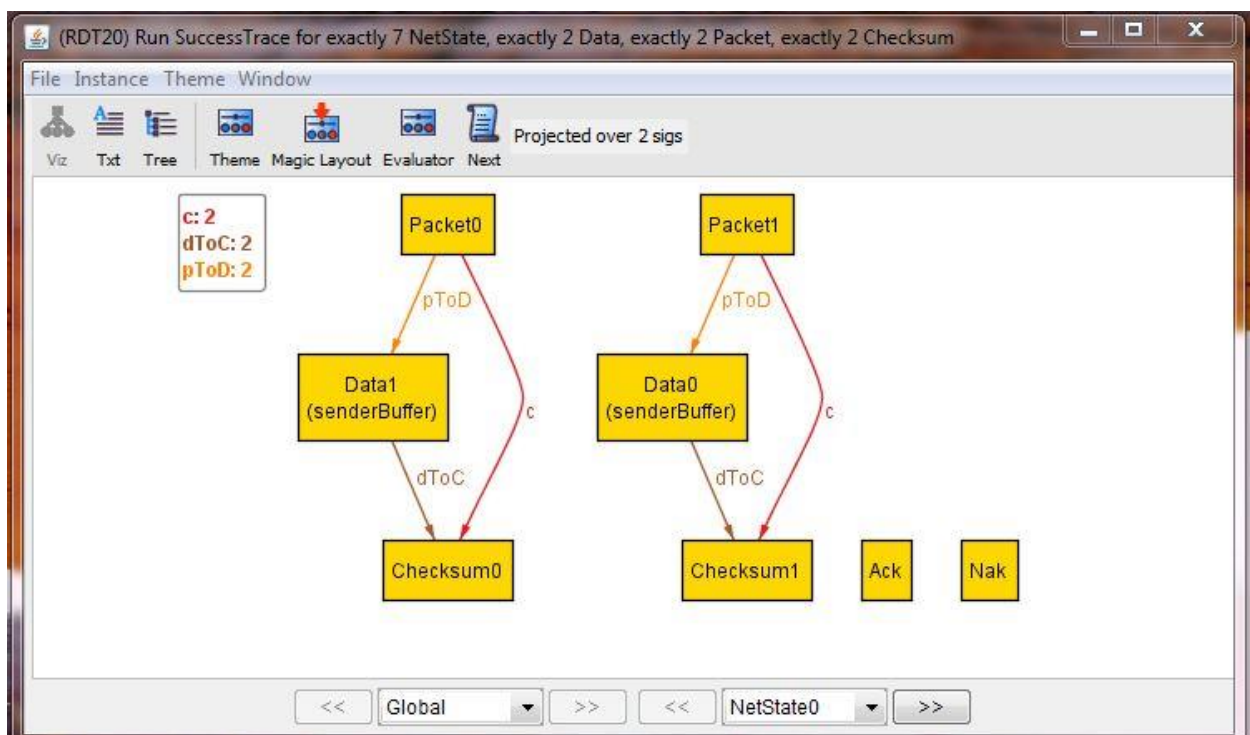


Team: Moore Hazzard
Members: Gordon Hazzard & Jordan Moore

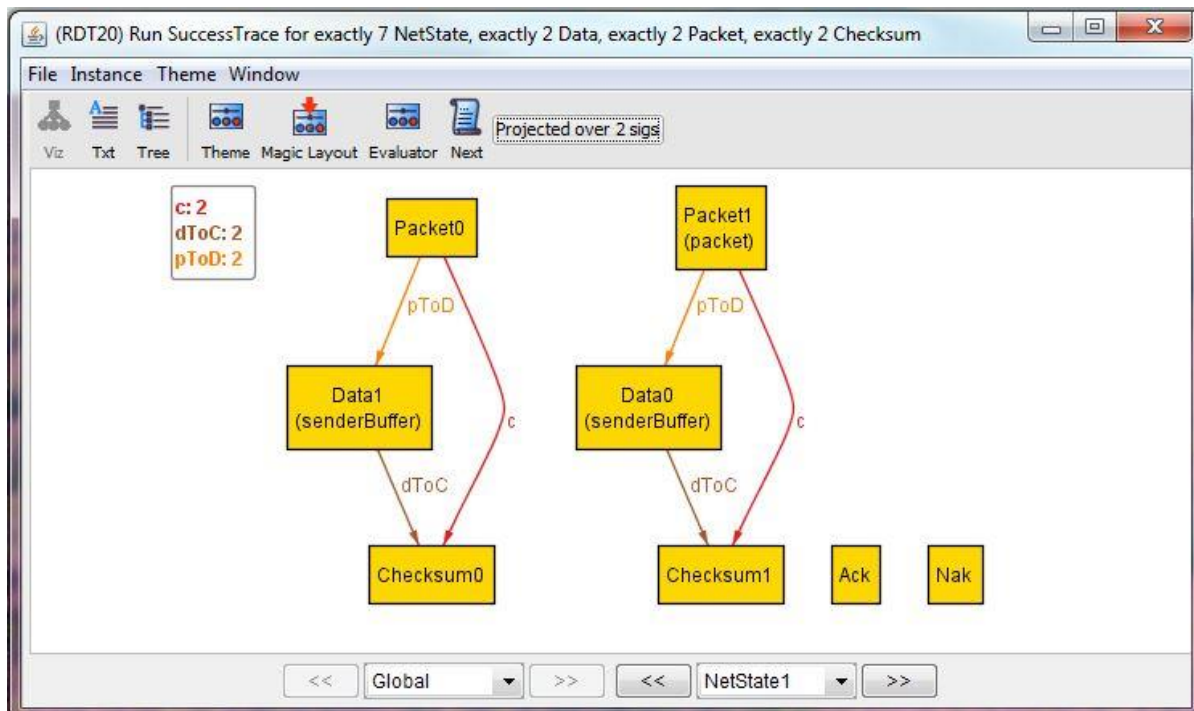
Sprint 2 – Checking RDT 2.0

Property 1: “It is possible to transmit all of the data in the sender’s buffer to the receiver’s buffer.”

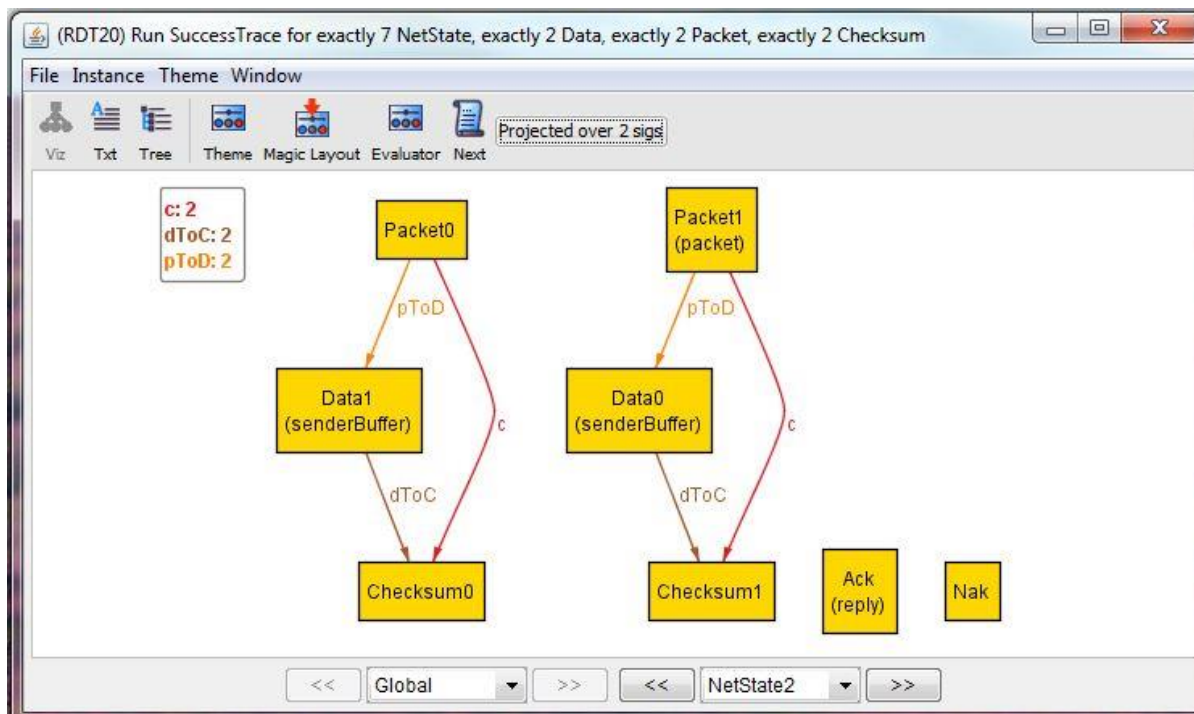
Below is a series of snapshots that confirms the first property. The images depict the transmission of 2 sets of Data, each with a unique Checksum, via 2 Packets.



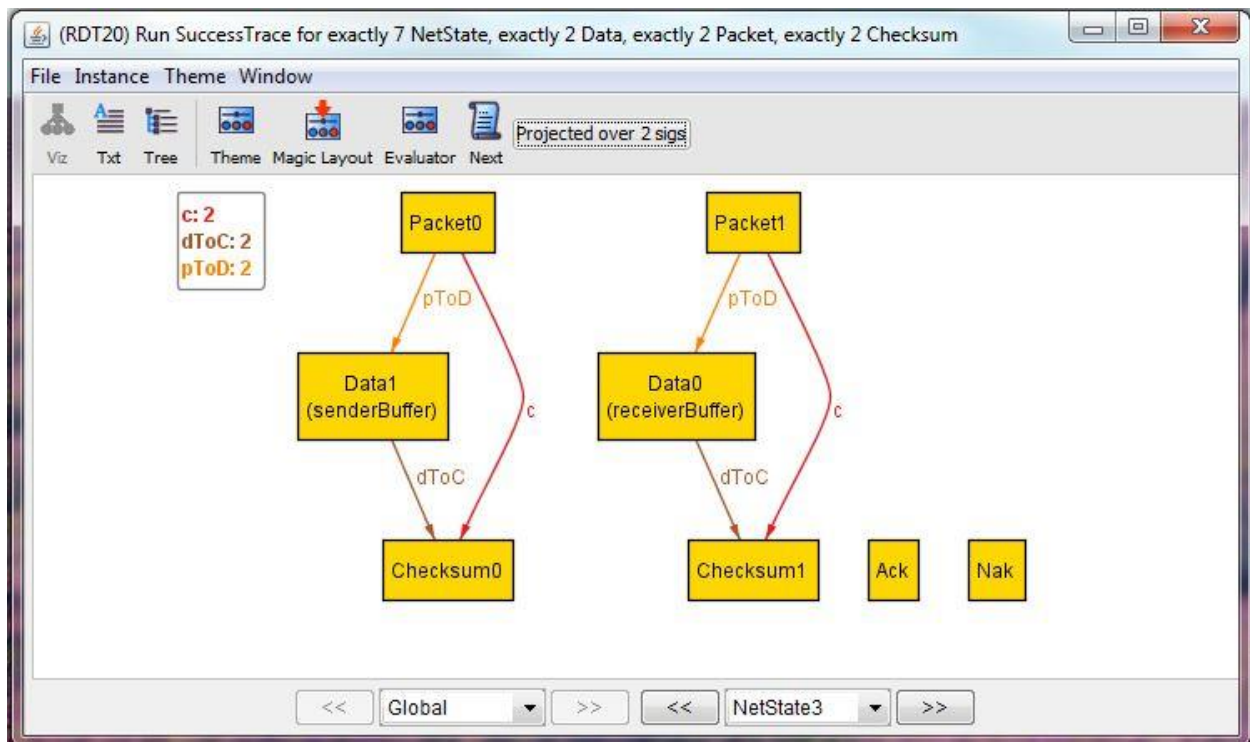
This is the initial state. All Data objects are in the senderBuffer and none are in the receiverBuffer. Additionally none of the Data have been packaged into a Packet yet. Both Data object's Checksum value has not been calculated nor has that Checksum been associated with that Data's packet yet. The future mappings of such relationships are only shown above via the Global signature which has knowledge of all States. Ack and Nak represent acknowledgements, and since nothing has yet been sent or received, Ack and Nak are not associated with this state.



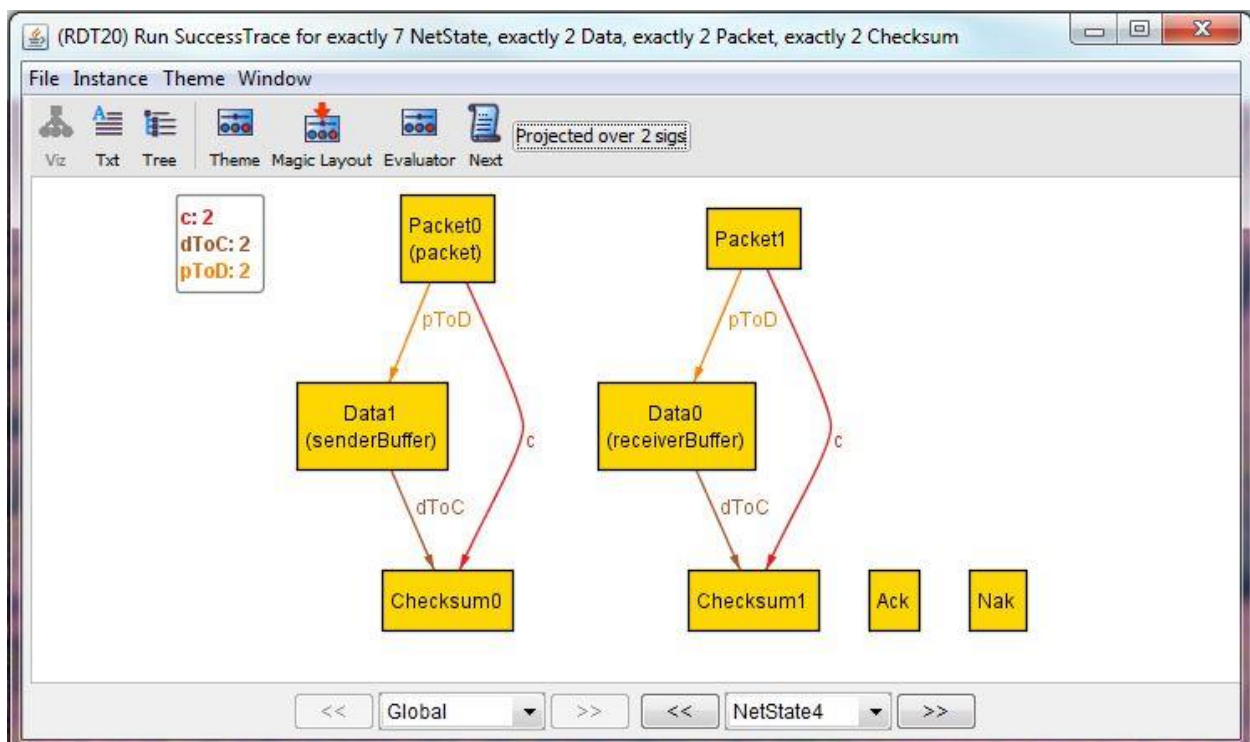
In the next state, Data0 has been made into a Packet, specifically the Packet1 which it is designated to map to in the Global signature. By now, its Checksum value would have been calculated and associated with the packet. This state shows that Data0 is in the process of being transmitted. Data0 will remain in the senderBuffer until an acknowledgement is received.



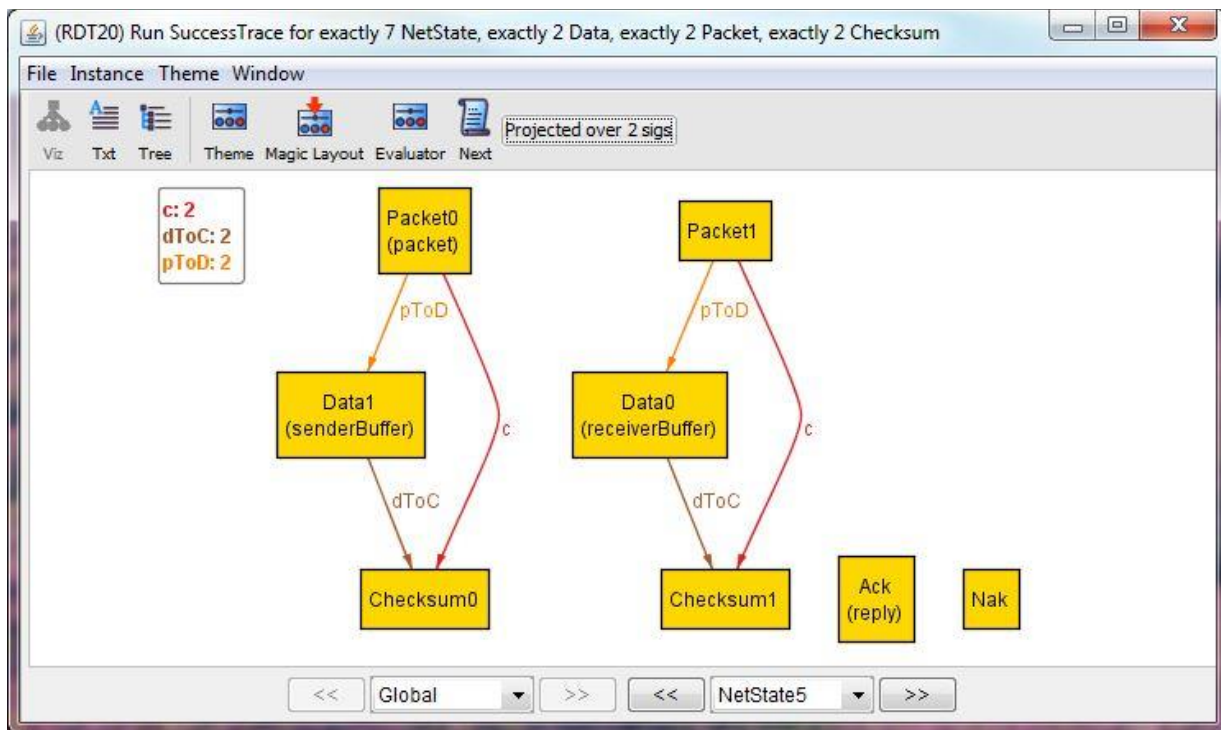
This state shows an acknowledgement through its 'reply' field which is Ack, meaning that Data0 and Packet1 are not corrupt (i.e. Data0 maps to Checksum1 and Packet1's checksum field is also Checksum1).



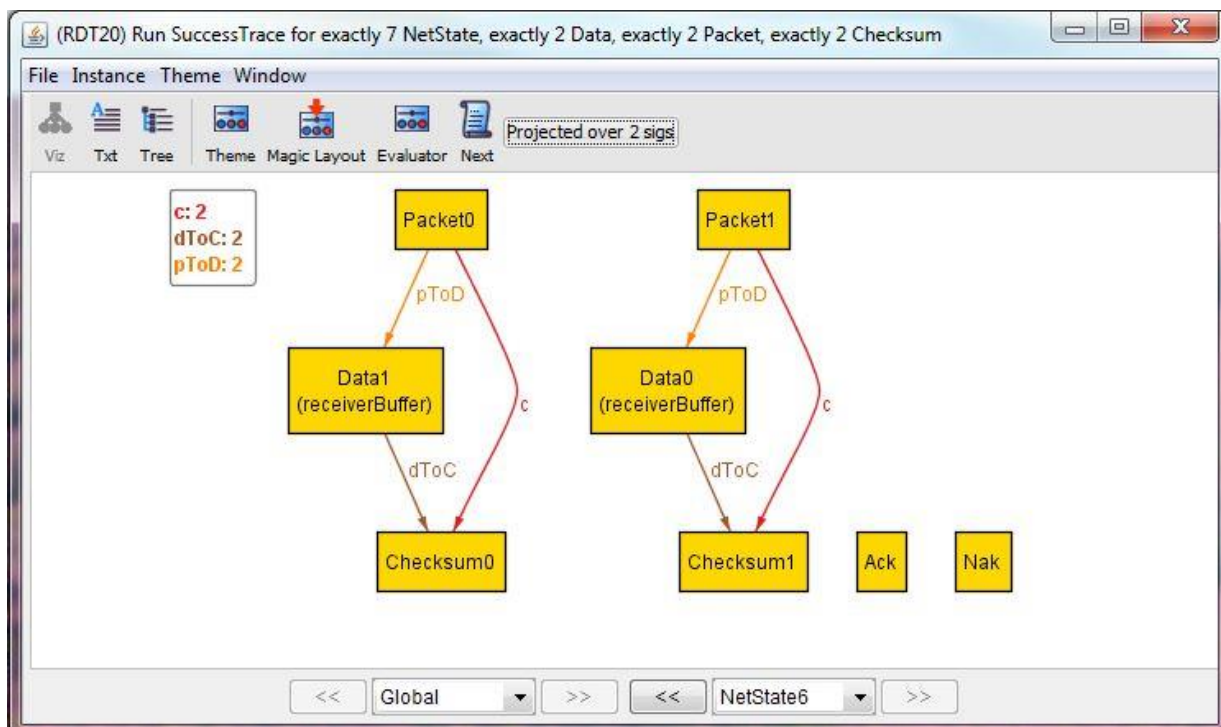
Next up is a receive state. Data0's transmission has just been successfully acknowledged in the previous state, so the receiver removes Data0 from its packet and places it in the receiverBuffer. The sender, having received the Ack acknowledgement message, removes Data0 from the senderBuffer. There are no packets in transmission or acknowledgments in this state.



This state is semantically identical to NetState1 and shows the sending of Data1.



This state is semantically identical to NetState2 and shows the acknowledgement after receiving Data1.



Finally, this state is semantically equivalent to NetState3 and shows the outcome of successfully delivery of Data1. This is also the end state since all Data has successfully be transmitted and is now in the receiverBuffer, no Data remain in the senderBuffer, and no extra packets/acknowledgement messages are floating around in transmission.

Property 2: “It is *always* possible to transmit all of the data in the sender’s buffer to the receiver buffer.”

Below is a series of snapshots of a counterexample that refutes the claim of Property 2. Since RDT2.0 uses an unreliable channel where data can become corrupted, there is a possible situation in which a certain packet becomes corrupted repeatedly and its data is never successfully transmitted. This phenomenon is shown below as Data0 has been garbled and never gets transmitted.

