

Vernetzte Systeme, Übungsstunde 3: Hinweise

Gabor Cselle, gabor@student.ethz.ch, 22. April 2005, HG F 26.3

Interaktiv

Ich habe Übung 3 gelöst in ...:

- A. Java
- B. Eiffel
- C. Mir war beides zu einfach und ich habe Übung 3 ausgelassen.

Nachbesprechung Übung 1

Allgemein gut gemacht, auch interessante Ansätze, z.B. POP3-ähnliches Protokoll.

Muster, nach den man vorgehen sollte:

1. Überlegen, welche Anforderungen Protokoll erfüllen sollte. (Für die Übung zwar nicht notwendig, aber man lernt ja für das Leben).

- stateful / stateless?
- Datenmenge abschätzen
- Befehlsformat: sollten Kommandos Human-Readable sein
- Kann meine Leitung abgehört werden? → Verschlüsselung
- Kann es evtl. bösartige Teilnehmer geben?

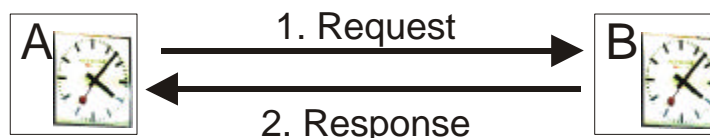
2. Protokollnachrichten, Grund des Sendens *und Reaktionen darauf* spezifizieren - Beispiel:

Nachrichtenformat	Wann gesendet?	Reaktion Zentralrechner
WITHDRAW <amount>	Kunde eingeloggt, wählt "Abheben" im Menü.	Sende OK, falls genügend Geld auf Konto, sonst ERROR.

3. Erst jetzt Beispielabläufe zeigen!

Nachbesprechung Übung 2

1a1) Wieso ist RTT einfacher zu messen als die Verzögerung?



Bei der RTT-Messung kann man die Zeit des Absendens und der Ankunft auf derselben Uhr messen. Für die Messung der Verzögerung müsste man auf A und B messen, und die Uhren müssten perfekt synchron sein.

1a2) Wieso ist RTT nicht 2x Verzögerung?

- Empfänger muss Paket verarbeiten, bevor er Antwort sendet.
- Hin- und Rückweg evtl. auf verschiedenen Routen.

1b1) Bestimmung der Bandbreite mit Ping: Herleitung.

$$T = 2 \left(\frac{L}{R} + V \right)$$

L: Grösse des Pakets

R: Bandbreite

V: Andere Verzögerungen

Faktor 2: Paket propagiert zum Ping-Ziel und zurück

Zwei verschiedene Messungen machen – man erhält:

$$T_1 = 2 \left(\frac{L_1}{R} + V \right)$$

$$T_2 = 2 \left(\frac{L_2}{R} + V \right)$$

Jetzt nach V auflösen:

$$V = \frac{T_1}{2} - \frac{L_1}{R}$$

$$V = \frac{T_2}{2} - \frac{L_2}{R}$$

Gleichsetzen:

$$\frac{T_1}{2} - \frac{L_1}{R} = \frac{T_2}{2} - \frac{L_2}{R}$$

Und nach R auflösen:

$$R = 2 \frac{L_1 - L_2}{T_1 - T_2}$$

So kann man nun aus 2 Messungen mit verschiedenen Paketgrössen die Bandbreite abschätzen.

- Jemand fleissiges hat viele Messungen gemacht, aus deren Kombination jeweils R bestimmt und gemittelt.
- Und es gab auch noch eine Least-Squares-Lösung:

$$2LR^{-1} + 2V = T$$

$$\begin{bmatrix} 2L_1 & 2 \\ 2L_2 & 2 \\ 2L_3 & 2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} R^{-1} & V \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$

1c) Wie funktioniert Traceroute wirklich?

1. Traceroute sendet UDP-Paket an beliebigen Port des Ziel-Rechners, Time-to-live=1
2. Wenn Rechner nach 1 Hop "nett" ist, sendet dieser ein ICMP (Internet Control Message Protocol) Paket zurück, dass TTL exceeded wurde. Wenn er nicht "nett" ist, schickt er kein Paket → in Traceroute erscheinen * * * Sternchen und keine IP für den Hop.
3. Traceroute erhöht TTL so lange, bis ICMP-Antwort von Ziel-Rechner kommt.

Manchmal gibt es auf dem Weg zum Zielrechner eine Firewall, der alle ICMP-Pakete ausfiltert. Deshalb gibt es manchmal das Phänomen, dass man ab einem gewissen Hop nur noch Sternchen und Timeouts bekommt.

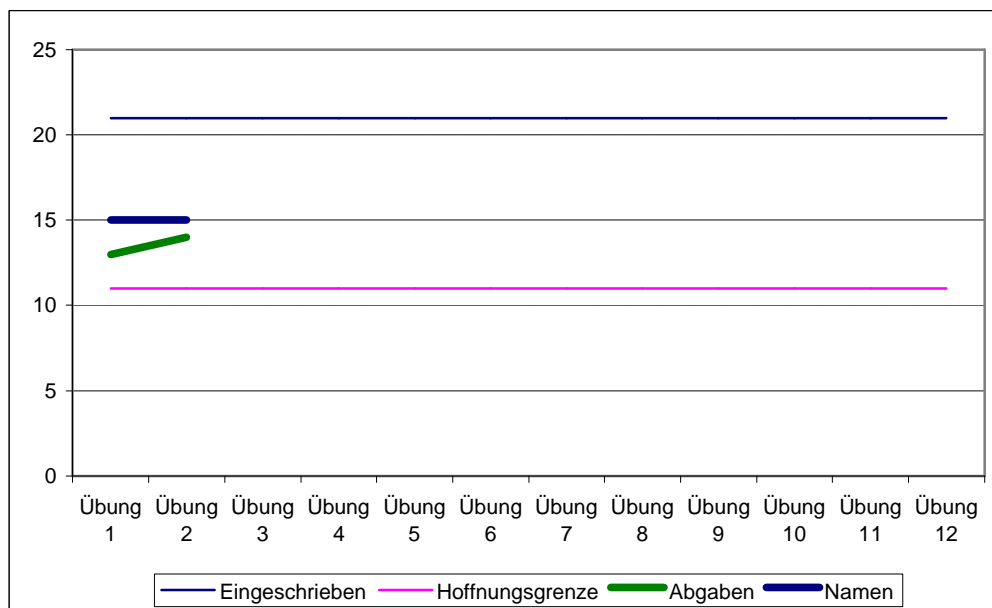
1c2) Gewinner ist Christian Regg, der für `traceroute www.abc.net.au` erst nach 28 Hops von einer Firewall (s.o.) ausgeblockt wird.

2) Diese Aufgabe wurde gut gelöst. Probleme nur bei:

2b) (Enorme Datenmenge transferieren). Hier ausrechnen, wie viel das Glasfaserkabel übertragen kann, während die 747 *einen* Flug macht. Wenn man mehr als das zu transferieren hat, ist die 747 schneller.

2c) (Antwort schneller von Schiff oder aus New York?) Oft wurde hier nur die Zeit für die Verzögerung eines Weges berechnet. Bis man die Antwort hat, muss aber eine Nachricht hin und eine zurückfließen.

Statistik Abgaben



Hinweise Übung 3

Allgemeine Bemerkungen:

- Bitte den gesamten Quellcode anschauen, nicht nur Teil, den man ausfüllen muss. (Das GUI-Zeug in Java braucht ihr euch natürlich nicht anzusehen.)
- Eiffel: Hier muss man zusätzlichen Clusters (Library) in der Eiffel-Entwicklungsumgebung einbinden. Auf der Vorlesungs-Website gibt es eine Anleitung.

Aufgabe 1: Java-Tipps

1a) `sendMessage()` implementieren: Was muss man machen?

1. Notwendige Informationen zusammentragen:
 - a. Länge des Messagetextes?
 - b. IP-Adresse und Port des Empfängers
2. Byte-Array für UDP-Paketinhalt erstellen, entsprechend Protokolldefinition füllen.
3. `DatagramPacket`-Objekt erstellen mit dem vorbereiteten Inhalt und Empfänger-IP, Port von oben.

Siehe auch Folien 2/64 und 2/65 im Skript.

1b) `receiveMessage()` implementieren: Was muss man machen?

1. Neues `DatagramPacket`-Objekt erstellen, mit `socket.receive(packet)` auf Empfang warten.
2. Message auseinandernehmen.
3. Mit `InstantMessenger.IM.getUser(...)` herausfinden, von welchem User die Message kam.
4. Mit `InstantMessenger.IM.displayMessage()` die Nachricht anzeigen.

Siehe auch Folien 2/66 und 2/67 im Skript.

Aufgabe 2: Flussteuerung

2a) Wenn man Verzögerung V mit Bandbreite R multipliziert, welche Grösse bekommt man?

2b+c) Hier muss man einfach nur überlegen.