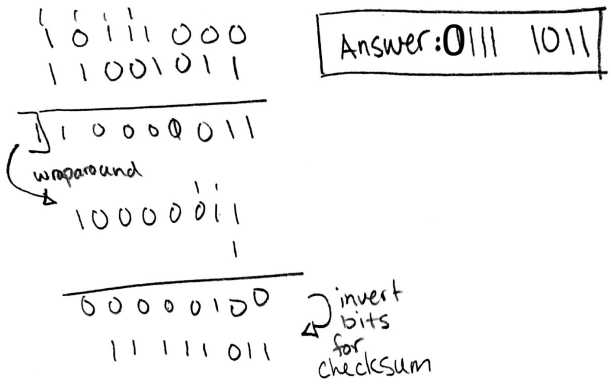
Gavin Forsberg
CIS430 - Dr. Lin
11/18/2020

1) TCP vs. UDP

Characteristic	TCP	UDP
Connectionless		X
Reliable Transfer	X	
Less Overhead		X
Has congestion/flow control	X	
Used by DNS/SNMP protocol		X
Error Checking	X	X
Has ACK	X	
Has Handshake	X	

2) Given two 8-bit integers (1 0 1 1 1 0 0 0) and (1 1 0 0 1 0 1 1). Calculate the checksum (as in UDP/ TCP). (4 pts)



3) Why does it mean by the “stop-and-wait” operation in the RDT protocols? Under such operations, how many packets can be sent out simultaneously? (2 pts)

Stop-and-wait refers to that fact that until the receiver correctly receives the current packet, the sender will not send any new pieces of data. Therefore, it only sends one packet at a time.

4) Compare RDT versions (5 pts)

RDT Version	Assumption(s)	Feature Introduced
1.0	Underlying channel is perfect	Reliable transfer
2.0	Underlying channel may flip bits	ACK/NAK feedback, error detection
2.1	ACK/NAK corruption can occur	Sequence number
2.2	Same functionality as 2.1	ACK ONLY, no nak
3.0	Underlying channel can lose packets	Timeout and retransmission

5) RDT 1.0 (use (A) – (F) to answer the questions, 2pts/each)

   a. What are the states in the given finite state machines?

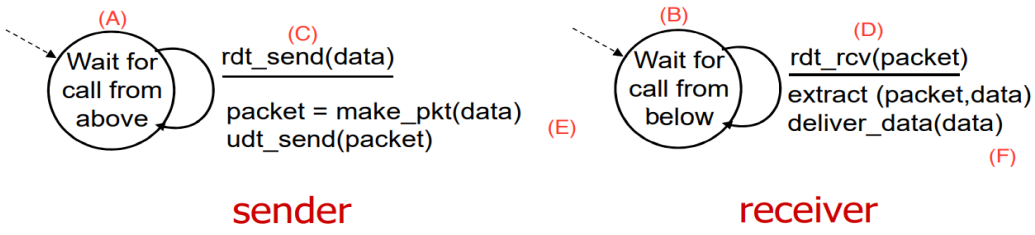
      AB.

   b. Which event is triggered when the sender receives data from the Application layer?

      C.

   c. Which action is taken when receiver receives the packet?

      F.



6) RDT 2.0, 2pts/each

   a. What is the meaning of (L)?

      It is where the data is extracted from the packet and data is delivered.

   b. If the receiver receives a corrupted packet, which action will be taken?

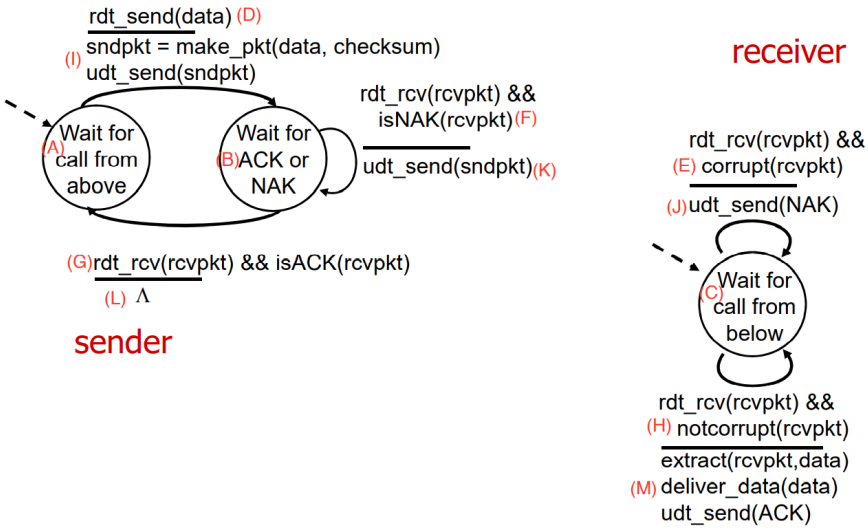
      Receiver sends back NAK to sender, sender resends packet and eventually there will be no corruption. Then open it up and sends back ACK. Triggers the received packet, verifies acknowledgment, Transitions back to wait state.

   c. What will happen when either the ACK or NAK is lost on transmission?

      Sender doesn't know what happened at receiver. If it receives something unknown, not ACK or NAK, it will drop it and continue in the wait state until is does receive ACK/NAK.

   d. Following the question above, what are the current states of the sender and receiver?

      Sender will continue wait status until it receives an ACK or NAK.



7) (RDT 2.1) We learn in class that the behaviors in higher versions of RDT protocols are symmetric. That is, the events and actions to handle the state with sequence number 0 is almost the same as those with sequence number 1.

   a. Explain the event (in plain English) for (X). Also, write down the pseudo code for (X). Note: Try to use the rest of the diagram to identify the event. (4 pts)

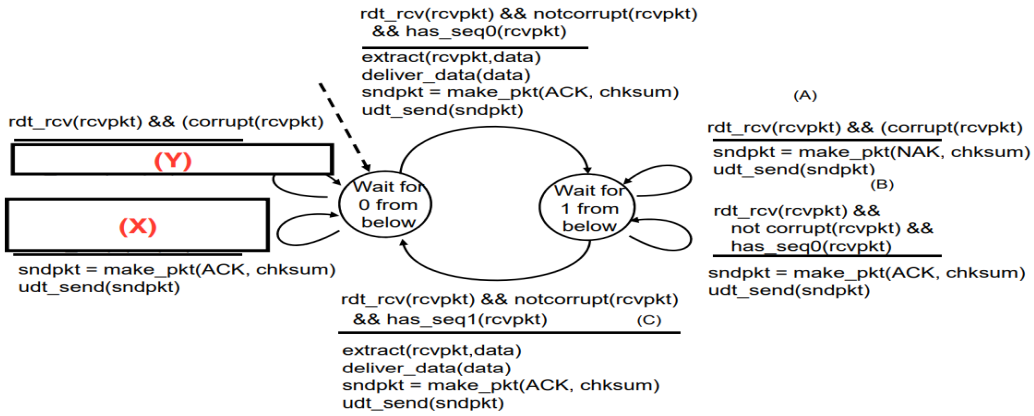
      rdt_rcv(rcvpkt) && not corrupt (rcvpkt) && has_seq1(rcvpkt)

      It checks before transitioning that the packet was received with no corruption and is not a duplicate (has seq number one). If what's received is unknown, it's assumed it is the already sent packet and it's dropped.

   b. Explain the actions (in plain English) for (Y). Also, write down the pseudo code for (Y). Note: Try to use the rest of the diagram to identify the actions. (4 pts)

      sndpkt = make_pkt(NAK, chksum) udt_send(sndpkt)

      If you receive the packet and it is corrupt, send a NAK back to receiver.



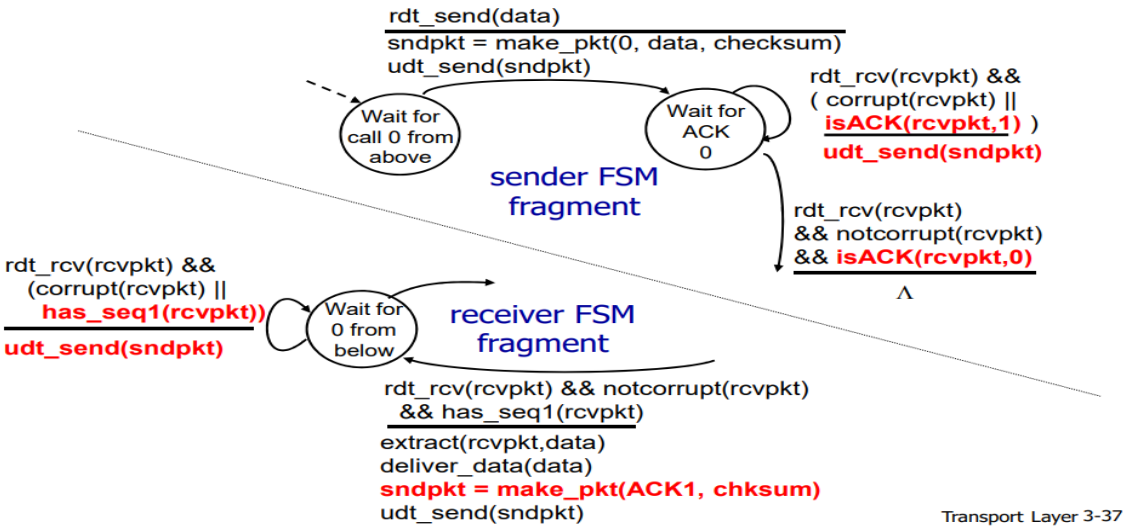
8) (RDT 2.2) Missing pieces. (2pts/each)

   a. Identify the other half of the diagram for the sender's FSM (finite state machine).

      We need to wait for call 1 from above, then do another rdt_send(data), finally wait for ACK1 by using rdt_rcv && (corrupt) as well as not corrupt functions.

   b. What are the differences between the two halves?

      The difference is whether the call/ACK is 1 instead of 0.



9) (RDT 3.0) When do you start and stop the timer? (2 pts)

   Whenever a packet is sent, the timer starts. It stops when the packet is received correctly with no corruption. If there is a timeout or a duplication in retransmission, the timer will reset.

10) (RDT 3.0) When we go from RDT 2.2 to RDT 3.0, shall we have another FSM for the receiver? Explain why or why not. (4 pts)

    The FSM for the receiver is the same as 2.2. They both handle duplicate packets. The time-out can be implemented from the sender as well.

11) (RDT 3.0) If the time set in the timer is too short, how does RDT 3.0 handle the duplicated packets? (2 pts)

    If duplication happens during retransmission, the timer will reset. It is expected an acknowledgement will come before timeout so it resends the ACK.

12) All RDT protocols share the same issue that they may send the next packet only after receiving an ACK. How do we improve their performance? (2 pts)

    Improves performance with pipelining - two protocols are go-back-n and selective repeat.

13) Why do we need to have both sequence and acknowledge numbers? Briefly explain the dilemma of having only the sequence number in Selective Repeat. (4 pts)

The receiver individually acknowledges correctly received packets. Therefore the sender only resends packets that an ACK was not received for. There can be many consecutive sequence numbers and this will cut down on sequence numbers sent/unACKed packets. The receiver's side thinks that the ACKS go through/keep moving on while the server just keeps resending packets.

14) (TCP) What are the required number of bytes in the TCP header? (2 pts)

TCP header can be as small as 20 bytes and as large as 60 bytes. 5 rows by four bytes is 20 for the min. The 6th row can go up to 40 bytes, max is 60. Source port, dest port, seq number, ack number, DO, RSV, Flags, window, checksum, urgent pointer are all required.

15) Compare the differences between Go-back-N (GBN) and Selective Repeat (SR) (answer true or false). (4 pts)

Characteristic	GBN	SR
Can send multiple packets in the pipeline	TRUE	TRUE
Receive individual ACK per packet	FALSE	TRUE
Has timer for the oldest unACKed packet	TRUE	FALSE
Does not have a receiving buffer for out-of-order packet	TRUE	FALSE