RDT example

<u>Question: A reliable data transfer protocol (20 points)</u>

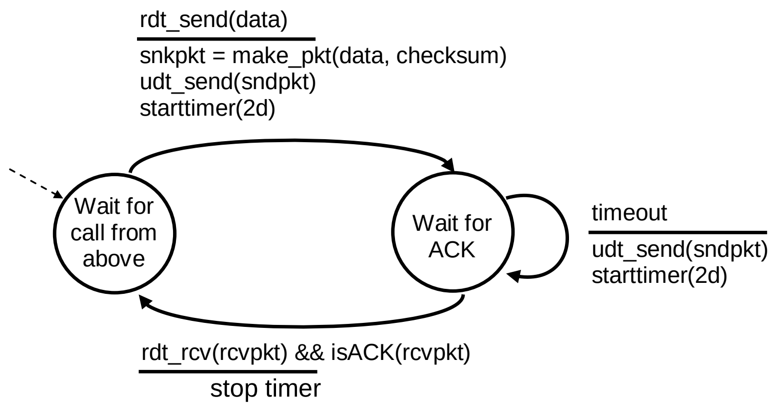
In class, we have seen a number of mechanisms used to provide for reliable data transfer:

- checksum
- ACKs
- timers
- sequence numbering

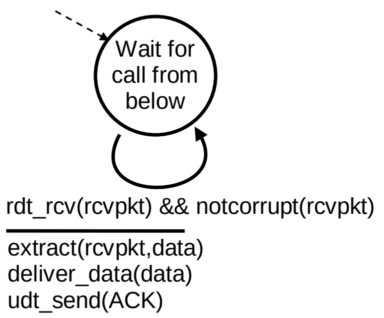
Consider a sender and receiver that are connected by a sender-to-receiver channel that can **corrupt** and **lose** packets. The receiver-to-sender channel is **perfect** (i.e., it will not lose or corrupt packets). Neither channel will reorder packets. [Note: re-read the channel properties just described and make sure you understand them!]

You are to design a reliable data transfer protocol (no pipeline, stop-and-wait style) for this scenario *using only those mechanisms (among the four listed above) that are absolutely required*. That is, if you use a mechanism that is not required, your answer will not be given full credit, even if the protocol works. Your protocol should be as simple as possible but have the functionality to reliably deliver data under the stated assumptions. Your solution need not be efficient; it must work correctly.

a) Draw the sender and receiver FSMs.



(a) sending side



(b) receiving side

Ignore the (2d) used in starttimer() function. We do not test on this item.

b). For each of the mechanisms (among the four listed above) that you use in your protocol, explain the role/purpose of the mechanism and why can't you get by without it? If you do not use one mechanism, explain why you do not need it.

Because the sender-to-receiver channel can corrupt packets, the data-sent on the sender-to-receiver channel will need a checksum to detect bit errors. Because the sender-to-receiver channel can lose packets, we will need to have a timer to timeout and retransmit packets that have not been received by the receiver. The receiver will need to indicate which packets it has received by using an ACK message; if a packet is not received or is received corrupted, no ACK is sent. There is no need for sequence numbers, since there will be no unneeded (and unexpected at the receiver) retransmissions.

The FSM for the sender and receiver are shown below.

As an example of the mistakes in answers to watch out for:

- *Do not need to use sequence numbers*
- *Do not need checksums on receiver-to-sender channel*
- *Separate FSMs are needed for the sender and the receiver!*
- *Do not need NAKs (absence of ACK serves as a NAK)*