11. Übung zur VL Betriebs- und Kommunikationssysteme

Tutor: Thomas Tegethoff

Bearbeiter: Etienne Jentzsch, Carola Bothe

1a) Unterschied zwischen TCP und UDP

Die Aufgabe der Transport Schicht (4. des OSI Modells) besteht darin, Datenpakete zwischen Prozessen hin und her zu schicken. Dazu benötigt es Protokolle, die spezifizieren, wie genau und an wen das Pakte geschickt wird. Die am weitesten verbreiteten sind TCP (Transmission Control Protocol) und UDP (User Datagram Protocol). TCP ist ein verbindungsorientiertes Protokoll, d.h. bevor ein Paket geschickt wird, wird zunächst eine Verbindung aufgebaut (vergleichbar mit einem Anruf übers Telefonnetz). Dies benötigt 3 Pakete, dafür garantiert TCP, dass die Daten ankommen und auch in der richtigen Reihenfolge (durch die Angabe von Sequenznummern). Die Header Größe ist mit 20 Bytes größer als die von UDP (8 Bytes) und TCP ist langsamer bei der Datenübertragung. UDP hingegen verfolgt den „best effort“ Ansatz. Vergleichbar mit einer Postkarte schickt UDP einfach einzelne Pakete an den angegebenen Empfänger. Daten können so schneller übermittelt werden aber UDP garantiert nicht für das Ankommen der Daten und erkennt zwar Fehler aber korrigiert sie nicht, sondern verwirft das Paket. TCP hingegen schickt eine Nachricht an den Absender, wenn das Paket fehlerhaft ist, sodass es erneut übertragen werden kann.

Trotz allem haben TCP und UDP natürlich Gemeinsamkeiten, denn sie verfolgen das gleiche Ziel, Daten zu übermitteln. In beiden Headern finden sich daher Absender und Ziel Port genau wie in allen weitern Transport-Schicht Protokollen (DCCP, SCTP). In allen Headern gibt es zudem eine Checksumme, die das Erkennen von Fehlern ermöglicht.

1b) TCP Verbindungsaufbau und -abbau

TCP baut vor der Datenübertragung eine Verbindung über einen sogenannten 3-way-handshake auf. Dabei wird vom Sender zunächst eine Verbindungsanfrage an den Empfänger geschickt (SYN). Dieser kann die Verbindung dann akzeptieren (SYN/ACK), was der erste Sender wiederum bestätigt (ACK). Die dritte Nachricht kann aus Effizienzgründen bereits Daten mit übertragen.

Für den Abbau mit Einverständnis (beide haben alles gesendet und empfangen) braucht es wieder drei Wege: ein Host sendet Anfrage, die Verbindung zu trennen (DR1), der andere sendet gleiche Anfrage zurück (DR2) und der erste bestätigt diese (ACK). Wenn die erste oder zweite Anfrage verloren geht, wird einfach die erste noch einmal geschickt. Geht das ACK verloren, wird die Verbindung irgendwann per time-out geschlossen.

c) Beispielanwendungen für TCP und UDP

TCP ist für Anwendungen sinnvoll, bei denen die Übertragungszeit weniger wichtig sind aber die hohe Zuverlässigkeit brauchen. Ein Beispiel wäre daher die Übertragung von Emails. Jeder möchte, dass seine Emails auch genauso ankommen, wie man sie abgeschickt hat und dass man weiß, falls etwas nicht geklappt hat. Dabei spielt die geringe Zeitverzögerung keine Rolle.

UDP ist sinnvoll für Anwendungen, die schnelle, effiziente Datenübertragung brauchen wie viele multiplayer Onlinespiele oder Videokonferenzen. Wenn da mal ein Paket verloren geht, ist das zumeist verkraftbarer als eine Verzögerung der Übertragung.

d)

Livestreams nutzen idR UDP (mit RTSP), da bei Livestreams nur wenige Informationen vorher zur Verfügung stehen (wäre bei TCP hilfreich) und je mehr gepuffert werden muss, desto höher wäre die Verzögerung zum Event, das man verfolgt. UDP hat im Gegensatz zu TCP keine eingebaute Staukontrolle und nicht die Fairness Eigenschaft, dass es die verfügbare Bandbreite mit allen Verbindungen teilt. Hätten wir also eine sehr geringe Bandbreite, wäre dies durch den Livestream komplett mit UDP belegt und wir können unseren Übungszettel nicht herunterladen, denn das geschieht normalerweise über TCP.

Quellen:

<http://www.diffen.com/difference/TCP_vs_UDP>

<https://de.wikipedia.org/wiki/Transmission_Control_Protocol>

<https://de.wikipedia.org/wiki/User_Datagram_Protocol>

VL 11 Folien und Mitschrift