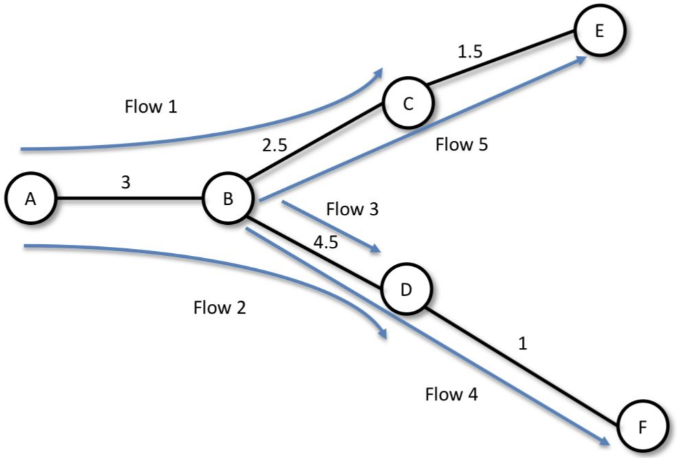
Example

Consider the network on the right consisting of 5 nodes (A to E). Each link has a maximal bandwidth indicated in red. 7 flows (1 to 7) are using the network at the same time. You can assume that they have to send a lot of traffic and will use whatever bandwidth they will get. Apply the max-min fair allocation algorithm to find a fair bandwidth allocation for each flow.



For each flow, what is the bottleneck link?

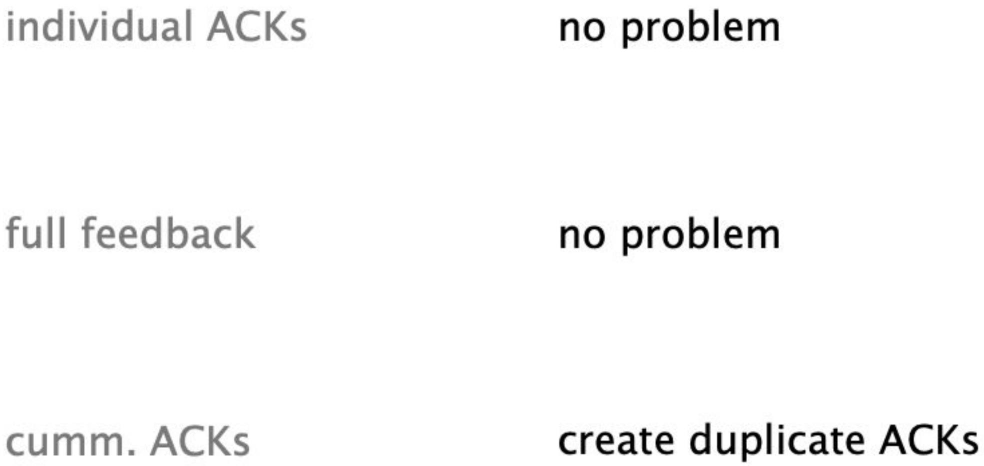
Design a *correct*, *timely*, *efficient* and *fair* transport mechanism

knowing that

packets can get lost
corrupted
reordered
delayed
duplicated

Dealing with corruption is easy:
Rely on a checksum, treat corrupted packets as lost

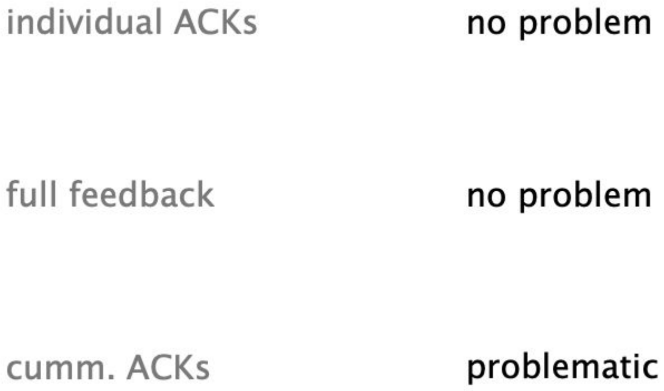
The effect of reordering depends on the type of ACKing mechanism used



individual ACKs	no problem
full feedback	no problem
cumm. ACKs	create duplicate ACKs

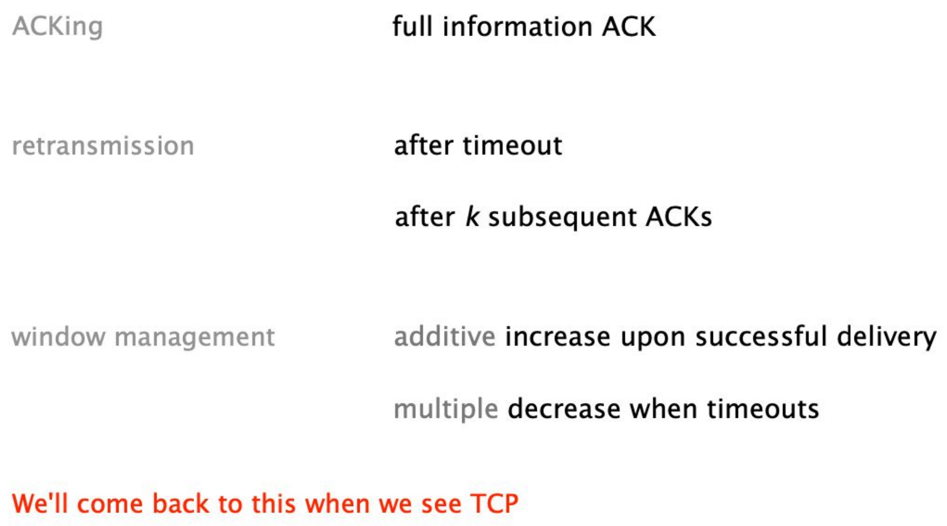
Long delays can create useless timeouts, for all designs

Packet duplicates can lead to duplicate ACKs whose effects will depend on the ACKing mechanism used



individual ACKs	no problem
full feedback	no problem
cumm. ACKs	problematic

Here is one correct, timely, efficient and fair transport mechanism

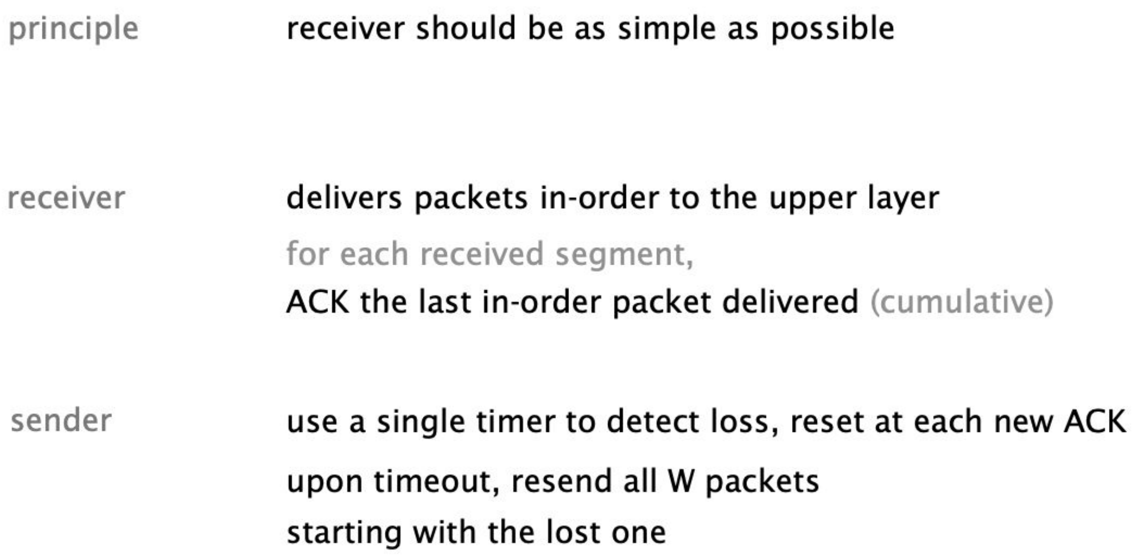


ACKing	full information ACK
retransmission	after timeout
	after *k* subsequent ACKs
window management	additive increase upon successful delivery
	multiple decrease when timeouts

We'll come back to this when we see TCP

Reliable Transport Examples

Go-Back-N (GBN) is a simple sliding window protocol using cumulative ACKs



principle

receiver should be as simple as possible

receiver

delivers packets in-order to the upper layer

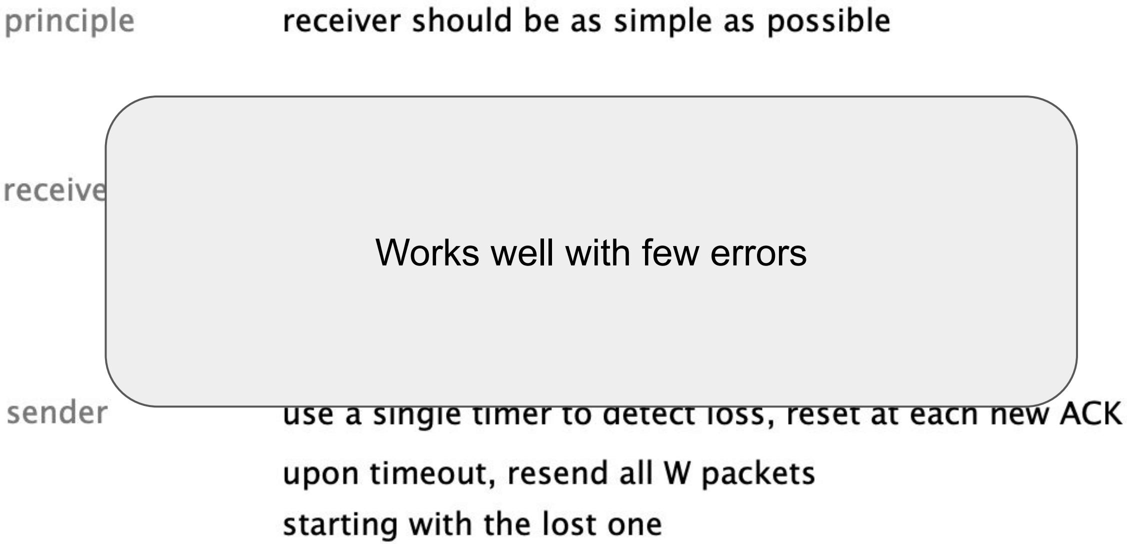
for each received segment,
ACK the last in-order packet delivered (cumulative)

sender

use a single timer to detect loss, reset at each new ACK

upon timeout, resend all W packets
starting with the lost one

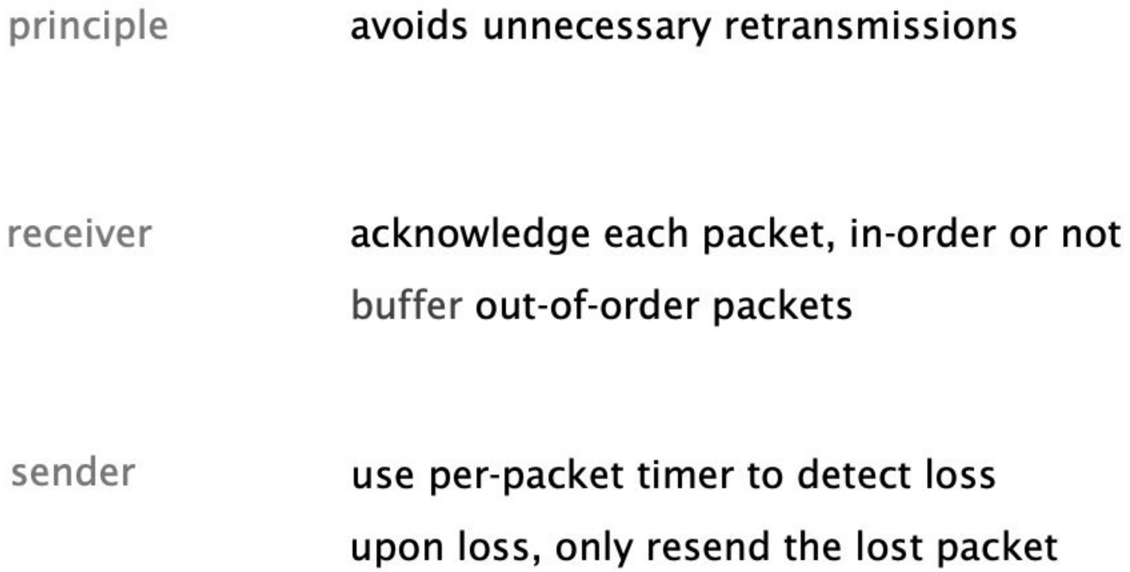
Go-Back-N (GBN) is a simple sliding window protocol using cumulative ACKs



principle	receiver should be as simple as possible
receive	
sender	use a single timer to detect loss, reset at each new ACK
	upon timeout, resend all W packets starting with the lost one

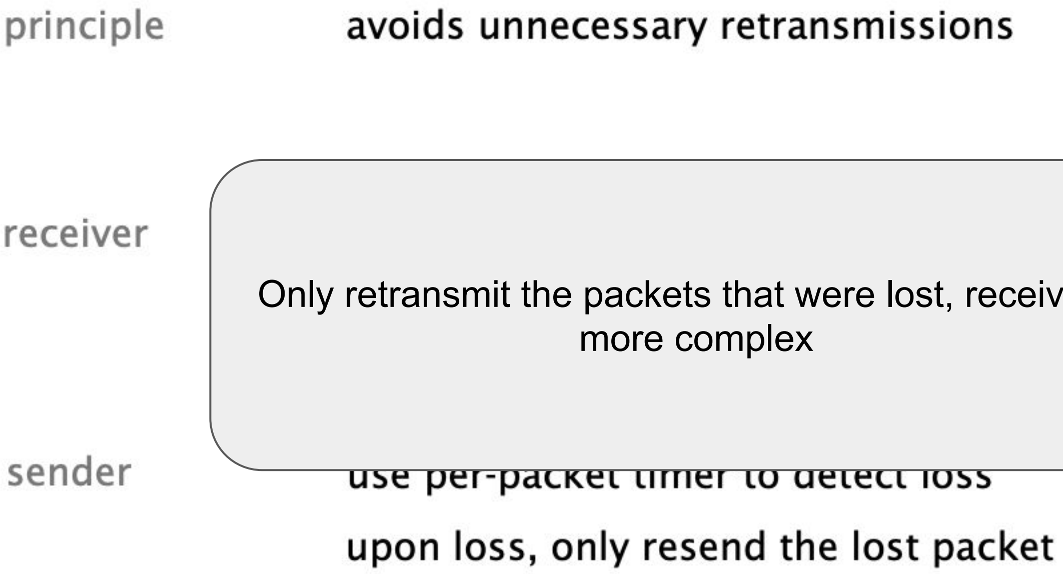
Works well with few errors

Selective Repeat (SR) avoids unnecessary retransmissions by using per-packet ACKs



principle	avoids unnecessary retransmissions
receiver	acknowledge each packet, in-order or not buffer out-of-order packets
sender	use per-packet timer to detect loss upon loss, only resend the lost packet

Selective Repeat (SR) avoids unnecessary retransmissions by using per-packet ACKs



principle	avoids unnecessary retransmissions
receiver	
sender	use per-packet timer to detect loss upon loss, only resend the lost packet

Only retransmit the packets that were lost, receiver is more complex

Illustration

https://www2.tkn.tu-berlin.de/teaching/rn/animations/gbn_sr/

GBN Question

Assume you have a Go-Back-N (GBN) sender and receiver. The receiver acknowledges each data segment with a cumulative ACK which indicates the next expected data segment. Furthermore, it saves out-of-order segments in a buffer. The sender and receiver buffer can contain four segments each. The time-out period is much larger than the time required for the sender to transmit four segments in a row.

- The sender wants to transmit 10 data segments (0,. . . ,9) to the receiver. Assume that exactly one segment is lost. How many segments has the sender to transmit in the best and worst case? For each case, indicate which segment was lost

GBN Question

Assume you have a Go-Back-N (GBN) sender and receiver. The receiver acknowledges each data segment with a cumulative ACK which indicates the next expected data segment. Furthermore, it saves out-of-order segments in a buffer. The sender and receiver buffer can contain four segments each. The time-out period is much larger than the time required for the sender to transmit four segments in a row.

- The sender wants to transmit 10 data segments (0,. . . ,9) to the receiver. Assume that exactly one segment is lost. How many segments has the sender to transmit in the best and worst case? For each case, indicate which segment was lost
  - Best case: 11 segments, the last segment is dropped.
  - Worst case: 14 segments, e.g., if the second segment is dropped. GBN will retransmit all packets in the current window

GBN Question

Assume you have a Go-Back-N (GBN) sender and receiver. The receiver acknowledges each data segment with a cumulative ACK which indicates the next expected data segment. Furthermore, it saves out-of-order segments in a buffer. The sender and receiver buffer can contain four segments each. The time-out period is much larger than the time required for the sender to transmit four segments in a row.

- Once again, the sender wants to transmit 10 data segments (0,..., 9) to the receiver. This time, assume that exactly one ACK is lost. How many segments does the sender have to transmit in the best and worst case and which ACK was lost?

GBN Question

Assume you have a Go-Back-N (GBN) sender and receiver. The receiver acknowledges each data segment with a cumulative ACK which indicates the next expected data segment. Furthermore, it saves out-of-order segments in a buffer. The sender and receiver buffer can contain four segments each. The time-out period is much larger than the time required for the sender to transmit four segments in a row.

- Once again, the sender wants to transmit 10 data segments (0,..., 9) to the receiver. This time, assume that exactly one ACK is lost. How many segments does the sender have to transmit in the best and worst case and which ACK was lost?
    - Best case: 10 segments, e.g., the ACK for segment 5 is lost. Since GBN uses cumulative ACKs, the ACK for segment 6 implicitly also acknowledges segment 5.
    - Worst case: 11 segments, the ACK for the very last segment is lost.