technische universität dortmund

Fakultät für Informatik
Lehrstühle 4, 11, 12

Alexander Lochmann, Nils Kriege
Nils Dunker, Felix Homa, Simon Koschel, Sebastian Lau, Benedikt Maus

Wintersemester 2018/19

Rechnernetze und verteilte Systeme
Übungsblatt 3

Ausgabe: 23. Oktober 2018 **Besprechung:** 30. Oktober – 2. November 2018

Allgemeine Informationen zu Feiertagen

Bitte denken Sie daran, dass Donnerstag, der 1. November, ein Feiertag ist und daher an diesem Tag keine Übungen stattfinden. Wenn Ihre Übungsgruppe hiervon betroffen ist, haben Sie die Möglichkeit einmalig in eine andere Gruppe Ihrer Wahl zu wechseln.

Quizfragen

1. Kann es bei Peer-to-Peer zentrale Server geben? Wenn ja, warum?
2. Mehrere Nutzer greifen gleichzeitig mithilfe eines herkömmlichen Browsers auf die Website der TU Dortmund zu. Welche Art von Architektur wird dafür verwendet?
3. A. Nonymous schaut sich bei der beliebten Videoplattform Duröhre Einhornvideos in 4K-Auflösung an. Ist die Nachrichtenlaufzeit oder Bandbreite wichtiger für ihn?
4. Welche Gründe gibt es für Signalverzögerung in Paketnetzen?

Aufgabe 3.1 (2 Vortragspunkte)

a) Grenzen Sie die Begriffe *Schicht*, *Dienst* und *Protokoll* ab.

b) Geben Sie für den TCP/IP-Protokollstack beispielhaft Protokolle der einzelnen Schichten an. Welche Dienste werden zur Verfügung gestellt?

c) Nennen und vergleichen Sie drei Datenübertragungsmedien.

Aufgabe 3.2 (2 Vortragspunkte)

a) Erläutern und vergleichen Sie das Client/Server- und Peer-to-Peer-Paradigma. Nennen Sie beispielhafte Anwendungen für beide Paradigmen.

b) Grenzen Sie die Begriffe *Prozess*, *Socket*, *IP-Adresse* und *Port* ab. Geben Sie für mindestens drei Anwendungen die Protokolle der Anwendungs- und Transportschicht sowie die Standard-Ports an.

c) Erläutern Sie die Socket-Systemaufrufe bei TCP.

Aufgabe 3.3 Programmierung mit Sockets in Java

Erstellen Sie ein kurzes Java-Programmbeispiel aus den Klassen `Server` und `Client`. Sie können davon ausgehen, dass Server und Client parallel laufen (Es müssen also keine Threads o. ä. beachtet werden). Als Zieladresse dürfen Sie <*Server*> und <*Client*> verwenden, weitere Variablen wie Ports o. ä. dürfen ebenfalls als Wort geschrieben werden.

Der Client soll eine Verbindung zum Server aufbauen. Der Server soll diese Verbindung annehmen und den Text "Hello World!" an den Client in *UTF-8* Kodierung senden. Der Client soll diesen lesen und danach die Verbindung beenden.

Sie sollten in der Lage sein, die Methodenaufrufe zu erklären. Diese Aufgabe dient zur Vorbereitung auf die Programmieraufgabe. Hinweis: Verwende Sie `java.net.ServerSocket` und `java.net.Socket`, die Fehlerbehandlung darf vernachlässigt werden.