

Theorie 2

Fachgruppe Telekommunikationsnetze (TKN)

10. November 2023

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

Beantworten Sie folgende Fragen rund um den TCP/IP Stack:

1. Welche Layer gibt es im TCP/IP Modell?
2. Angenommen Sie öffnen einen Browser und senden eine Web-Anfrage (HTTP) über TCP über IP über Ethernet. In welcher Reihenfolge werden die Header gesendet?
3. Angenommen Sie haben zwei Browser gleichzeitig geöffnet und rufen vom gleichen Server gleichzeitig über HTTP das selbe Dokument ab.
 - a) Wie unterscheidet der Server zwischen den beiden Anwendungen?
 - b) Was bringt das *Hourglass Modell* zum Ausdruck?

Übung 2

Die Zuordnung von Domainnamen zu IP-Adressen erfolgt im Internet mit Hilfe des Domain Name Systems (DNS). Beschreiben Sie anhand der folgende Fragen den Aufbau und die Funktionsweise von DNS:

1. Was ist der Unterschied zwischen einem Namen und einer Adresse?
2. Wie sind die Verantwortlichkeiten im Namenssystem aufgeteilt?
3. Wie funktioniert das rekursive bzw. iterative Auflösen von Namen mit DNS?
4. Welche Funktion erfüllen die TTL-Einträge?
5. Überprüfen Sie mittels des Kommandozeilen-Tools `dig` (Linux/OS X) bzw. `nslookup` (Windows), welche DNS-Server für die Domain `tu.berlin` sowie deren Subdomains zuständig sind und welche TTL diese Einträge haben.

Übung 3

Diskutieren Sie Vor- und Nachteile der Client-Server-Architektur, auch im Vergleich zu Peer-to-Peer Systemen.

Übung 4

Nutzen Sie das Beispiel des Abrufs von E-Mails durch Clients bei einem Server, um die folgenden Fragen für das Client-Server-Prinzip im Allgemeinen zu beantworten:

1. Muss der betreffende Rechner (Client/Server) immer angeschaltet und ans Netz angeschlossen sein? Was hätte es für Konsequenzen, wenn beide Kommunikationspartner gleichwertig sind, also es keinen expliziten Server bzw. Client gibt?
2. Welche Art von Adresse haben Client und Server? Was hätte es für Konsequenzen, wenn beide Kommunikationspartner gleichwertig sind, also es keinen expliziten Server bzw. Client gibt?
3. Mit wem kommunizieren Clients in der Regel direkt? Mit wem der Server?

Übung 5

Welche Protokolle verwenden Portnummern? Welchem Zweck dienen Ports dabei?

Übung 6

Angenommen Sie möchten ein Programm schreiben, das eine Funktion auf einem entfernten Server per Socket-API aufruft und dabei als Argument ein Paar aus ID und Namen übergibt. Das Programm speichert das Paar intern wie hier dargestellt:

```
1 |     typedef struct ValueType {  
2 |         int      id;  
3 |         const char* name;  
4 |     } ValueType;
```

1. Kann der dargestellte **struct** ValueType direkt an `send()` übergeben werden? Spielt die Prozessorarchitektur hierbei eine Rolle?
2. Funktionsaufrufe auf entfernten Systemen werden oft als sog. Web Services über HTTP ausgeführt (siehe REST). Wie löst HTTP die Probleme aus der ersten Teilaufgabe?
3. Wie können komplexe Datenstrukturen übertragen werden, bspw. verkettete Listen oder Binärbäume?

Übung 7

Beantworten Sie im Kontext vom Hypertext Transfer Protocol (HTTP) folgende Fragen:

1. Welche HTTP Methoden gibt es und was machen diese?
2. Welche HTTP Methoden sind idempotent?
3. Was sind persistente und nicht-persistente Verbindungen?
4. HTTP ist ein zustandsloses Protokoll. Was bedeutet das? Mit welcher Technik können Server z.B. trotzdem den Warenkorb einer Nutzers speichern?

Übung 8

Gegeben sei ein RESTful Webservice, der Teilnehmer in einer Veranstaltung verwaltet. Der Service läuft auf dem Rechner mit dem DNS-Namen `api.tkn.tu-berlin.de`. Folgendes Verhalten ist bereits vorgegeben und implementiert, dabei sind `<courseName>` und `<student>` jeweils Platzhalter für die jeweilige Veranstaltung bzw. den Teilnehmer:

Ressource	Methode	Verhalten
<code>/courses/<courseName></code>	GET	gibt Liste der Teilnehmer zurück
<code>/courses/<courseName></code>	DELETE	Löscht die Veranstaltung
<code>/courses/<courseName>/<student></code>	GET	Gibt bestimmten Teilnehmer zurück
<code>/courses/<courseName>/<student></code>	DELETE	Löscht Teilnehmer

1. Ist eine der Ressourcen eine Collection-URL und falls ja, geben sie die komplette URL an (inkl. verwendender Zugriffsmethode und Hostnamen)?
2. Der Webservice soll von Ihnen nun um die Funktionalität erweitert werden, die Klausurnote eines Teilnehmers zu ändern. Welche HTTP-Methode muss dazu auf welche URL aufgerufen werden?
3. Was wäre allgemein der Inhalt (Payload) der Anfrage aus Teilaufg. 2.
4. Der Webservice soll von Ihnen nun um die Funktionalität erweitert werden, einen neuen Teilnehmer zur Veranstaltung hinzuzufügen, welcher vorher noch nicht angemeldet war. Welche HTTP-Methode muss dazu auf welche URL aufgerufen werden?

Übung 9

In dieser Aufgabe versuchen wir nachzuvollziehen wie ein Aufruf von YouTube abläuft und wie das verwendete CDN arbeitet. Rufen Sie dazu ein YouTube Video auf, während Sie den Netzwerktraffic mit Wireshark mitschneiden.

Hinweis: Durch gecachten Inhalt kann es sein, dass sie nicht die volle Anfrage sehen, falls Sie die Website bereits besucht haben. Um dies zu vermeiden starten Sie gegebenenfalls das System vor dem Test neu. Eure Teampartner erhalten unter Umständen andere Ergebnisse, vergleicht diese.

1. Welche DNS Anfragen werden ausgelöst? Welcher der Hosts liefert vermutlich die eigentliche Website aus? Welcher Host stellt die Videodaten bereit?
2. Wieso werden mehr Anfragen als nur `youtube.com` gestellt?
3. Welche CDN Mechanismen werden verwendet?
4. In der Vorlesung wurde besprochen, dass große Datenmengen (wie z.B. Videos) möglichst nah beim Nutzer gespeichert werden um Netzwerkkapazität zu sparen. Versuchen Sie herauszufinden, wo sich der Server befindet, der diese Daten ausliefert.¹ Überprüfen Sie ob dieser Standort plausibel ist, beispielsweise indem Sie die Latenz via `ping` testen.

¹Bspw. über dieses Online-Tool: <https://www.ip-tracker.org/locator/ip-lookup.php>