Project Sprint 3

# Property 1A – Successful Transfer without Corruption

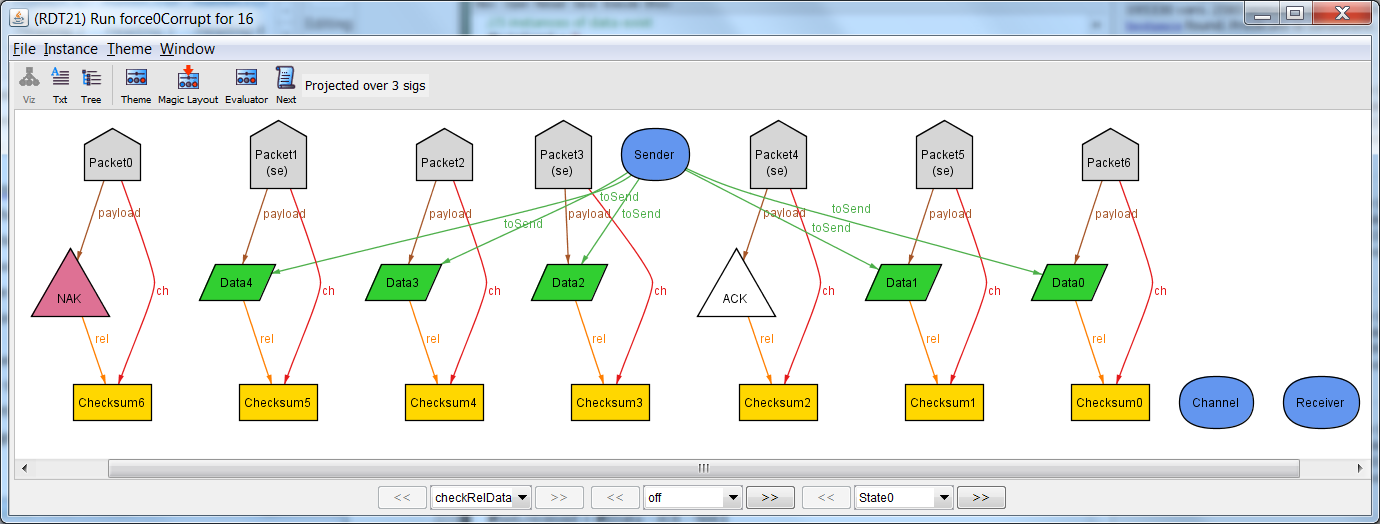
It is possible for the data to be successfully transferred. In this case, no corruption of data occurred during transfer.

For this trace, five instances of data (green parallelograms) will be transferred. These pieces of data are initially held in Sender (blue circle). Packets (grey pentagons) each contain a piece of data as well as a checksum (yellow boxes). The Receiver (blue circle) will contain data after it is successfully transferred. The Channel (blue circle) will contain packets while they are being transferred. Finally, the Ack (white triangle) and Nak (red triangle) are sent after each receive operation to indicate if the transfer was successful or corrupted. For this trace, no corruption will occur, so the Nak will never be sent.

These states follow the pattern of Send Data, Receive Data, Send Response. In Send Data, a data from Sender is transferred through the Channel. In Received Data, the transferred packet’s checksum is compared to the packet’s contents. If they match, then the data is added to the Receiver. If they do not, the data is ignored. In Send Response, if the previous receive was successful, then an Ack is sent through the Channel. Otherwise, a Nak is sent through the Channel. The process then repeats, with the Data selected by Sent Data being chosen based on whether an Ack or a Nak was sent in the previous step.

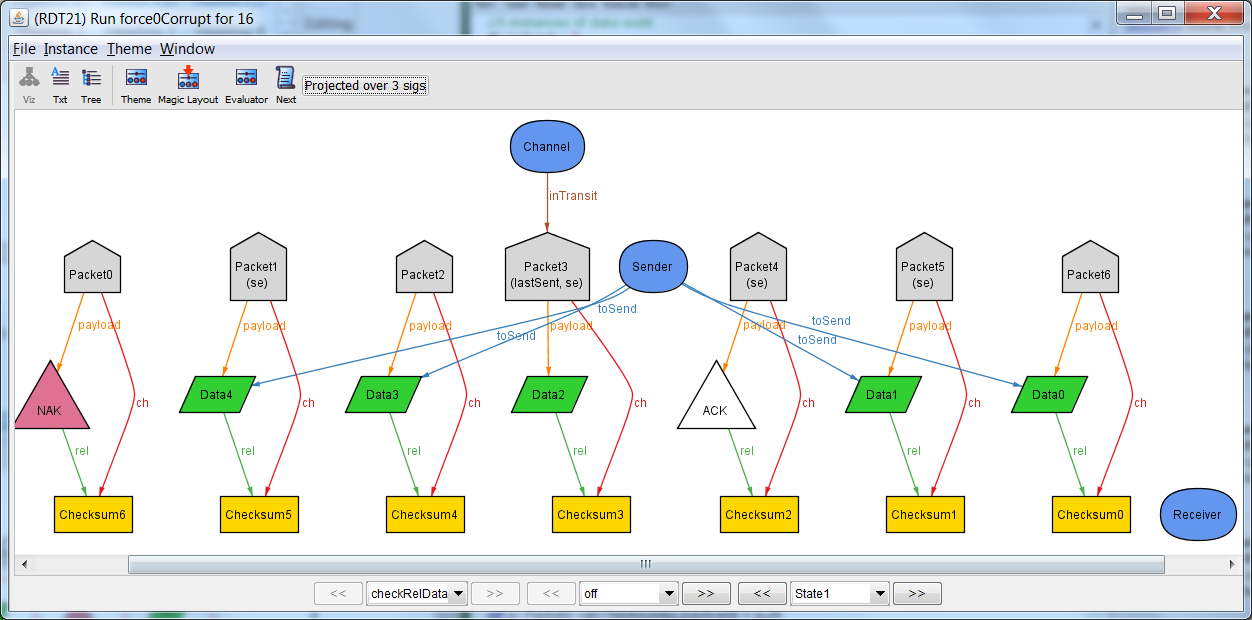
Each Packet also has a sequence bit which starts off at 0. This sequence bit alternates each time a NEW piece of data is sent. If the Ack or Nak is corrupted, the Sender treats this like a Nak and resends the data with the same sequence bit as last time. If the receiver gets two Packets with the same sequence bit, it knows the new packet is duplicate information and can ignore the packet.

## State 0



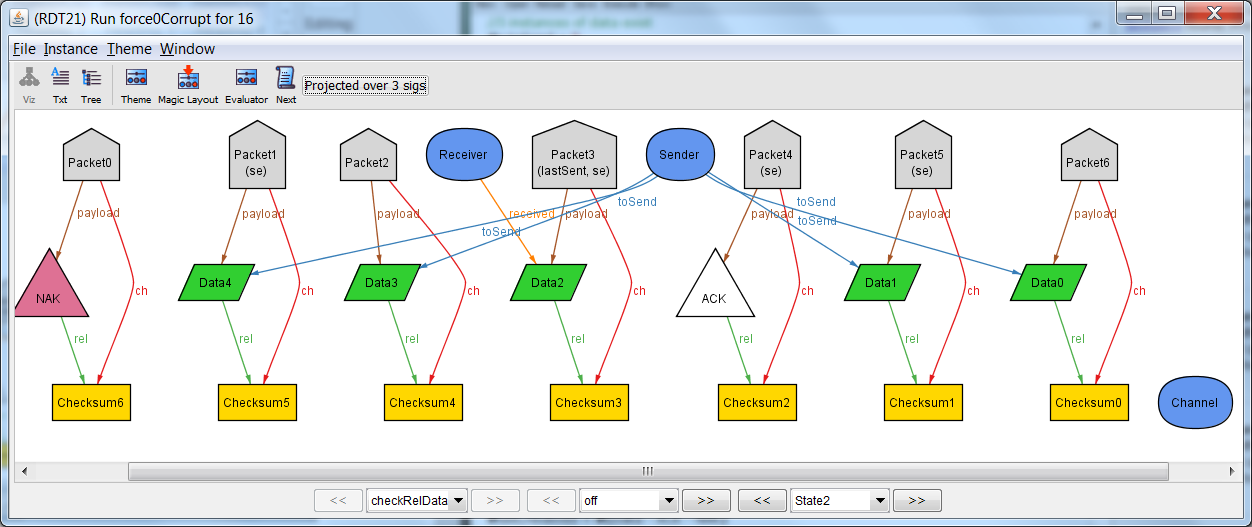
In the initial state, all five instances of data are contained in the Sender. Nothing is being transferred in the channel. Nothing has been received in the receiver.

## State 1

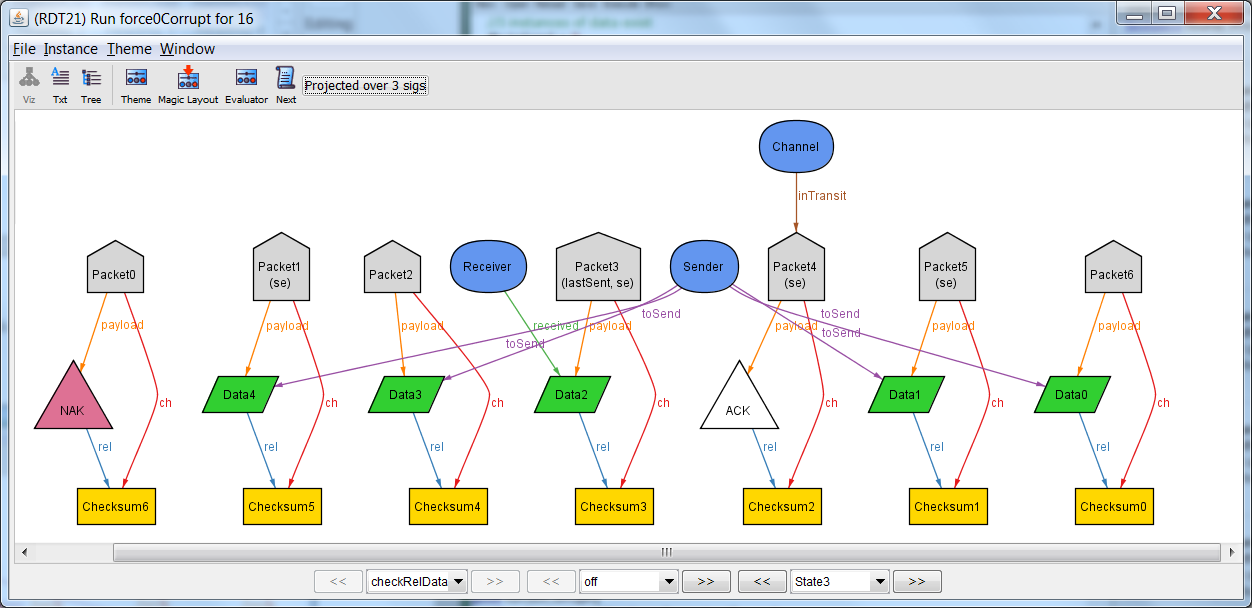


In this state, the first send occurs. Data 3 is no longer contained in the Sender. Packet 2, which contains Data 3, is now being transferred through the Channel.

## State 2

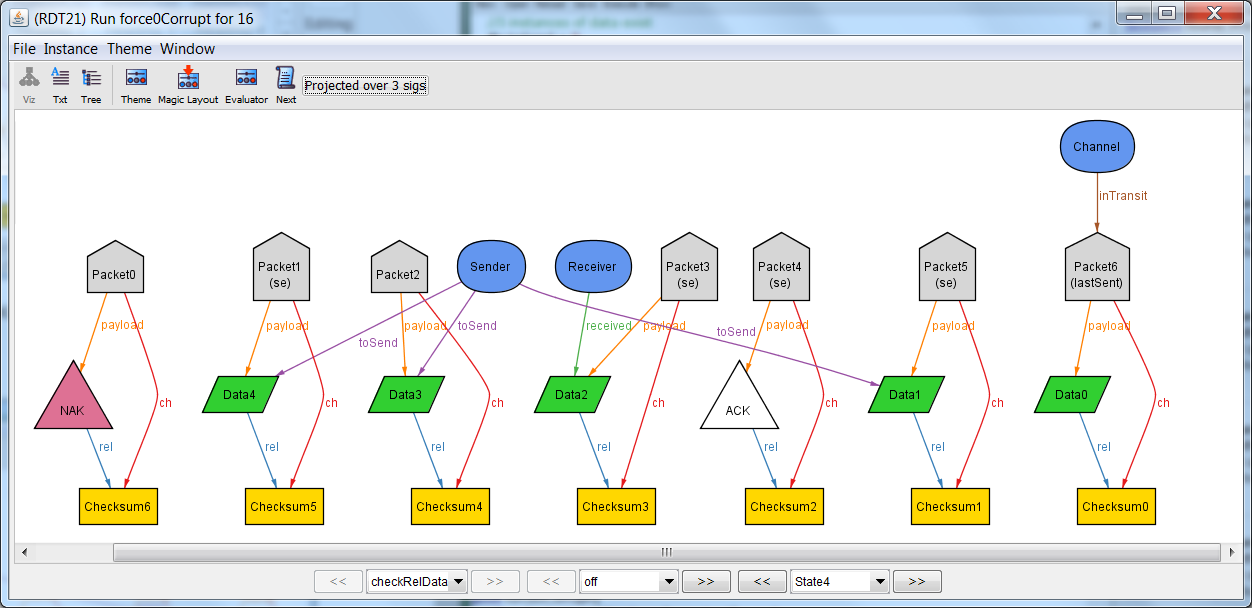
In this state, the first receive occurs. Data 3 is no longer being transferred through the Channel. The Checksum of the transferred Packet matches the contents of the Packet, so no corruption occurred. Data 3 is now contained in the Receiver.

## State 3



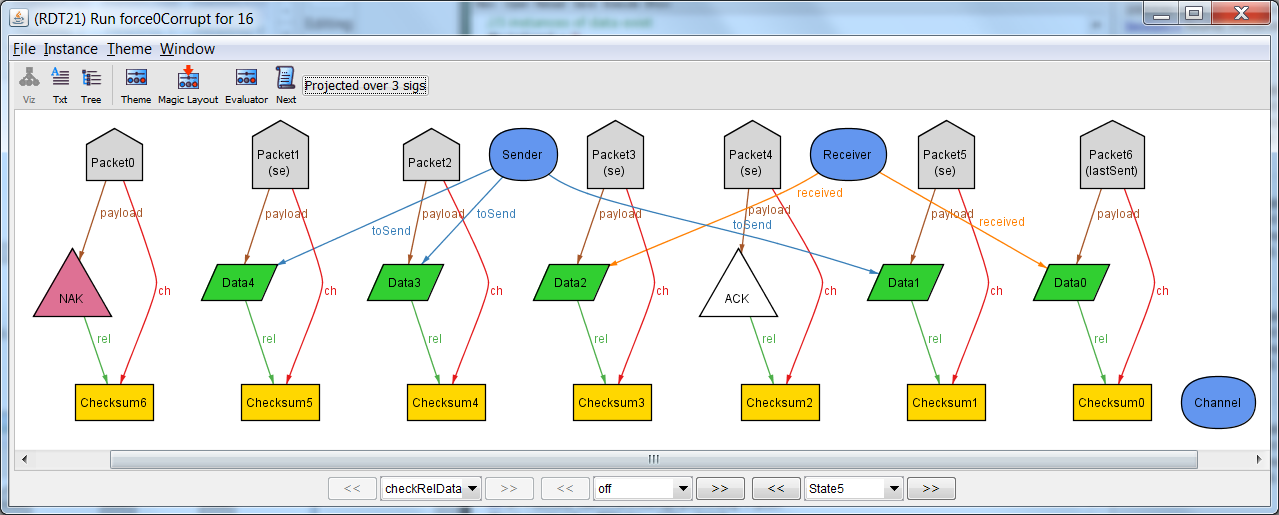
Since the data received in the previous state was not corrupted, the Ack is now sent through the Channel. A full cycle of steps has now come to an end.

## State 4



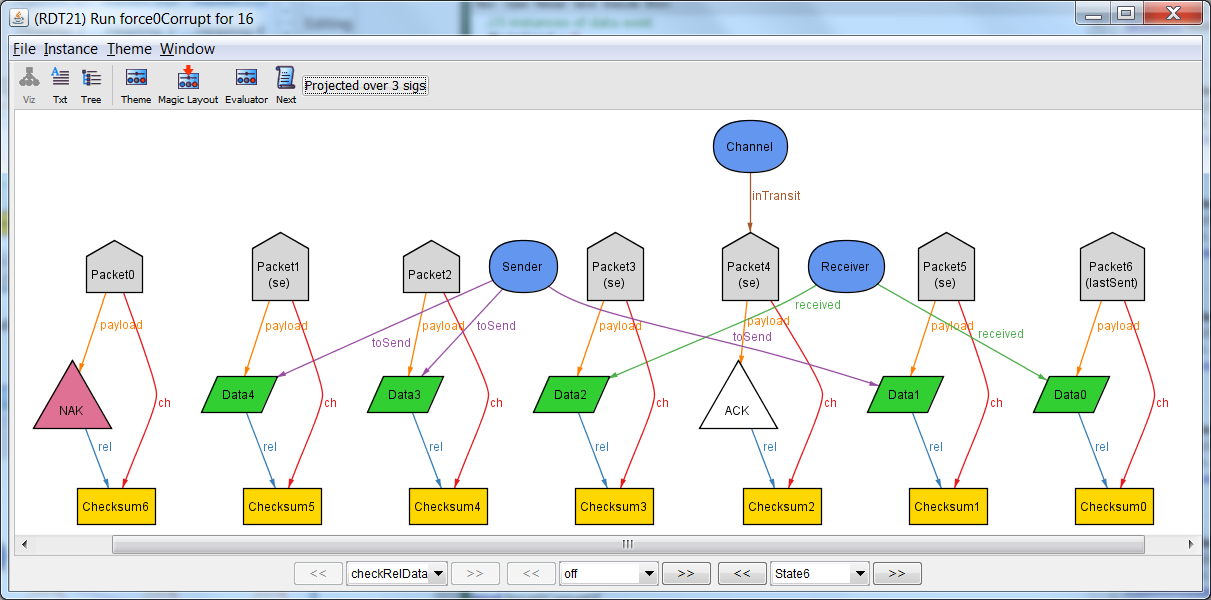
A new cycle starts with a send. Since the previous state transferred an Ack, a new piece of data is sent.

## State 5



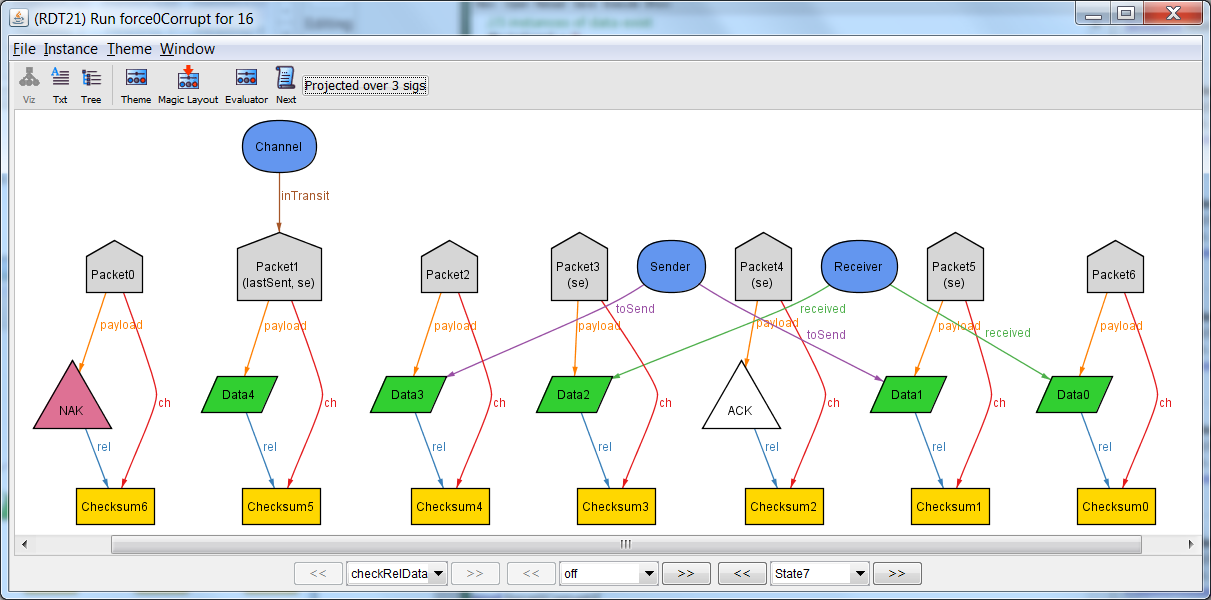
Receive: The checksum and contents matched, so the data is added to the receiver.

## State 6



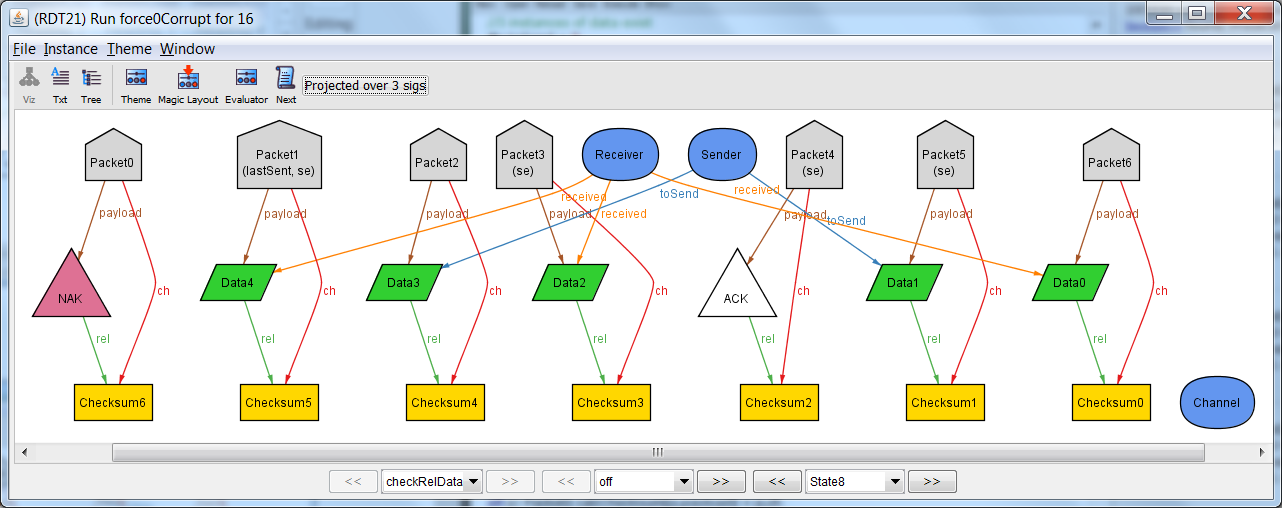
Response: The last transfer was successful, so Ack is sent back.

## State 7

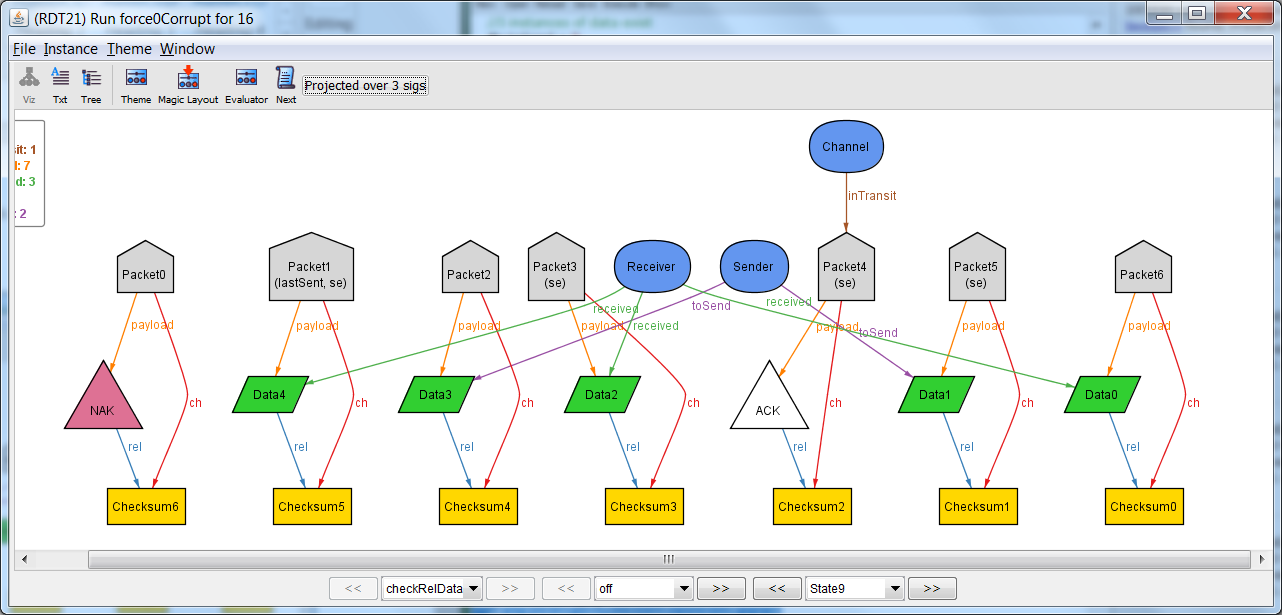


Send: Ack was received, so send a new data.

## State 8

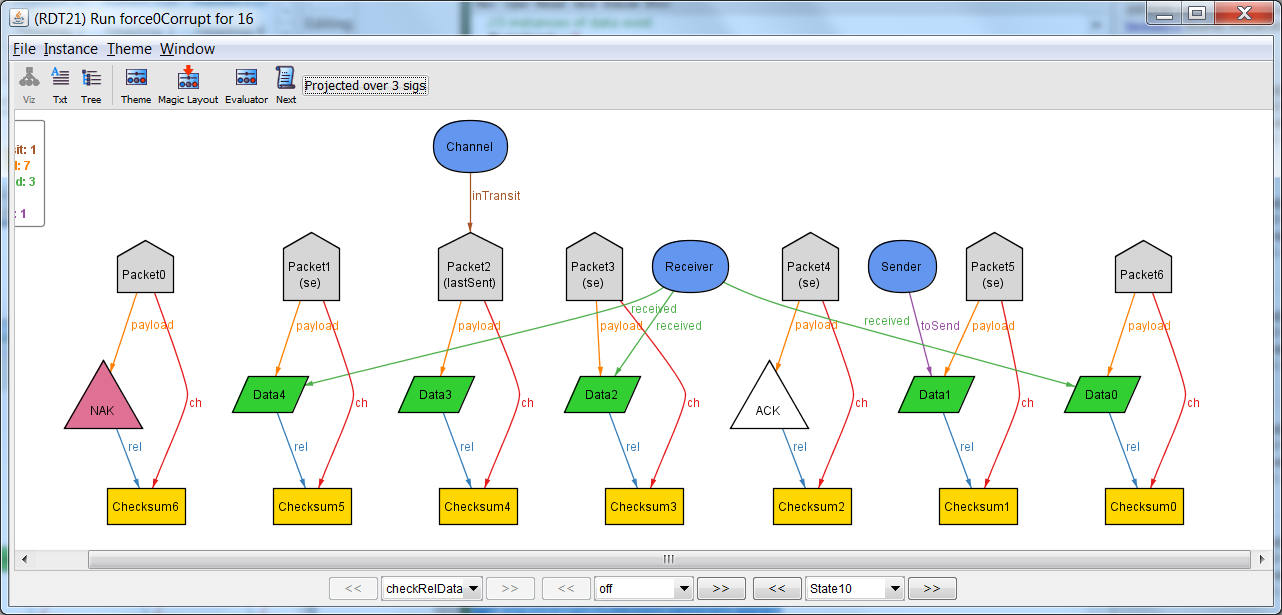
Receive: The checksum matched the contents, so the receive was successful.

## State 9

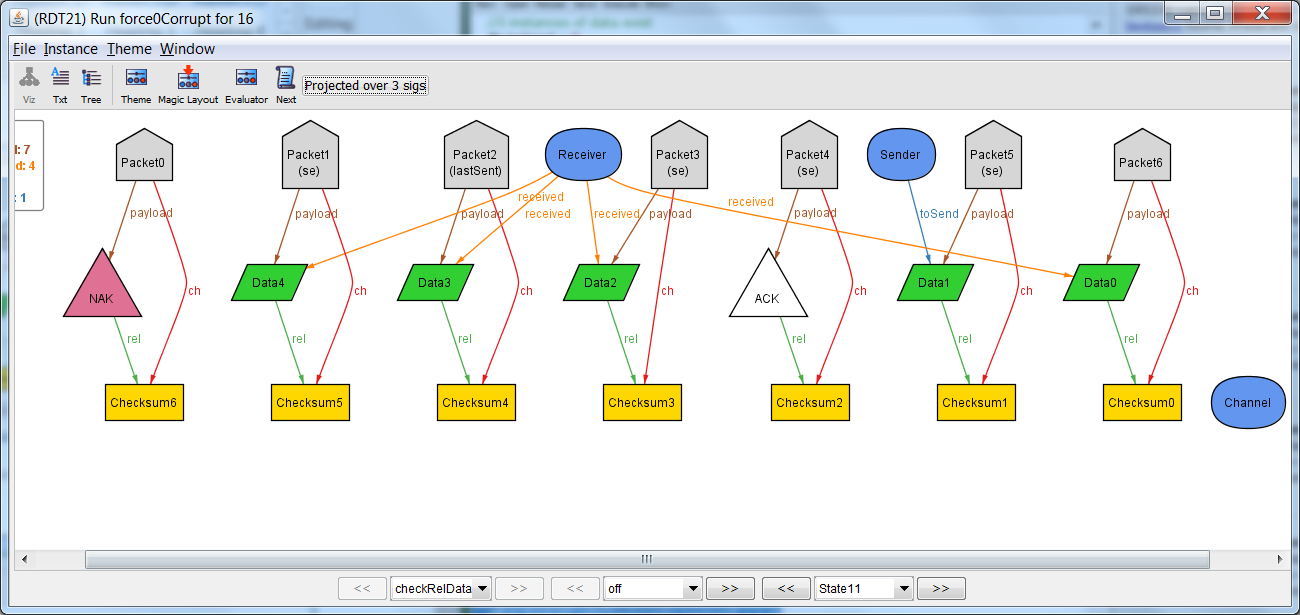


Send Response: The last receive was successful, so send Ack.

## State 10

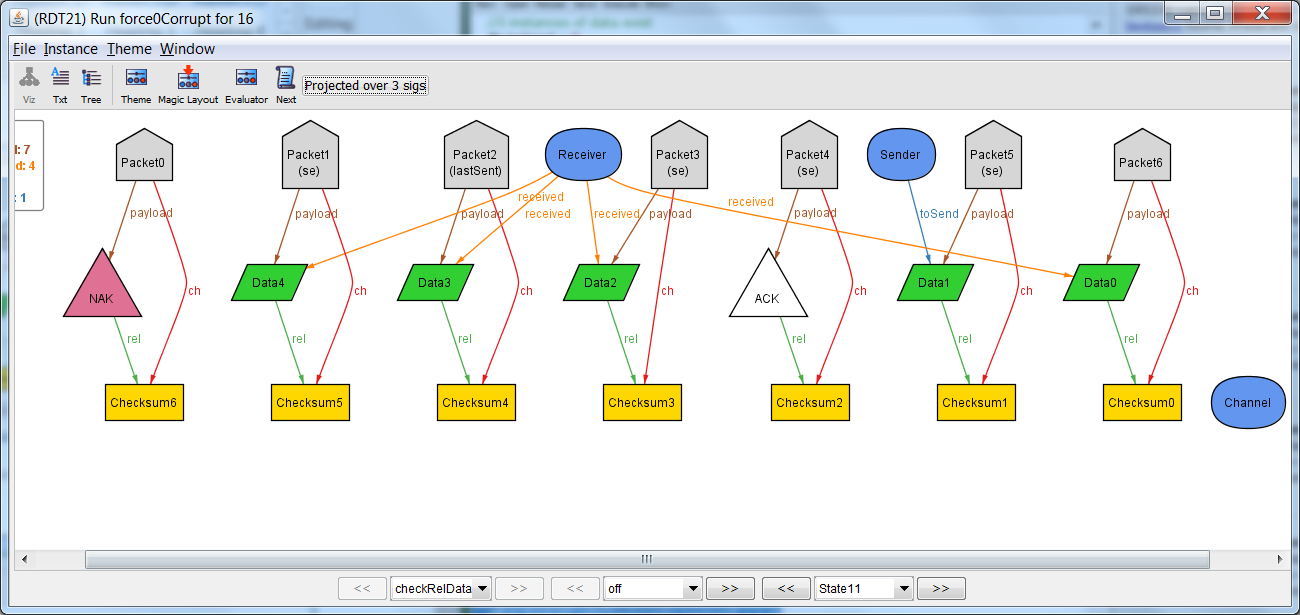
Send: Ack was received, so send new data.

## State 11



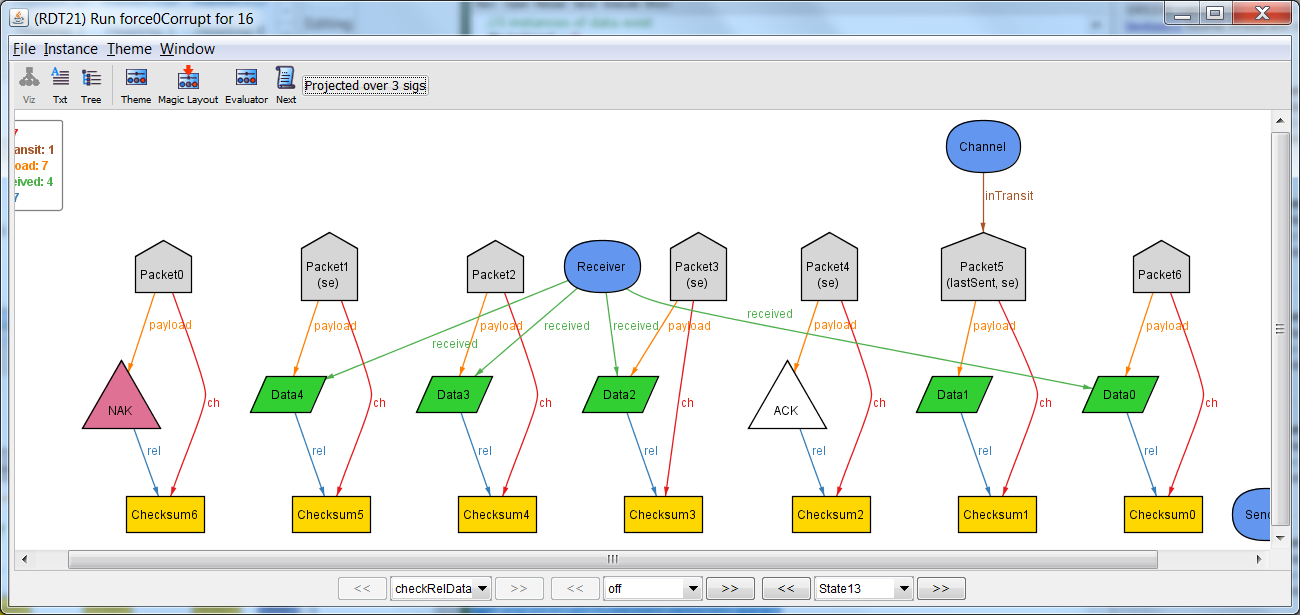
Receive: The checksum matches the contents, so the receive is successful.

## State 12



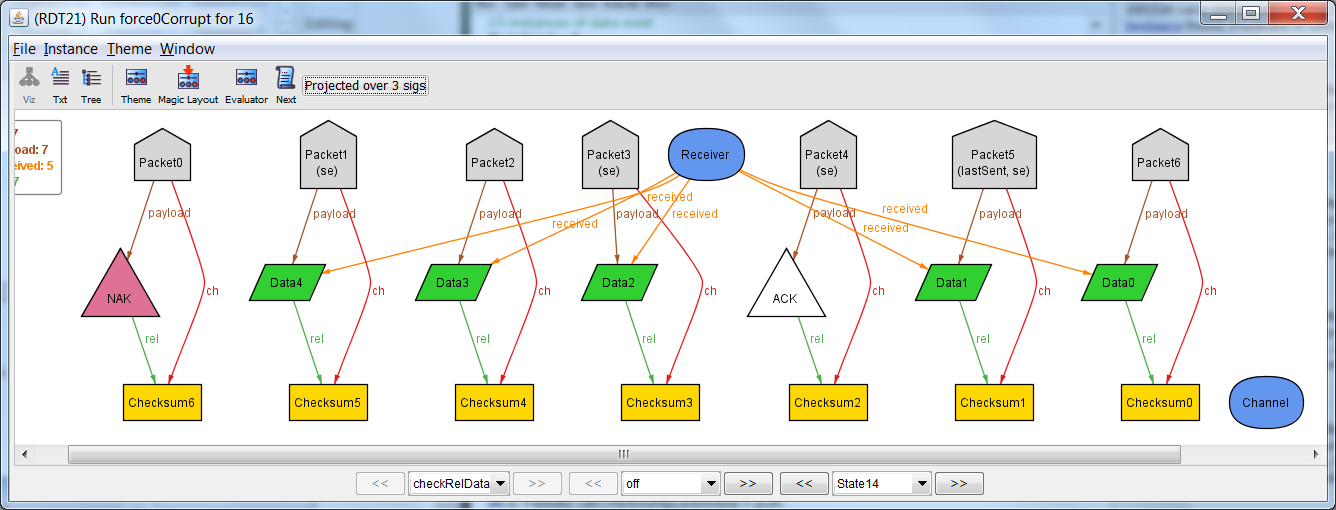
Send Response: The last receive was successful, so send Ack.

## State 13



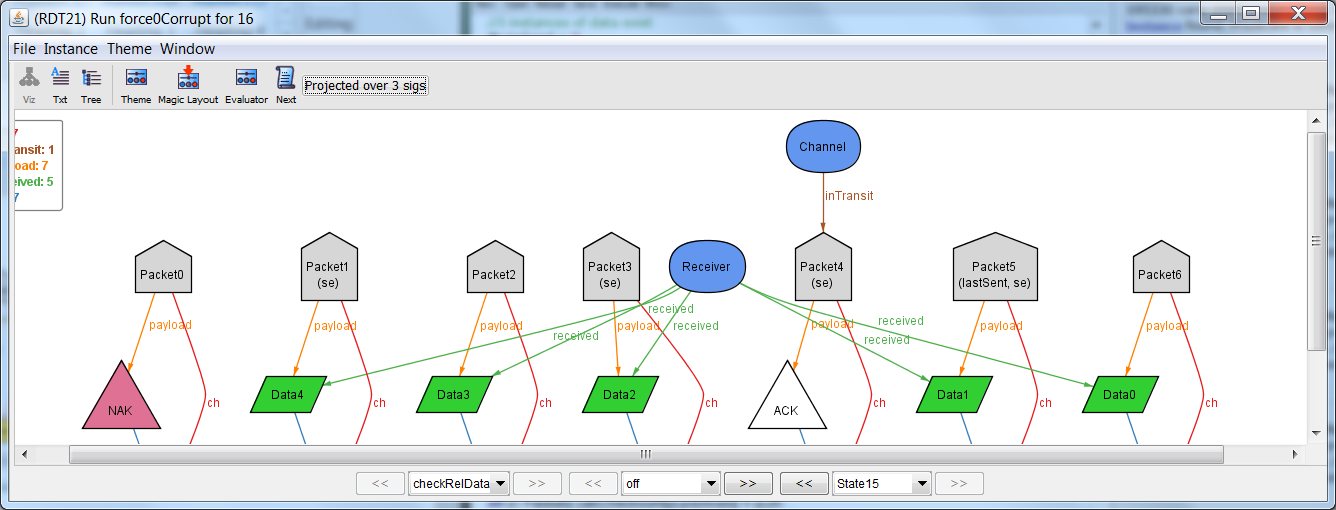
Send: Ack was received, so send new data.

## State 14



Receive: The checksum and contents match, so the receive was successful.

## State 15



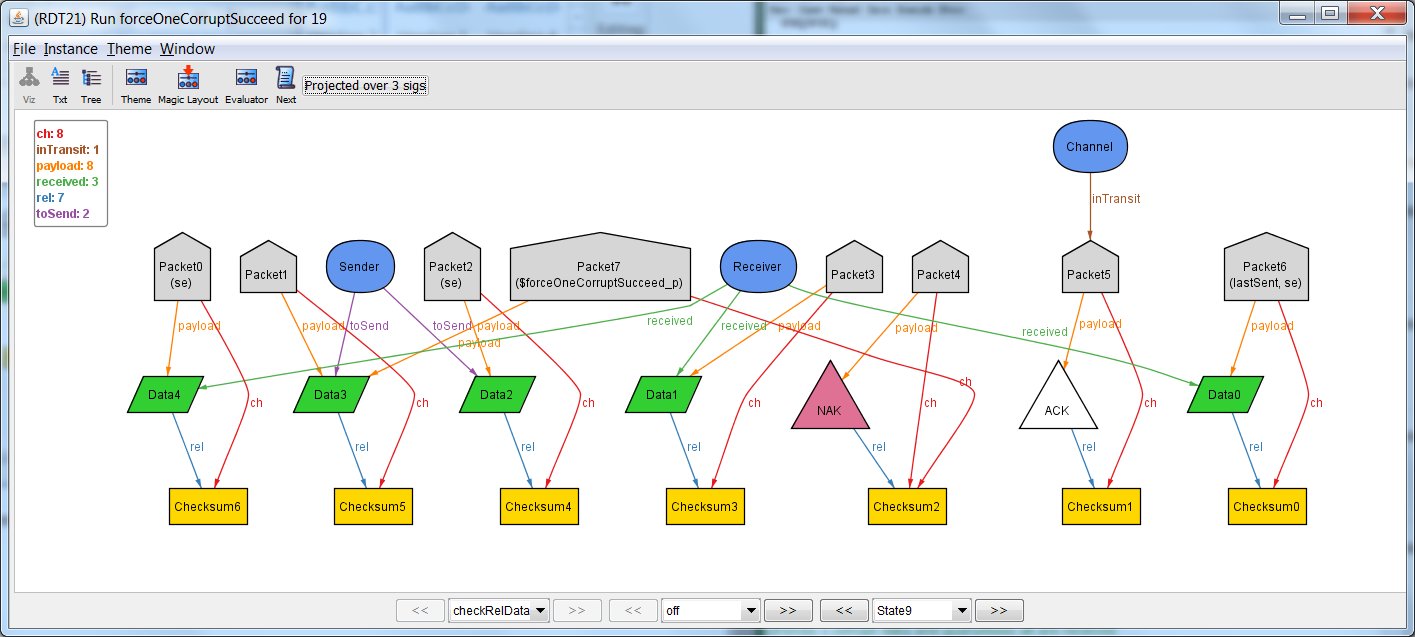
Send Response: The previous receive was successful, so send Ack.

All of the data was successfully received by the Receiver. Thus property 1 holds when there is no corruption.

# Property 1B – Successful Transfer with Data Corruption

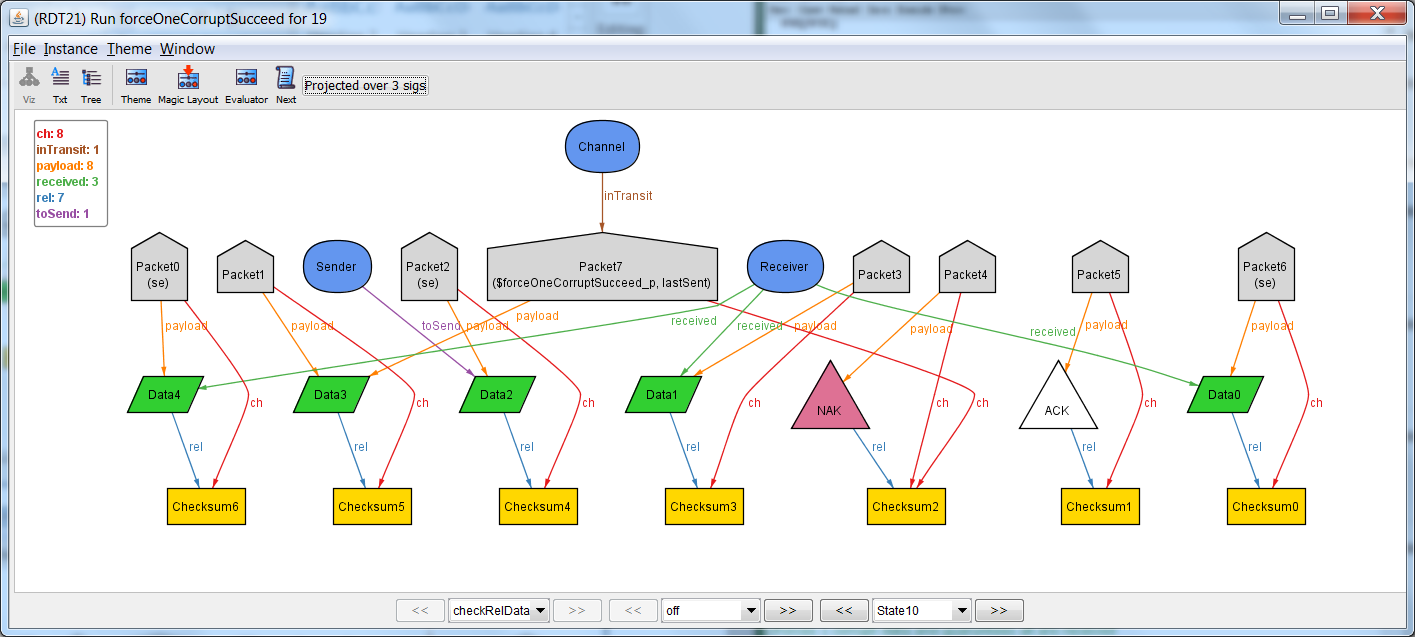
## State 0-9



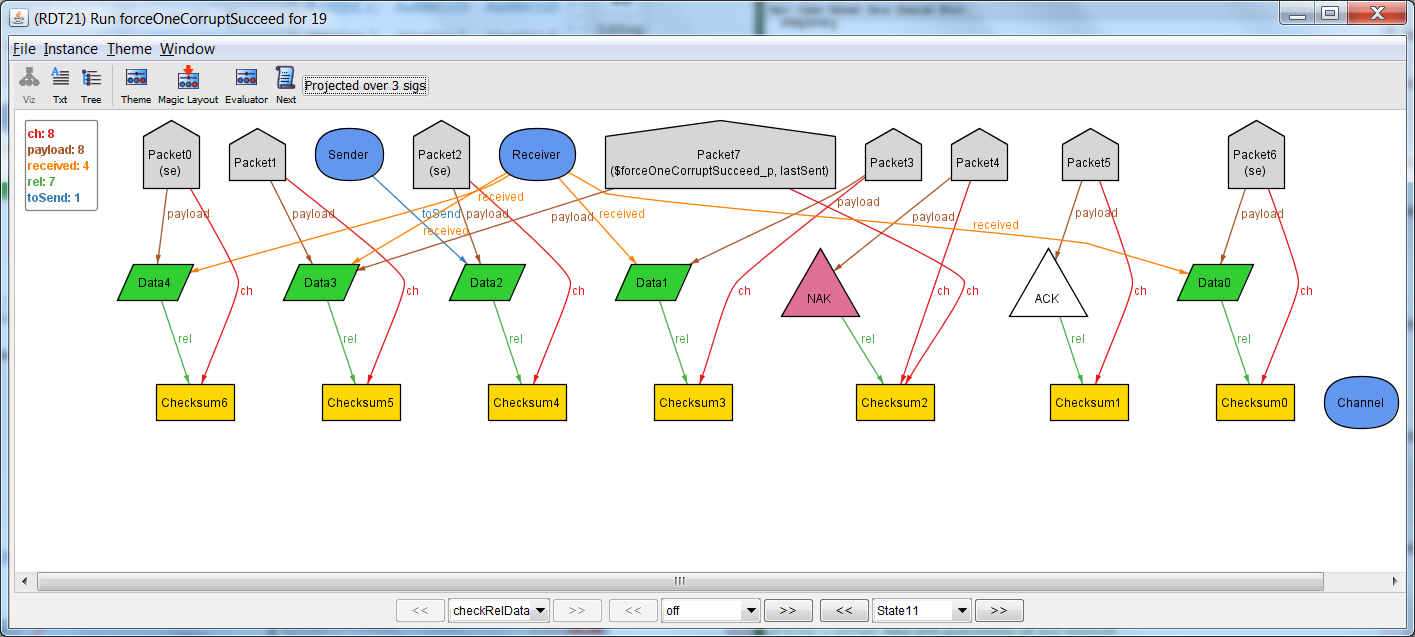


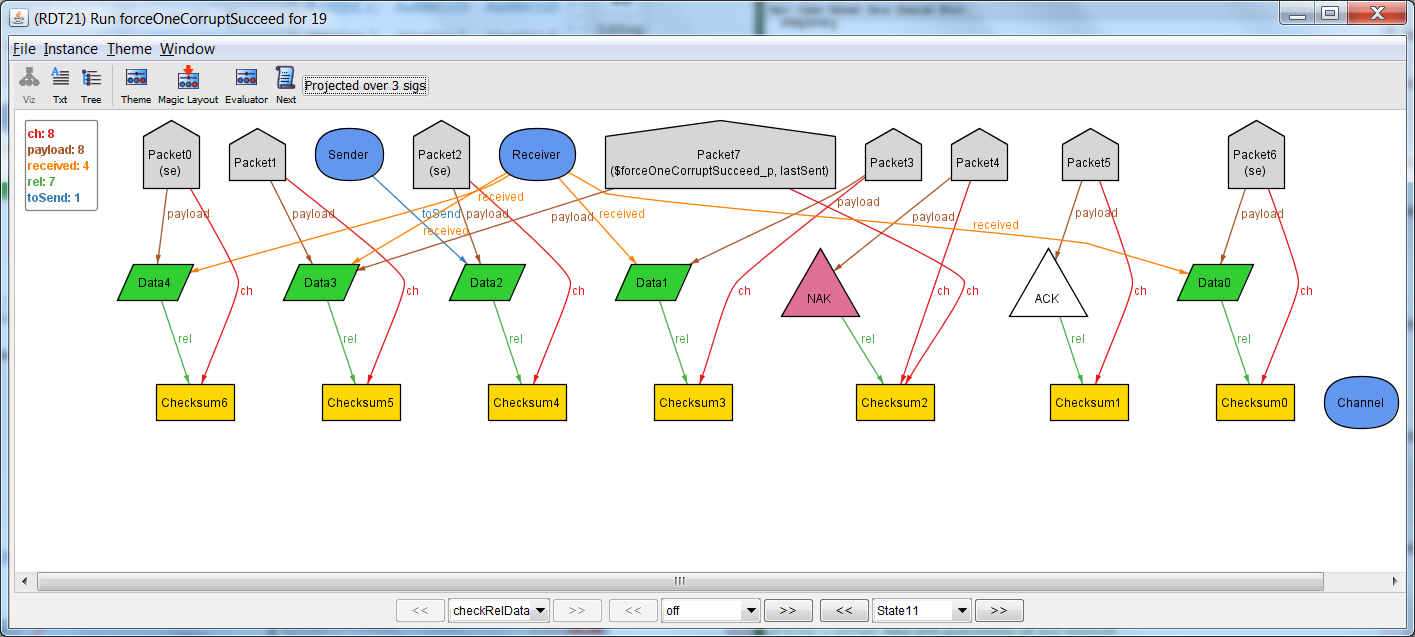
Nothing interesting happens

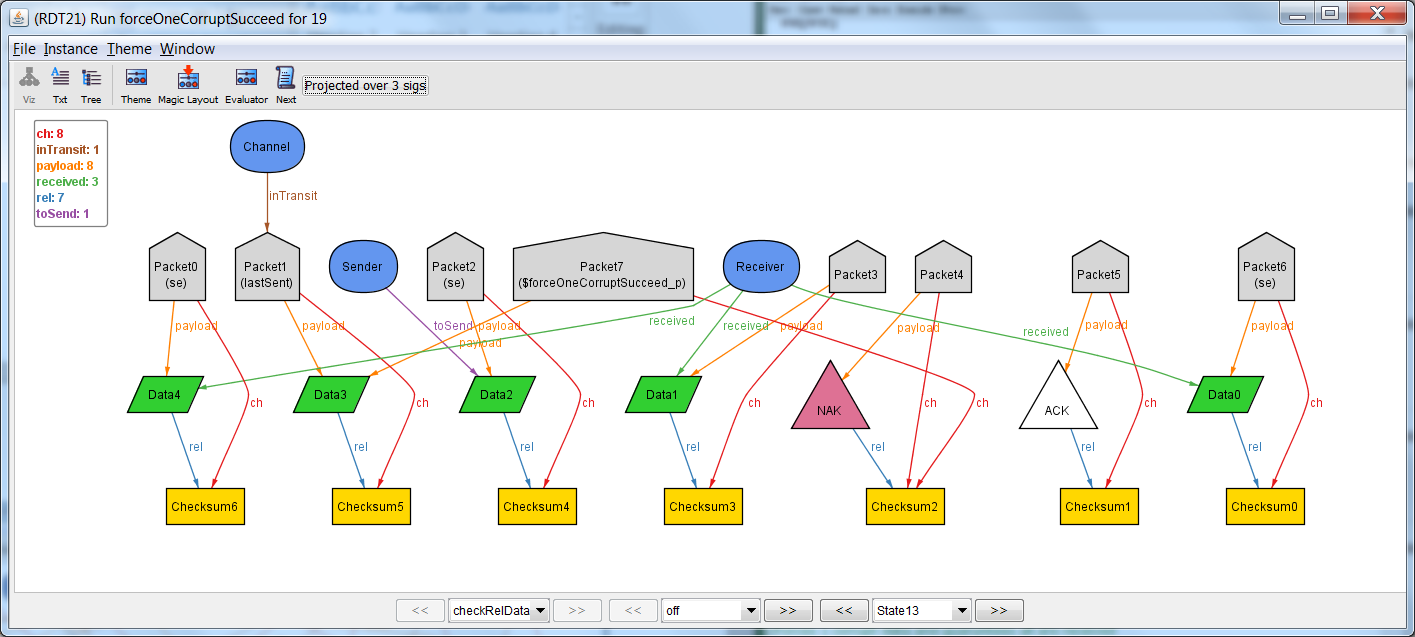
## State 7

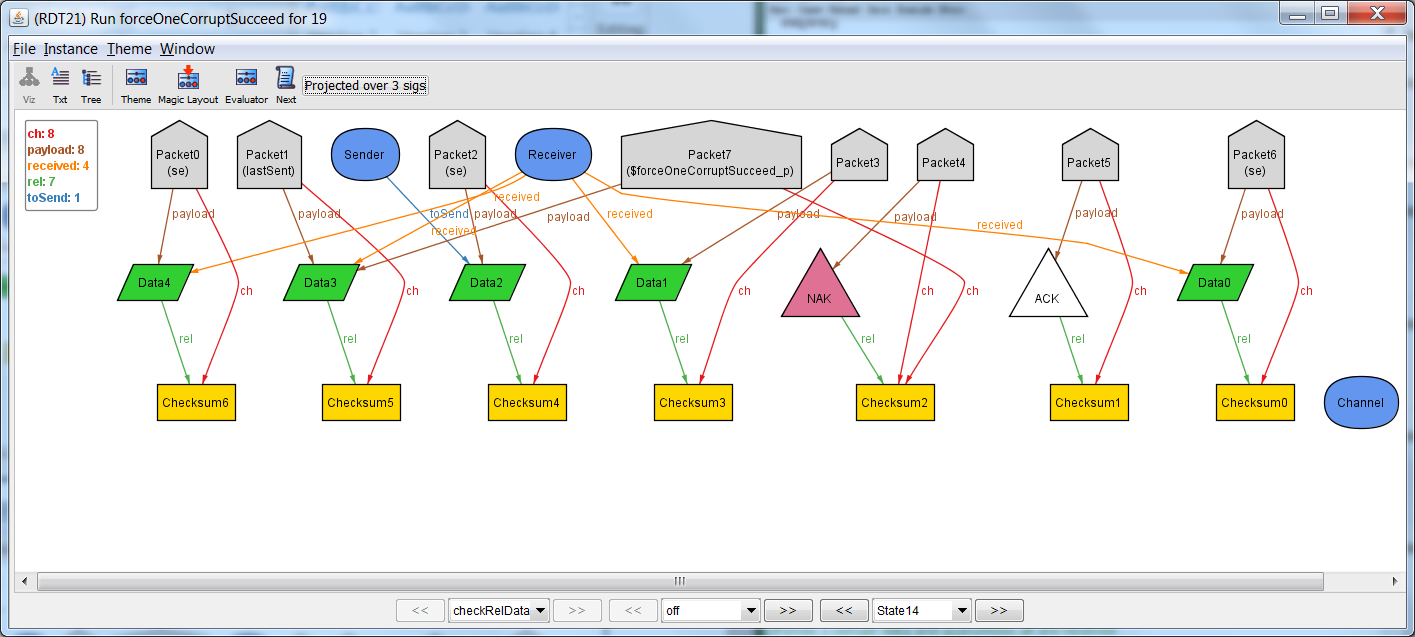


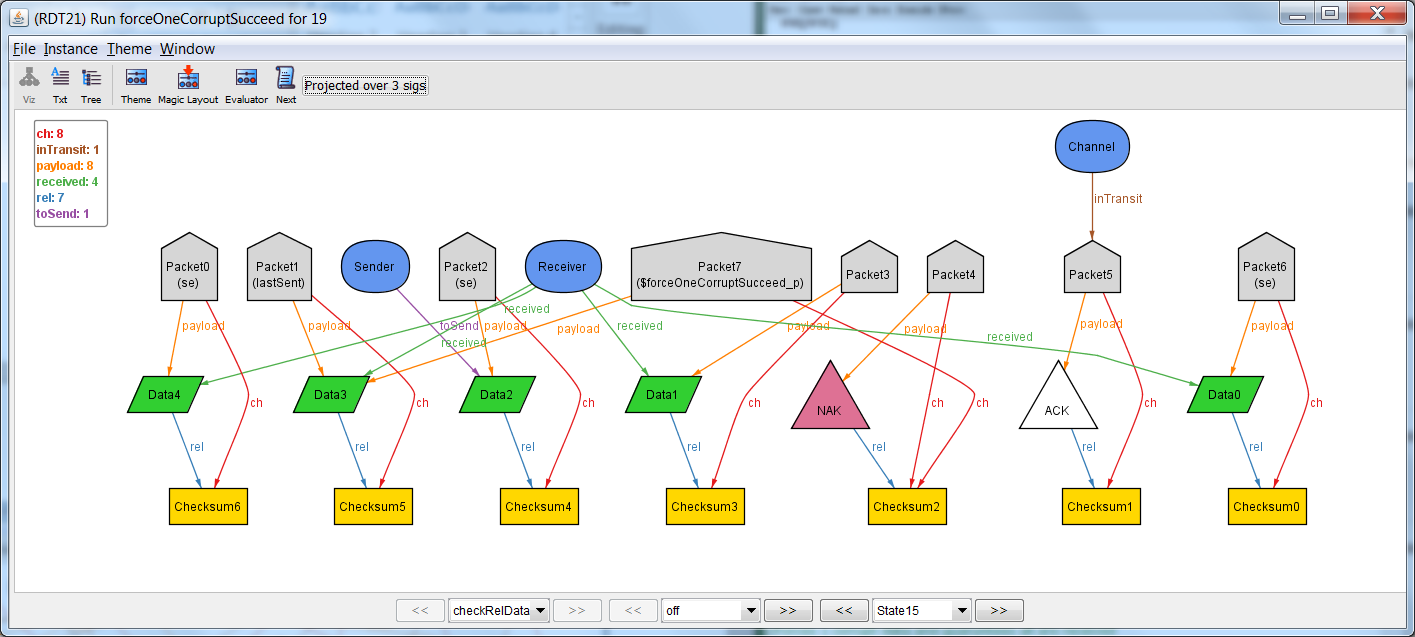
Suddenly a corrupt packet!









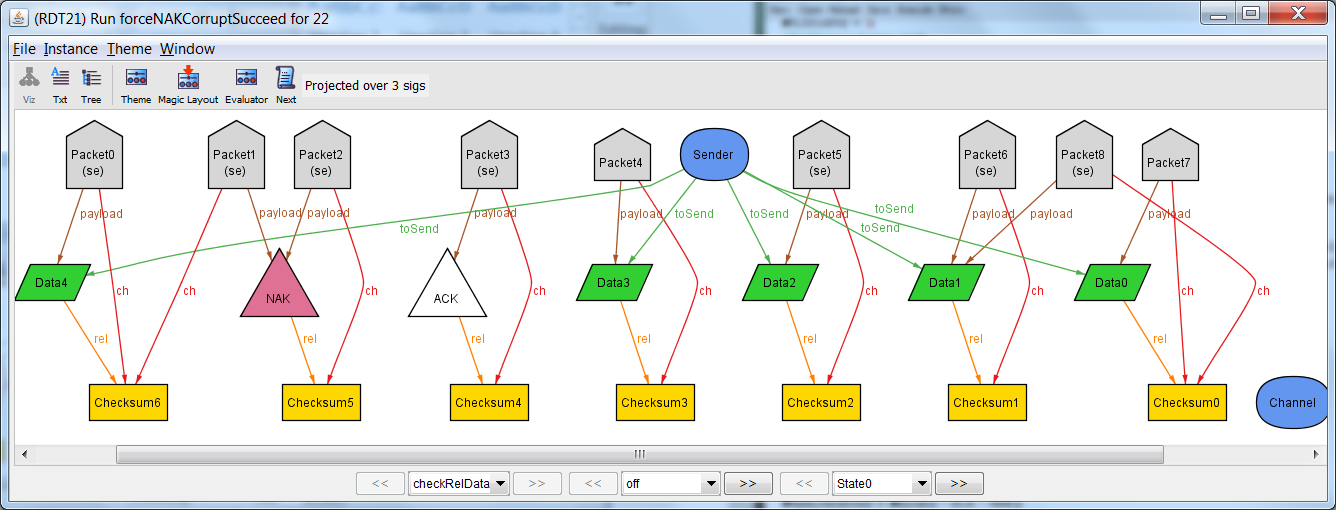


## State 16-18

The rest of the states are unremarkable. All data goes to receiver.

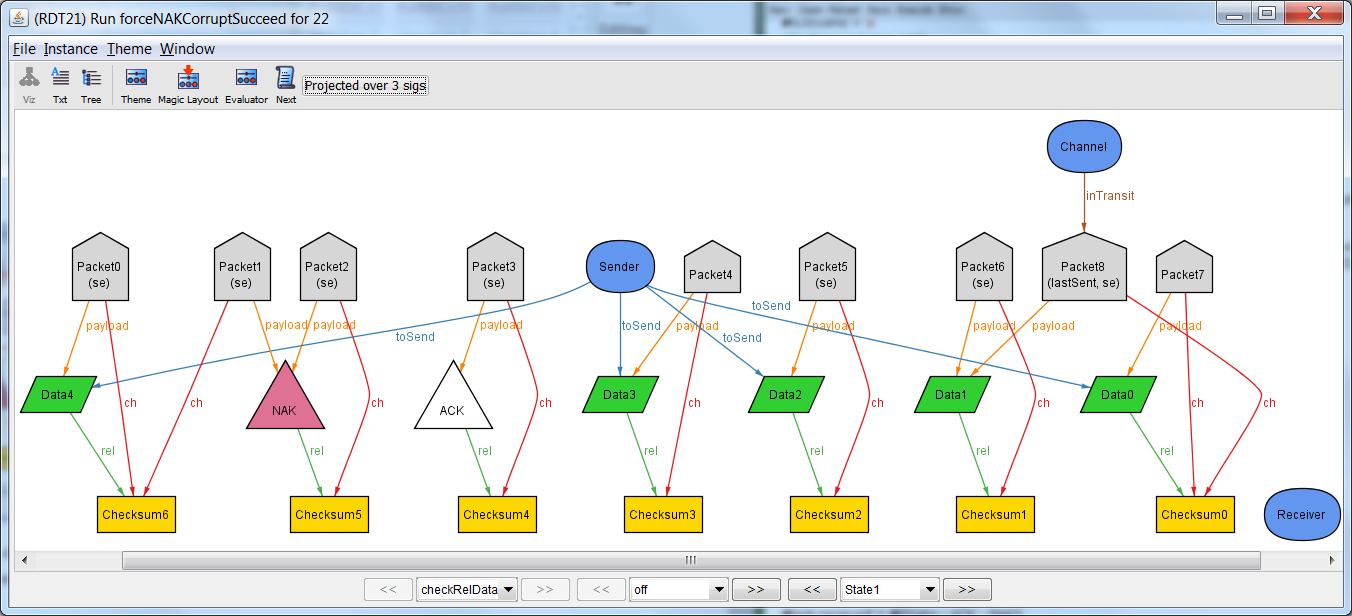
# Property 1C – Successful Transfer with NAK/ACK Corruption

## State 0



To start with, all of the data is contained in Packets and referenced by Sender. Packet 1 contains an error, as its checksum refers to Data4, while its data contains NAK.

## State 1



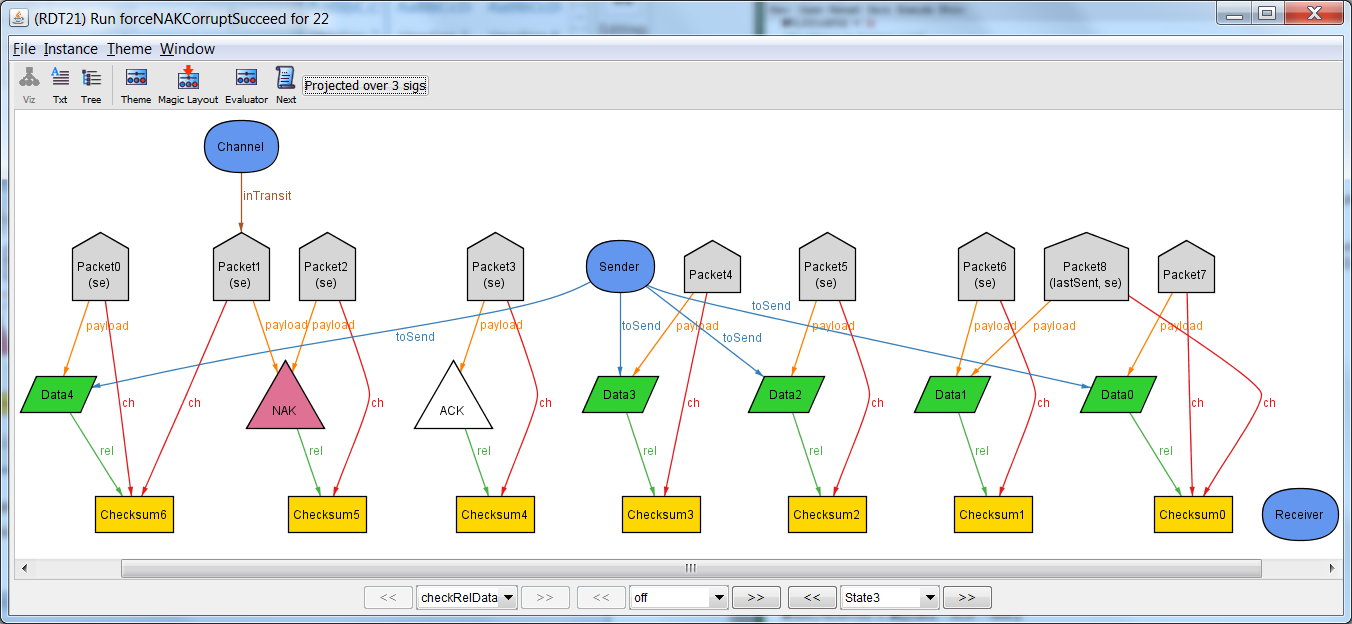
State 1 places packet 8 into the channel. Packet 8 is corrupt: its data does not match its checksum. (sequence bit 0)

## State 2



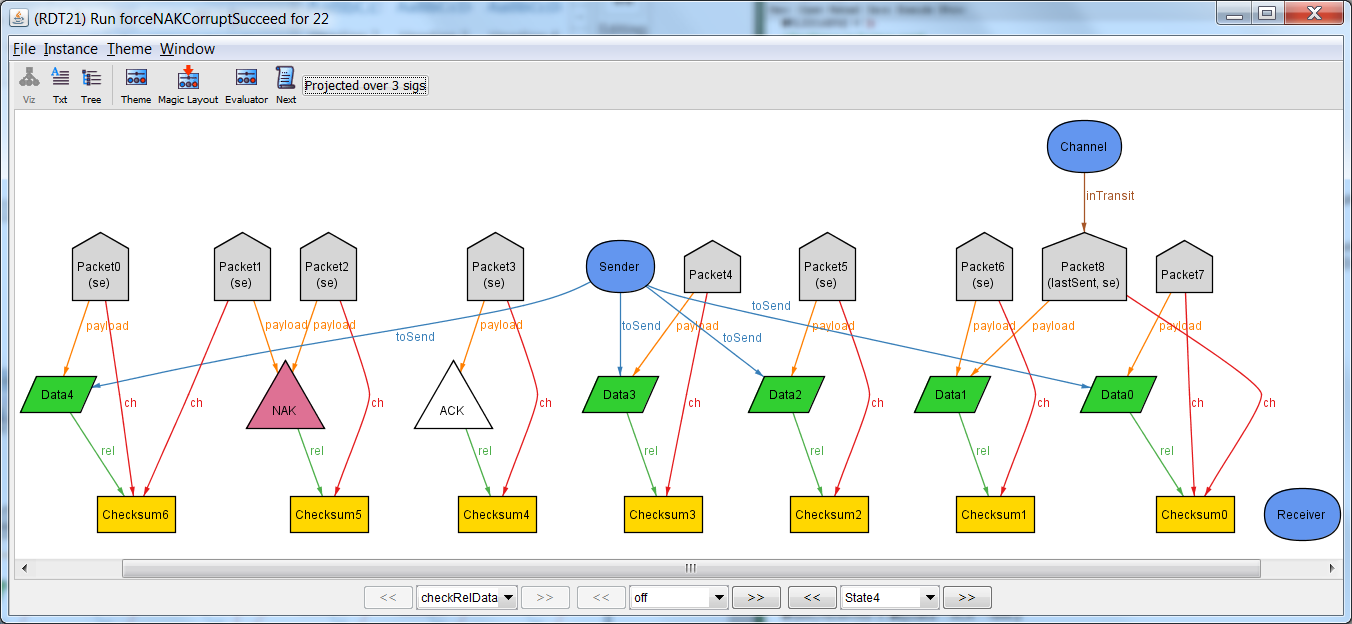
Here the receiver is attempting to receive Data 1, but the packet is corrupt.

## State 3



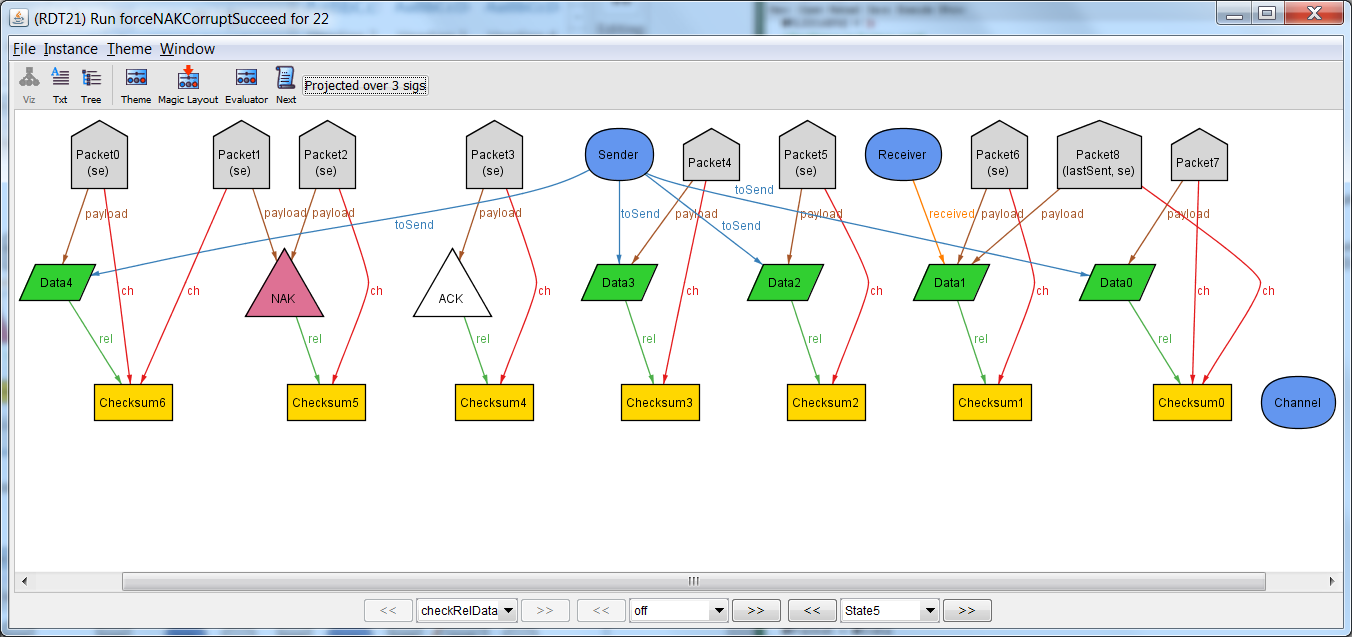
Receive replies with a NAK packet, but the packet is corrupt.

## State 4



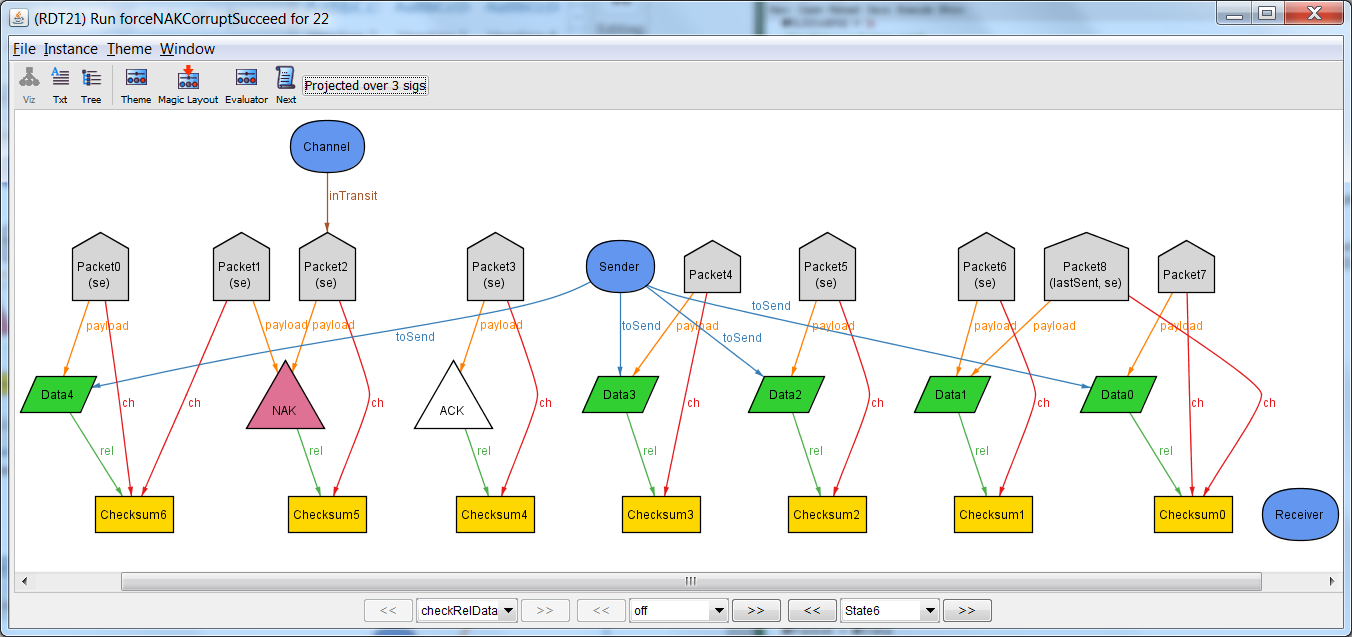
Because the NAK was corrupt, the Sender sends the same packet with the same sequence bit.

## State 5

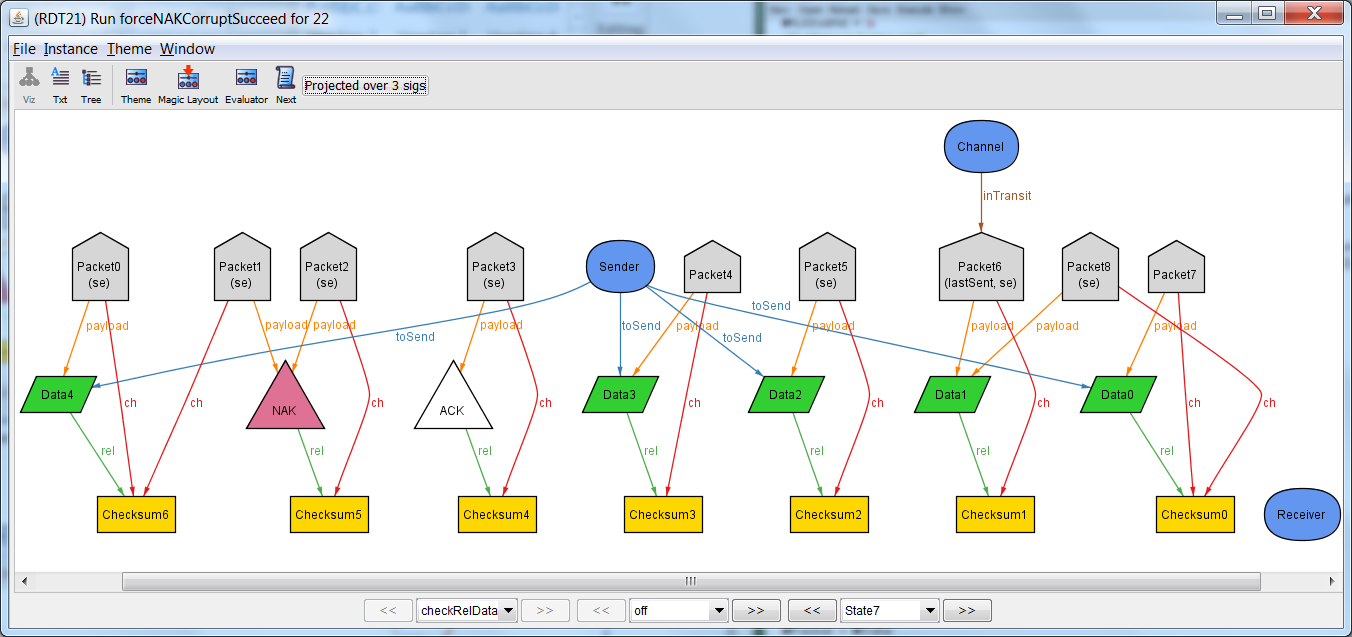


Receiver is attempting to receive Data 1, but the sequence bit is the same and the Packet is still corrupt. (Sequence bit still 0)

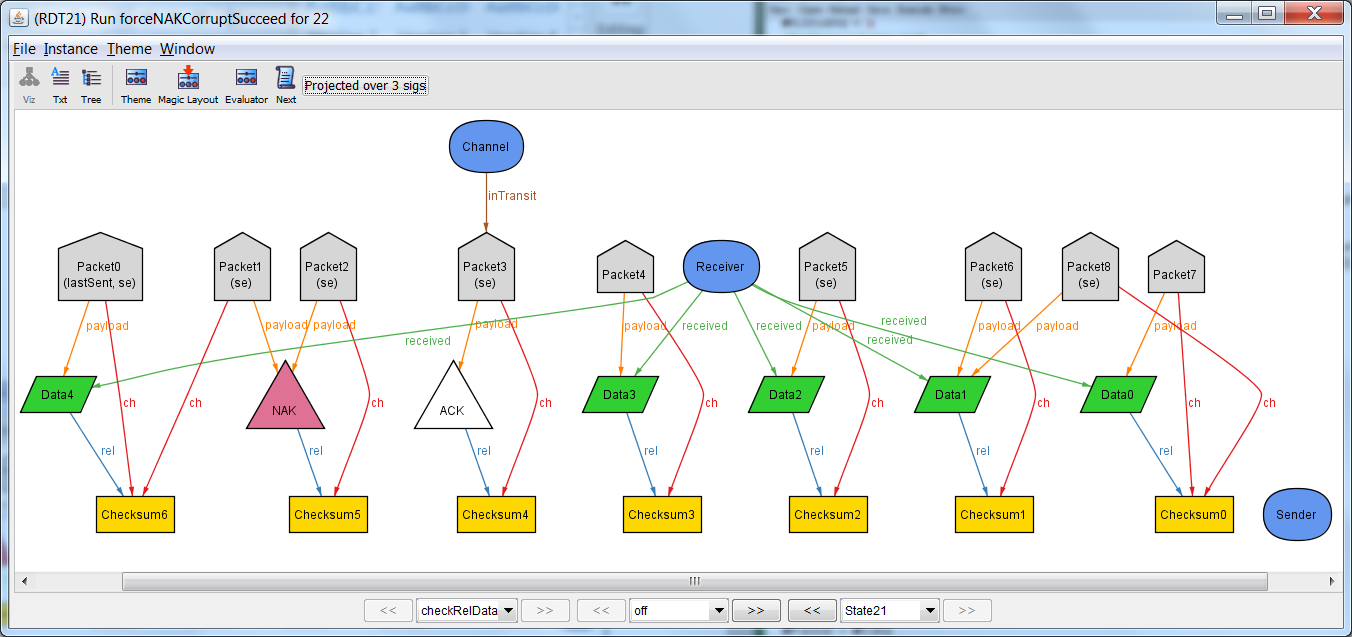
## State 6

It replies back the correct NAK packet this time

## State 7-21



In state 7 through 21, the rest of the packets are sent as normal. (As described in the 0 corrupt previously).



Here’s the final state

# Property 2-Unsuccessful Transfer

Alloy was unable to find a counterexample where all of the Data could not be sent regardless of how much corrupt data there was.

