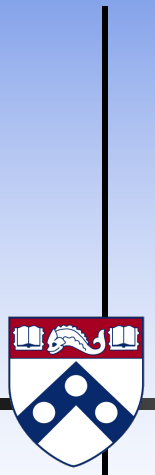


CIS 553: Networked Systems

Review

March 2, 2020

Announcements

Problem set: Today at 10pm

- Solutions will be released after the deadline

HW3: Friday at 10pm

- Project topics are still fair game

Midterm 1

- Date: Wednesday, March 4 in class
- Don't forget to bring a cheat sheet

Agenda

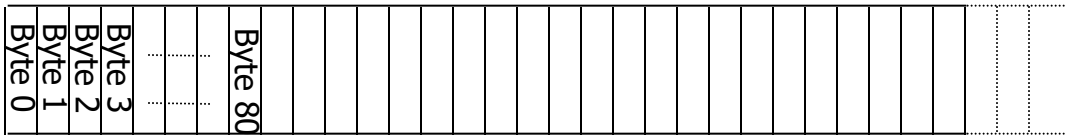
- Interdomain Routing ✓
  - BGP ✓
  - Issues with BGP ✓
- Transport Layer ✓
  - UDP ✓
  - TCP NEXT

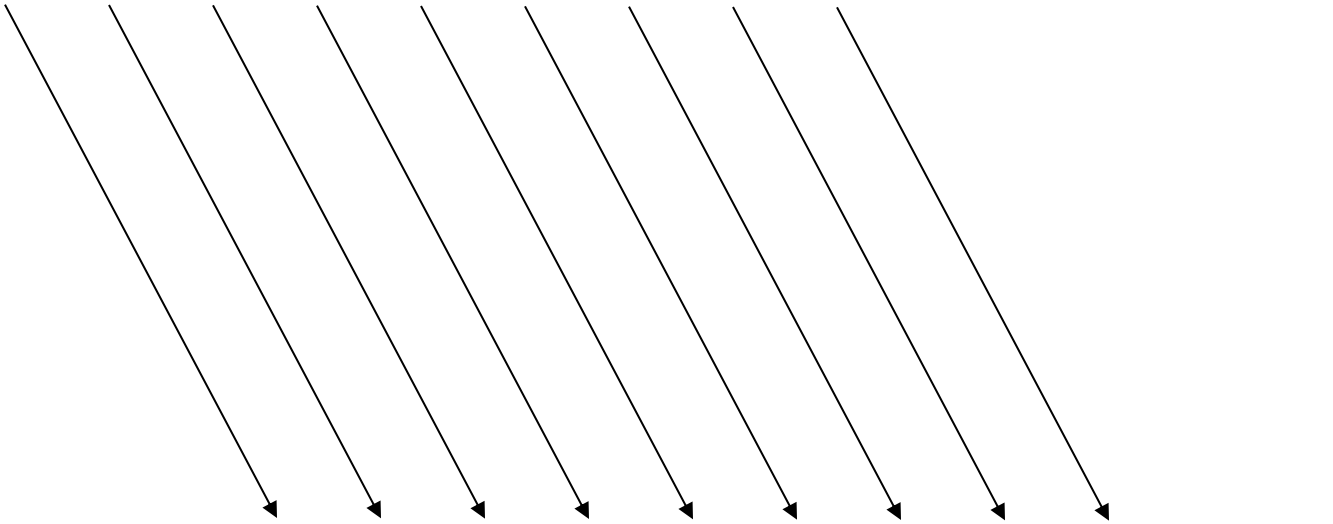
Transmission Control Protocol (TCP)

1. Stream-of-bytes service
   - Sends and receives a stream of bytes
2. Reliable, in-order delivery
   - Detect corruption, loss, and reordering
   - Reliable delivery: acknowledgments and retransmissions
3. Connection-oriented
   - Explicit set-up and tear-down of TCP connection
4. Flow control
   - Prevent overflow of the receiver's buffer space
5. Congestion control
   - Adapt to network congestion for the greater good

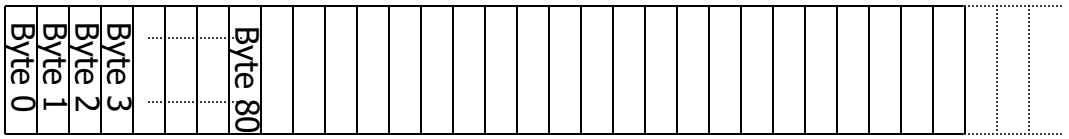
TCP's "stream of bytes" model

Host A



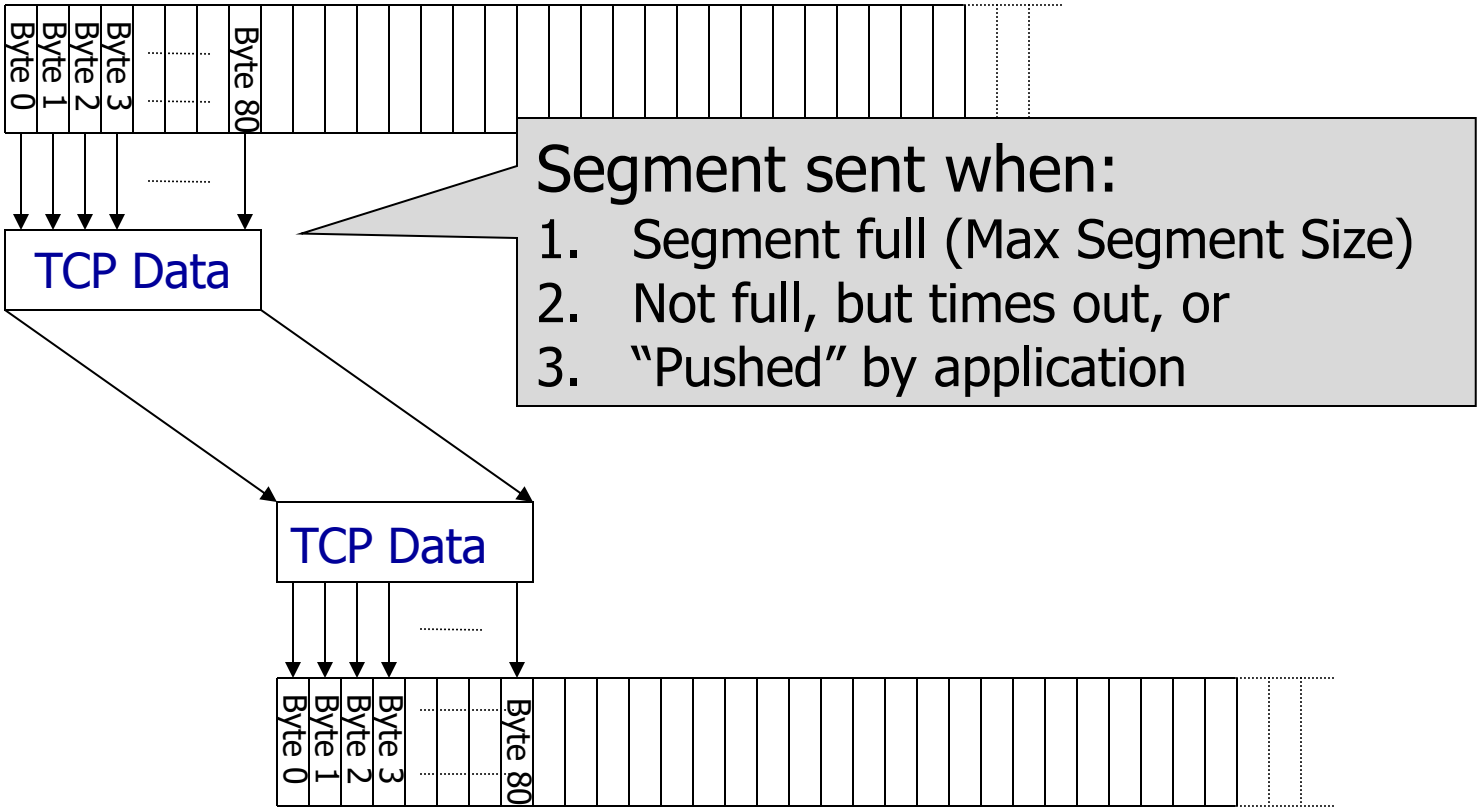


Host B



…Emulated Using TCP "Segments"

Host A



Host B

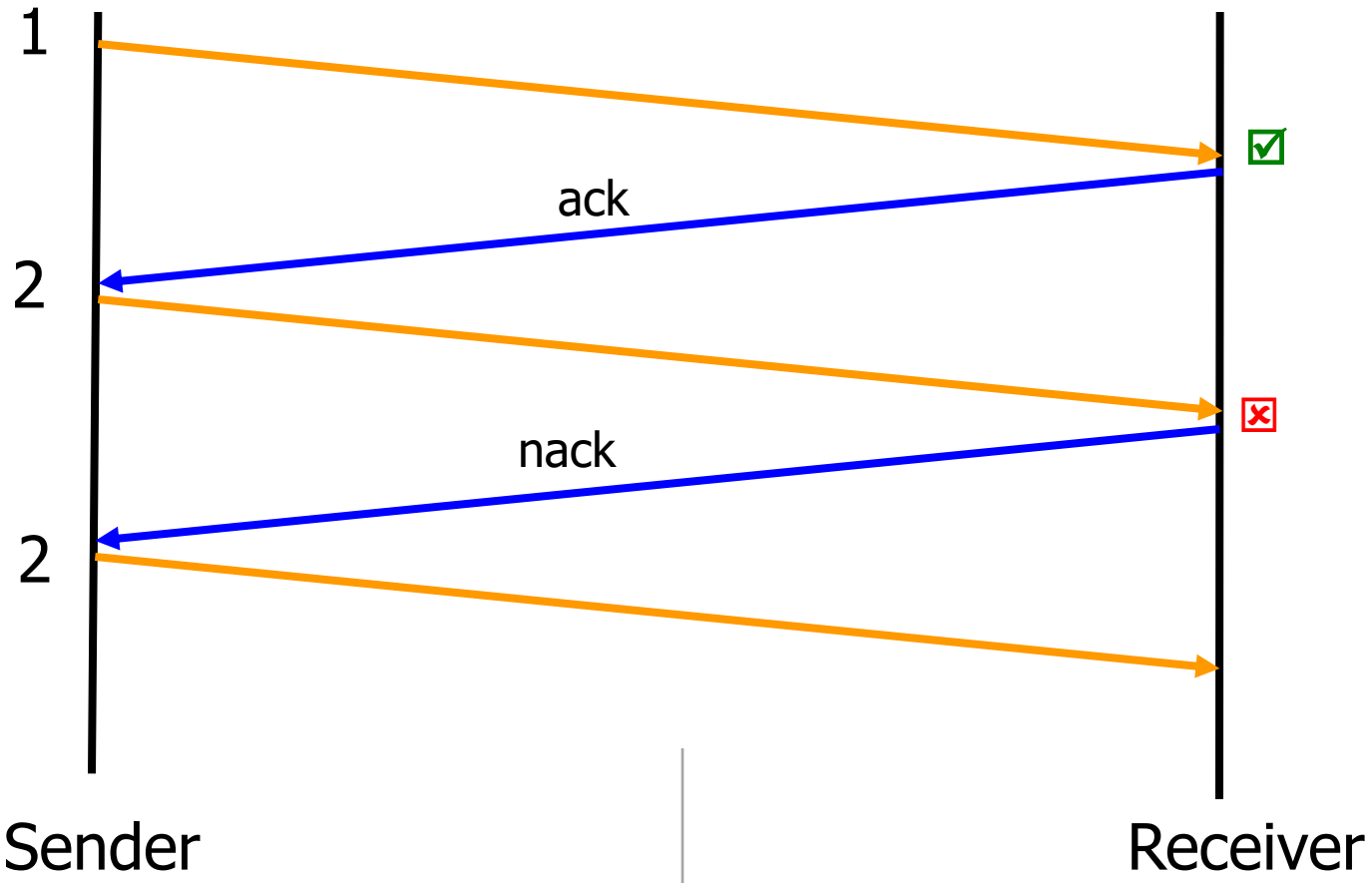
Transmission Control Protocol (TCP)

1. Stream-of-bytes service
   - Sends and receives a stream of bytes
2. Reliable, in-order delivery
   - Detect corruption, loss, and reordering
   - Reliable delivery: acknowledgments and retransmissions
3. Connection-oriented
   - Explicit set-up and tear-down of TCP connection
4. Flow control
   - Prevent overflow of the receiver's buffer space
5. Congestion control
   - Adapt to network congestion for the greater good

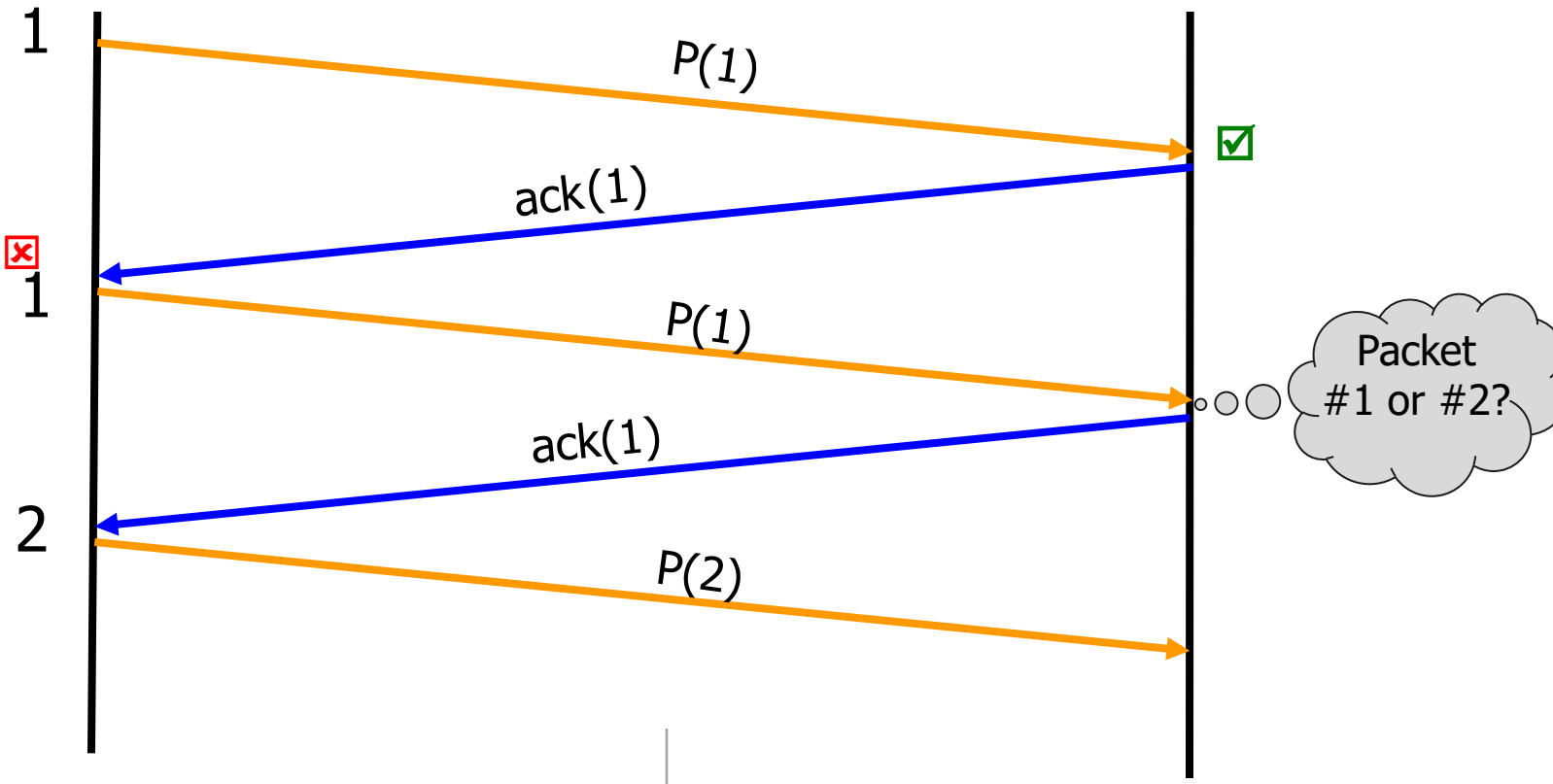
Challenges of Reliable Data Transfer

- Over a perfectly reliable channel? Done!
- What if packets can experience bit errors?
- What if packets can be lost?
- What if packets can be reordered?

Dealing with packet corruption



Dealing with packet corruption

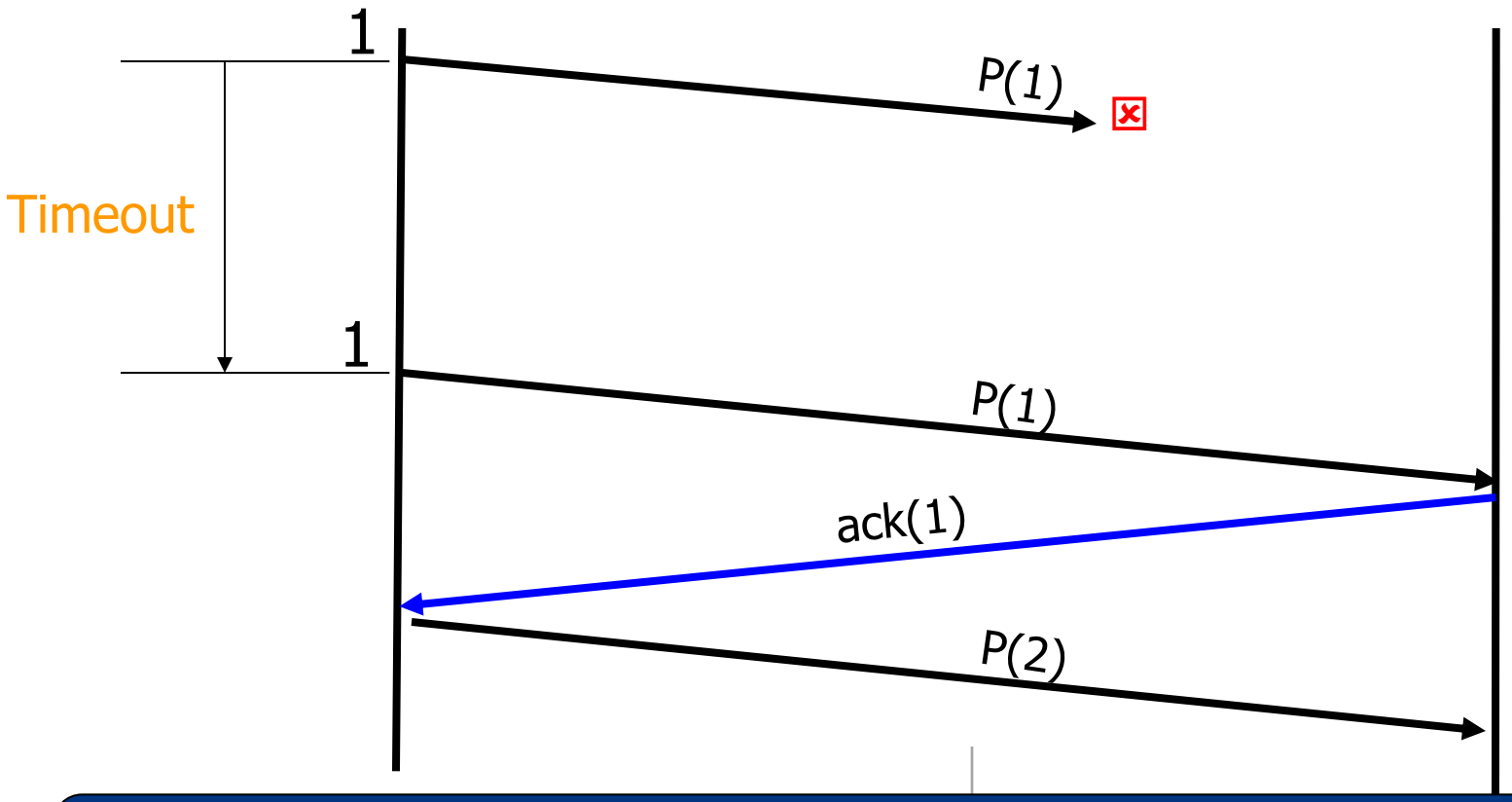


What if the ACK/NACK is corrupted?

Components of a solution

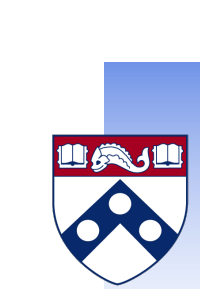
- Checksums (to detect bit errors)
- Acknowledgements (plus retransmissions)
- Sequence numbers (to deal with duplicates)

Dealing with packet loss

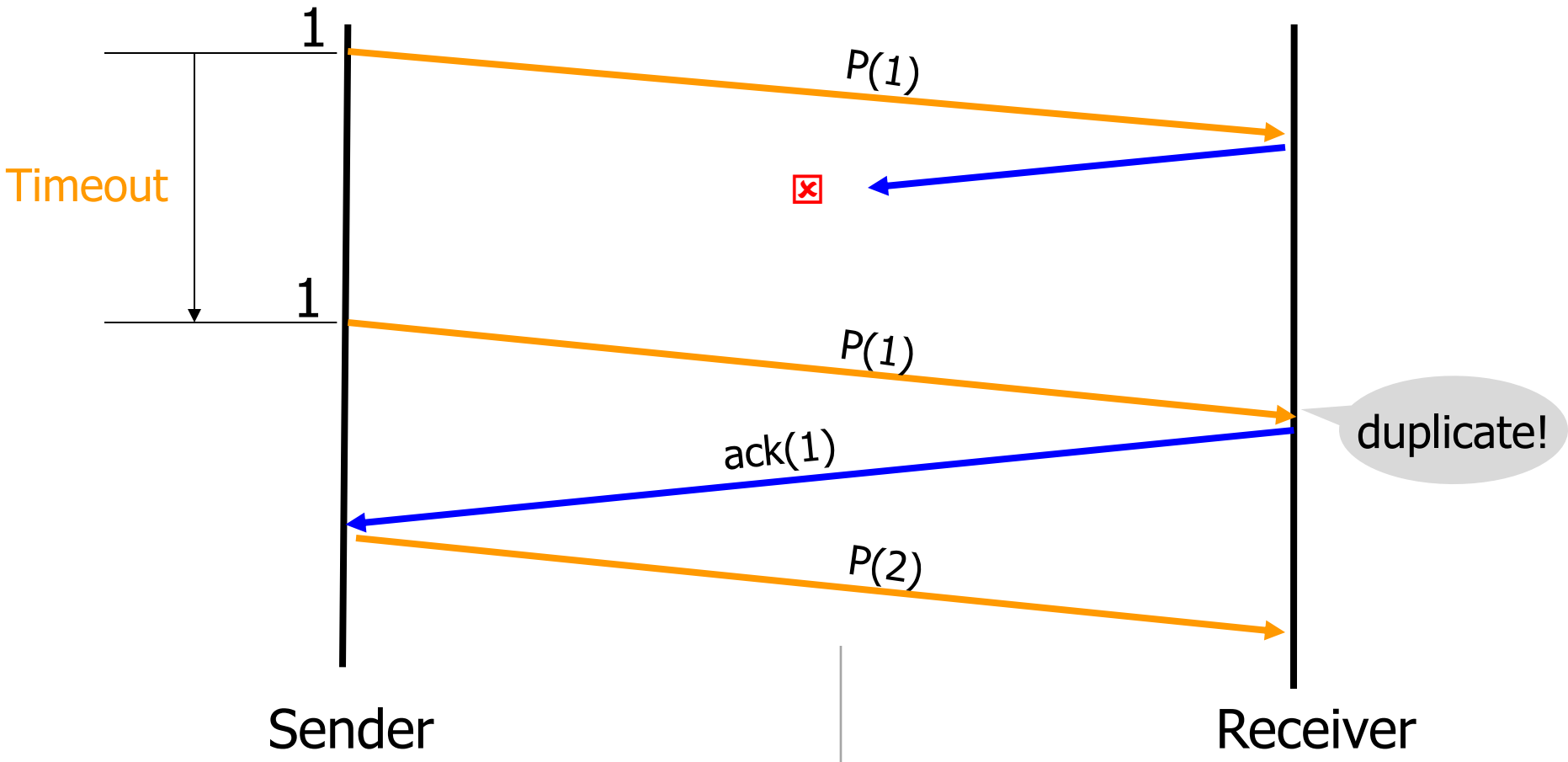


Timer-driven loss detection

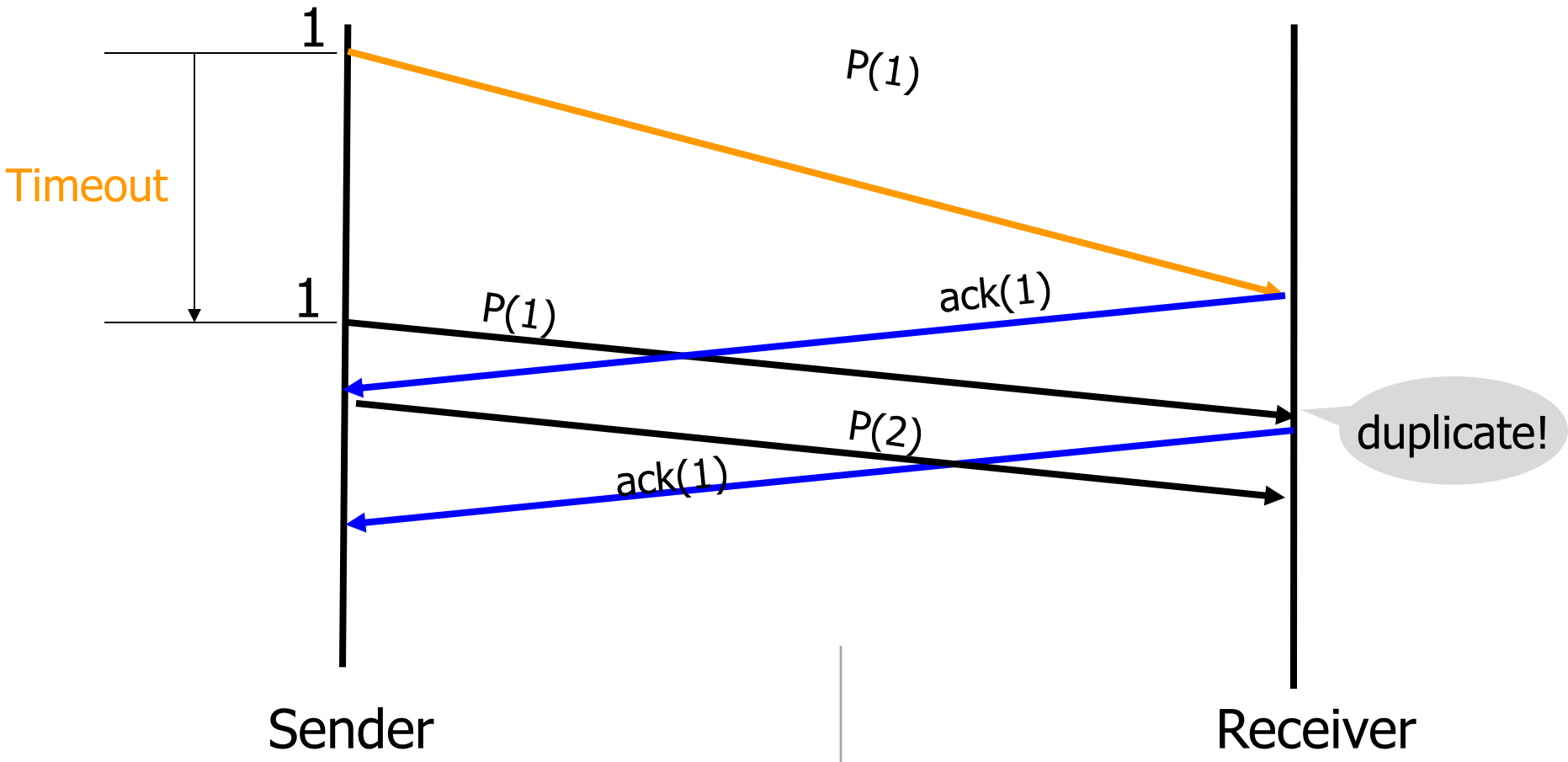
Set timer when packet is sent; retransmit on timeout



Dealing with packet loss (of ack)



Dealing with packet loss



Components of a solution

- Checksums (to detect bit errors)
- Acknowledgements (plus retransmissions)
- Sequence numbers (to deal with duplicates)
- Timers (to detect loss)

A Solution: "Stop and Wait"

@Sender

- Send packet(I); (re)set timer; wait for ack
- If (ACK)
  - I++; repeat
- If (NACK or TIMEOUT)
  - repeat

@Receiver

- Wait for packet
- If packet is OK, send ACK
- Else, send NACK
- Repeat