

Rechnernetze und Verteilte Systeme

Block 1

Einführung

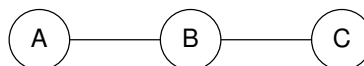
21. — 26.10.2019

1 Präsenzaufgaben

Die folgenden Aufgaben werden gemeinsam im Tutorium bearbeitet.

Aufgabe 1:

Eine Nachricht mit einer Nutzdatenlänge von $p = 10000$ Bits soll von A über B nach C verschickt werden (siehe Grafik). Jede der beiden Verbindungen hat eine Datenrate von $r = 100$ kbps (1 kbps=1000 bits per second) und eine Verzögerung von $d = 10$ ms. Es treten keine weiteren Verzögerungen auf. Zum Versenden muss jedes Paket mit einem Header von $h = 100$ Bits versehen werden.



- (a) Wie lange benötigt die Nachricht vom Beginn des Versendens bei A bis sie komplett bei C angekommen ist, wenn sie in einem Paket geschickt wird?
- (b) Wie lange benötigt die Nachricht vom Beginn des Versendens bei A bis sie komplett bei C angekommen ist, wenn sie in 5 Paketen geschickt wird?
- (c) Stellen Sie bitte die **symbolische** Formel für die Gesamtverzögerung $T(n)$ in Abhängigkeit von der Anzahl der Pakete n und mit den Parametern p , h , r und d auf (keine Zahlenwerte!).
- (d) Begründen Sie bitte **kurz** warum es **nicht** sinnvoll ist, die Nachricht in sehr viele Pakete aufzuteilen!

- (e) In wie viele Pakete sollte man die Nachricht im Allgemeinen (symbolisch rechnen) aufteilen, damit die Gesamtverzögerung minimal wird? Welche Paketanzahl ergibt sich daraus für die oben angegebene Nutzdaten- und Header-Länge?

Lösung:

(a) $T = 2d + 2\frac{p+h}{r} = 222 \text{ ms}$

(b) $s = h + p/5 = 2100 \text{ bits}$, $T = 2d + 6\frac{s}{r} = 146 \text{ ms}$

(c) $T(n) = 2d + (n+1)\frac{h+p/n}{r}$

(d) Zur eigentlichen Payload kommt pro Paket noch der Header hinzu.

(e) $\frac{dT(n)}{dn} = \frac{1}{r}(h - \frac{p}{n^2})$ Nach gleichsetzen mit 0 ergibt sich: $n = \sqrt{\frac{p}{h}} = 10$

Aufgabe 2:

Die Webseite <https://iperf.fr/iperf-servers.php> listet Server auf, die einen kostenlosen Service zur Durchsatzmessung anbieten.

- (a) Wählen Sie einen der Server aus und schätzen Sie zunächst Roundtrip-Time anhand des angegebenen Standorts ab. Gehen Sie dabei davon aus, dass sich die Signale durchschnittlich mit 200.000 km/s ausbreiten.
- (b) Verwenden Sie nun das Kommandozeilenprogramm `ping` um die Roundtrip-Time (RTT) zwischen Ihrem Rechner und dem gewählten Server zu messen. Versuchen Sie die Unterschiede zu Ihrer Schätzung zu erklären.
- (c) Verwenden Sie das Kommando `iperf3 -c [HOSTNAME]` um mit dem `iperf3` Programm den Durchsatz von ihrem Rechner zu dem oben gewählten Server über Streaming-Sockets zu messen.
- (d) Berechnen Sie, wie viele Daten bei einer Streaming-Socket-Verbindung zwischen ihrem Rechner und dem oben genannten Server durchschnittlich zu jeder Zeit unterwegs sind.
- (e) Wo sind diese Daten gespeichert?

Lösung:

(a) Beispiel: `iperf.he.net`

Entfernung Freemont - Berlin = ca. 9100 km

$$2 \cdot \frac{9100 \text{ km}}{200.000 \text{ km/s}} = 0,091 \text{ s} = 91 \text{ ms RTT}$$

(b) RTT TU ca.: 150ms

Mögliche Antworten zur Differenz:

1. Pakete nehmen vermutlich nicht den direkten Weg
2. Pakete werden auf dem Weg (vor allem über den Atlantik) gequeued

(c) Durchsatz TU ca.: 123 Mbit/sec

$$(d) \ d * r = 123 \text{ Mbit/s} * \frac{0,150 \text{ s}}{2} = 9.225 \text{ kbit}$$

(e) Diese Daten sind gerade unterwegs, also entweder gespeichert auf:

1. Der Leitung.
2. In einer Warteschlange in einem Router.

Aufgabe 3:

Das Programm `netcat` (kurz `nc`) ist ein einfaches Netzwerkprogramm, das eine Verbindung zu einem entfernten Server aufbaut und die Eingabe von `stdin` über das Netzwerk an den entfernten Server sendet. Das Programm kann auch als Server fungieren, wobei auf eingehende Verbindungen gewartet wird und alle empfangenen Daten nach `stdout` geschrieben werden. Netcat ist deshalb gut geeignet, um die eigene Implementierung von einfachen Netzwerkprotokollen zu testen. Probieren Sie deshalb die folgenden Anwendungsbeispiele aus:

(a) Benutzen Sie das `nc` Kommandozeilen-Programm und rufen Sie eine Webseite auf der Konsole ab. Beispiel:

```
$ echo -e "GET / HTTP/1.1\r\nHost:whatismyip.akamai.com\r\n" | \
nc whatismyip.akamai.com 80
```

- (b) Benutzen Sie das `nc` Kommandozeilen-Programm um einen Server anzubieten.

Zum Beispiel:

```
$ nc -l 4711
```

Sie können sich dann mit `netcat` von einem anderen Konsolenfenster zum Server verbinden:

```
$ echo "Hello World" | nc localhost 4711
```

- (c) Benutzen Sie das Programm `wireshark` um die Kommunikation aus den vorherigen Teilaufgaben nachzuverfolgen.

2 Praxisaufgaben

Im diesem Block gibt es noch keine klassischen Praxisaufgaben zur Abgabe. Die folgenden Aufgaben dienen lediglich dazu sich an den *Workflow* des Praktikums zu gewöhnen und sich auf die kommenden Wochen vorzubereiten.

Aufgabe 4:

Sämtliche Programmieraufgaben des Praktikums sind in der Programmiersprache C auszuführen. Wir schreiben Ihnen nicht vor mit welchen Werkzeugen Sie arbeiten sollen, aber die Erfahrung zeigt, dass gerade Fehler aus Nachlässigkeit/Unaufmerksamkeit durch moderne IDEs (Integrated Development Environments) verringert werden können.

- (a) Richten Sie eine IDE Ihrer Wahl auf Ihrem PC (vorzugsweise unter Linux) ein.
- Falls Sie an ihrem Rechner arbeiten, können Sie bspw. *CLion* benutzen.¹
 - Auf den Arbeitsplatzrechner in den Praktikumsräumen ist *KDevelop* bereits vorinstalliert.

- (b) Erstellen Sie ein neues Projekt namens "*Block1*"

- i. Ihr Projekt sollte folgende Struktur aufweisen:

```
Block1
+--- build #Verzeichnis
+--- CMakeLists.txt
+--- main.c
```

- ii. Schreiben Sie in der `main.c` ein einfaches *Hello-World* Programm, kompilieren Sie es und führen Sie es aus. Achten Sie dabei darauf nicht versehentlich C++ Compiler zu benutzen oder C++ Header einzubinden.
- (c) Setzen Sie an einer beliebigen Stelle Ihres Codes einen Breakpoint und vergewissern Sie sich, dass dieser beim ausführen auch wirklich ausgelöst wird. Probieren Sie aus, wie Sie Ihr Programm schrittweise ausführen können, sobald ein

¹Das Programm ist leider nicht quelloffen, aber als TU-Student erhalten Sie über Ihre Uni-Email eine kostenlose Lizenz.

Breakpoint ausgelöst hat bzw. wie Sie die aktuellen Variablenwerte angezeigt bekommen.

(d) Erstellen Sie die zwei Unterverzeichnisse `src` und `include`.

- i. Erzeugen Sie unter `src` die Datei `foo.c` und den entsprechenden Header `foo.h` im Verzeichnis `include`.
- ii. Schreiben Sie eine beliebige Funktion in `foo.c` und definieren Sie die Signatur `foo.h`. Binden Sie dabei auch den Header mit in der Quelldatei ein.
- iii. Binden Sie `foo.h` in `main.c` ein und versuchen Sie Ihre neue Funktion dort aufzurufen. Damit dies funktionieren kann muss ihre IDE wissen wo Sie `foo.h` finden kann. Wenn Sie dies erreicht haben, sollten auch die Auto-Vervollständigungsfeatures der IDE Ihre Funktion samt Signatur erkennen.

3 Vertiefungsaufgaben

Die folgenden Aufgaben sind zur selbstständigen Vertiefung des vermittelten Stoffs gedacht. Die Lösung der Aufgaben ist freiwillig, der Stoff ist trotzdem prüfungsrelevant.

Aufgabe 5:

Eine wesentliche Eigenschaft von Kommunikationsnetzen ist, dass Daten verzögert übertragen werden. Die Verzögerung eines Datenpakets setzt sich aus verschiedenen Teilen zusammen. Erläutern Sie bitte kurz die folgenden Begriffe:

- (a) Propagation Delay (Ausbreitungsverzögerung)
- (b) Transmission Delay (Übertragungsverzögerung)
- (c) Processing Delay (Verarbeitungsverzögerung)
- (d) Queueing Delay (Warteschlangenverzögerung)

Lösung:

- (a) Durch die endliche Ausbreitungsgeschwindigkeit von Signalen (maximal die Vakuumlichtgeschwindigkeit, in der Praxis meist nur 200 000 km/s) kommt ein Signal nicht sofort nach dem Absenden am Empfänger an, sondern wird abhängig von der Länge des Weges verzögert. Die Ausbreitungsverzögerung gibt an, wie lange der gleiche Teil des Signals/der Nachricht (z.B. Beginn des ersten Bits) vom Sender zum Empfänger benötigt. Sie ist nur abhängig von der Länge des Weges und der Ausbreitungsgeschwindigkeit ($T_p = d/v$).
- (b) Die einzelnen Bits einer Nachricht werden i.d.R. seriell, d.h. nacheinander übertragen. Jedes Bit wird dabei durch ein physikalisches Signal einer gewissen zeitlichen Länge repräsentiert (z.B. Spannungsverlauf). Aus physikalischen Gründen kann die Länge dieses Signals nicht beliebig klein gewählt werden. Ein Übertragungskanal kann daher nur eine begrenzte Anzahl an Datenbits pro Zeiteinheit übertragen; wie viele wird durch die Übertragungsrate angegeben. Die Übertragungsverzögerung ist nun die Zeit, die benötigt wird,

alle Bits eines Pakets auf die Übertragungsleitung zu schicken. Sie ist nur abhängig von der Länge des Pakets und der Übertragungsrate ($T_t = l/R$). Sie ist insbesondere **nicht** von der Länge der Leitung abhängig.

- (c) Der Sender braucht ggf. Zeit ein Paket zusammenzubauen, der Empfänger um es zu interpretieren und eine passende Antwort zu generieren.
- (d) Es kann immer nur ein Paket gesendet werden. Sollen mehrere Pakete verschickt werden, so muss dies nacheinander geschehen. Zu diesem Zweck werden die Pakete z.B. in eine Warteschlange eingereiht und nacheinander abgearbeitet. Die Zeit, die die Pakete in der Warteschlange verbringen, nennt man Queuing Delay.

Aufgabe 6:

Nehmen Sie an, zwei Hosts, A und B, sind 20000 km voneinander entfernt und werden durch eine direkte Verbindung mit $r = 2\text{Mbit/s}$ verbunden. Angenommen, die Ausbreitungsgeschwindigkeit über den Link sei 200000 km/s.

- (a) Berechnen Sie das Bandbreiten-Delay-Produkt, $r * d$.
- (b) Betrachten Sie die Übertragung einer Datei von 800000 Bits von Host A zu Host B. Angenommen, die Datei wird kontinuierlich als eine große Nachricht gesendet. Was ist die maximale Anzahl von Bits, die zu einem bestimmten Zeitpunkt jeweils unterwegs sind?
- (c) Geben Sie eine Interpretation des Bandbreiten-Delay-Produkts.
- (d) Leiten Sie einen allgemeinen Ausdruck für die Breite eines Bits in Bezug auf die Ausbreitungsgeschwindigkeit, die Übertragungsrate r und die Länge der Verbindung m her.

Lösung:

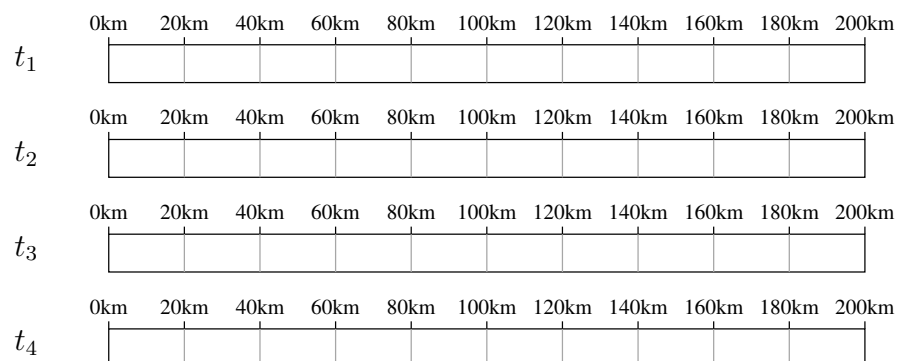
- (a) $0,2\text{Mbit} = 200\text{kbit} = 200000\text{bit}$
- (b) 200000bit

- (c) Die Bits, die gerade zwischen den Host unterwegs sind bzw. auf der Leitung.
- (d) Allgemein: $w = v/r$ (w = Bitbreite, v = Ausbreitungsgeschwindigkeit, r = Bandbreite) (Ergebnis in Aufgabe: 100m)

Aufgabe 7:

Zwei Rechner seien durch eine Leitung von 200 km Länge miteinander verbunden. Vom Sender sollen Pakete mit einer Größe von 10 000 Bits bei einer Übertragungsrate von 100 Mbps ($1 \text{ Mbps} = 10^6 \text{ bits per second}$) übertragen werden. Processing und Queueing Delay seien vernachlässigbar klein. Zum Zeitpunkt $t_0 = 0 \mu\text{s}$ wird damit begonnen, ein einzelnes Paket zu senden. Die Ausbreitungsgeschwindigkeit der Signale beträgt $v = 200\,000 \text{ km/s}$.

- (a) Wann beendet der Sender die Übertragung dieses Pakets (t_1)?
- (b) Wann kommt das erste Bit am Empfänger an (t_3), wann das letzte (t_4)?
- (c) Zeichnen Sie in folgendes Diagramm die Position des Paketes zu denen von Ihnen berechneten Zeitpunkten sowie zum Zeitpunkt $t_2 = 500 \mu\text{s}$ ein, indem Sie den Bereich vom ersten bis zum letzten Bit des Pakets schraffieren.

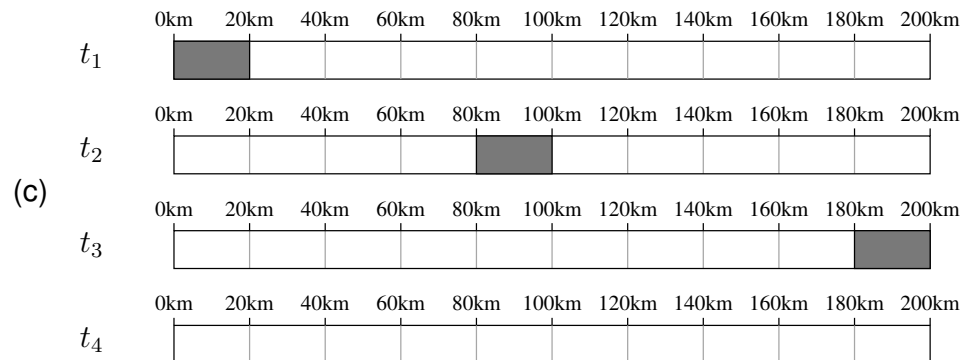


- (d) Wie viele vollständige Pakete können sich gleichzeitig auf dem Übertragungskanal befinden?
- (e) Wann käme das letzte Bit am Empfänger an, wenn die Übertragungsrate 10 Mbps beträgt? Was ergibt sich für eine Leitungslänge von 20 000 km und Übertragungsraten von 10 Mbps und 100 Mbps?

Lösung:

(a) $t_1 = 10\,000 \text{ bits} / 100 \text{ Mbps} = 100 \mu\text{s}$

(b) $t_3 = 200 \text{ km} / 200\,000 \text{ km/s} = 1\,000 \mu\text{s}$, $t_4 = t_1 + t_3 = 1\,100 \mu\text{s}$



(d) $N = t_3 / t_1 = 10$

(e) $t_{4b} = 2 \text{ ms}$, $t_{4c} = 101 \text{ ms}$, $t_{4d} = 100,1 \text{ ms}$,