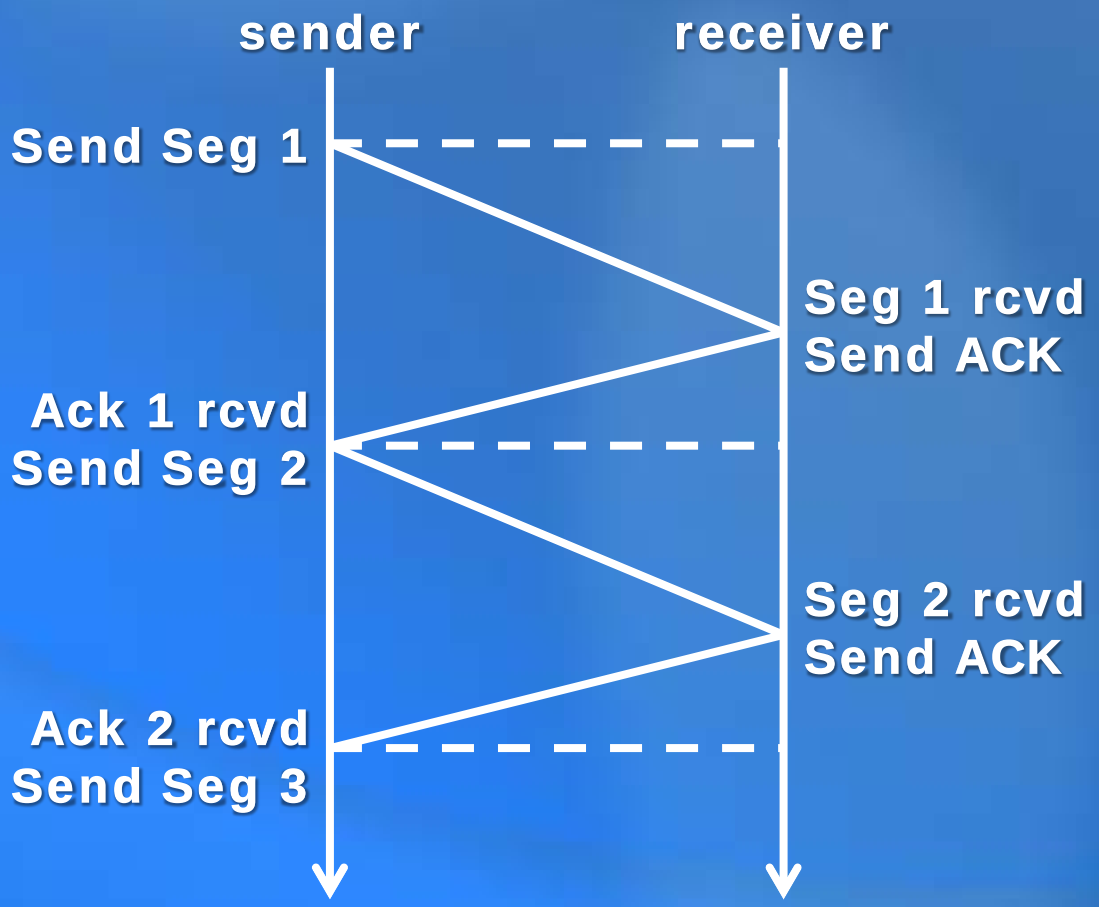
Stop-and-wait Protocol

Stop-and-wait (no losses)



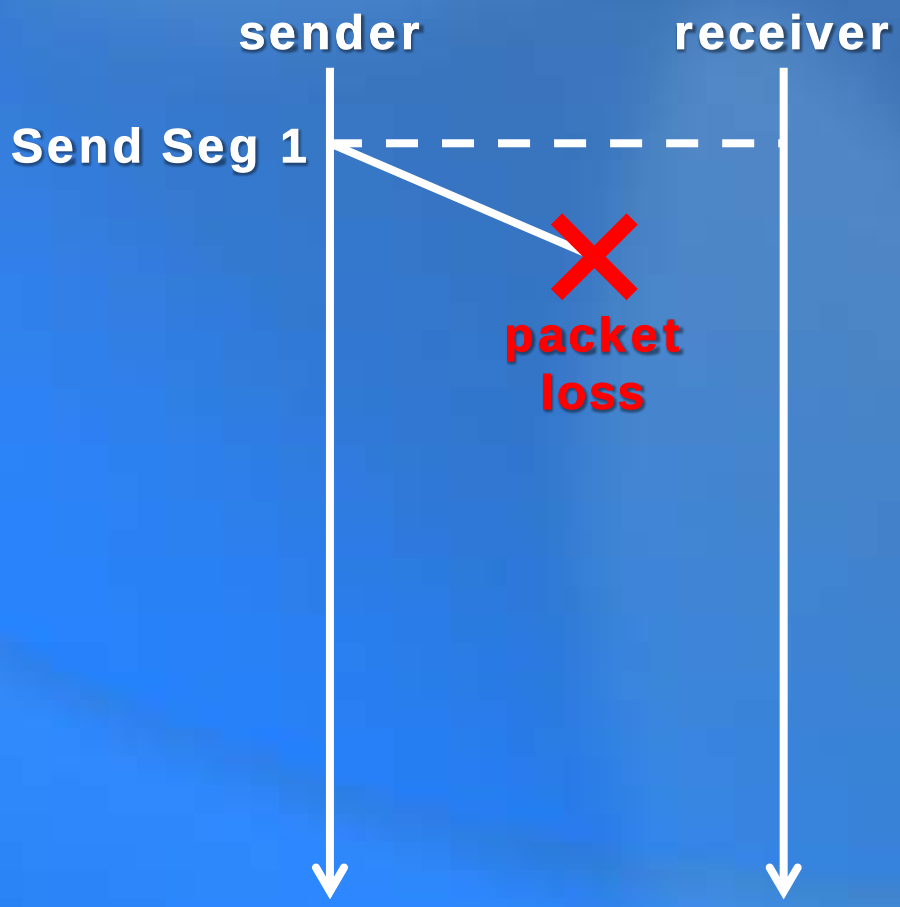
Sender

1. Send segment n
2. Wait for ACK for segment n
3. n = n+1, go to 1

Receiver

1. Wait for segment
2. When receive, send ACK with segment number
3. Deliver packet to application
4. Go to 1

Effect of packet loss



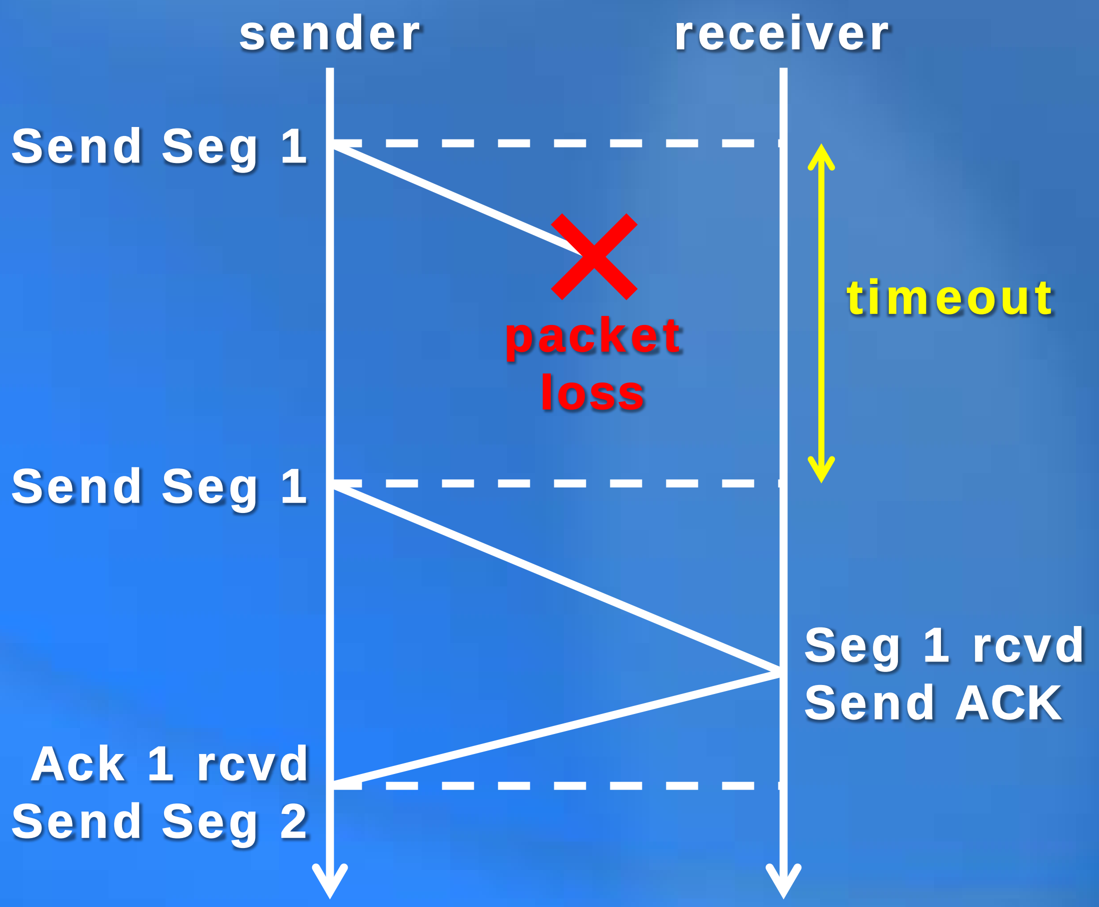
Sender

1. Send segment n
2. Wait for ACK for segment n
3. n = n+1, go to 1

Receiver

1. Wait for segment
2. When receive, send ACK with segment number
3. Deliver packet to application
4. Go to 1

Revised protocol



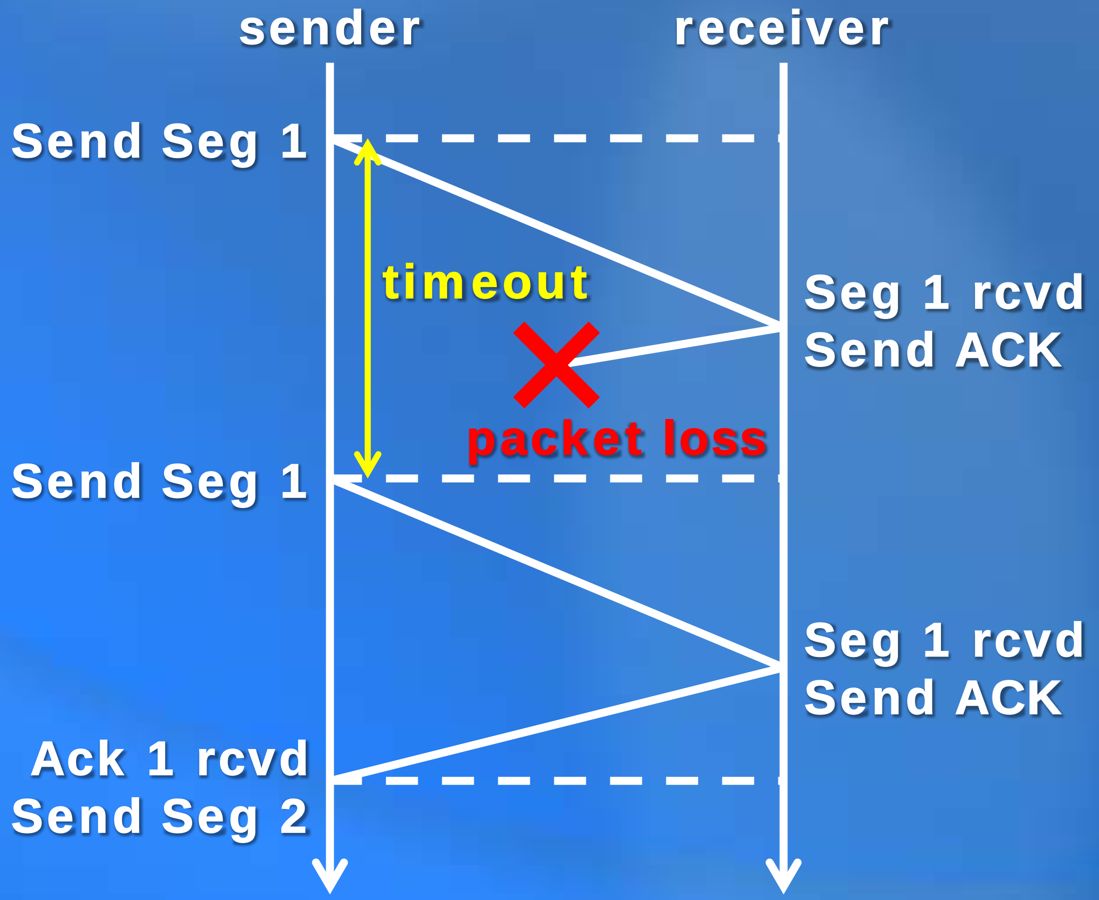
Sender

1. Send segment n
2. Wait for ACK for segment n
   1. If no ACK received before timeout, leave n unchanged
   2. If ACK received, set n = n+1
3. Go to 1

Receiver

1. Wait for segment
2. When receive, send ACK for segment number
3. Deliver packet to application
4. Go to 1

Effect of ACK loss



Sender

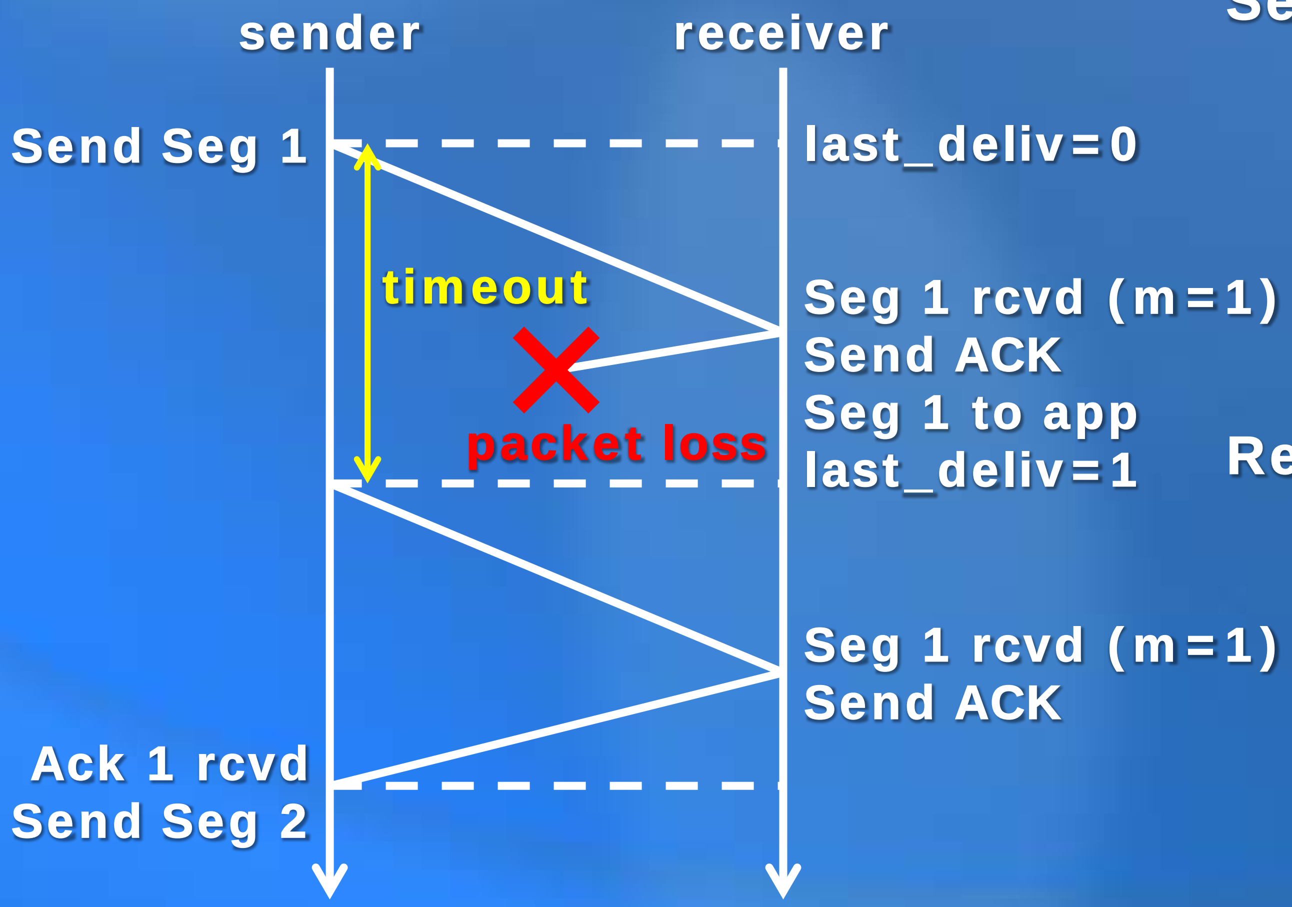
1. Send segment n
2. Wait for ACK for segment n
    1. If no ACK received before timeout, leave n unchanged
    2. If ACK received, set n = n+1
3. Go to 1

Receiver

1. Wait for segment
2. When receive, send ACK for segment number
3. Deliver packet to application
4. Go to 1

wanted "exactly once", got "at least once"

Revised Protocol



Sender

1. Send segment n
2. Wait for ACK for segment n
    1. If no ACK received before timeout, leave n unchanged
    2. If ACK received, set n = n+1
3. Go to 1

Receiver

1. Wait for segment
2. When receive, check segment number (m)
    1. send ACK m
    2. If m = last_deliv + 1, deliver segment to application and set last_deliv = m
    3. Otherwise, discard segment
3. Go to 1