

Internet-Transport TCP-Mechanismen



STUDIERN
AUF HÖCHSTEM
NIVEAU

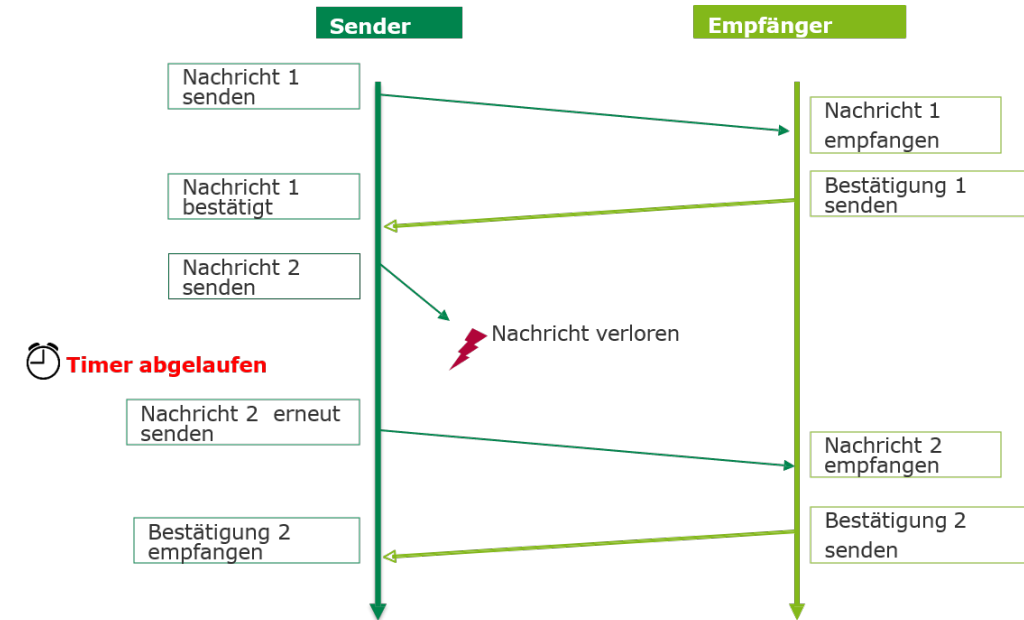
Prof. Dr. Jürgen Anders, Hochschule Furtwangen
Fakultät Digitale Medien

Retransmission

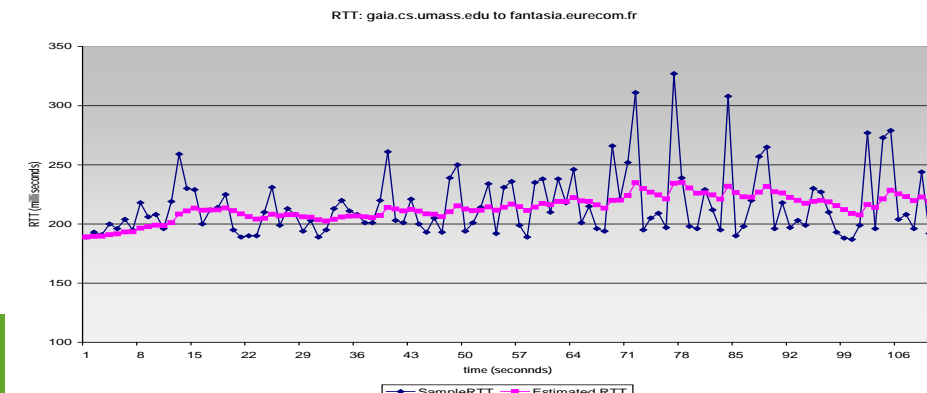
Die von TCP zur Korrektur im Fehlerfall initiierte Neuübertragung von Datenpaketen ist Grundtechnik zur Gewährleistung fehlerfreier Übertragung – **Retransmission**

- Gesendete TCP-Nachrichten werden dem Sender nach Empfang vom Empfänger bestätigt – **Acknowledgement**
- Woher weiß aber Empfänger überhaupt, dass eine TCP-Nachricht gesendet wurde?
- Sender startet vor Übertragung eines Datenpakets eine Uhr – **Timer**:
 - Läuft Timer vor Eintreffen einer Empfangsbestätigung ab, gilt Datenpaket als verloren und wird neu übertragen
- Um Neuübertragung zu erzwingen, sendet Empfänger im Fall beschädigter Nachrichten kein Acknowledgement und wartet einfach auf automatische Retransmission nach Timer-Ablauf
- Wahl der Zeitschranke für Timer bestimmt Effizienz des Datendurchsatzes im Internet
- Innerhalb eines LANs sind kürzere Zeiten sinnvoller als bei Weitverkehrsverbindungen:
- Erfolgt Neuübertragung ...
 - zu früh, wird Internet mit überflüssigen Duplikaten überflutet -> Durchsatz sinkt
 - zu spät, müssen viele Nachrichten zwischengespeichert werden -> Überlauf der Warteschlangen
- Fixe Timer-Zeitschranke kann nie allen Situationen gerecht werden
- TCP bietet deshalb komplexen adaptiven Mechanismus zur Anpassung der Zeitschranke: **Adaptive Retransmission**
- TCP überwacht dazu Netzlast für jede Verbindung mittels der Paketumlaufzeit – **Round-Trip-Time** – Zeitspanne zwischen Aussendung eines Pakets und Eintreffen dessen Empfangsbestätigung:
 - Messung wird für jede Sendung ausgeführt und mit vorhergehenden Messungen für gleiche Verbindung verglichen
 - Über gleitenden Mittelwert – **Smoothed Round-Trip-Time** – wird Zeitspanne adaptiv an momentane Lastsituation angepasst

TCP-Retransmission bei Zeitüberschreitung



Round-Trip-Time - RTT

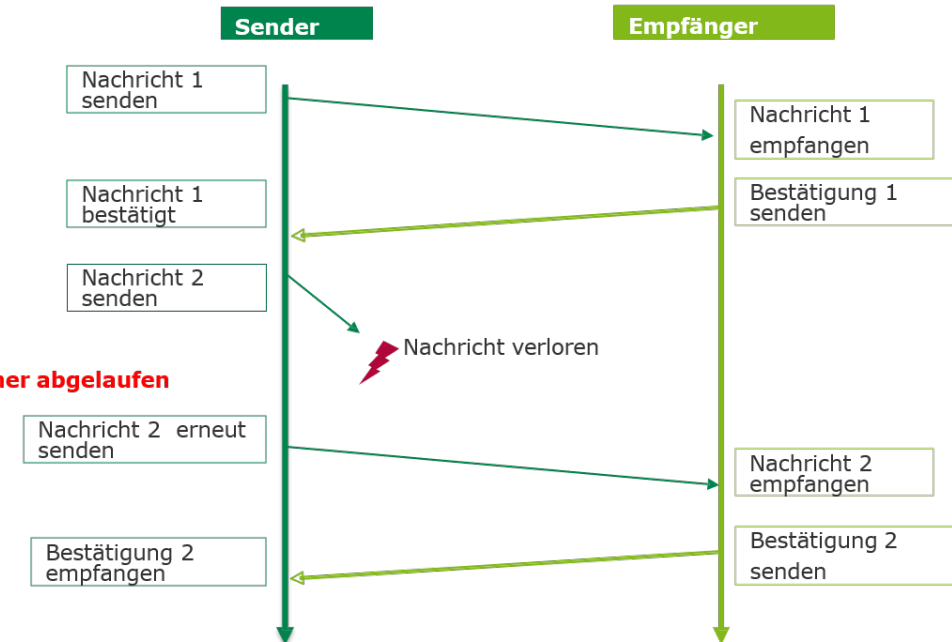


Retransmission

Die von TCP zur Korrektur im Fehlerfall initiierte Neuübertragung von Datenpaketen ist Grundtechnik zur Gewährleistung fehlerfreier Übertragung – **Retransmission**

- Gesendete TCP-Nachrichten werden dem Sender nach Empfang vom Empfänger bestätigt – **Acknowledgement**
- Woher weiß aber Empfänger überhaupt, dass eine TCP-Nachricht gesendet wurde?
- Sender startet vor Übertragung eines Datenpakets eine Uhr – **Timer**:
 - Läuft Timer vor Eintreffen einer Empfangsbestätigung ab, gilt Datenpaket als verloren und wird neu übertragen
- Um Neuübertragung zu erzwingen, sendet Empfänger im Fall beschädigter Nachrichten kein Acknowledgement und wartet einfach auf automatische Retransmission nach Timer-Ablauf
- Wahl der Zeitschranke für Timer bestimmt Effizienz des Datendurchsatzes im Internet
- Innerhalb eines LANs sind kürzere Zeiten sinnvoller als bei Weitverkehrsverbindungen:
- Erfolgt Neuübertragung ...
 - zu früh, wird Internet mit überflüssigen Duplikaten überflutet -> Durchsatz sinkt
 - zu spät, müssen viele Nachrichten zwischengespeichert werden -> Überlauf der Warteschlangen
- Fixe Timer-Zeitschranke kann nie allen Situationen gerecht werden
- TCP bietet deshalb komplexen adaptiven Mechanismus zur Anpassung der Zeitschranke: **Adaptive Retransmission**
- TCP überwacht dazu Netzlast für jede Verbindung mittels der Paketumlaufzeit – **Round-Trip-Time** – Zeitspanne zwischen Aussendung eines Pakets und Eintreffen dessen Empfangsbestätigung:
 - Messung wird für jede Sendung ausgeführt und mit vorhergehenden Messungen für gleiche Verbindung verglichen
 - Über gleitenden Mittelwert – **Smoothed Round-Trip-Time** – wird Zeitspanne adaptiv an momentane Lastsituation angepasst

TCP-Retransmission bei Zeitüberschreitung



TCP regelt Datenfluss über eine Verbindung mit Hilfe eines Schiebefenster-Protokolls – **Sliding Window Protocol**

- Schiebefenster-Protokoll arbeitet mit adaptiven, lastabhängigen Fenstern – **Windows**, wobei Quittierung und Zuweisung eines Fensters voneinander entkoppelt sind

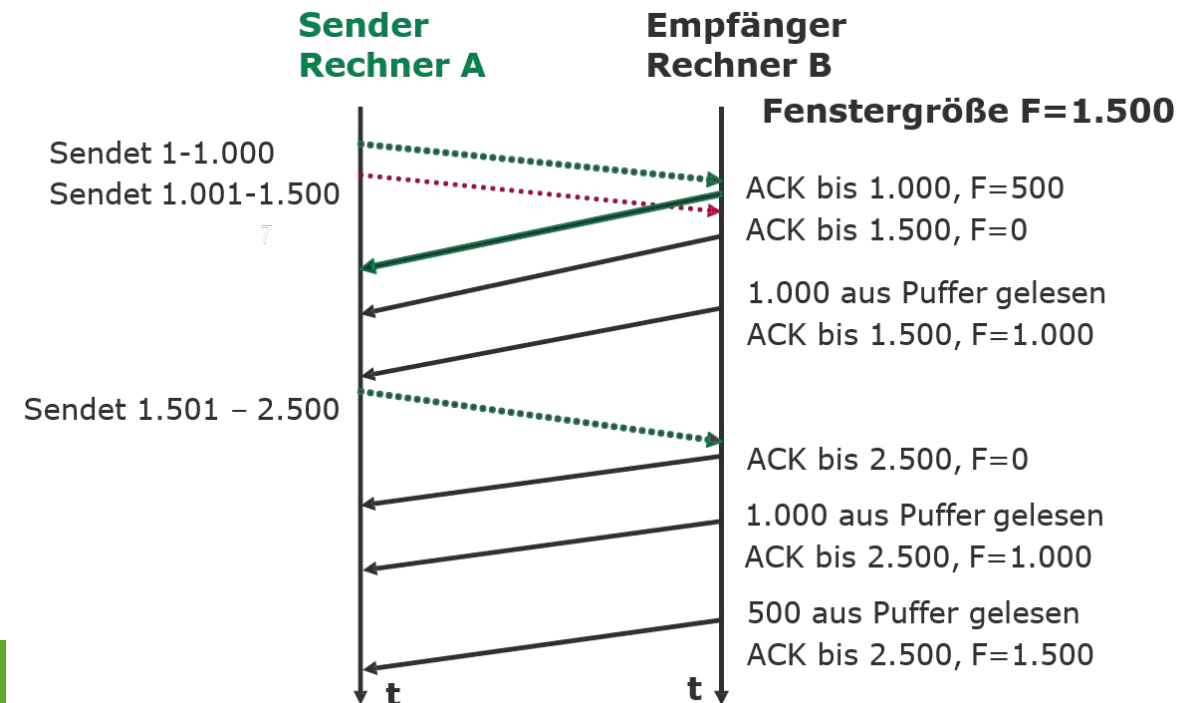
Beispiel:

Nachricht der Länge 2.500 Byte wird per TCP-Verbindung mit Fenstergröße 1.500 Byte ($F = 1.500$) von A nach B übertragen

- A sendet erste 1.000 Byte
- B nimmt diese entgegen, schreibt sie in Eingangspuffer, setzt Fenstergröße auf $1.500 - 1.000 = 500$ ($F=500$) und bestätigt Annahme der ersten 1.000 Byte ($ACK=1.000$) und Fenstergröße ($F=500$)
- A überträgt gemäß erlaubter Fenstergröße nächste 500 Byte
- B nimmt diese entgegen, setzt Fenstergröße auf 0 und bestätigt Empfang und neue Fenstergröße
- A muss nun warten bis Anwendung auf B aus Eingangspuffer 1.000 Byte ausgelesen hat, Fenstergröße wieder auf 1.000 Byte gesetzt und entsprechende Mitteilung geschickt ($ACK=1.500$, $F=1.000$) ist
- A kann nun den Rest der Nachricht (1.000 Byte) schicken

Silly Window Syndrom

- Ohne zusätzliche Vorkehrungen würde sich bei schnellen Verbindungen die Fenstergröße nahe 0 einpegeln und die Übertragungsstrecke nur unzureichend genutzt
- Empfänger gibt deshalb Rückmeldung über verfügbare Fenstergröße erst dann, wenn wieder mindestens 50% der maximalen Fenstergröße zur Verfügung stehen
- Sender sorgt dafür, dass die verfügbare Fenstergröße nicht vollständig ausgenutzt wird



TCP Überlastkontrolle (Congestion Control)

Überlaststeuerung – **Congestion Control** – ist eines der schwierigsten Probleme für TCP, da TCP Überlastsituationen nicht direkt erkennen kann:

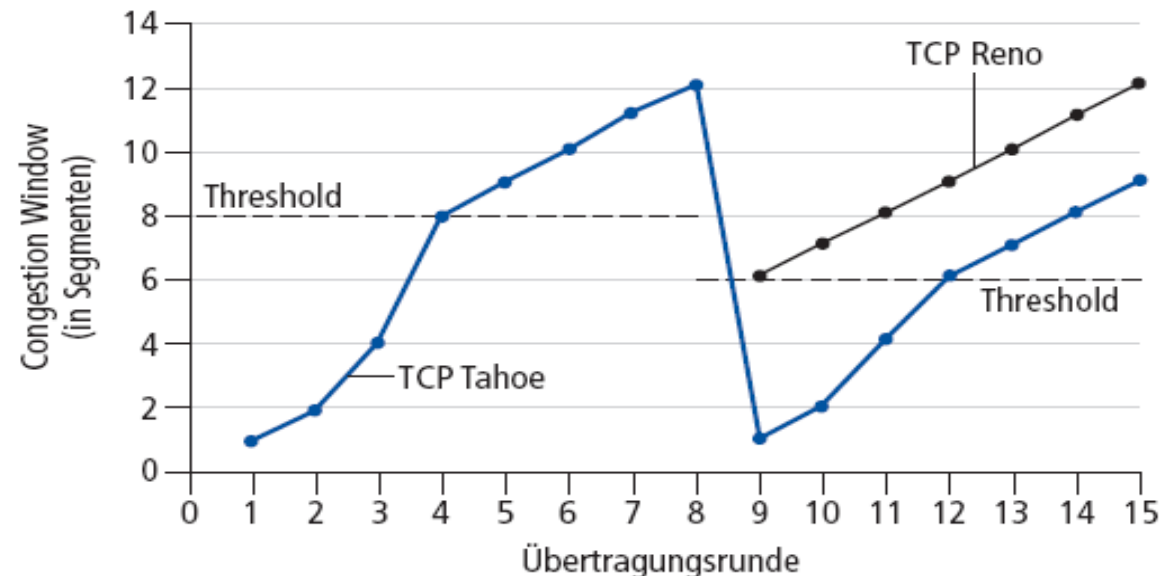
- Datenpakete werden per IP auf unterschiedlichen Wegen durch das Internet befördert, sodass keine direkten Rückschlüsse auf Überlast bestimmter Zwischensysteme gezogen werden können
- Verschiedene TCP-Instanzen können nicht miteinander kooperieren

Idee: TCP interpretiert Zahl der verlorenen (also nicht bestätigten) Segmente als Maß für Überlast und Parameter für Bestimmung des Congestion Windows zur Regelung der Überlast

- **Slow-Start Algorithmus** – zur schnellen Anpassung des Congestion Windows:
- Verbindungen werden mit kleiner Fenstergröße gestartet, die dann solange **exponentiell gesteigert wird**, bis Pakete verloren gehen...
- Nach Erreichen eines Grenzwertes erfolgt die Steigerung nur noch linear
- Der Grenzwert wird dynamisch angepasst
- Bei Überschreiten der maximal Fehlerrate erfolgt Zurücksetzen des Fensters
- Hierfür gibt es verschiedene Algorithmen (z.B. TCP Tahoe oder Reno)

Congestion-Avoidance Algorithmus – zum Abbau von Überlastsituationen:

Häuft sich der Segmentverlust – d.h. Quittungen erreichen Sender nicht innerhalb der festgesetzten Zeitspanne – **senkt TCP die Sende- Datenrate**, um eine Überlastung des Netzwerkes nicht mit ständiger Wiederholung von Sendungen zu verschärfen



Internet-Transport TCP-Mechanismen



STUDIERN
AUF HÖCHSTEM
NIVEAU

Prof. Dr. Jürgen Anders, Hochschule Furtwangen
Fakultät Digitale Medien