CSSE373

Milestone 2

Fred Zhang, Songyu Wang

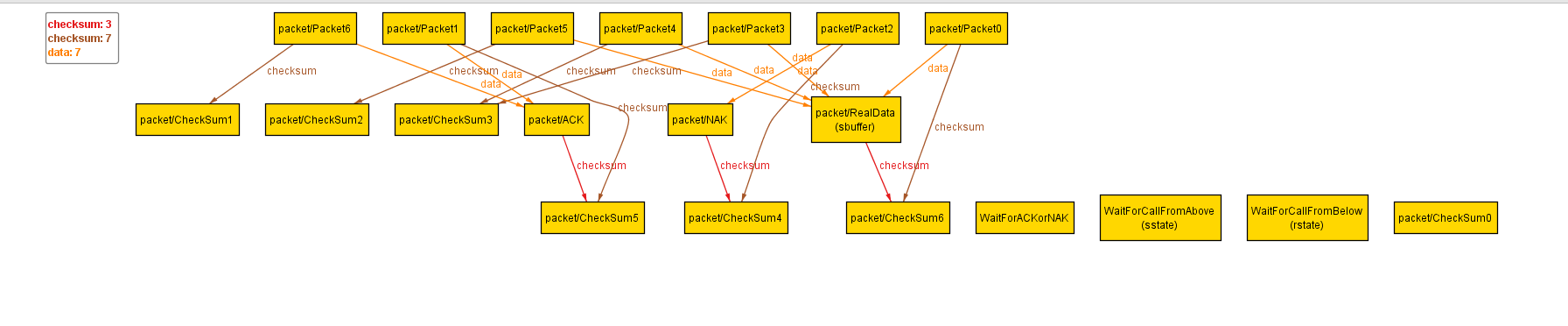
**Using the given protocol, it is possible to transmit all of the data in the sender’s buffer to the receiver’s buffer.**

We ran:

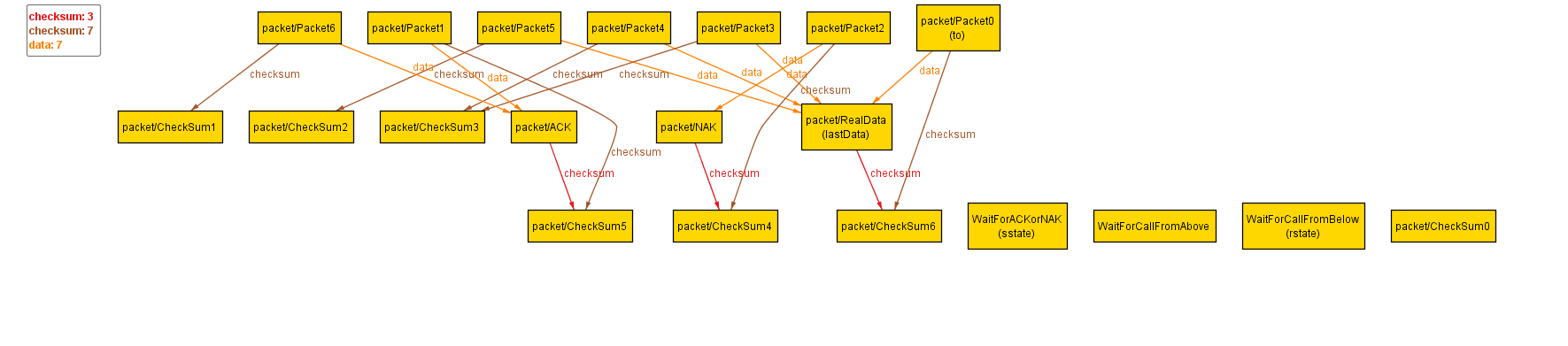
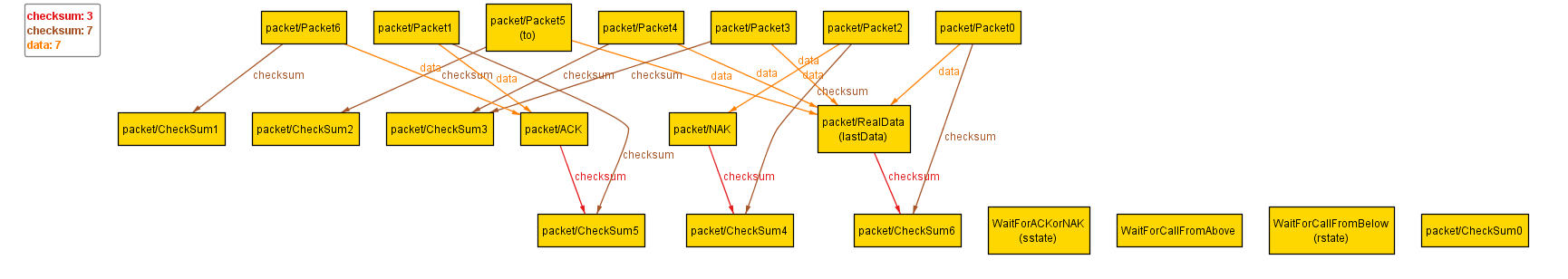
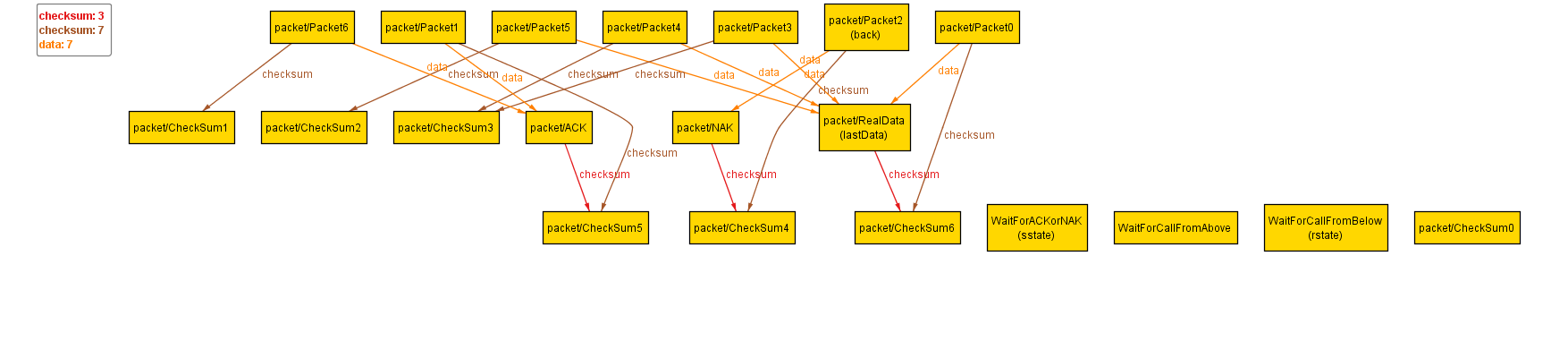
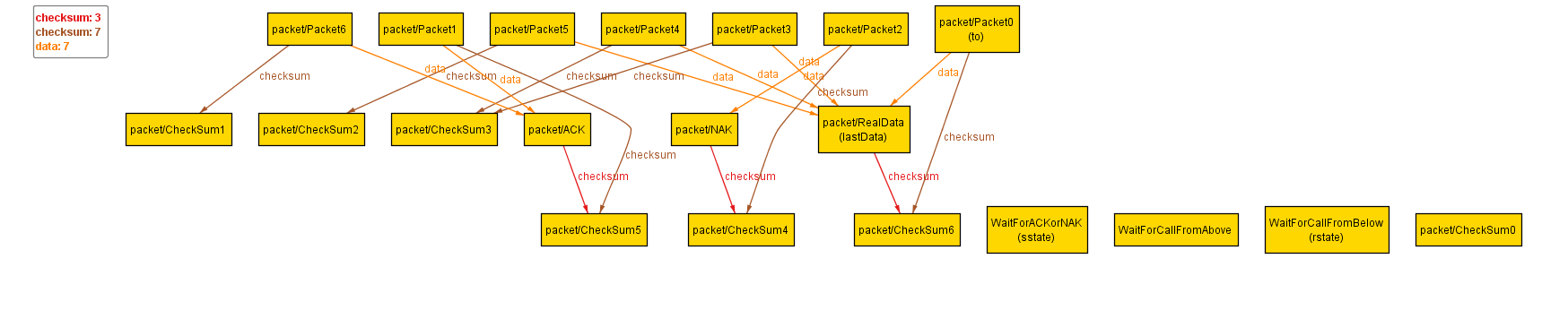
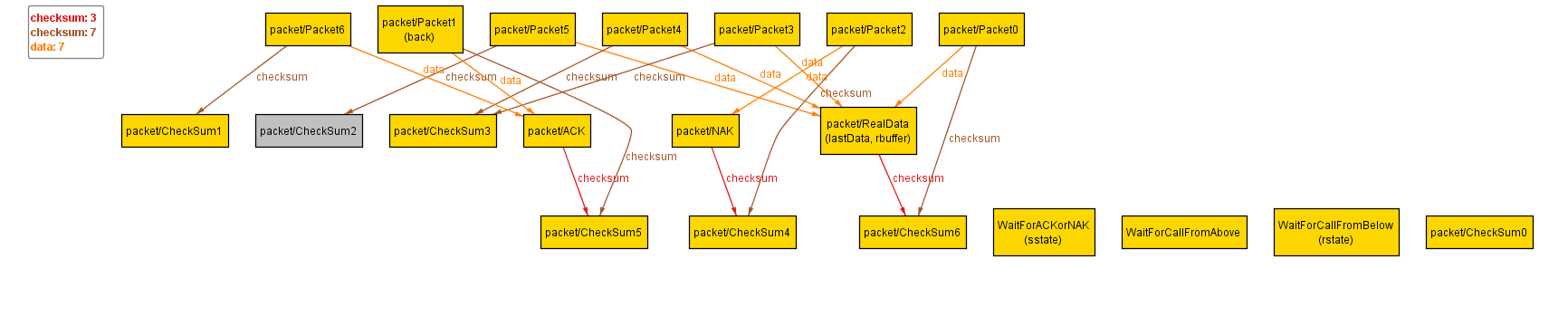
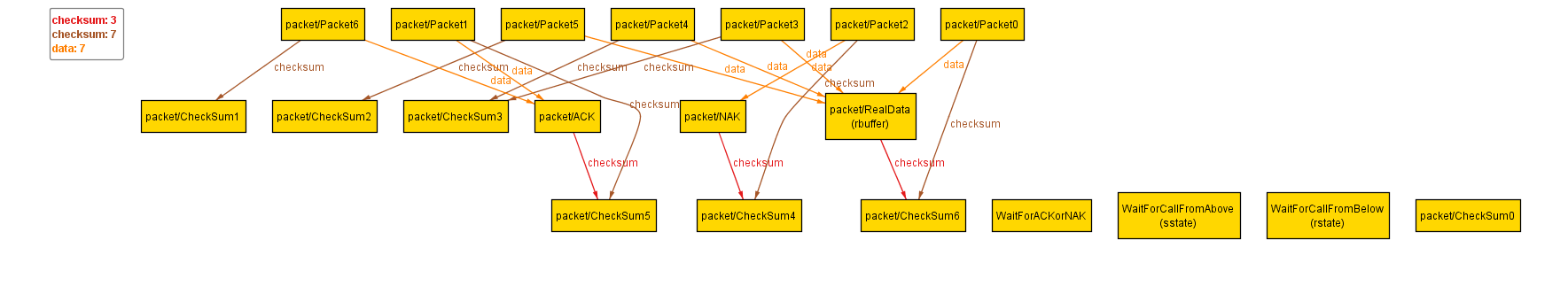
run possibleReliabe for 7 but exactly 1 RealData

The result is:

Time0



Time1

Time2 Time3 Time4  Time5 Time6 

At Time 0, The RealData is in sender buffer.

Sender state is Wait for Call From Above

Receiver state is Wait for Call From Below

At Time 1, The RealData is packed in Packet0 and put in the channel. It is waiting for being received by Receiver.

Sender state is Wait for ACK or NAK

Receiver state is Wait for Call From Below

At Time 2, The RealData is corrupted(Packet0 becomes Packet5)

Sender state is Wait for ACK or NAK

Receiver state is Wait for Call From Below

At Time 3, A NAK packet is sent by receiver due to the corruption of RealData.

Sender state is Wait for ACK or NAK

Receiver state is Wait for Call From Below

At Time 4, Sender receives the NAK data and Resend RealData/Packet0 into the channel.

Sender state is Wait for ACK or NAK

Receiver state is Wait for Call From Below

At Time 5, Receiver receives the uncorrupted data. The data is in the receiver buffer. An ACK message is sent by receiver.

Sender state is Wait for ACK or NAK

Receiver state is Wait for Call From Below

At Time 6, Sender receives the ACK message. Since there is no more data to send. This is the last state.

Sender state is Wait for Call From Above

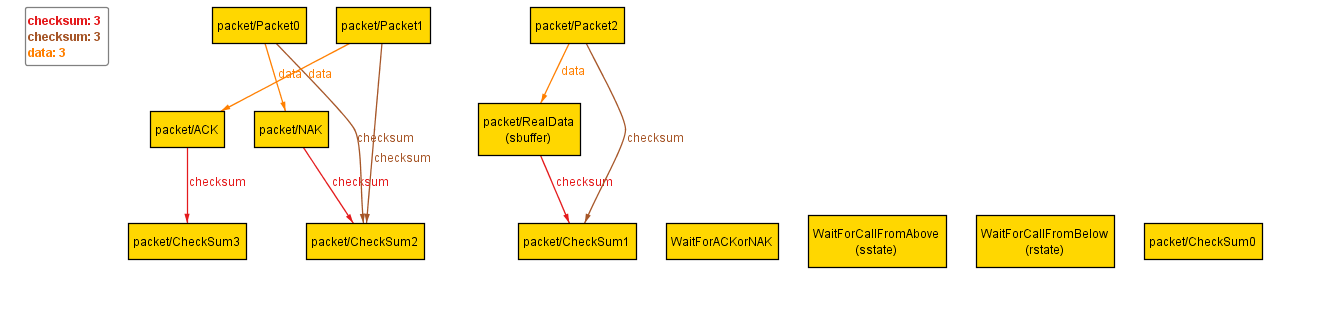
Receiver state is Wait for Call From Below

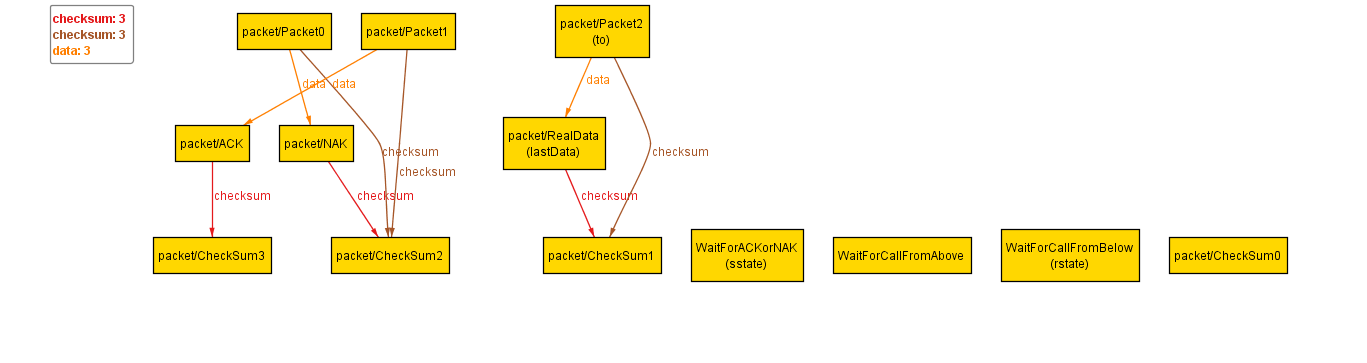
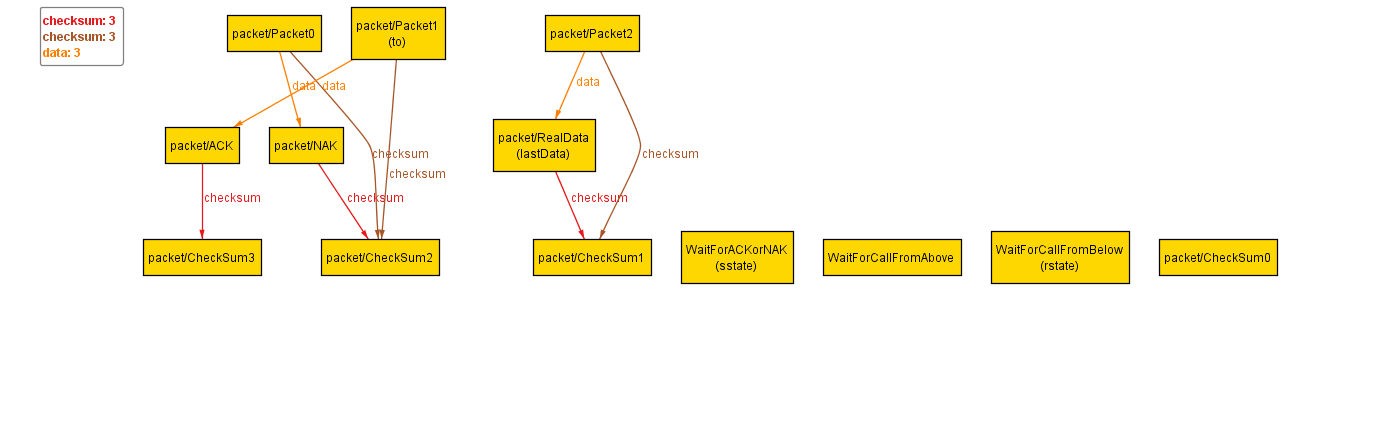
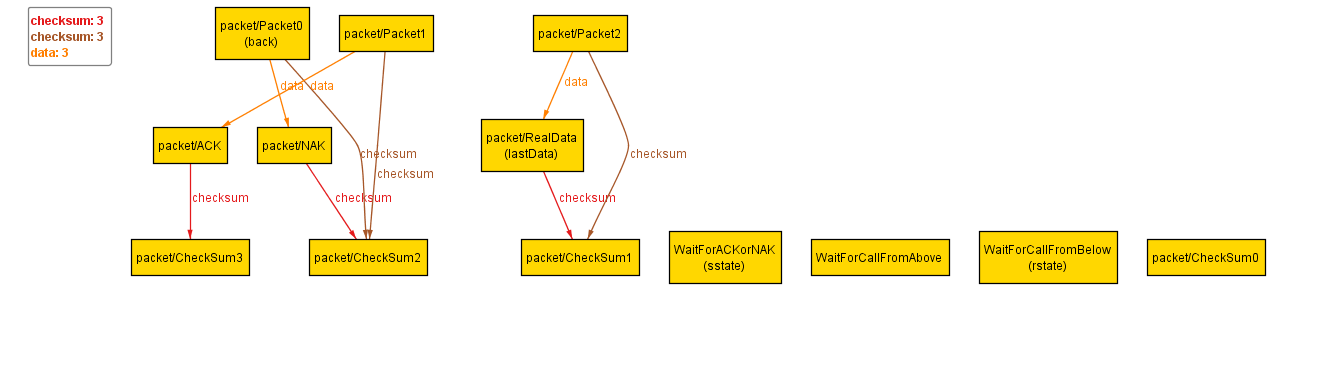
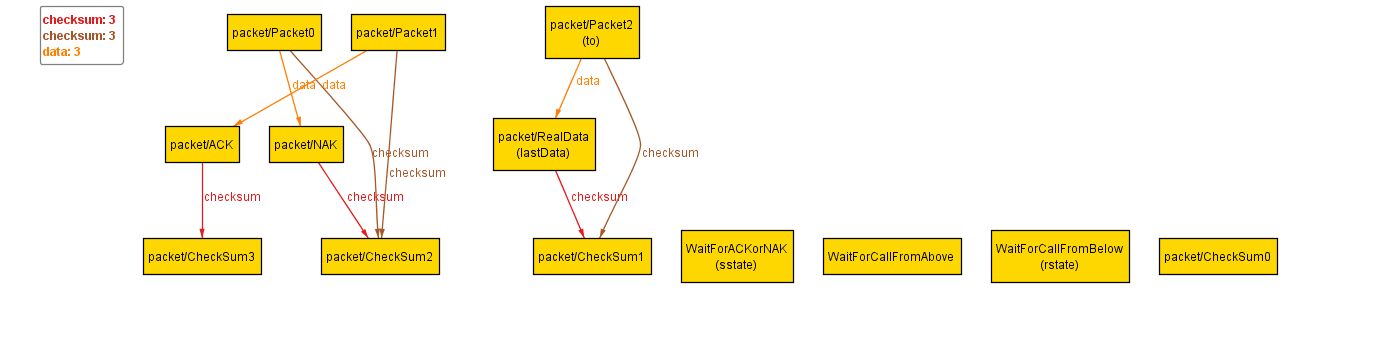
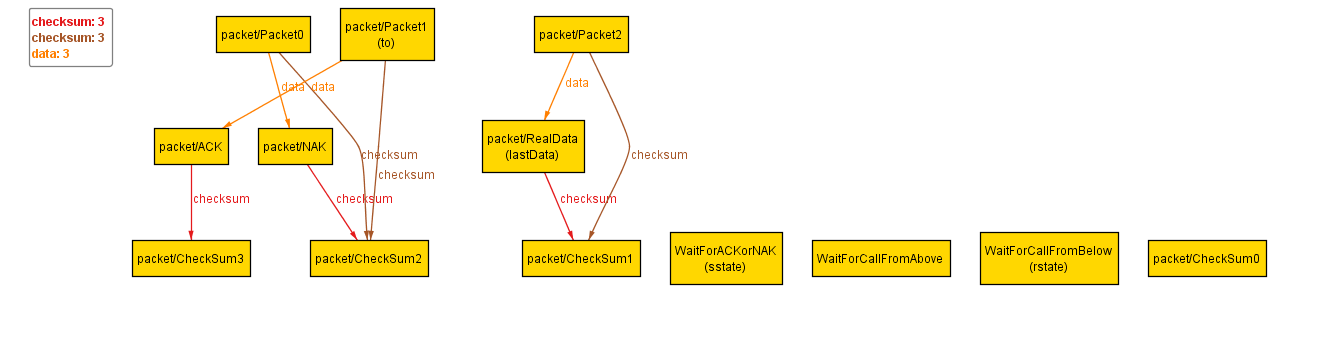
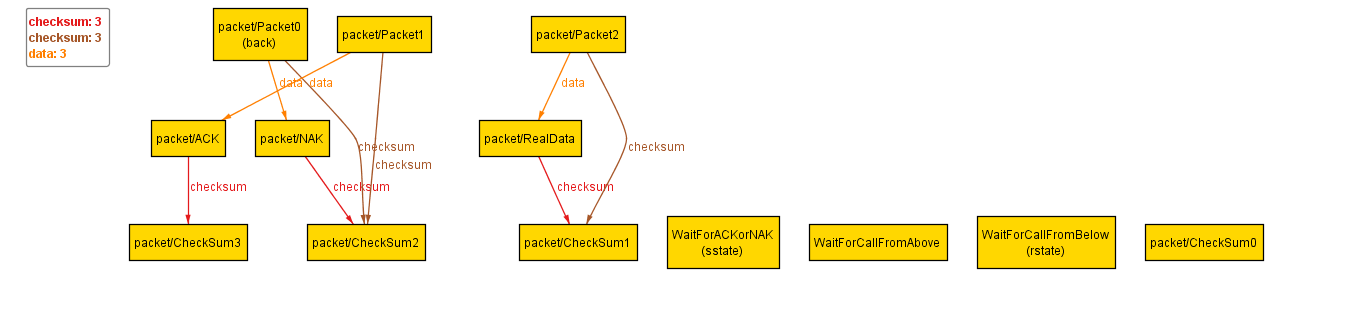
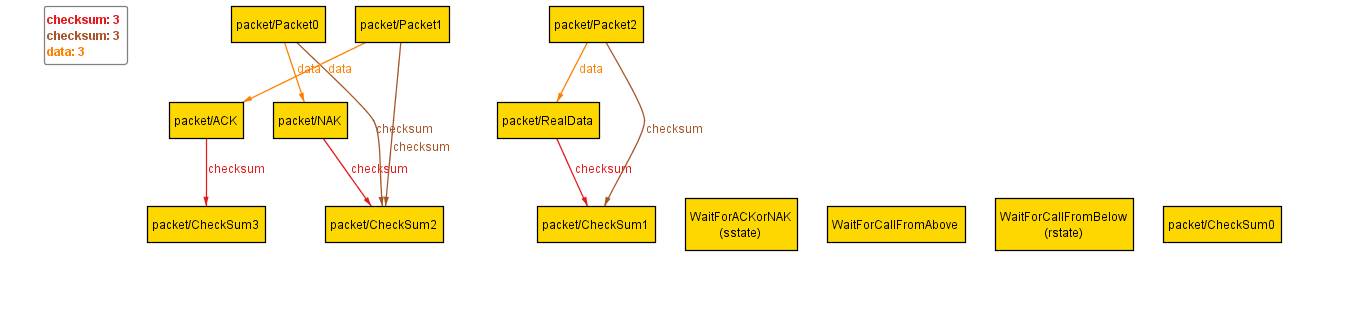
**Using the given protocol, it is** always **possible to transmit all of the data in the sender’s buffer to the receiver buffer.**

We ran

check alwaysReliable for 5 but exactly 8 Time, 1 RealData

And we have a counterexample:



We think 8 Time step is reasonable for one data to be transmitted.

However, as you can see above, There exist an “loop”:

Sender sends the data

The packet gets corrupted (In our case, the packet corrupted and became an ACK packet)

Receiver sends NAK to indicate sender that the data is corrupted

Sender send the data

The packet gets corrupted

…

…

At the end, the data magicly disappeared.

Consider this scenario:

Assume the sender is not functional properly. All the message sended by it will be corrupted or disappeared no matter how many time it sends the message.

That is the same scenario described by Alloy.

This protocal cannot guarantee sending all the data from sender to receiver. If there is one packet cannot be sent properly, that data may stuck in the channal forever and block the rest of data in sender. Or the packet can be lost and sender/reciever will wait forever because none of them know the data is gone.