

Übungsblatt 12

Abgabe am 16.07.2010 in der Vorlesung, oder im Briefkasten in der Oettingenstraße 67 (bis 11:00 Uhr).

Hinweis: Schreiben Sie unbedingt Ihre Übungsgruppe auf Ihre Abgabe!

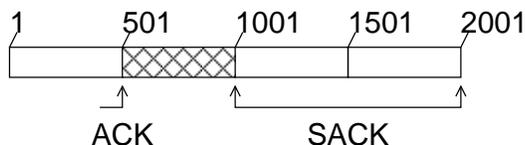
1. TCP-Verbindung (H)

Ein Protokoll der Anwendungsschicht (z.B. HTTP) führt einen Anfrage-Antwort-Dialog aus, der über eine TCP-Verbindung zwischen einem Client- und einem Serverprozess transportiert werden soll. Die Netzverzögerung zwischen Client und Server betrage 150 ms, unabhängig von der Nachrichtenlänge. Ferner betragen die Größe der Anfrage (Request) 50 Byte und die Größe der Antwort (Response) 1000 Byte.

- Zeichnen Sie ein Sequenzdiagramm des gesamten TCP-Austausches zwischen Client und Server! Beschriften Sie dabei die Pfeile mit den dabei relevanten Teilen der TCP-Segmentstruktur (relevante Flags, Sequenznummer, ACK-Nummer). Initiale Sequenznummern seien 6000 für den Client und 9000 für den Server.
- Zeitverhältnisse
 - Wie lange dauert es, bis die Antwort (Response) beim Client angekommen ist?
 - Um welchen Faktor schneller wäre der Austausch von Anfrage/Antwort mittels UDP-Datagrammen?
 - Wie lange besteht die TCP-Verbindung mindestens (wann wird das letzte Segment empfangen)?

2. Selektive Quittungen (H)

Der Verlust einzelner Segmente kann zur unnötigen Wiederholung großer Datenmengen führen, insbesondere bei Pfaden mit hoher Netzverzögerung. Durch die Einführung selektiver Quittungen (SACK, siehe RFC 1888) kann dieses Problem gemildert werden.



Statt wie bei den „normalen“ kumulativen TCP-Quittungen den bis dahin korrekt empfangenen *zusammenhängenden* Byte-Strom zu quittieren, kann ein Empfänger mit selektiven Quittungen zusätzliche Segmente (oder zusammenhängende Folgen von Segmenten, sogenannte Blöcke) im Options-Feld des TCP-Headers als empfangen notieren. Hierzu wird ein Bereich des Byte-Stroms mit Anfangs- und End-Byte notiert. Auf Grundlage der obigen Abbildung würde etwa eine Quittung mit AckNr=501 und SACK-Block=(1001,1500) gesendet; das verlorene Segment mit Bytes 501–1000 (schraffiert) wird so ausgespart. Es können mehrere solche Blöcke in den Optionen angegeben werden.

Gehen Sie von einem Sender mit aktueller SeqNr=5000 und einem Empfänger (AckNr=2001) aus. Der Sender sendet 8 Segmente von jeweils 500 Byte Länge. Wie werden kumulative und selektive Quittungen benutzt, wenn:

- die ersten vier Segmente empfangen werden, die letzten vier aber verloren gehen?
- das zweite, vierte und sechste und achte Segment verloren gehen?

3. Staukontrolle (H)

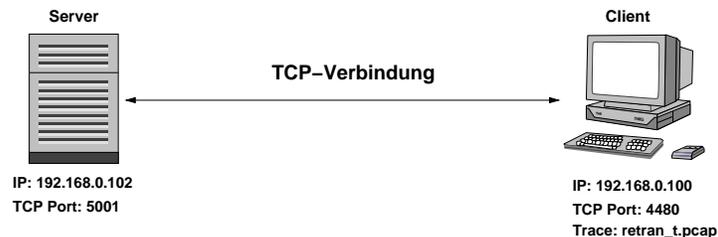
Bei TCP kommt Slow-Start als Mechanismus zur Staukontrolle zum Einsatz. Im folgenden soll die Auswirkung des Slow-Starts auf die Übertragung von Dateien am konkreten Beispiel behandelt werden. Hierbei werden einige Vereinfachungen vereinbart:

- Zwischen den zwei Systemen besteht eine direkte Verbindung über einen zuverlässigen Kanal. Es sollen also keine Paketverluste auftreten.

- Die Endsysteme besitzen unendlich große Puffer und keinerlei Verarbeitungsverzögerung.
 - Jeglicher Overhead durch Protokollheader wird ignoriert. Steuerpakete (z.B. Bestätigungen) werden also sofort übertragen, sie besitzen keine Übertragungsverzögerung. (Im Sequenzdiagramm durch Striche darstellbar).
 - Der Verbindungsabbau wird ignoriert.
 - Die Slow-Start-Phase beginnt **nach** dem Verbindungsaufbau!
 - Die RTT beträgt 200ms, also 100ms in jede Richtung.
 - Segmente, die Daten enthalten, sind $S = 500$ Bytes groß. (Im Sequenzdiagramm ist dies durch Balken darstellbar).
 - Der Grenzwert (**Threshold**) der TCP Verbindung ist so hoch, dass er nie erreicht wird, der Sender bleibt also die ganze Zeit im Slow-Start.
- (a) Ein FTP-Server überträgt eine Datei zum Client. Geben Sie mit Hilfe eines Sequenzdiagramms den Ablauf für die Übertragung einer Datenmenge von 7500 Byte (also 15 Pakete zu je 500 Byte) an. Die maximale Übertragungsrate beträgt $R = 20\,000$ Byte/s. Schreiben Sie an das Sequenzdiagramm jeweils die Größe des aktuellen Überlastfensters (**CongWin**). *Hinweis:* nach dem Verbindungsaufbau durch den FTP-Clienten beginnt der Server sofort mit der Datenübertragung.
- (b) Bestimmen Sie die Zeit, die diese Übertragung braucht (vom Absenden des SYN des Clients bis alle Daten beim Client angekommen sind):
- mit Slow-Start
 - mit reinem Go-Back-N mit Fenstergröße 20 (also "ohne Slow-Start".)
- (c) Wie lange würde die Übertragung mit/ohne Slowstart dauern, wenn eine höhere Übertragungsrate von 500 000 Byte/s (statt 20 000 Byte/s) zur Verfügung stände?

4. Analysen von TCP Trace

Der Datenverkehr einer TCP-Verbindung zwischen zwei Hosts (Client und Server, siehe Abbildung) wurde in einer Trace-Datei *retran_t.pcap* festgehalten



Laden Sie die Trace-Datei in *Wireshark* und führen Sie die folgenden Analysen durch:

- (a) Finden Sie die TCP-Segmente, die den *Three-Way-Handshake* beinhalten.
- Welche TCP-Parameter werden beim *Three-Way-Handshake* festgelegt. Was bedeuten sie? Geben Sie die Werte der Parameter an!
 - Welche sind die relativen bzw. absoluten Sequenznummern, die von Sender und Empfänger als initiale Sequenznummern angewendet werden? Wie kommen die initialen Sequenznummern zustande? *Hinweis:* Wireshark zeigt standardmäßig relative Sequenznummern (fängt also bei 0 zu zählen an). Die absolute Sequenznummer kann in der Hex-Ansicht abgelesen werden.
- (b) Aufgrund verlorener Segmente kommen selektive Quittungen zum Einsatz. Finden Sie die Segmente, die *Selective Acknowledgment* Optionen beinhalten (Filter `tcp.options.sack`). Welches ist der größte ausstehende Block? Wie viele Segmente beinhaltet er?