

# Rechnernetze-Praktikum

## WS 2003/04

Jürgen Vogel

Lehrstuhl für Praktische Informatik IV

Universität Mannheim

# Prüfung

mündliche Prüfung

- ca. 15 Minuten
- Prüfer: Prof. Effelsberg, Beisitzer: J. Vogel
- Anmeldung im Studienbüro
- schriftliche Vereinbarung des Termins mit LS PI IV
- (ebenso schriftliche Absage)

**Termin: 19.02.03**  
**14:00 + n \* 30 min**

**Anmeldung: ab sofort bei mir**

# Fünfte Aufgabe

## TCP-Verbindungsmanagement

- Verbindungsabbau realisieren

## Datenübertragung

- Paketstrom wie im Eventfile definiert senden
- Empfänger versendet TCP-ACK's, verzögert um x ms (200 ms)
- Gesendete Pakete verbleiben bis zur Bestätigung in der Sende-Queue

## Code-Abgleich

- Trace-Files mit gegebenen Beispiel-Szenarien abgleichen:  
network02.ns, events01.ns

# Sechste Aufgabe (1)

## Bestätigungen (ACKs)

- Bestätigung um 200 ms verzögern
- Nur ein globaler ACK-Timer zu jeder Zeit
- Jedes x-te Paket sofort bestätigen (meist  $x = 2$ )
- Das höchste lückenlose Segment mit ACK bestätigen

## Paketverluste

- Sender stellt adaptiven Retransmit-Timer für jedes gesendete Paket
- Wenn dieser abläuft, Go-Back-N mit Pufferung

# Retransmit-Timer (1)

## Exponential Backoff

- $\text{timer} = 2^i * \text{RTO}$  ,  $i = 1, 2, \dots$

## RTT Measurement

- Match ACK and data segment (wait for ACK including SN of data)
- Only one measurement at a time
- Measure RTT only for data that was not retransmitted (Karn)

## 1) Exponential Smoothing

$$\text{RTT}_n = \alpha \text{RTT}_{n-1} + (1 - \alpha)M$$

$$\text{RTO} = \beta \text{RTT}_n$$

$$\alpha = 0.9 \quad \beta = 2$$

M = current measurement

## Retransmit-Timer (2)

### 2) Jacobson's Algorithm

$$\text{Err} = M - A_n$$

$$A_n = A_{n-1} + g \text{ Err}$$

$$D_n = D_{n-1} + h ( |\text{Err}| - D_{n-1} )$$

$$\text{RTO} = A_n + 4 D_n$$

$$g = 0.125$$

$$h = 0.25$$

A estimates average RTT

D estimates mean deviation

Initialization:

1) On startup:  $A_0 = 0, D_0 = 3 \text{ s}$

$$\Rightarrow \text{RTO}_0 = A_0 + 2 D_0 = 6 \text{ s}$$

2) First M:  $A_1 = M_0 + 0.5, D_1 = A_1/2$

$$\Rightarrow \text{RTO}_1 = A_1 + 4 D_1 = 3 A_1$$

## Sechste Aufgabe (2)

### Fast Retransmit

- Sofortige Bestätigung, wenn unerwartete Segmente eintreffen
- Empfänger bestätigt eintreffende Segmente weiterhin mit der höchsten empfangenen SN
- nach dem 3.ten Duplicate ACK sofortige Übertragungswiederholung

### Paketverluste beim Verbindungsmanagement

### Messung

- Vergleich Laufzeit Retransmit-Timer:  
exponentielle Glättung vs. Jacobson's Algorithmus  
verschiedene Szenarien auf Basis von network02.ns