TCP FLOW CONTROL & CONGESTION AVOIDANCE

AUTOR: MARIO KAPPERT

INHALT

- TCP Problem
- TCP Flow Control
- TCP Congestion Avoidance

TCP PROBLEM UND AUSWIRKUNG

- Datenstau
- Datenverlust
- Überlastung im Netzwerk oder Socket

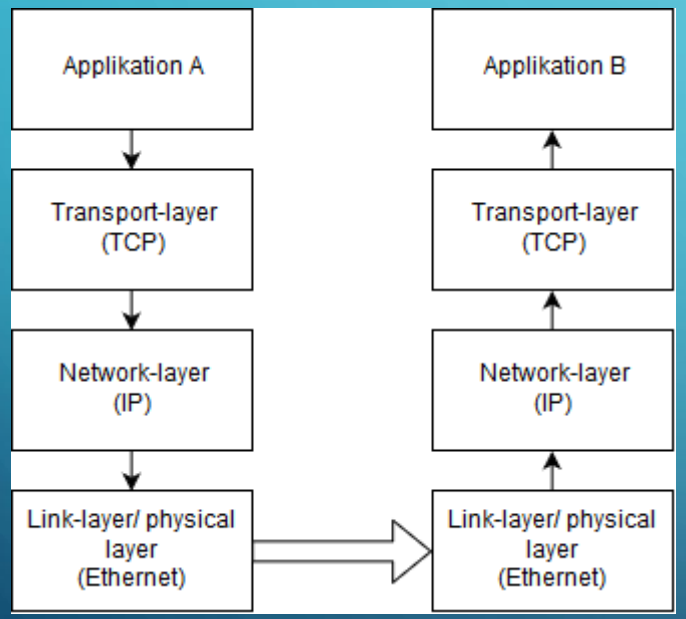
=> 1986 erster Internet Zusammenbruch

TCP FLOW CONTROL

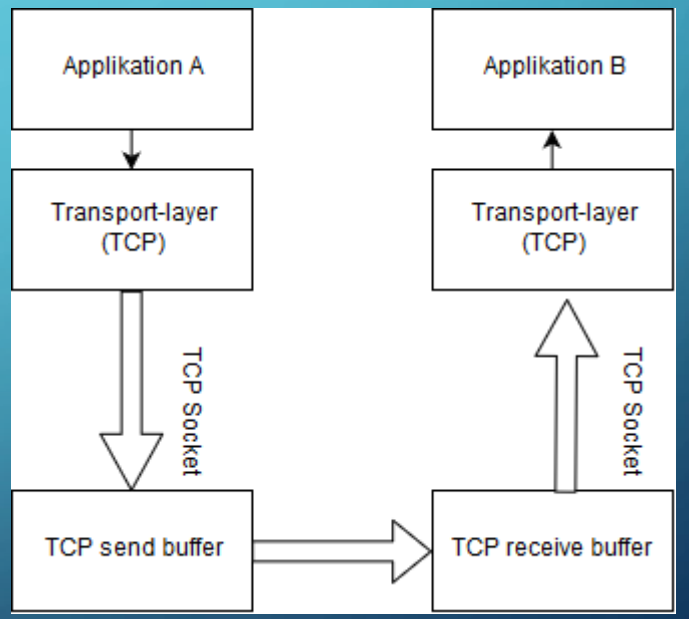
- Verhindert Datenstau
  - Nicht mehr Pakete senden als gelesen werden
  - Empfänger sendet Feedback über den Auslastungsstatus
  - Ähnlich wie Congestion Avoidance, dennoch unterschiedlich

TCP FLOW CONTROL

- TCP Datenaustausch:



Zoom-in TCP:

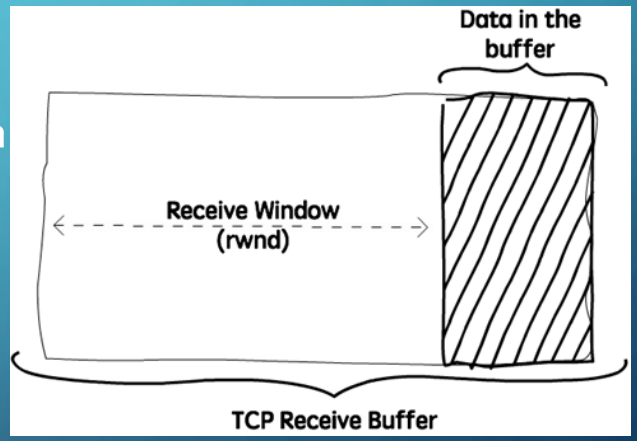


TCP FLOW CONTROL

- Receiver bestimmt wie groß Buffer ist
- Sender darf nur dementsprechend viele Pakete auf einmal senden
- Ansonsten würden Pakete verloren gehen

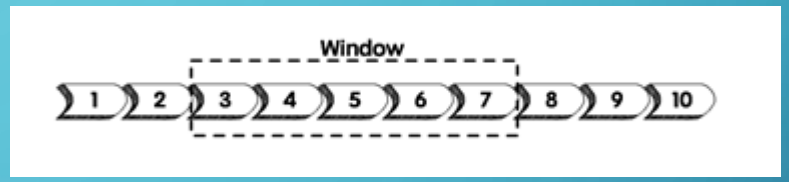
TCP FLOW CONTROL

- Wie funktioniert jetzt Flow Control?
  - Der Empfänger bestimmt das sogennante „Receive Window (rwnd)
  - Receive Window enthält die Größe des Buffers
  - Sender darf Nachrichten <= rwnd ohne ACK senden



TCP FLOW CONTROL

- TCP benutzt dafür das „Sliding Window“



- Beispiel:
- Empfänger kann 5 Datenpakete a 100byte empfangen
- Der Sender schickt sofort 5 los und wartet auf Ack‘s
- Sobald Packet 3 Acknowledged wird, kann Paket 8 gesendet werden

TCP FLOW CONTROL

- Das rwnd ist nicht konstant und kann sich ändern.
  - Verbindungsprobleme
  - Delays
- Feedback nach jedem ACK über die Größe

TCP FLOW CONTROL

- Problem: Verlorene ACK‘s
- => Deadlock
  - receiver wartet auf Pakete, Sender denkt das window wäre voll
  - Für dieses Probleme gibt es einen Persist timer
  - Sobald zero-window gesendet wurde wird periodisch ein kleines Probedatenpaket gesendet um zu testen ob das window auch wirklich voll ist.

CONGESTION AVOIDANCE

- Ähnelt Flow Control
- Beobachtet nicht den Endpunkt
- Überwacht ob ein oder mehrere Sockets nicht das Netzwerk überlasten
- Verhindert packet loss, delays, network collapse

CONGESTION AVOIDANCE

- Besteht aus 4 Teilen:
  - Slow Start
  - Congestion Avoidance
  - Fast Recovery
  - Fast Transmit

- Ssthresh (Slow Start Threshold) variable zur Bestimmung des Zustandes

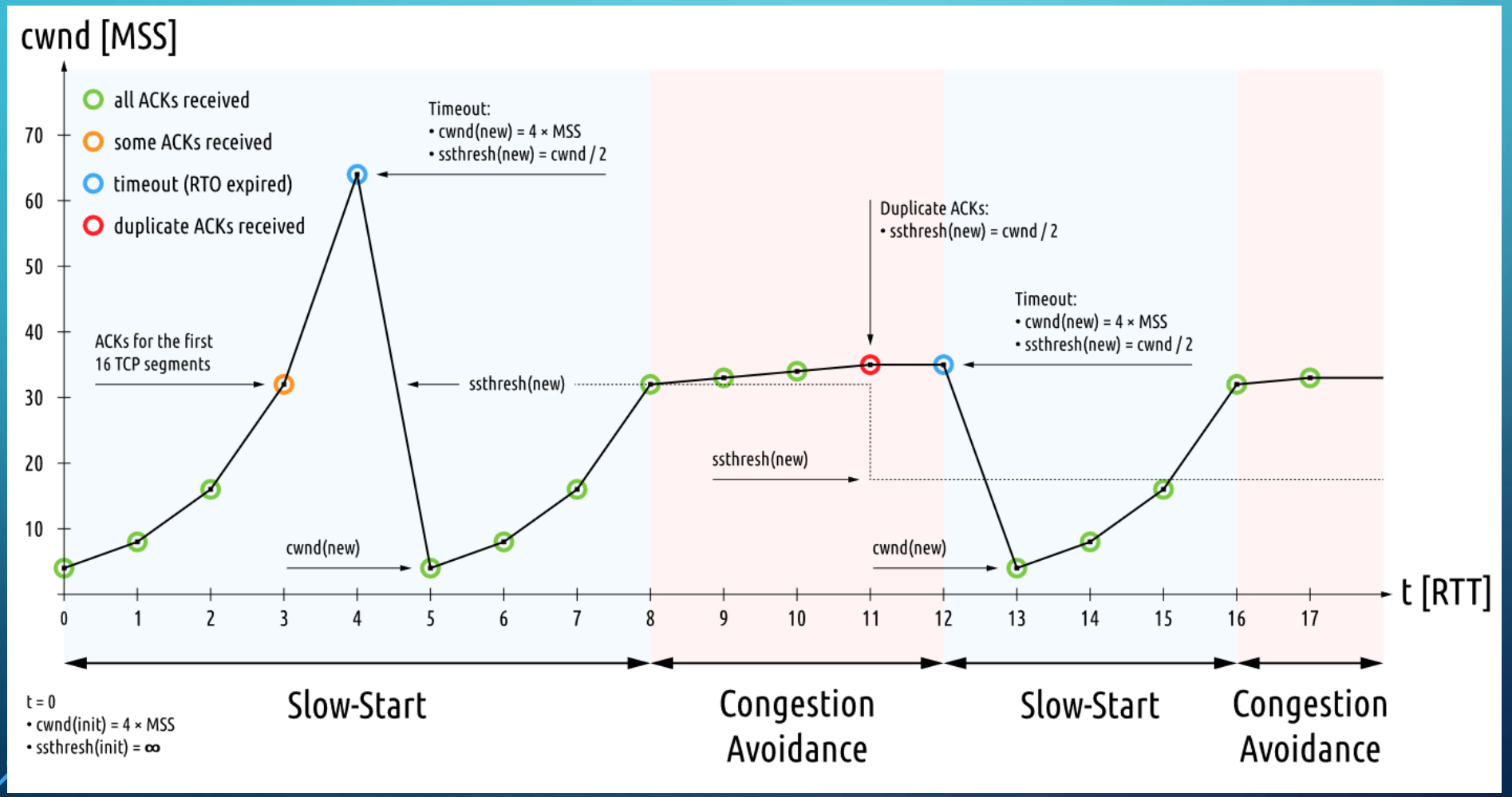
CONGESTION AVOIDANCE

- Slow Start:
  - CWND = Congestion Window
  - Übertragung startet langsam mit 1 MSS(Maximum Segment Size, TCP Header)
  - Nach jedem ACK; ACK++
  - Erhöhung des windows exponentiell
  - MSS = CWND = 2^(n*ACK)

CONGESTION AVOIDANCE

- Congestion Avoidance
- CWND > ssthresh
- Lineare Steigerung der Übertragungsgeschwindigkeit
- CWND += 1 nach jedem ACK
- Sobald ein Stau entsteht wird ssthresh durch 2 dividiert.

CONGESTION AVOIDANCE



CONGESTION AVOIDANCE

- Bei einem timeout wird das cwnd auf init-value oder 1 gesetzt und man beginnt von vorne
- Wachstum wird beendet wenn das von Empfänger festgelegte window erreicht wurde.

CONGESTION AVOIDANCE

- Fast Retransmit und Fast-Recovery:
  - Schnelleres reagieren auf Stau Situationen bei Paketverlust
  - Receiver meldet wenn Pakete in falsche Reihenfolge kommen
  - Sender wird über das letzte korrekte Paket informiert (ACK)
  - Dup-ACK (duplicate acknowledgments)
  - Sender realisiert die duplizierte Bestätigung und nach dem dritten Duplikat erfolgt ein retransmit

CONGESTION AVOIDANCE

- Detaillierter:
- Empfänger informiert Sender über die falsche Reihenfolge
- Nach dreimal falschen ACK wird reagiert
- Die Dup-Acks weisen auf Paketverlust hin, aber das Datenpakete ankommen
- Deswegen CWND/2, anstatt Slowstart

CONGESTION AVOIDANCE

- Das window kann noch um die Anzahl der Dup-Acks erhöht werden
- Jedes Ack steht für ein Datenpaket
- Geschwindigkeit steigt schneller

=> Fast-Recovery

VIELEN DANK FÜR DIE AUFMERKSAMKEIT