RDT2.1 Writeup

Team Carry Harder

John Krasich

Matt Lash

Caleb Post

1. Is it possible to transmit all the data in the sender’s buffer to the receiver’s buffer?

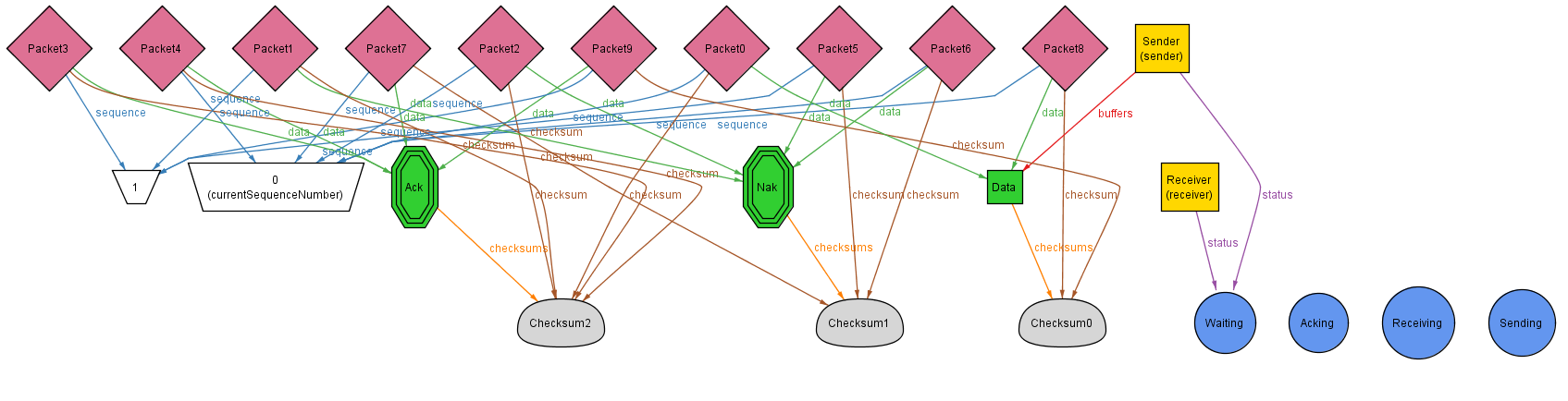
Yes – RDT 2.1 can handle three different types of data issues and still transfer all of the data from one buffer to the other. These situations are when there is corruption in the data being transferred, an error with an Ack message, or an error with a Nak message.

For all of the situations below, alloy produces 10 packets. Four are dedicated for Ack to cover all combinations of sequence numbers and checksum assignments (one correct and one incorrect). There are four additional to account for Nak in the same manner. The remaining two are data packets.

Data Corruption

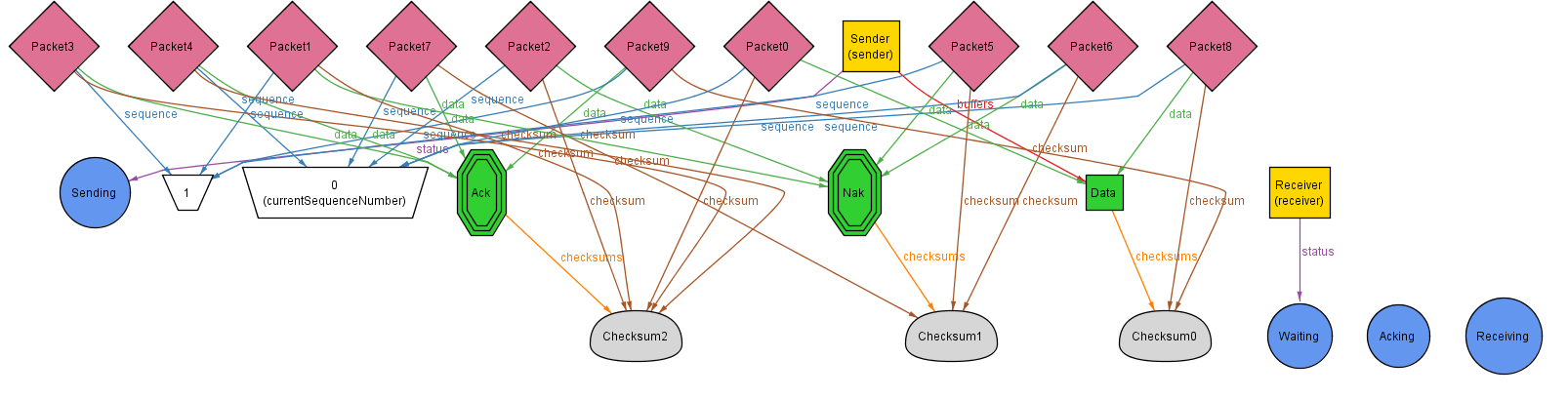
Initial State:

There is one piece of data to be sent from the sender’s buffer to the receiver’s buffer. The system’s currentSequenceNumber is initialized to 0, so any packets transferred in this example will have a 0 sequence as well.



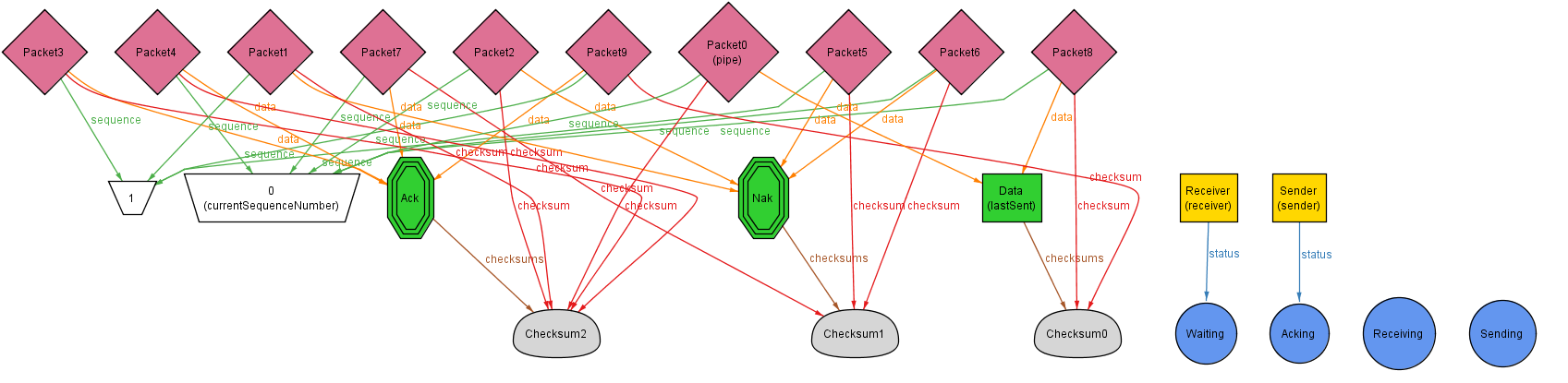
State 2:

The sender transitions into the sending state and prepares to place the data in the pipe.



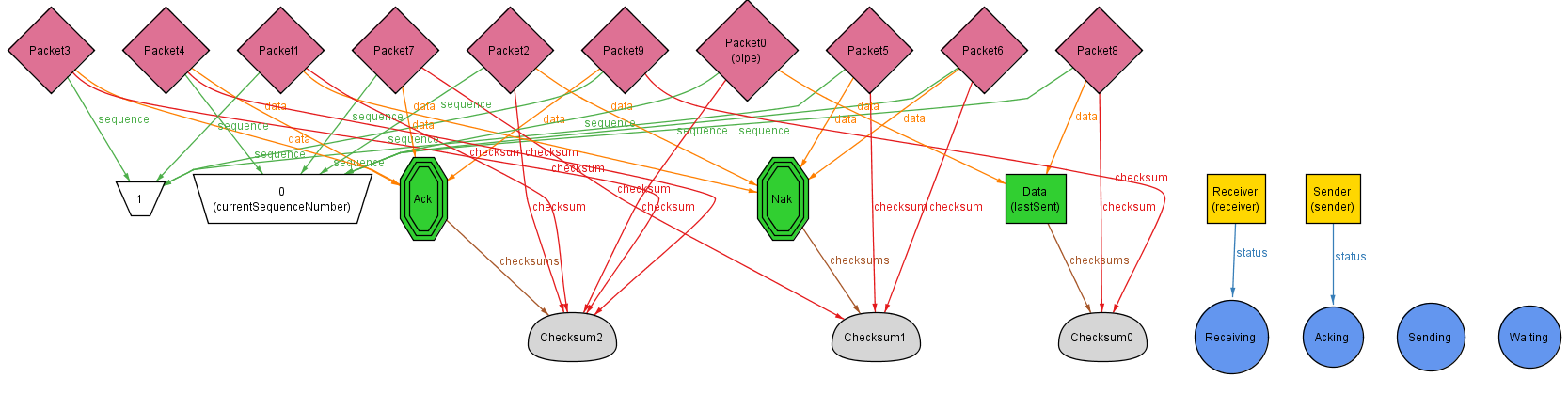
State 3:

The sender has sent the data into the pipe. The last sent data is updated to be Data. The sender enters the Acking state because it is now waiting on a reply from the receiver that it has gotten the data. Note that the checksum for the packet used to send the data is incorrect.



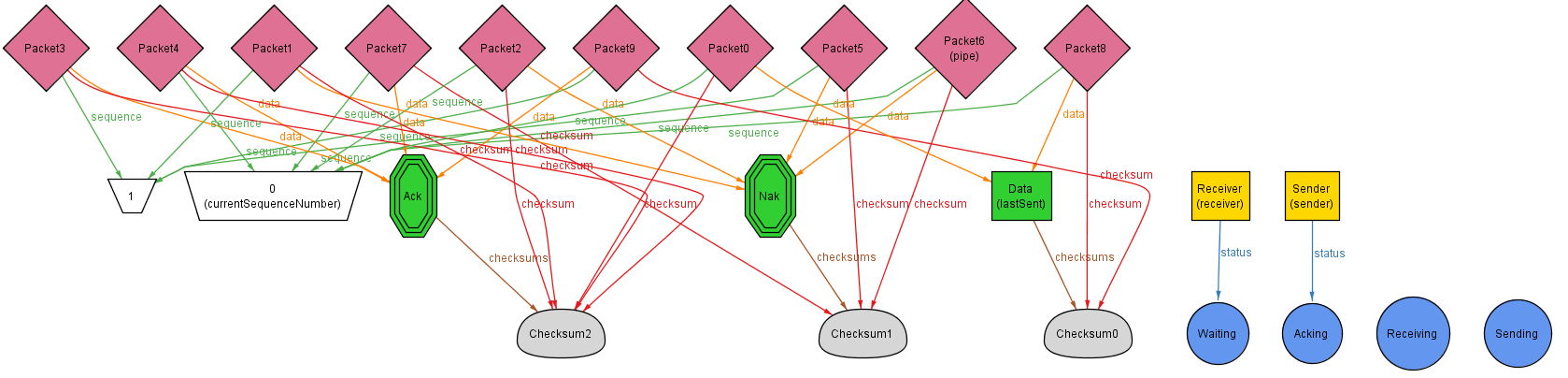
State 4:

The receiver sees that there is data in the pipe, and transitions into the receiving state to prepare to get the data out of the pipe.



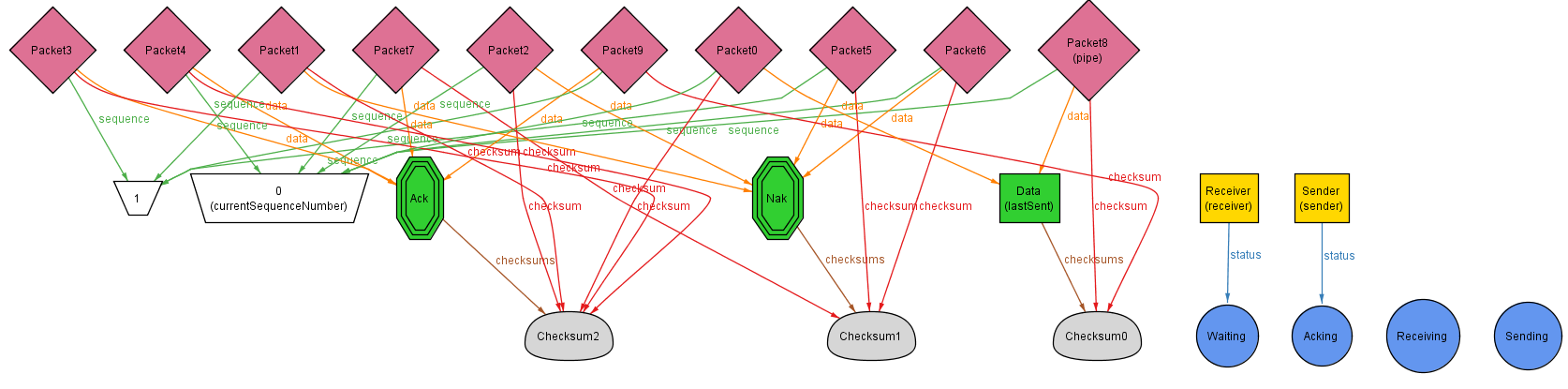
State 5:

The receiver pulls the data packet from the pipe and sees that it is a corrupt packet. The data is not brought into the buffer, and a Nak packet is placed into the pipe to indicate the failed transfer.



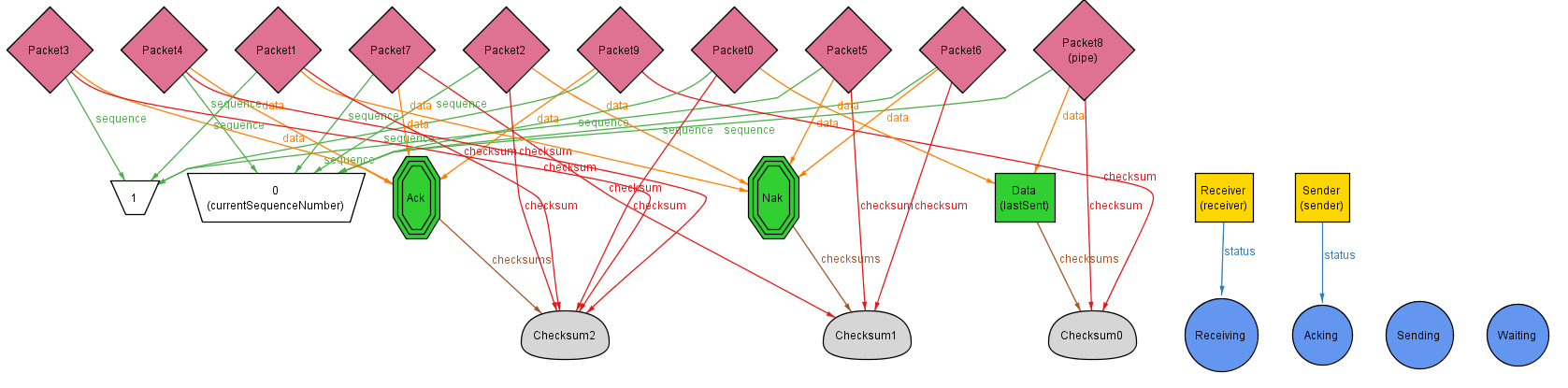
Step 6:

Reading the Nak packet, the sender sends the data into the pipe. Note that this time, the packet used for the data has the correct chechsum.



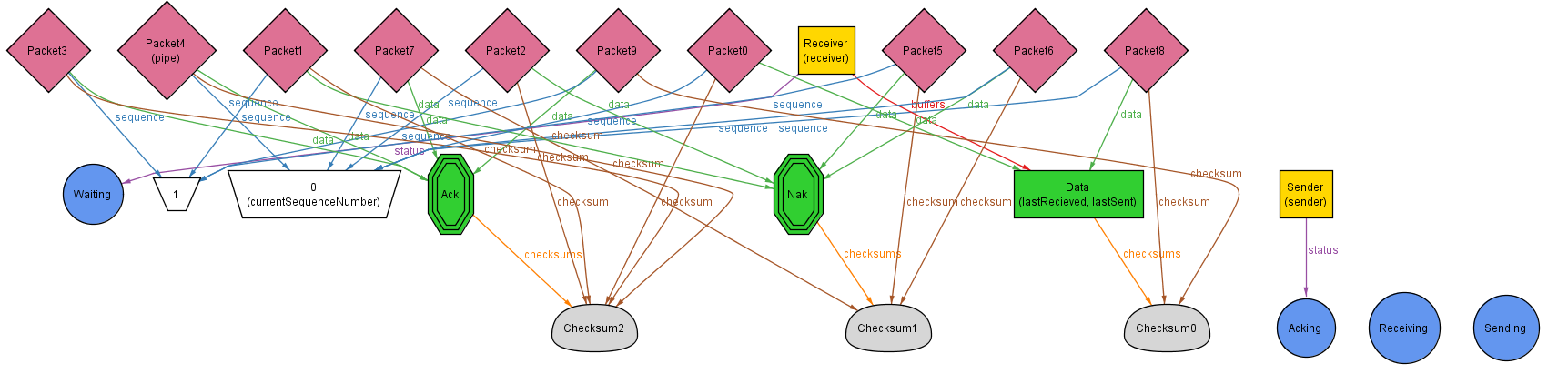
Step 7:

The receiver sees that there is data in the pipe, and transitions into the receiving state to prepare to get the data out of the pipe.



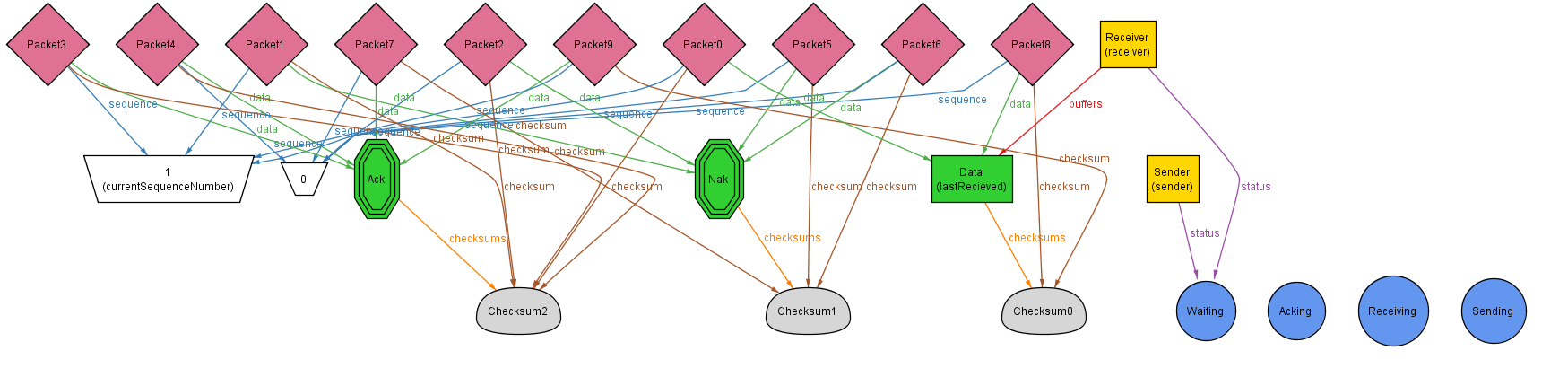
Step 8:

The receiver pulls the data packet from the pipe and sees that it is an uncorrupt packet. As a result, the data is brought into the buffer and the correct Ack packet is placed into the pipe.

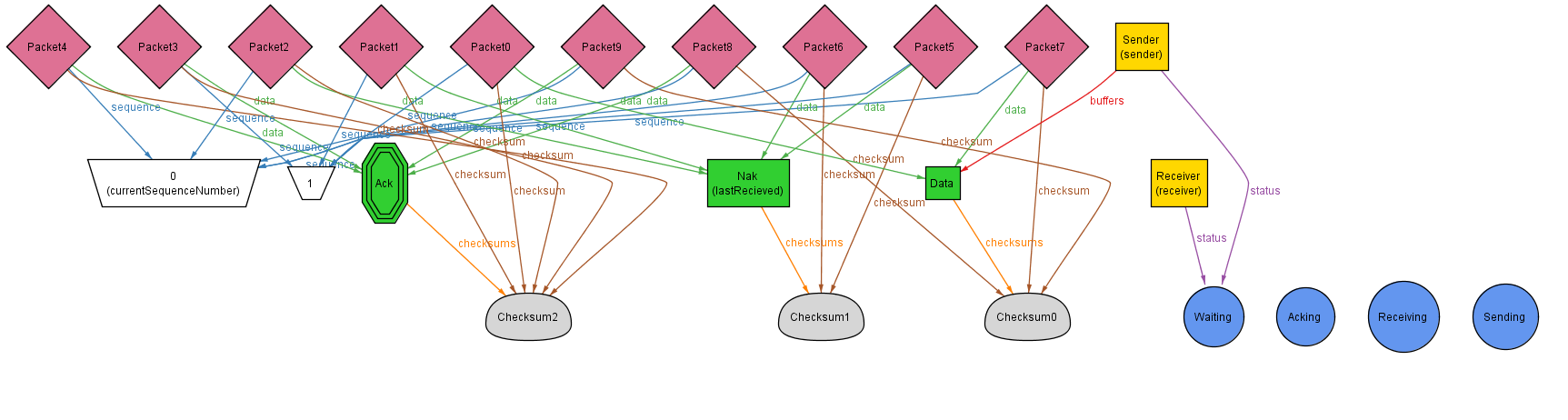


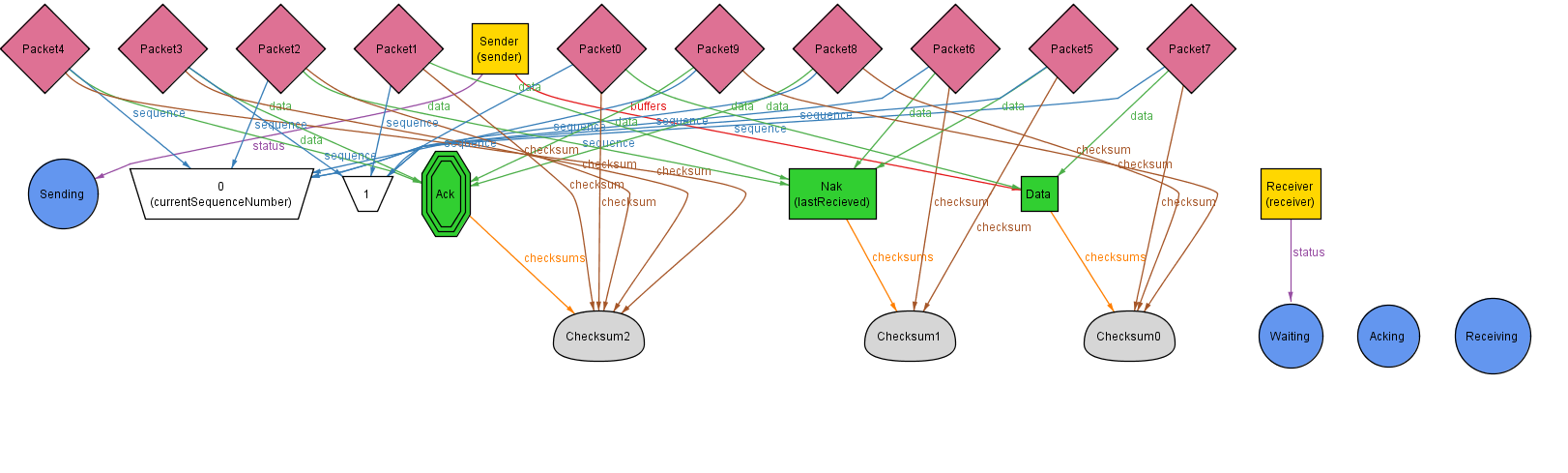
Step 9:

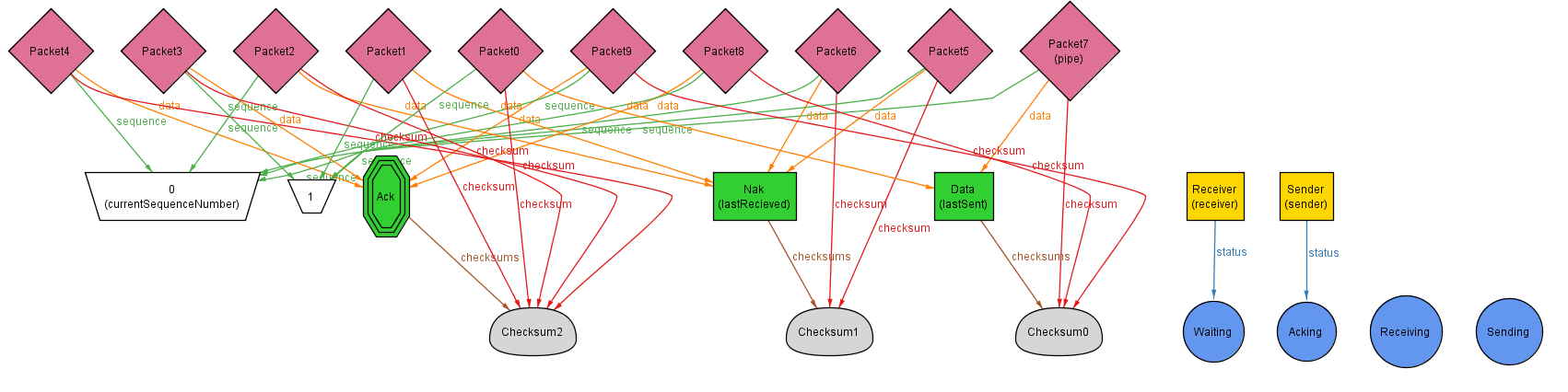
Having received the noncorrupt Ack, the transfer is complete, and the currentSequenceNumber is switched.

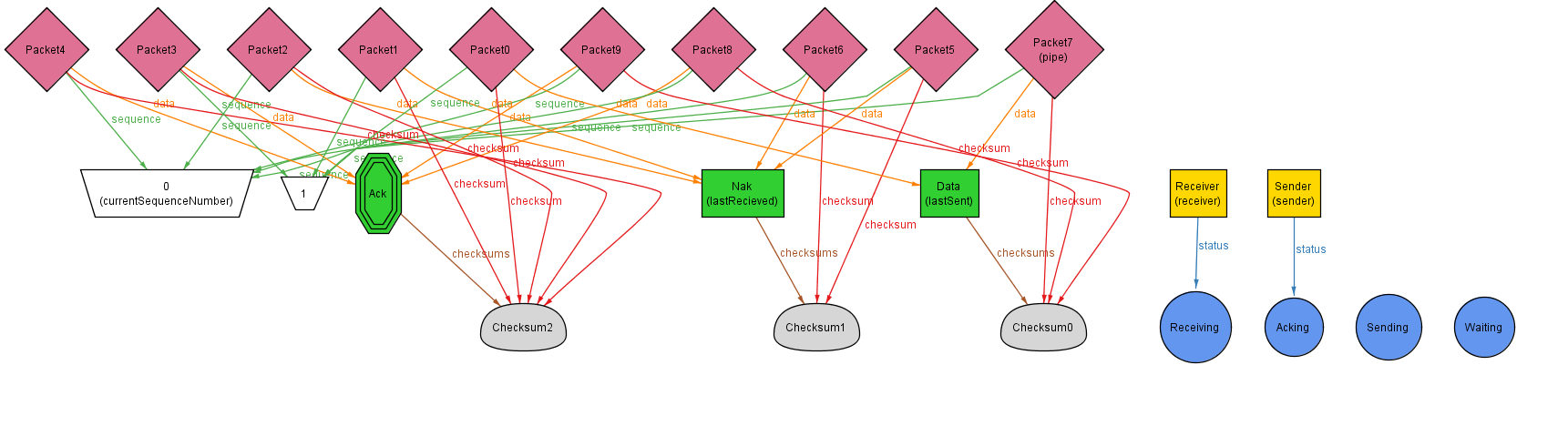


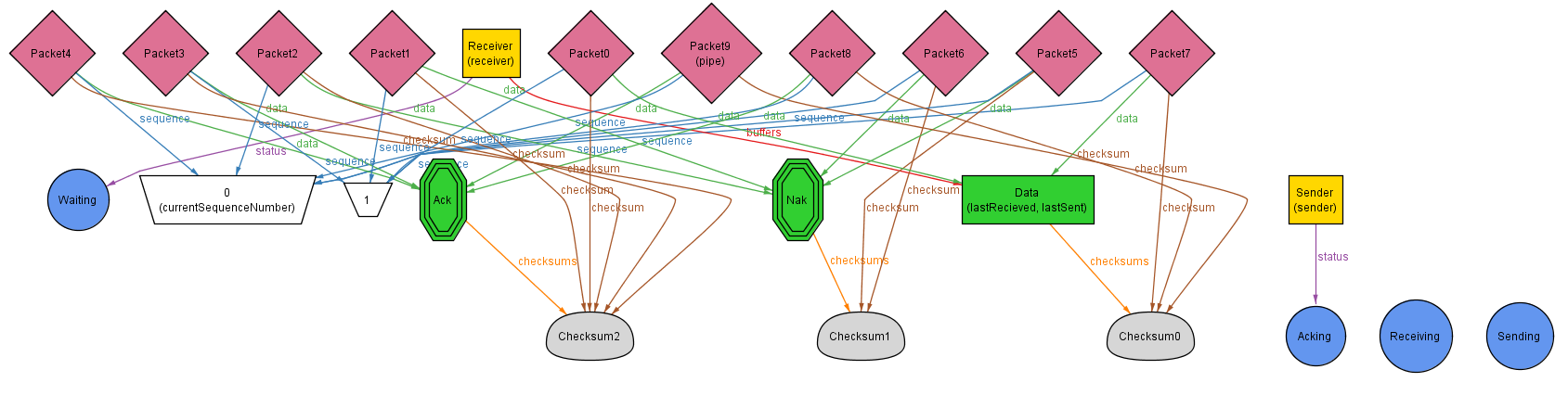
Incorrect Ack

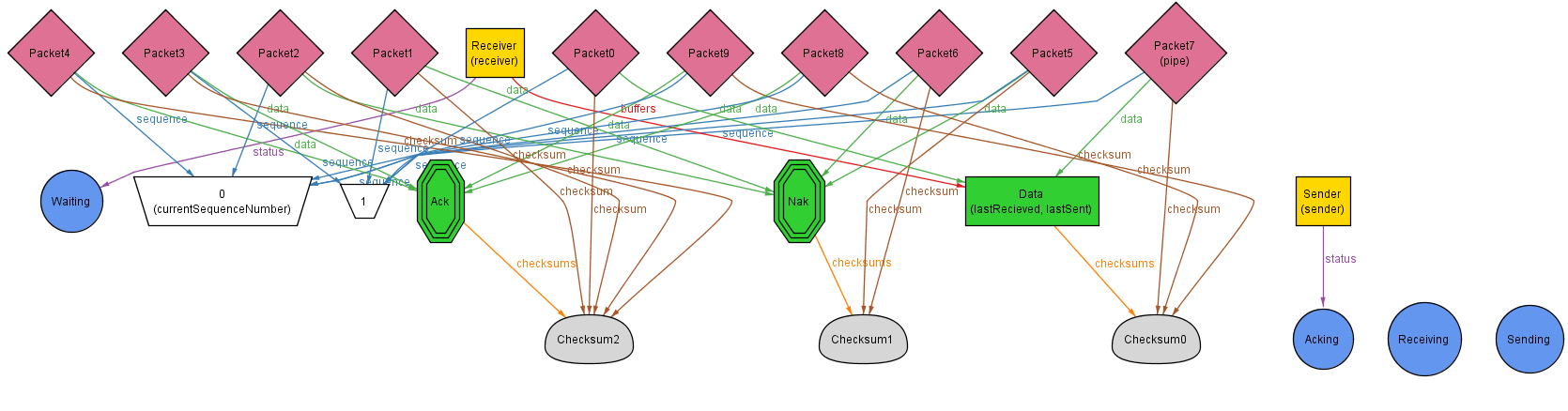


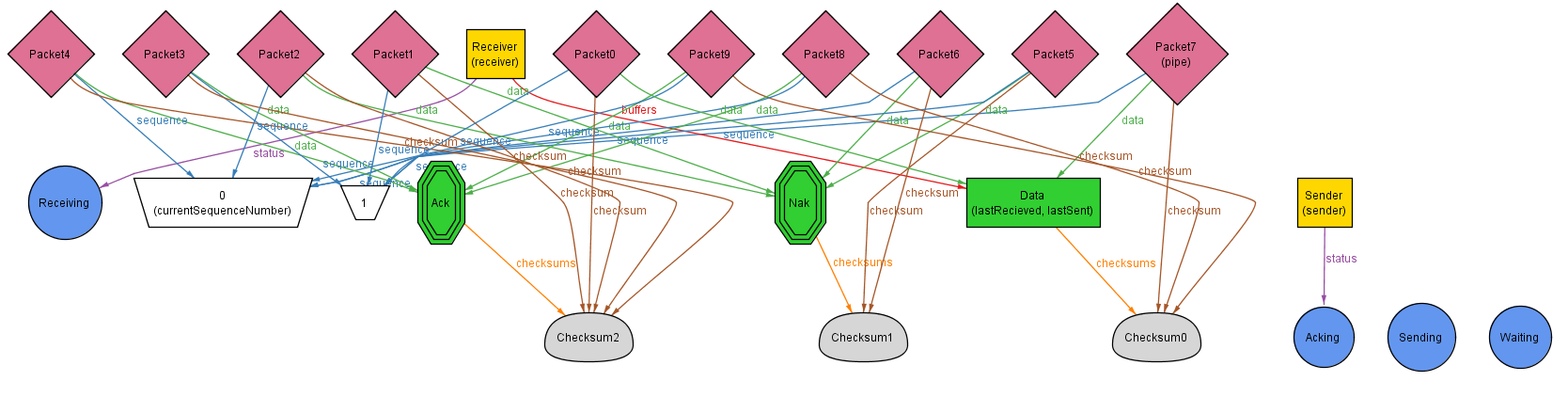


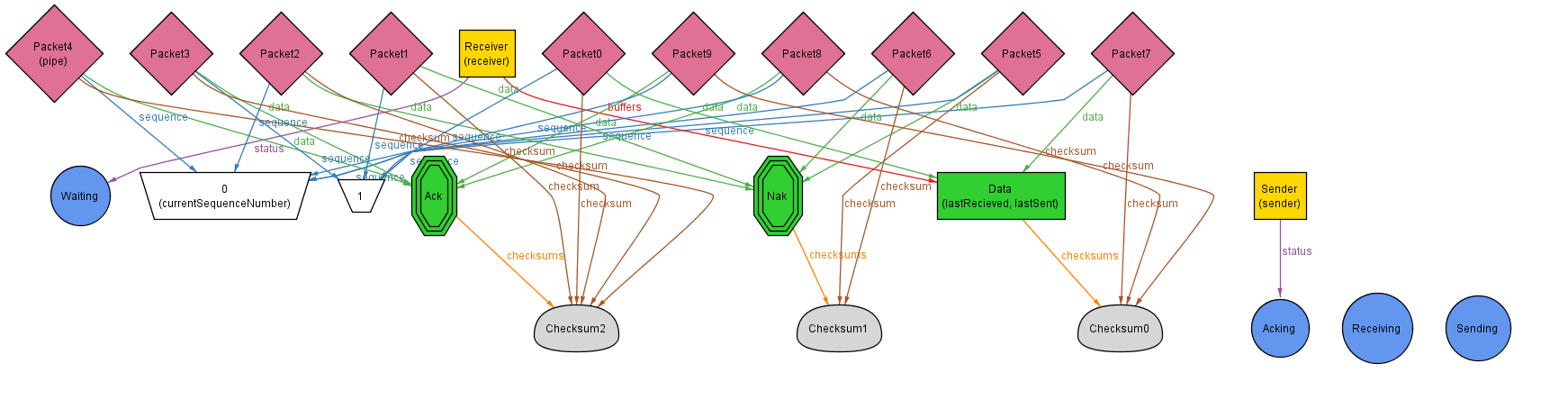


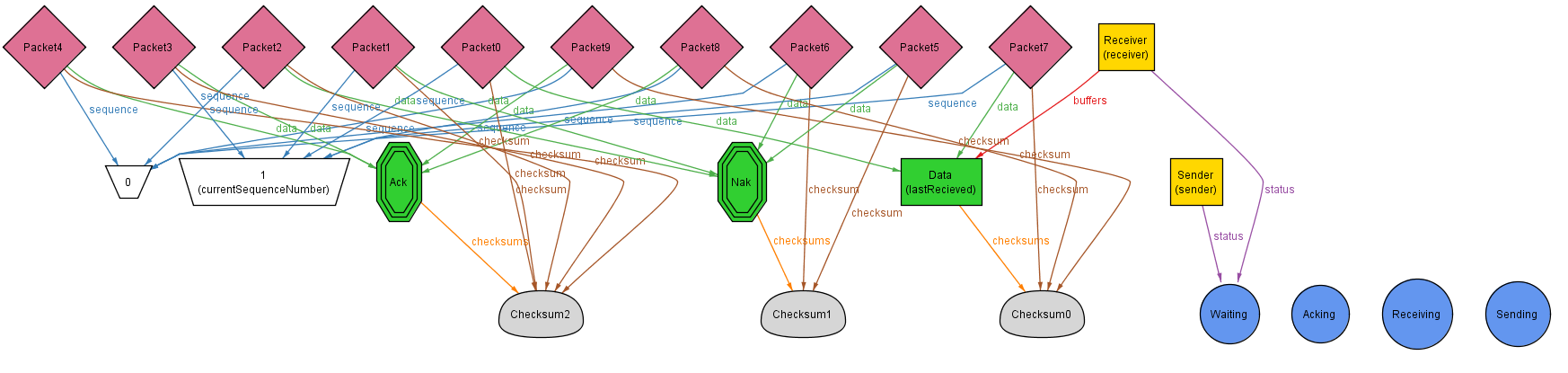












Incorrect Nak

