

Rechnernetze und verteilte Systeme

Theorie 5: Transportschicht

Fachgruppe Telekommunikationsnetze (TKN)

12. Februar 2024

Einleitung

Die folgenden Aufgaben werden gemeinsam im Tutorium bearbeitet. In der Veranstaltung Rechnernetze wird die SI-Notation verwendet. Beispiele für Präfixe: $m = 10^{-3}$, $k = 10^3$, $M = 10^6$, $ki = 2^{10}$, $Mi = 2^{20}$. „B“ bezeichnet Bytes, „bit“ Bits.

Übung 1 *TCP Basics*

Sie haben in der Vorlesung die Funktionsweise von TCP kennengelernt. Beantworten Sie in diesem Kontext die folgenden Fragen:

1. Welche Pakete werden beim TCP-Handshake ausgetauscht?
2. Welche Pakete werden ausgetauscht, wenn der eigene Rechner die TCP-Verbindung abbauen möchte? Was passiert, wenn der entfernte Rechner noch ausstehende Daten hat?
3. Das Internet gilt als sog. Best-Effort-Netzwerk. Konkret heißt das, dass Pakete verfälscht werden können, komplett verloren gehen oder die Sendereihenfolge nicht erhalten wird. Überlegen Sie sich wie diese Effekte zustande kommen können!
4. Welche Mechanismen nutzt TCP um die Probleme aus c) zu lösen?

Übung 2 *ARQ*

Sie haben in der Vorlesung verschiedene Fehlerkontrollmechanismen (ARQ) kennengelernt. Bearbeiten Sie in diesem Kontext folgende Aufgaben:

1. Wie erkennt ein ARQ-Mechanismus überhaupt Fehler?
2. Was ist ein kumulatives ACK?
3. Wie funktioniert das GoBackN-Verfahren? Erklären Sie es in eigenen Worten Ihren Kommiliton:innen!

4. Vergleichen Sie die vorgestellten ARQ-Mechanismen anhand ihrer Komplexität sowohl für Sender als auch Empfänger!
5. Angenommen es wird TCP genutzt und der Empfänger hat gerade Byte 4711 empfangen. Welche Sequenznummer schreibt der Empfänger in sein ACK-Paket?

Übung 3 *TCP Window*

TCP benutzt einen sog. Schiebefenstermechanismus, nicht nur zur Fehlerkontrolle sondern auch zur Fluss- und Überlast/Staukontrolle.

1. Was ist der Unterschied zwischen Flow- und Congestion-Control?
2. Wie wird der Timeout in TCP gewählt? Ist dieser konstant?
3. Was sind die Auswirkungen einer zu großen, bzw. einer zu kleinen Schätzung der RTT?
4. Wie interpretiert TCP Paketverluste? Geben Sie ein Beispiel, wann diese Interpretation falsch ist.

Übung 4 *GoBackN*

Nutzen Sie das verlinkte Onlinetool¹ um eine Simulationen von GoBackN durchzuführen. Nehmen Sie eine Fenstergröße von 5 Paketen an und spielen Sie verschiedene Szenarien durch (z.B. Verlust eines oder mehrerer Pakete und/oder Acknowledgements)

Interessant dürften extreme Fälle sein, z.B.:

- Das erste Paket kommt nicht an, aber die restlichen 4
- Alle 5 Pakete kommen an. Die ersten 4 ACKs gehen verloren, nur das 5. kommt an.

Hinweis

Das Onlinetool setzt den Timer komplett zurück für jedes ACK das ankommt. Im Tanenbaum und in der Musterlösung ist das nicht der Fall. Dort wird der Timer auf den Timeout des ältesten Pakets gesetzt. Ersteres ist zwar effizienter, z.B. falls das erste ACK verloren geht, aber letzteres hat höhere Effizienz wenn das letzte Paket im Fenster verloren geht, da es schneller wiederholt wird.

¹https://www.tkn.tu-berlin.de/teaching/rn/animations/gbn_sr/

Übung 5 *Bit Flips*

Ein Paket der Länge n soll über einen fehleranfälligen Kanal übertragen werden. Die Bitfehlerwahrscheinlichkeit betrage p_b .

1. Wie groß ist die Wahrscheinlichkeit, dass das Paket ohne Fehler übertragen wird?
2. Wie groß ist der Erwartungswert der Anzahl der Versuche die unternommen werden müssen, bis das Paket einmal ohne Fehler übertragen wurde?
3. Welche Annahme liegt Ihrer Berechnung zugrunde und weshalb ist sie realistisch, bzw. unrealistisch?

Übung 6 *Effizienz des Go-Back-N Protokolls*

1. Bestimmen sie die Effizienz des Go-Back-N-Mechanismus, wenn bei der Übertragung von Daten keine Fehler auftreten. Die Effizienz eines Protokolls ist das Verhältnis zwischen erreichtem Durchsatz (Achtung: Netto! Also Goodput², nicht Throughput) und dem theoretischen Maximum.
 - Die Größe des verwendeten Fensters ist w .
 - Die Größe jedes Datenpakets ist n_d Bits
 - die Größe der Acknowledgements ist n_a Bits.
 - Bei jeder Datenübertragung entsteht eine Verzögerung τ .
 - Die Datenrate des verwendeten Mediums ist R .

Fertigen sie eine Skizze an und geben sie die allgemeine Formel an.

2. Wie gross muss das Fenster mindestens sein, damit das Protokoll eine Effizienz von 1 erreicht?

Übung 7 *TCP vs. UDP*

Diskutieren Sie die Vor- und Nachteile von UDP und TCP für die folgenden Anwendungsbeispiele. Welches Transportprotokoll ist jeweils sinnvoller?

1. Fernzugriff auf eine Datenbank
2. Videostreaming
3. Videotelefonie
4. Übertragung großer Mengen von Archivdaten
5. Multiplayer Gaming
6. Ein P2P-Overlay-Netzwerk

²<https://de.wikipedia.org/wiki/Datendurchsatz>

Übung 8 3-Wege-Handshake

Einem Chatbot³ wurde folgende **Aufgabe** gestellt:

! Erklären Sie den Prozess des TCP 3-Wege-Handshake und den Zweck jedes Schrittes. Beziehen Sie in Ihrer Erklärung die spezifischen Nachrichten ein, die zwischen dem Client und dem Server während des Handshakes ausgetauscht werden.

Die **Antwort** des Chatbots war die folgende:

Der TCP-3-Wege-Handshake ist der Prozess, der verwendet wird, um eine zuverlässige, geordnete Verbindung zwischen zwei Geräten, typischerweise einem Client und einem Server, unter Verwendung des Transmission Control Protocol (TCP) herzustellen. Der Prozess besteht aus drei Schritten:

1. SYN (Synchronisieren): Der Client sendet eine Nachricht an den Server, in der das SYN-Flag auf 1 gesetzt ist, um anzuzeigen, dass er eine neue Verbindung aufbauen möchte. Diese Nachricht enthält auch die vom Client gewählte Anfangssequenznummer (ISN).
2. SYN-ACK (Synchronize-Acknowledge): Der Client empfängt die SYN-Nachricht und antwortet mit einer Nachricht, bei der sowohl das SYN- als auch das ACK-Flag auf 1 gesetzt sind. Diese Nachricht enthält die eigene ISN des Servers und bestätigt den Empfang der ISN des Clients.
3. ACK (Acknowledge): Der Server empfängt die SYN-ACK-Nachricht und sendet eine letzte Nachricht an den Client, in der das ACK-Flag auf 1 gesetzt ist und der Empfang der ISN des Clients bestätigt wird.

Der Zweck der SYN-Nachricht besteht darin, die Verbindung zu beenden und die Sequenznummern zwischen Client und Server zu synchronisieren. Die SYN-ACK-Nachricht dient dazu, den Empfang der SYN-Nachricht zu bestätigen und die eigene ISN des Clients zu übermitteln. Die abschließende ACK-Nachricht bestätigt, dass der Server die SYN-ACK-Nachricht erhalten hat und die Verbindung hergestellt ist.

Sobald der Drei-Wege-Handshake abgeschlossen ist, gilt die Verbindung als beendet, und es können keine Daten mehr zwischen Client und Server über die bestehende Verbindung ausgetauscht werden.

1. Leider haben sich einige Fehler in die Antwort des Chatbots eingeschlichen. Identifizieren Sie diese!
2. Visualisieren Sie den korrekten Ablauf des 3-Wege-Handshakes, inklusive der relevanten Flags die in den Nachrichten gesetzt sein müssen, sowie die verwendeten Sequenznummern/Acknowledgements.

³Diese Aufgabe wurde unter Zuhilfenahme von ChatGPT (<https://openai.com/blog/chatgpt/>) erstellt.

3. In welchem Paket dürfen jeweils die ersten Nutzdaten in beiden Richtungen versandt werden?

Übung 9 *TCP-Übertragung*

1. Berechnen Sie die Zeit zum Kopieren eines Objektes vom Server zum Client mit TCP. Nehmen Sie an, dass die TCP-Verbindung zwischen Server und Client gerade frisch aufgebaut wurde und das Objekt somit sofort übertragen werden kann. Nehmen Sie außerdem an, dass keine Paketverluste auftreten. Ignorieren Sie außerdem den Overhead von TCP-Headern sowie Protokollen tieferer Schichten.

Gehen Sie von den folgenden Werten aus:

- Objektgröße $O = 10 \text{ kB}$
 - Segmentgröße $MSS = 1 \text{ kB}$
 - Round-Trip-Time $RTT = 1 \text{ s}$
 - Datenrate $R = 10 \text{ kB s}^{-1}$
2. Diskutieren Sie, wie sich Änderungen der Vorgaben auf den Ablauf der Übertragung auswirken. Welchen Effekt hat eine Halbierung der Segmentgröße? Welchen Effekt hat eine Verdopplung der Objektgröße? Welchen Effekt hat eine Verdopplung der Datenrate?
 3. Diskutieren Sie nun, wie sich der Ablauf der Übertragung verändert, wenn Paketverluste auftreten. Nehmen Sie beispielsweise an, dass jedes 2. Paket sicher verloren geht.

Übung 10 *TIME-WAIT*

RFC 1122⁴ sieht vor, dass Sockets nach dem Schließen im TIME-WAIT Zustand verbleiben:

! When a connection is closed actively, it MUST linger in TIME-WAIT state for a time $2 \times \text{MSL}$ (Maximum Segment Lifetime). However, it MAY accept a new SYN from the remote TCP to reopen the connection directly from TIME-WAIT state, if it: [...]

Die MSL ist laut TCP 120 s, entsprechend verweilen Sockets für eine signifikante Zeit. Dies führt dazu, dass Anwendungen den Socket in diesem Zeitraum nicht verwenden können, da die Socket Address already in use ist.

Warum sieht TCP diesen Zustand vor? In welchem Fall ist das Verweilen des Sockets sinnvoll?

⁴<https://www.rfc-editor.org/rfc/rfc1122#page-88>

Übung 11 *Fluss- und Überlastkontrolle*

Beantworten Sie die folgenden Fragen im Kontext von Transportprotokollen:

1. Was ist der Unterschied zwischen Flusskontrolle und Überlastkontrolle?
2. Über welchen Mechanismus wird Flusskontrolle in TCP implementiert?
3. Über welchen Mechanismus wird Überlastkontrolle in TCP implementiert?
4. Inwiefern hängen die beiden Aspekte zusammen?