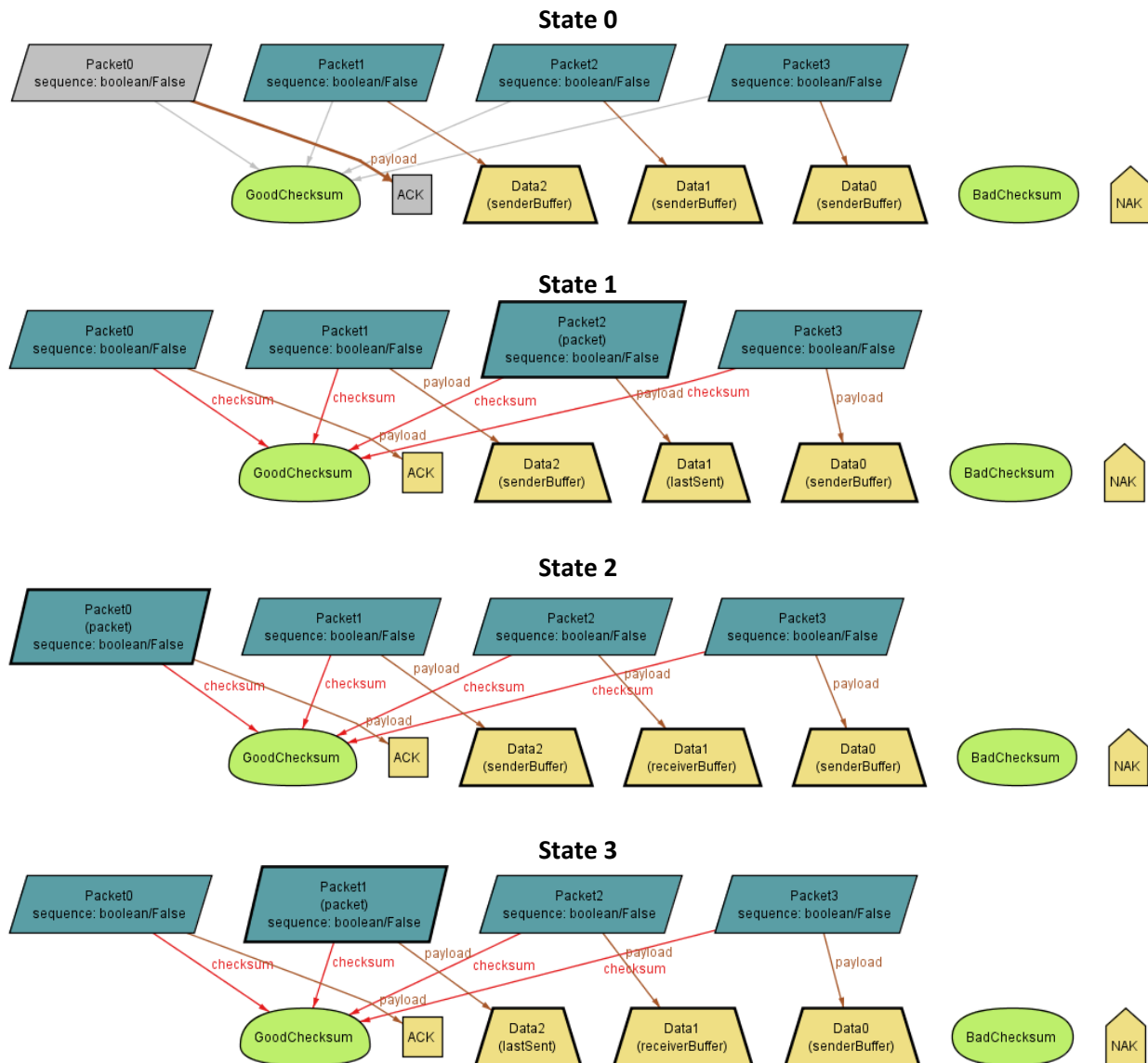
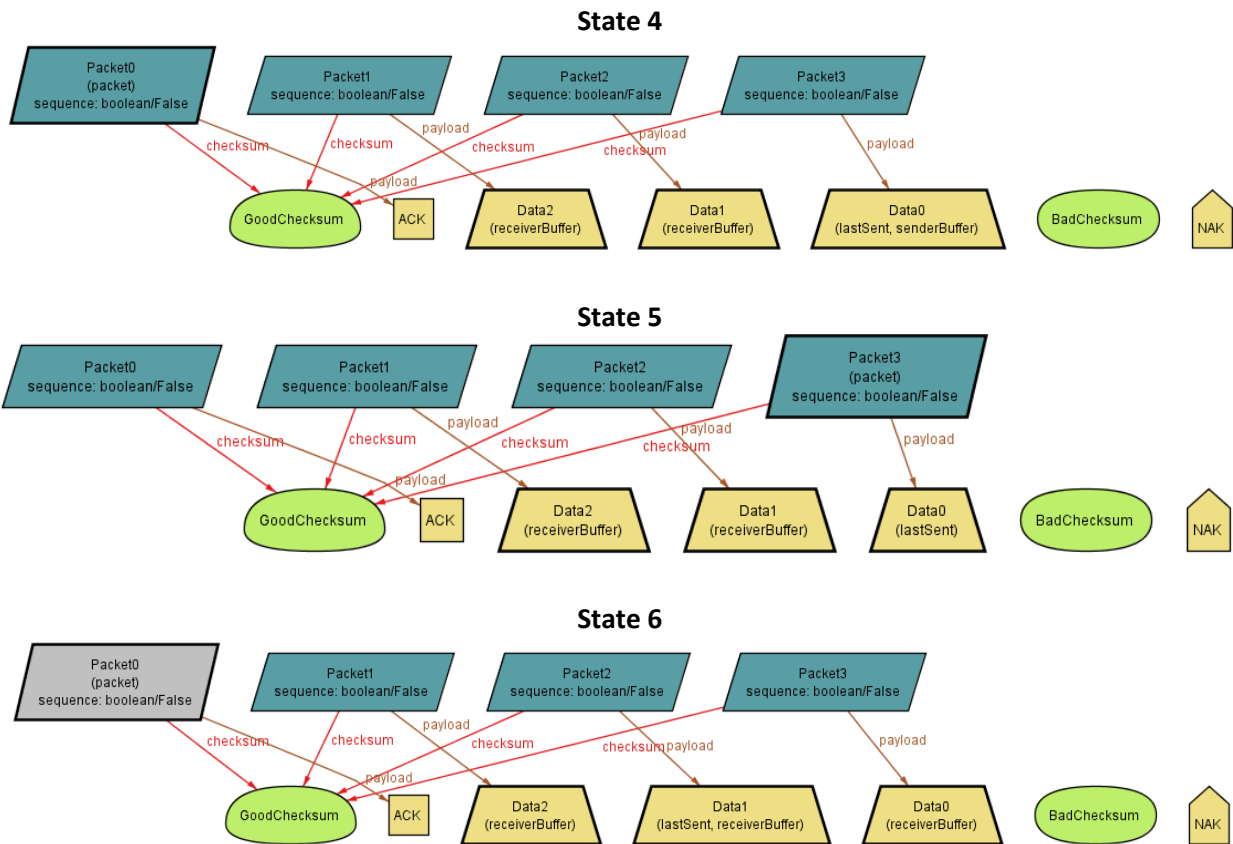


TEAM POTATO

Milestone 3

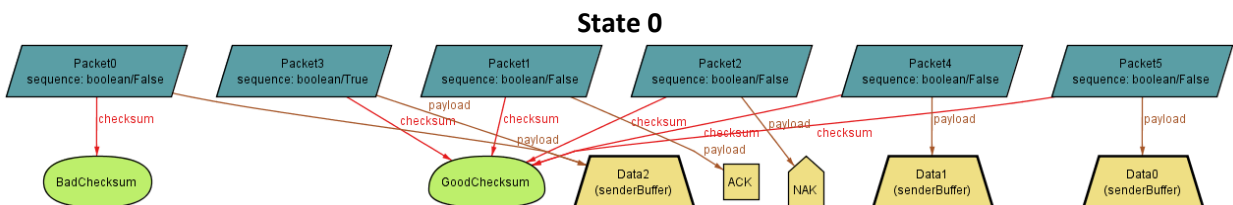
In this milestone, we were asked to model RDT2.1, which specifies that there is a sequence number attached to each data transfer (excluding ACK and NAK). If the data is retransferred, we are guaranteed that the transfer will be a success (again excluding ACK and NAK). Following is the trace where there are no failures. Then we will show a trace with one failure, but ultimate success.





State	Send Buffer	Receiver Buffer	Packet in Channel
0	0, 1, 2	-	-
1	0, 2	-	P2 -> Data1, GoodCs, Seq = false
2	0, 2	1	P0 -> ACK, GoodCs, Seq = false
3	0	1	P1 -> Data2, GoodCs, Seq = false
4	0	1, 2	P0 -> ACK, GoodCs, Seq = false
5	-	1, 2	P3 -> Data0, GoodCs, Seq = false
6	-	0,1,2	P0 -> ACK, GoodCs, Seq = false

Now we move on to the trace where the data is successfully transferred but has one failure to transfer involved.



[illegible]

```

graph TD
    P0[/Packet0  
sequence: boolean/False/]
    P3[/Packet3 (packet)  
sequence: boolean/True/]
    P1[/Packet1  
sequence: boolean/False/]
    P2[/Packet2  
sequence: boolean/False/]
    P4[/Packet4  
sequence: boolean/False/]
    P5[/Packet5  
sequence: boolean/False/]

    P0 -- checksum --> BC([BadChecksum])
    P3 -- checksum --> BC
    P1 -- checksum --> GC([GoodChecksum])
    P2 -- checksum --> GC
    P4 -- checksum --> GC
    P5 -- checksum --> GC

    P1 -- payload --> D2[/Data2 (lastSent)/]
    P2 -- payload --> D2
    P4 -- payload --> D2
    P5 -- payload --> D2

    P2 -- payload --> ACK[/ACK/]
    P4 -- payload --> ACK
    P5 -- payload --> ACK

    P2 -- payload --> NAK[/NAK/]
    P4 -- payload --> NAK
    P5 -- payload --> NAK

    P4 -- payload --> D1[/Data1 (senderBuffer)/]
    P5 -- payload --> D1

    P4 -- payload --> D0[/Data0 (senderBuffer)/]
    P5 -- payload --> D0
  
```

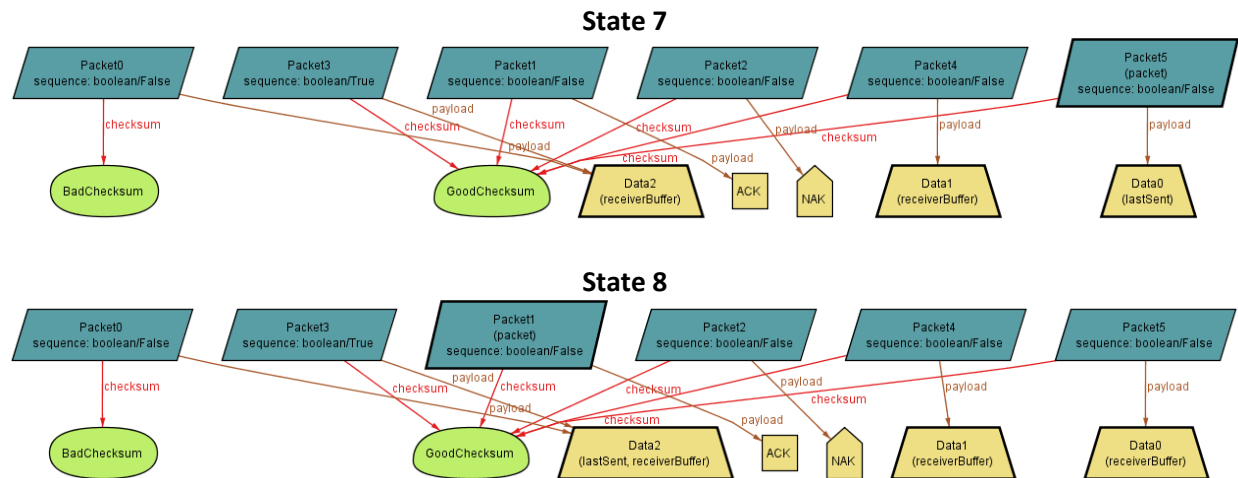
The diagram illustrates the receiver's buffer management in a Stop-and-Wait protocol. It shows six packets (Packet0 to Packet5) with their sequence numbers and checksum status. Packet0 (sequence: boolean/False) leads to 'BadChecksum'. Packet1 (sequence: boolean/False) leads to 'GoodChecksum'. Packet2 (sequence: boolean/False) leads to 'Data2 (receiverBuffer)'. Packet3 (sequence: boolean/True) leads to 'Data2 (receiverBuffer)'. Packet4 (sequence: boolean/False) leads to 'Data1 (lastSent, senderBuffer)'. Packet5 (sequence: boolean/False) leads to 'Data0 (senderBuffer)'. Arrows indicate the flow of checksums and payloads from packets to the respective data structures.

The diagram illustrates the receiver's buffer management in a Stop-and-Wait protocol. It shows a sequence of packets (Packet0 to Packet5) and their corresponding checksums and payloads. The receiver buffer (Data2) is full, and the sender buffer (Data0) is empty. The receiver buffer (Data1) is the last sent packet. The receiver buffer (Data0) is the sender buffer.

- Packet0**: sequence: boolean/False. checksum: BadChecksum.
- Packet1**: sequence: boolean/False. checksum: GoodChecksum.
- Packet2**: sequence: boolean/False. checksum: ACK.
- Packet3**: sequence: boolean/True. checksum: NAK.
- Packet4**: (packet) sequence: boolean/False. checksum: Data1 (lastSent).
- Packet5**: sequence: boolean/False. checksum: Data0 (senderBuffer).

The receiver buffer (Data2) is full, and the sender buffer (Data0) is empty. The receiver buffer (Data1) is the last sent packet. The receiver buffer (Data0) is the sender buffer.

The diagram illustrates the receiver's perspective of a Stop-and-Wait protocol. It shows six packets (Packet0 to Packet5) with their sequence numbers and acknowledgment status. Packet0 (sequence: boolean/False) has a checksum failure, leading to 'BadChecksum'. Packet1 (sequence: boolean/False) has a checksum success, leading to 'GoodChecksum'. Packet2 (sequence: boolean/False) has a checksum failure, leading to 'ACK'. Packet3 (sequence: boolean/True) has a checksum success, leading to 'Data2 (receiverBuffer)'. Packet4 (sequence: boolean/False) has a checksum failure, leading to 'Data1 (lastSent, receiverBuffer)'. Packet5 (sequence: boolean/False) has a checksum success, leading to 'Data0 (senderBuffer)'.



State	Send Buffer	Receiver Buffer	Packet in Channel
0	0, 1, 2	-	-
1	0, 1	-	P0 -> Data2, BadCs, Seq = false
2	0, 1	-	P2 -> NAK, GoodCs, Seq = false
3	0, 1	-	P3 -> Data2, GoodCs, Seq = true
4	0, 1	2	P1 -> ACK, GoodCs, Seq = false
5	0	2	P4 -> Data1, GoodCs, Seq = false
6	0	2, 1	P1 -> ACK, GoodCs, Seq = false
7	-	2, 1	P5 -> Data0, GoodCs, Seq = false
8	-	2, 1, 0	P1 -> ACK, GoodCs, Seq = false

With this model, if given enough states, the data may not always be transferred, because it possible that we will receive corrupted ACK's indefinitely.