CSSE373

Milestone 1

Fred Zhang, Songyu Wang

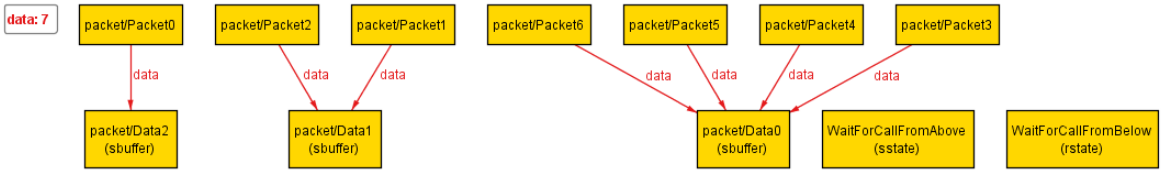
Using the given protocol, it is possible to transmit all of the data in the sender's buffer to the receiver's buffer.

We run:

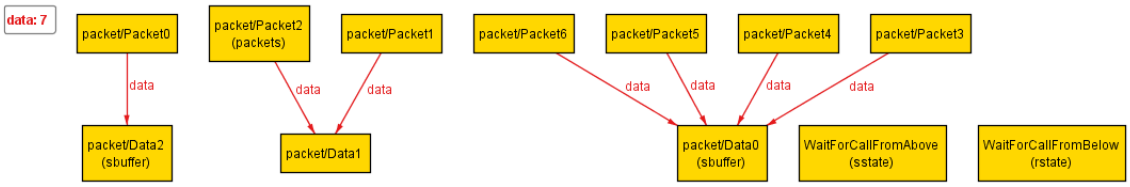
run possibleReliabe for 7 but 3 Data

The result is:

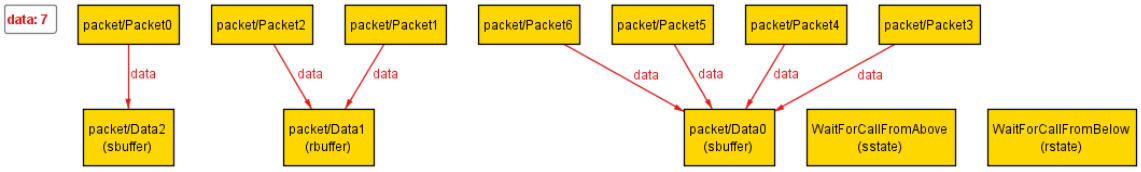
Time 0



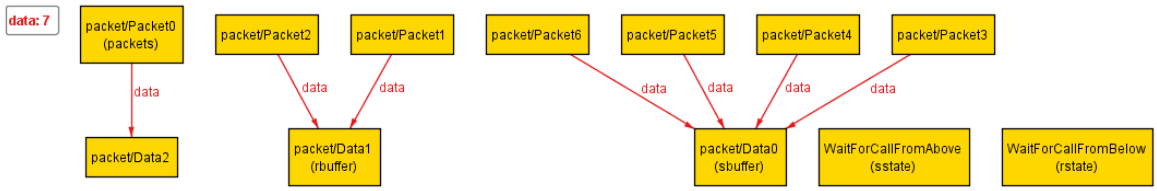
Time 1



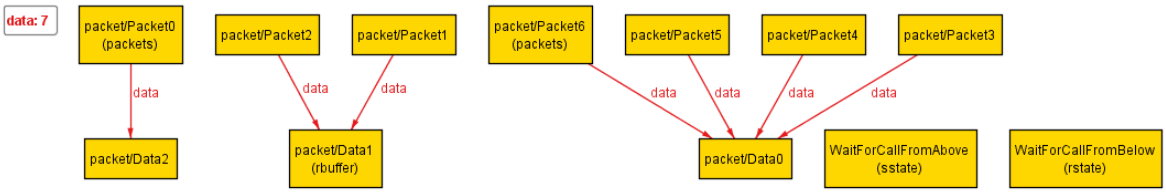
Time 2



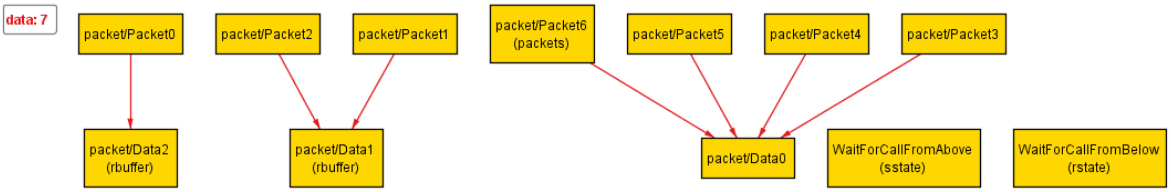
Time 3



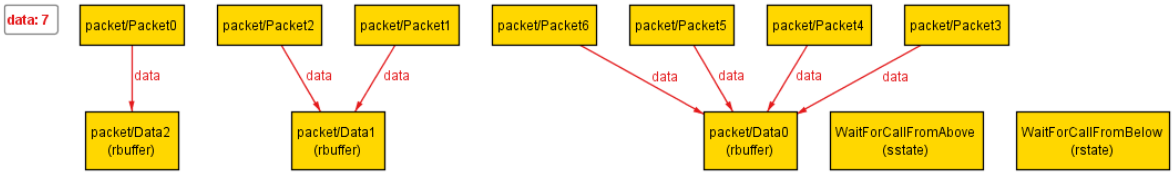
Time 4



Time 5



Time 6



At Time 0, all the data are in sbuffer (Sender buffer).

At Time 1, Data1 is removed from sbuffer and pack as Packet2. Packet2 is in the packets (Channel).

At Time 2, Packet2 is removed from packets and arrive at rbuffer(Receiver buffer).

At Time 3, Data2 is removed from sbuffer and pack as Packet0. Packet0 is in the packets.

At Time 4, Data0 is removed from sbuffer and pack as packet6. Packet6 is in the packets.

At Time 5, Packet0 is removed from packets and arrive at rbuffer.

At Time 6, Packet6 is removed from packets and arrive at rbuffer

At this time, all the data are in rbuffer.

There is one possible way to transmit all the data.

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

We run:

// produce a counter example, because there is not enough time elapsed

1. check alwaysReliable for 5 but exactly 10 Time, 5 Data

// produces no counter example, because all packets eventually arrive

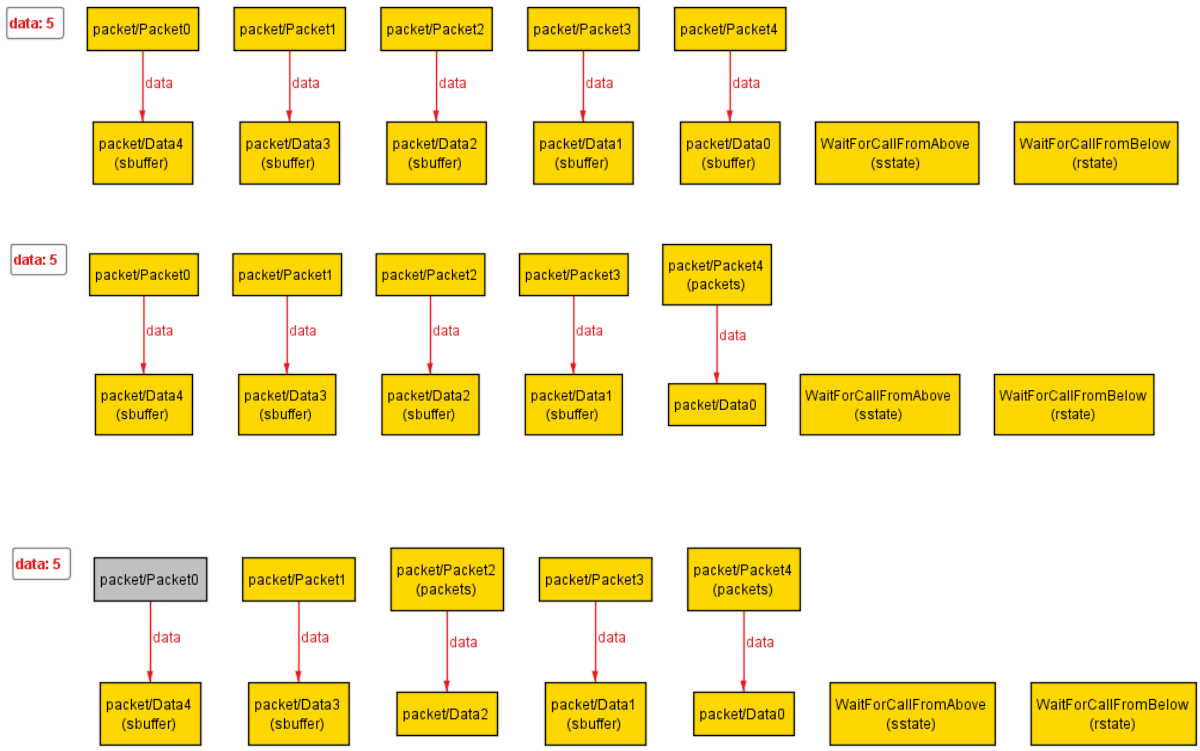
2. check alwaysReliable for 5 but exactly 11 Time, 5 Data

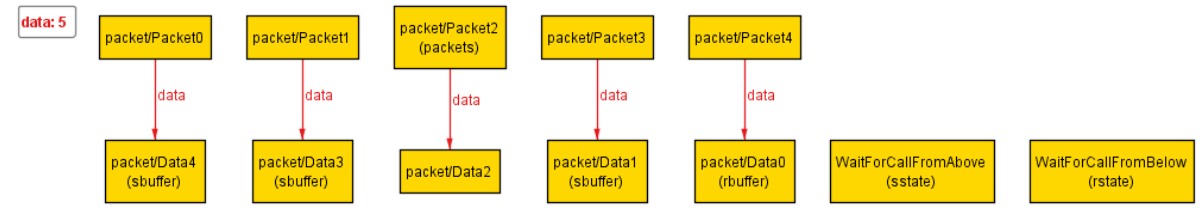
Here is the result:

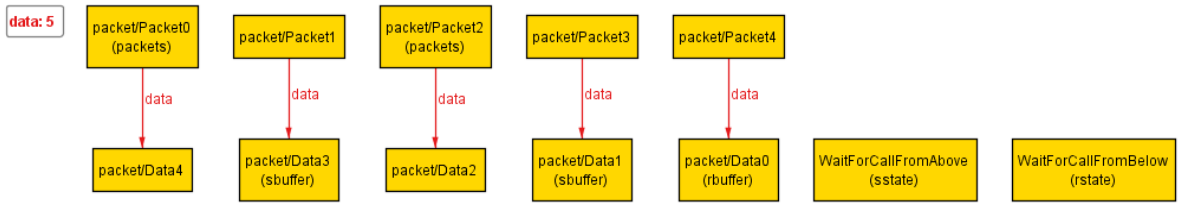
```
#4: Counterexample found. alwaysReliable is invalid.
#5: No counterexample found. alwaysReliable may be valid.
```

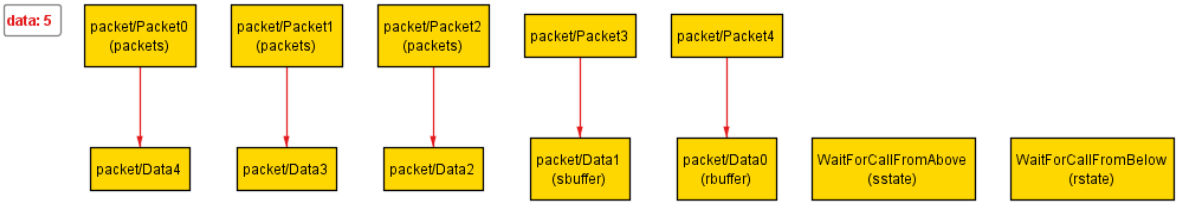
The second check found no counterexample.

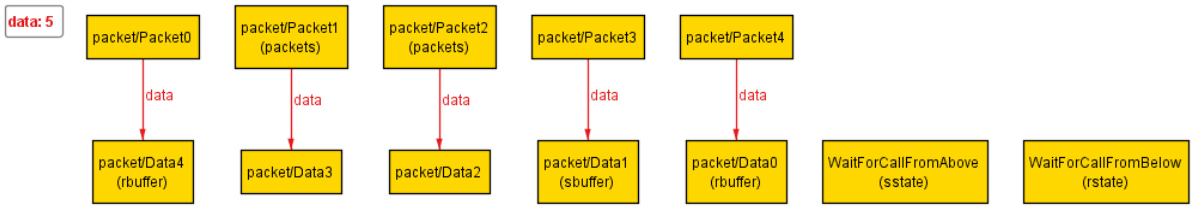
So we decide to exam the first check.

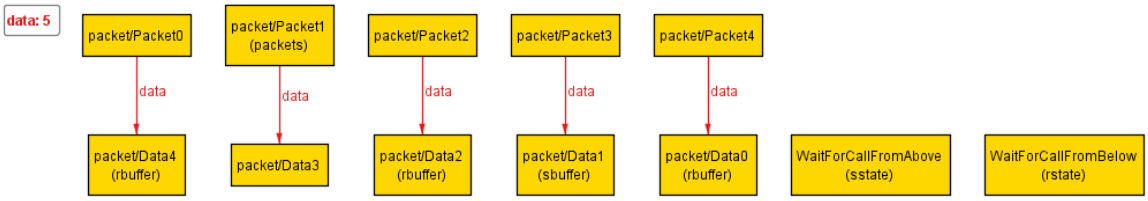


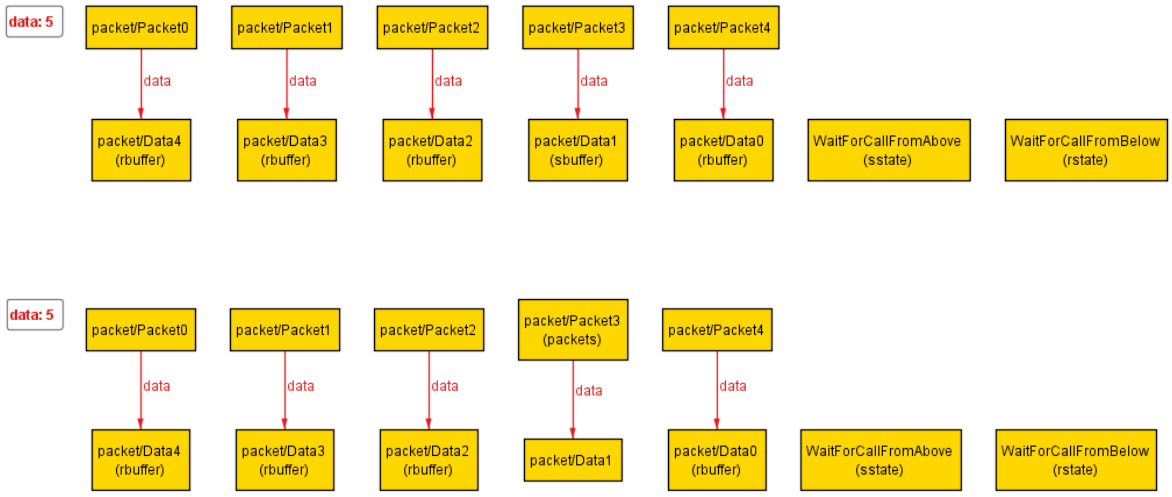

data: 5
packet/Packet0
packet/Packet1
packet/Packet2 (packets)
packet/Packet3
packet/Packet4
data
data
data
data
data
packet/Data4 (sbuffer)
packet/Data3 (sbuffer)
packet/Data2
packet/Data1 (sbuffer)
packet/Data0 (rbuffer)
WaitForCallFromAbove (sstate)
WaitForCallFromBelow (rstate)


data: 5
packet/Packet0 (packets)
packet/Packet1
packet/Packet2 (packets)
packet/Packet3
packet/Packet4
data
data
data
data
data
packet/Data4
packet/Data3 (sbuffer)
packet/Data2
packet/Data1 (sbuffer)
packet/Data0 (rbuffer)
WaitForCallFromAbove (sstate)
WaitForCallFromBelow (rstate)


data: 5
packet/Packet0 (packets)
packet/Packet1 (packets)
packet/Packet2 (packets)
packet/Packet3
packet/Packet4
packet/Data4
packet/Data3
packet/Data2
packet/Data1 (sbuffer)
packet/Data0 (rbuffer)
WaitForCallFromAbove (sstate)
WaitForCallFromBelow (rstate)


data: 5
packet/Packet0
packet/Packet1 (packets)
packet/Packet2 (packets)
packet/Packet3
packet/Packet4
data
data
data
data
data
packet/Data4 (rbuffer)
packet/Data3
packet/Data2
packet/Data1 (sbuffer)
packet/Data0 (rbuffer)
WaitForCallFromAbove (sstate)
WaitForCallFromBelow (rstate)


data: 5
packet/Packet0
packet/Packet1 (packets)
packet/Packet2
packet/Packet3
packet/Packet4
data
data
data
data
data
packet/Data4 (rbuffer)
packet/Data3
packet/Data2 (rbuffer)
packet/Data1 (sbuffer)
packet/Data0 (rbuffer)
WaitForCallFromAbove (sstate)
WaitForCallFromBelow (rstate)



After seeing the diagram, we found that it is doing the correct behavior just like what we described in previous section. However, since we only allow 10 Times, the transmitting process does not have enough time to finish. That is why we have this counterexample. We concluded that this is not a flaw in the protocol. It is a problem of how Alloy model the protocol.

So, there is no real counterexample that disprove the property. This property holds.