

Final Project

PROBLEM C:

Consider a scenario in which Host A wants to simultaneously send packets to Hosts B and C. A is connected to B and C via a broadcast channel – a packet sent by A is carried by the channel to both B and C. Suppose that the broadcast channel connecting A, B, and C can independently lose and corrupt packets (and so, for example, a packet sent from A might be correctly received by B, but not by C). Design a stop-and-wait-like error-control protocol for reliably transferring packets from A to B and C, such that A will not get new data from the upper layer until it knows that both B and C have correctly received the current packet. The FSM for B should be essentially the same as for C.

SOLUTION:

Figure 1 is the sender FSM for host A. The process starts with a call from the above layer, it then sends the data to hosts A and B with the sequence number 0. If A receives a corrupt packet from either host B or C it will resend the missed packet to both hosts. If A receives an ACK from both B and C then it will wait for a call from the above layer and send the packet with sequence 1 to B and C. The cycle will repeat with sequence 1, on error resend packet to both hosts, on ACK from both wait for above layer and set seq to 0. Because A has to wait for the ACK from both hosts before it checks the above layer, A will not pull new data until both hosts have ACK'ed without error.

Figure 2 shows the receiver FSM, hosts B and C will have similar FSM's because they both behave the same way. Host B will wait for the packet with sequence 0 from A, if the packet from A is corrupt, B will send a NAK to A, if the packet has a sequence number of 1, B will send an ACK and continue waiting for the sequence 0. If the packet sent by A is not corrupted and has a sequence of 0, B will extract and deliver the data and send an ACK to A, then wait for a sequence 1 from A. The process repeats for sequence 1.

The packet format will include either a one or a zero for the sequence number, the sender of the packet, a data section (for data, ACK or NAK), and a checksum section.

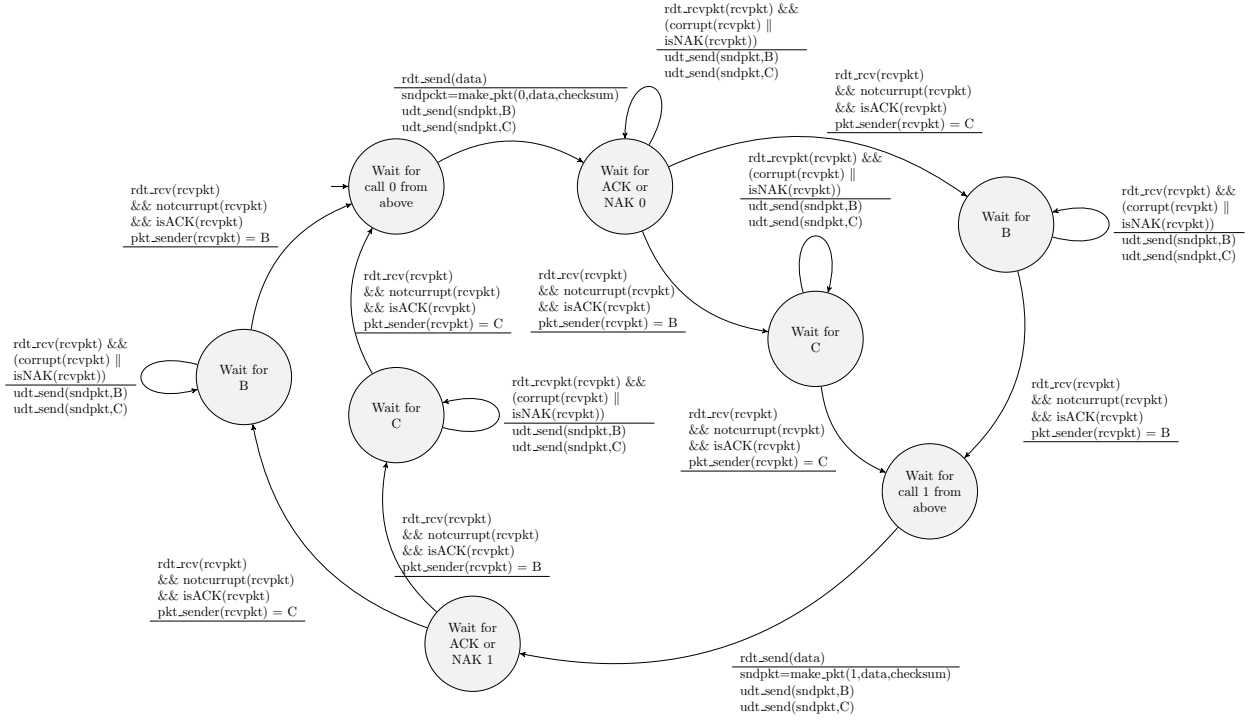


Figure 1: Sender FSM for host A.

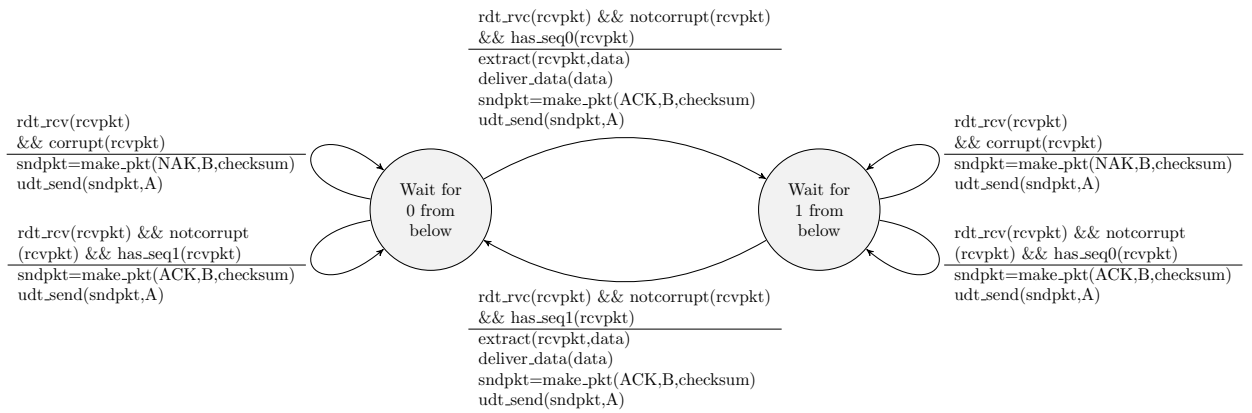


Figure 2: Receiver FSM for host B.