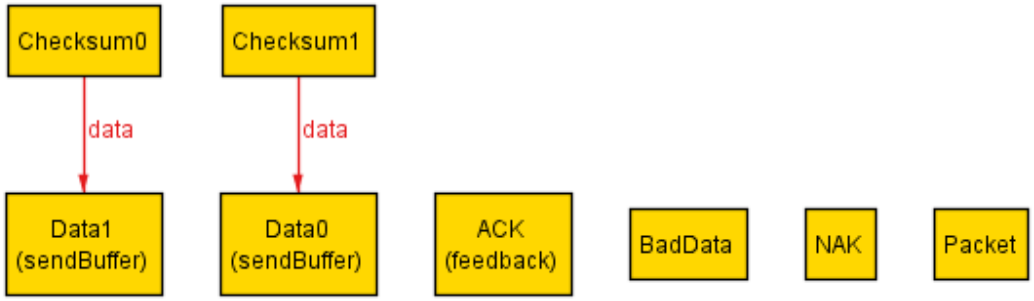
CSSE373: Project Milestone 2, RDT 2.0
Sean Carter and Austin Willis ("The Awesome Team")

Property 1 Screenshots

Key: sendBuffer is the sender's buffer, recvBuffer is the receiver's buffer

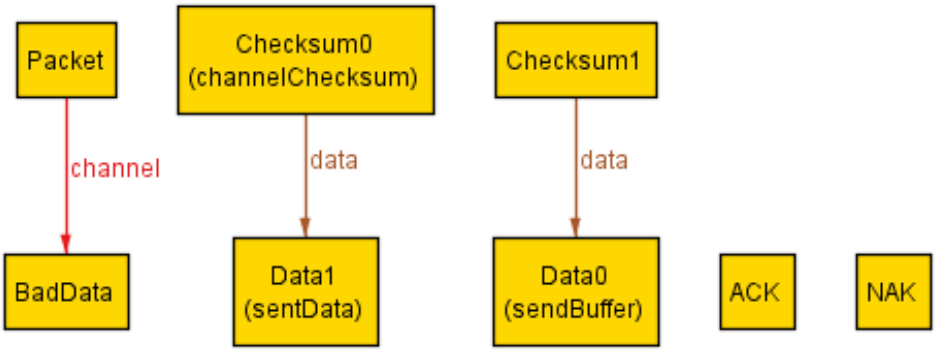
State0 (init)

All data is in sendBuffer, each Checksum points to exactly one piece of data, the feedback defaults to ACK.



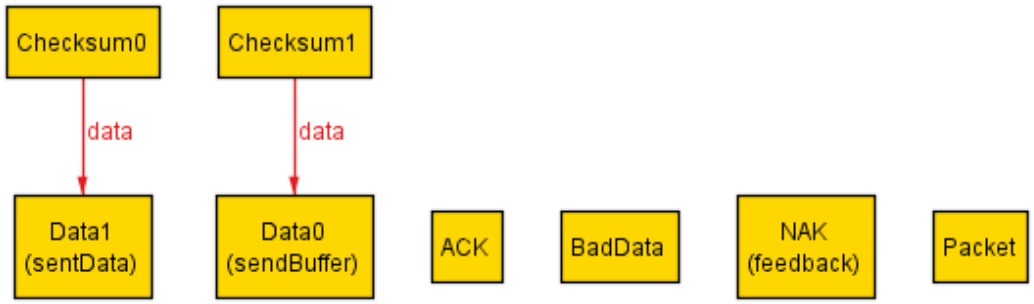
State1

Data1 is selected to be sent, but a piece of bad data is packaged instead (modeling data corruption in the packet). Checksum0 is in the channel along with the data.



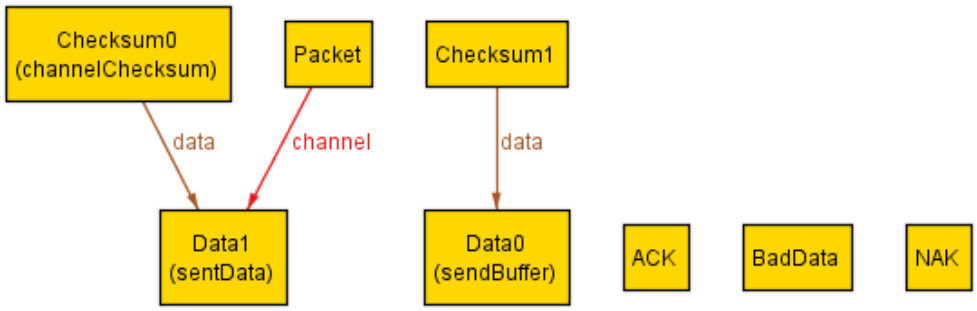
State2

The Receiver returns a NAK response, because Checksum0 did not match the BadData.



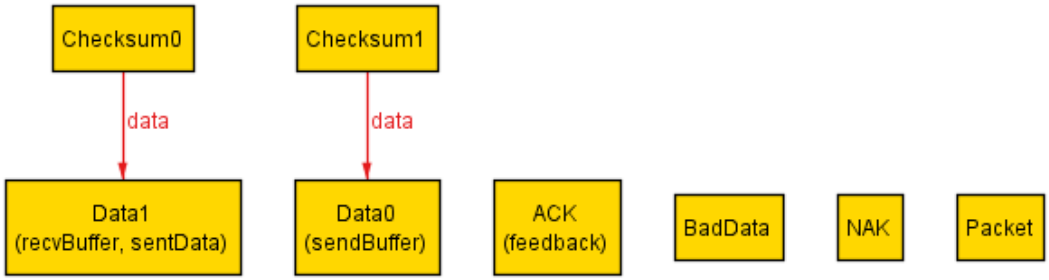
State3

Data1 is grabbed from the sentData copy, repackaged, and sent properly.



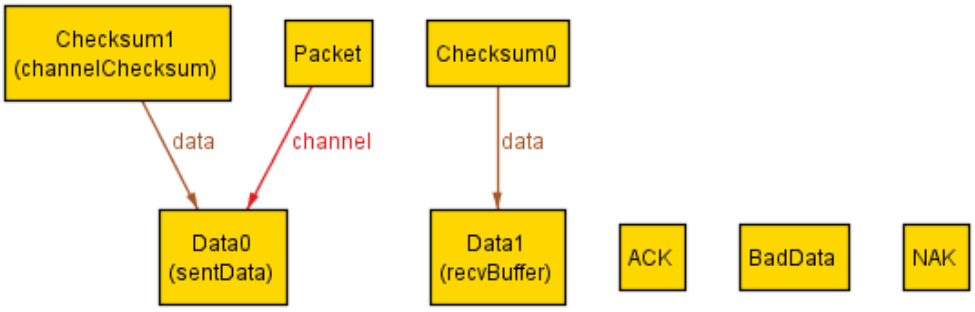
State4

Checksum0 is matched with Data1, Data1 is unpackaged and placed in recvBuffer, and the Receiver returns a NAK response.



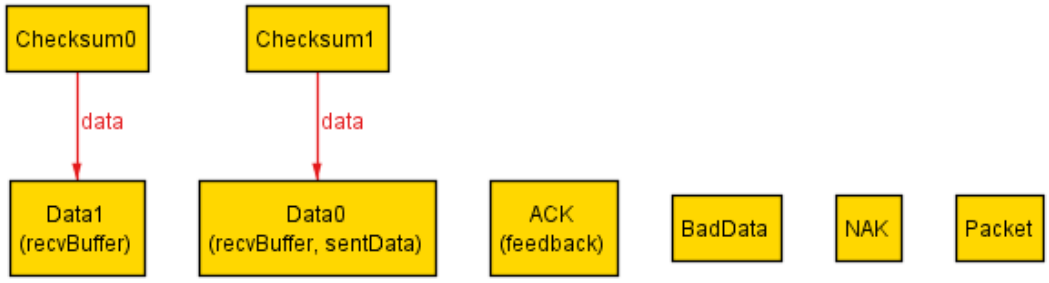
State5

Data0 and Checksum1 are packaged properly and placed in the channel.



State6

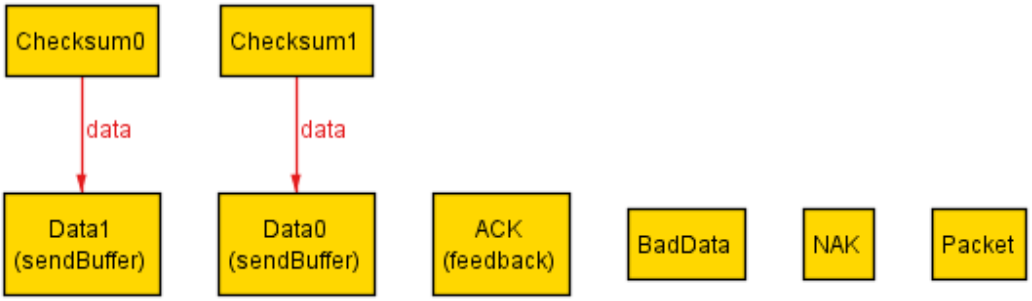
Checksum1 is matched to Data0, Data0 is placed in recvBuffer, and the Receiver sends an ACK response. All data has now been sent and received, and is available in recvBuffer.



Property 2

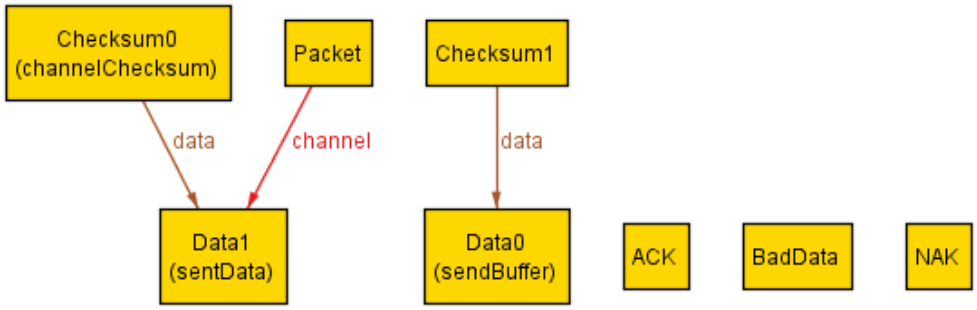
Our assertion found a counterexample in which data is continually corrupted on a send attempt.

State0 (init)



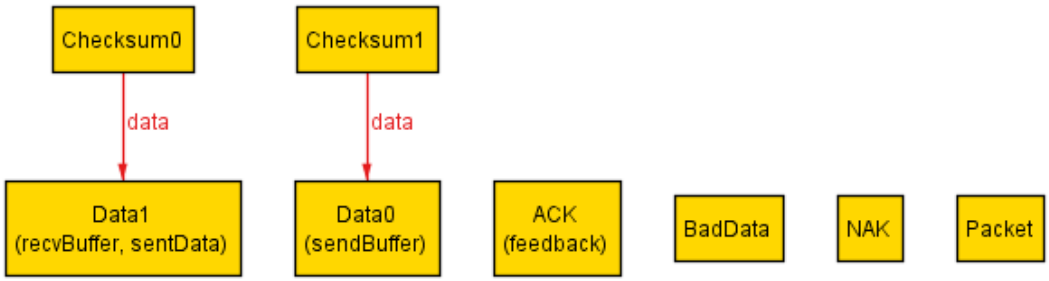
State1

Data1 is sent normally.



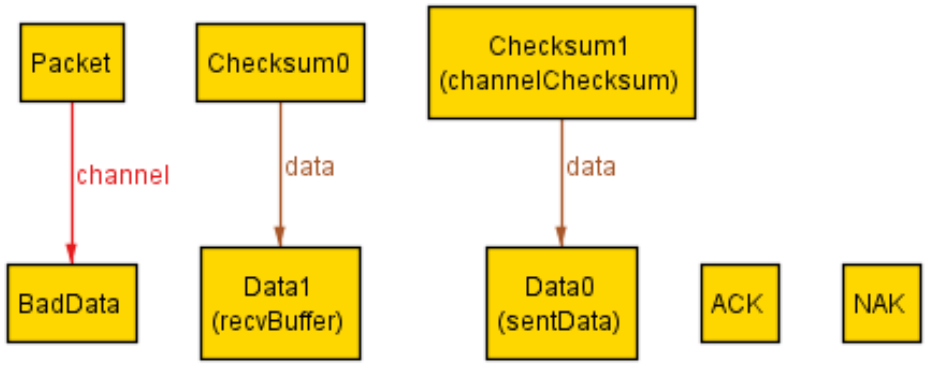
State2

Data1 is received normally.



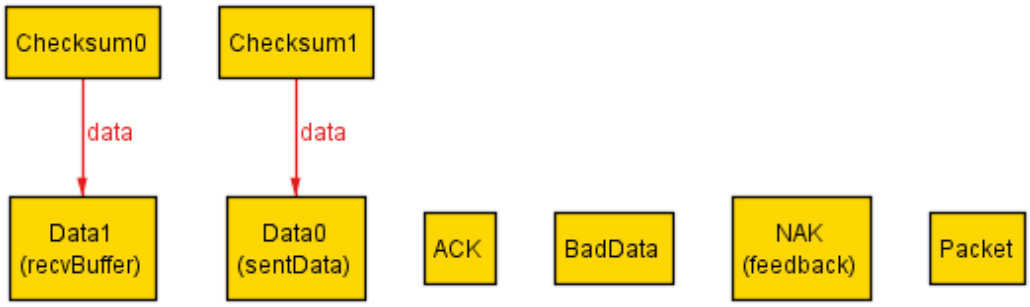
State3

A bad piece of data is sent.



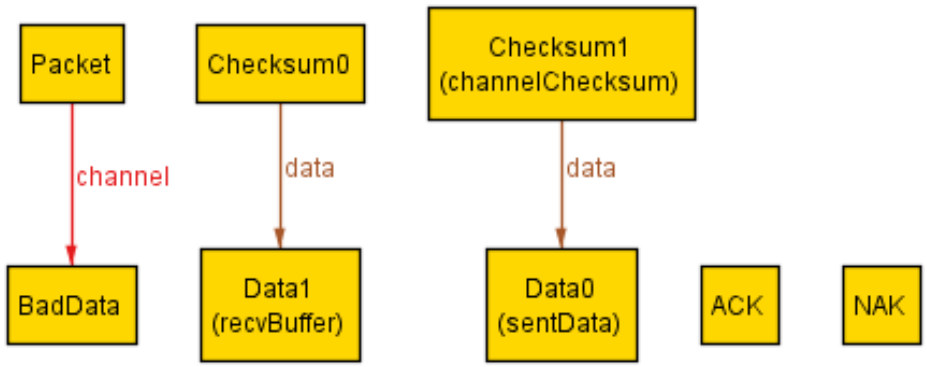
State4

A NAK response is sent.



State5

A bad piece of data is sent again.



State6

A NAK response is sent again.