
Leistungsnachweis in
Rechnerkommunikation

Sommersemester 2012

01. Oktober 2012

Name: _____

Matrikelnummer: _____

Geburtsdatum: _____

Studienfach: _____

Fachsemester: _____

Extra Nachweis: benoteter Schein unbenoteter Schein

- Bitte verwenden Sie einen blauen oder schwarzen Kugelschreiber (kein rot, keinen Bleistift).
- Schriftliche Aufzeichnungen (sowohl eigene Aufzeichnungen wie auch Bücher) sind als Hilfsmittel zugelassen. Auch ein Taschenrechner ist erlaubt und hilfreich. Nicht zugelassen sind dagegen Computer, PDAs, Mobiltelefone und sonstige Kommunikationsmittel.
- Legen Sie den Ausweis (mit Lichtbild) griffbereit auf den Platz.
- Bitte überprüfen Sie, ob Sie alle 16 Blätter erhalten haben.
- Schreiben Sie die Antworten jeweils in den freien Raum hinter den Fragen. Sollte dieser nicht ausreichen, steht noch freier Raum am Ende der Klausur zur Verfügung. Bitte kennzeichnen Sie dort deutlich, welche Aufgabe Sie bearbeiten. Gesondert beigelegte Blätter werden nicht bewertet!
- Schreiben Sie deutlich! Unleserliche Antworten gehen nicht in die Bewertung ein!
- Die Arbeitskopien können der Klausur entnommen werden und müssen nicht mit abgegeben werden.

Ich habe die Hinweise auf dieser Seite zur Kenntnis genommen und alle 16 Blätter der Klausur empfangen:

Unterschrift

Bewertung:	1	2	3	4	Σ

1 Allgemeine Fragen (21 Punkte)

1.1 Leitungskodierung (7 Punkte)

Geben Sie die in Abbildung 1 codierten Bitsequenzen an. Fehlerhafte Bits markieren Sie mit einem X.

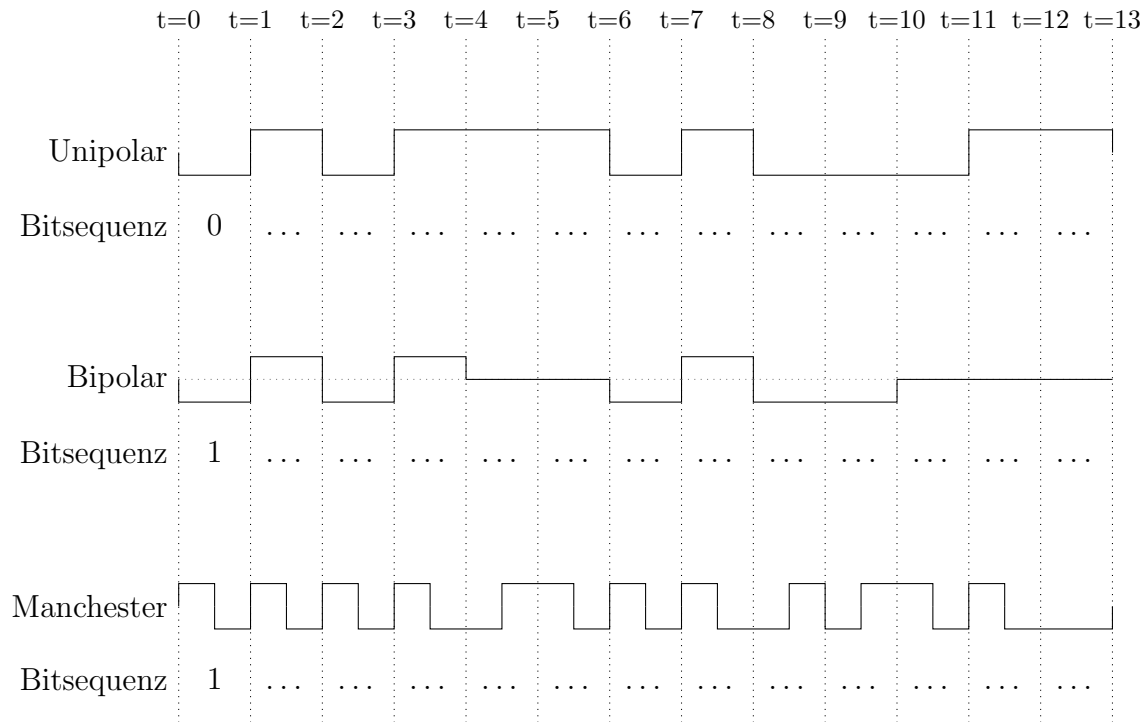


Abbildung 1: Leitungskodierungen

Nennen Sie 2 mögliche Fehlerquellen:

1.2 Datenaustausch (8 Punkte)

Gegeben sind zwei Hosts A und B, die über einen Link mit einer Bitrate $R = 1 \text{ Gbit s}^{-1}$ und einer Länge von $l = 20\,000 \text{ km}$ verbunden sind. Nehmen Sie an, dass die Ausbreitungsgeschwindigkeit $v = 2 \times 10^8 \text{ m s}^{-1}$ beträgt.

- a) Bestimmen Sie das Produkt b aus Bitrate und Ausbreitungsverzögerung. Was sagt dieser Wert aus?

$b =$

- b) Host A möchte nun ein Objekt der Größe $O = 25 \text{ MB}$ an Host B senden. Nehmen Sie an, dass die Nachricht ohne Unterbrechung (Wartezeit) versendet wird. Wie viele Bits können sich zu einem beliebigen Zeitpunkt maximal auf der Leitung befinden? Bestimmen Sie die gesamte Übertragungszeit. Nehmen Sie vereinfachend an, dass gilt: $1 \text{ MB} = 10^6 \text{ B}$.

1.3 Adressierung (6 Punkte)

In einem Netzwerk (basierend auf TCP/IP) möchte ein Host A einen anderen Host B eine Datei übermitteln. Host A kennt die IP-Adresse und den benötigten TCP-Port von Host B.

- a) Wie kann Host A in einem lokalem Ethernet-Netzwerk (alle Hosts sind über einen Switch miteinander verbunden) die physikalische Adresse (MAC-Adresse) von Host B herausfinden? Geben Sie das zugrunde liegende Protokoll an und beschreiben Sie stichpunktartig dessen Funktionsweise. Auf welcher Schicht ist dieses Protokoll angesiedelt?

- b) Muss ein Host grundsätzlich die MAC-Adresse seines Kommunikationspartners kennen um mit ihm kommunizieren zu können? Begründen Sie ihre Antwort.

2 Latenzzeiten (21 Punkte)



Abbildung 2: Beispiel Netzwerk mit $E = 3$ Links

Abbildung 2 zeigt eine beispielhafte Netzwerktopologie, in dem sich zwischen Host A und Host B 2 weitere Knoten und somit $E = 3$ Links befinden. Die Rate jedes Links beträgt R . Im folgenden ist die Anzahl der Links E zwischen den Hosts variabel. Ein Objekt der Größe O soll von Host A auf Host B übertragen werden. Pro Paket werden L Byte Nutzdaten und zusätzlich h Byte Header übertragen.

Tipp: Store-and-Forward berücksichtigen!

2.1 Minimale Übertragungszeit (8 Punkte)

Wie muss die Größe der Nutzdaten L in Relation zu der Objektgröße O gewählt werden damit die Übertragungszeit d_{min} möglichst klein ist? Nehmen Sie vereinfachend an, dass immer gilt: $\frac{O}{L} \in \mathbb{N}$.

2.2 Maximale Übertragungszeit (4 Punkte)

Wie muss die Größe der Nutzdaten L in Relation zu der Objektgröße O gewählt werden, damit die Übertragungszeit d_{max} möglichst groß ist? (Aber: $d_{max} \neq \infty$). Vernachlässigen Sie für diese Aufgabe die h Byte Header ($h = 0$). Geben Sie eine allgemeine Formel für die Übertragungszeit d_{max} in Abhängigkeit von E und ohne L an.

2.3 TCP (9 Punkte)

Im folgenden wird nun eine TCP Verbindung zwischen den Hosts aufgebaut. Ändern Sie die in der Vorlesung hergeleitete Gleichung (1) für die Gesamtverzögerungszeit d , von dem Fall einer direkten Verbindung auf die in Abbildung 2 gezeigten Netzwerk-Topologie mit einer variablen Anzahl mit E Links um und erweitern Sie die Gleichung (1) so, dass die h Byte Header pro Paket (mit L Byte Nutzdaten) mit berücksichtigt werden. Wie verändert sich die Anzahl der Wartezeiten P (Formel für K und Q)? Gehen Sie davon aus, dass die Round-Trip-Time (RTT) als Konstante bekannt ist.

$$d = 2RTT + \frac{O}{R} + P \left(RTT + \frac{L}{R} \right) - (2^P - 1) \frac{L}{R} \quad (1)$$

3 TCP-FIN: Statechart-Programmierung (22 Punkte)

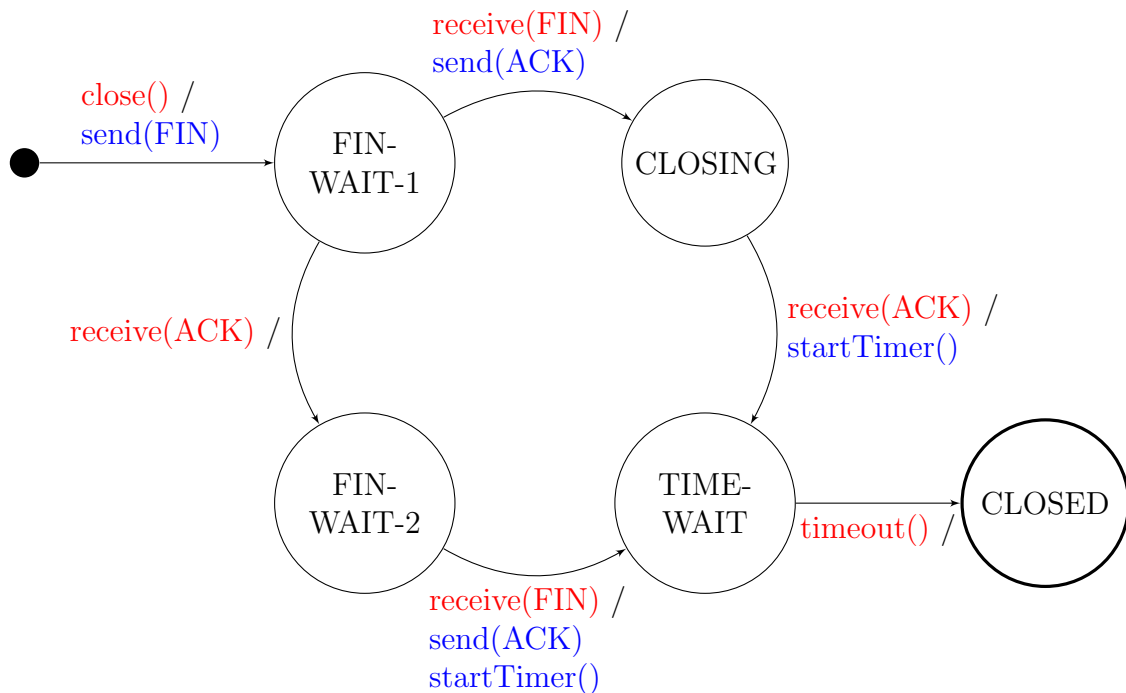


Abbildung 3: TCP-FIN Initiator Verhalten

Abbildung 3 zeigt das Verhalten für einen TCP-Verbindungsabbau auf der Seite des Initiators im fehlerfreien Fall. Realisieren Sie das gezeigte Verhalten blockierungsfrei in der Klasse *TCPFIN*. Leiten Sie die Klasse *TCPFIN* von *Base* ab und verwenden Sie die in *Base* bereit gestellten Methoden. Führen Sie eine Zustandsvariable ein, die klar die Zustände aus Abbildung 3 widerspiegelt. Bei Fehlern werfen einige Methoden eine *WrongStateException* bzw. eine *IOException*. Diese Exceptions müssen nicht behandelt werden und führen zum Abbruch des Programms. Ein fehlerfreies Kompilieren muss dennoch gewährleistet sein. Die *WrongStateException* wird von der Methode *timeout()* geworfen, wenn diese aufgerufen wird ohne dass sich das Programm im Zustand *TIME-WAIT* befindet. Beachten Sie die Kommentare von *Base* in Listing 1.

Listing 1: Abstrakte Klasse Base

```
1 import java.io.IOException;
2
3 public abstract class Base throws Exception {
4
5     /* Flags */
6     final int FIN = 0;
7     final int ACK = 1;
8
9     /* Exception fuer einen falschen Systemzustand */
10    class WrongStateException extends Exception {...}
11
12    /* Sendet ein Flag zum Kommunikationspartner */
13    void send(int flag) throws IOException {...}
14
15    /* Startet einen internen Timer */
16    void startTimer() {...}
17
18
19    /**
20     * Folgendes muss in TCPFIN implementiert werden
21     **/
22
23    /* Die Applikation moechte die Verbindung beenden */
24    abstract void close(void) throws IOException;
25
26    /* Empfaengt ein Flag vom Kommunikationspartner */
27    abstract void receive(int flag) throws IOException;
28
29    /* Wird nach Ablauf des Timers aufgerufen */
30    abstract void timeout(void) throws WrongStateException;
31 }
```

```
public class TCPFIN
```

4 Routingverfahren (26 Punkte)

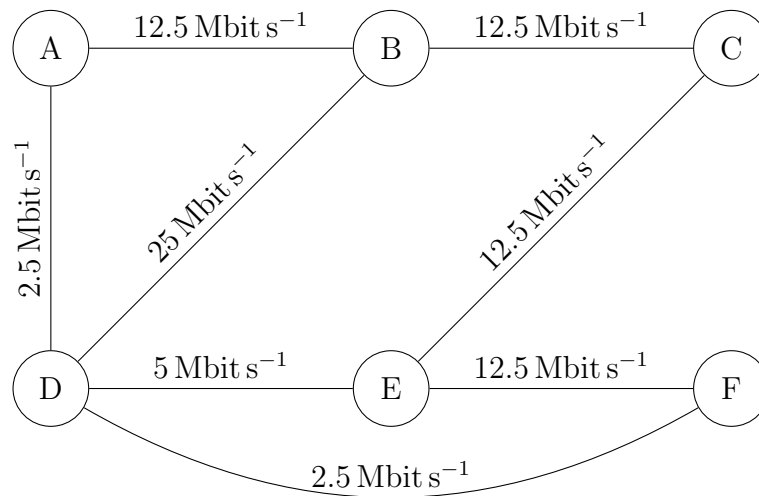


Abbildung 4: Netzwerktopologie

Abbildung 4 zeigt eine Netzwerktopologie bestehend aus sechs Knoten $K = \{A, B, C, D, E, F\}$. Für jede Verbindung (Kante) ist die jeweilige Übertragungsgeschwindigkeit $R_{u,v}$ mit $u, v \in K$ angegeben.

4.1 Kantengewichte (7 Punkte)

Kantengewichte für Routingverfahren können auf Grund verschiedener Kriterien festgelegt werden.

- a) Bestimmen Sie die Kantengewichte $w(u, v)$ mit $u, v \in K$ der Netzwerktopologie in Abbildung 4 so, dass ein Routingverfahren einen Baum mit einer **minimalen Übertragungszeit** aufspannt. Geben Sie Ihre Formel zur Bestimmung der Kantengewichte $w(u, v)$ an. Normieren Sie dabei die Werte so, dass natürliche Zahlen in einem Wertebereich von 1 bis 10 auftreten. Tragen Sie diese Gewichte in Abbildung 4 ein.

$$w(u, v) =$$

- b) Wie müssen die Kantengewichte $w(u, v)$ definiert werden, wenn ein Routingverfahren die Anzahl der verwendeten Kanten (Links) minimieren soll?

4.2 Link-State-Routing (6 Punkte)

Die Latenzzeit der in Abbildung 4 gezeigten Netzwerktopologie soll optimiert werden. Führen Sie dazu aus Sicht des **Knotens A** das Dijkstra-Verfahren in Tabelle 1 durch. Der resultierende aufspannende Baum soll **minimal hinsichtlich der Übertragungsverzögerung** sein (siehe Aufgabe 4.1; alternativ können Sie auch mit den Übertragungsgeschwindigkeiten als Kantengewichte arbeiten). Verwenden Sie bitte die gleiche Notation wie in der Vorlesung.

S	N'	$D(B), p(B)$	$D(C), p(C)$	$D(D), p(D)$	$D(E), p(E)$	$D(F), p(F)$
1						
2						
3						
4						
5						
6						
7						
8						

Tabelle 1: Dijkstra-Verfahren für Knoten A

4.3 Distance-Vector-Routing (13 Punkte)

Führen Sie das Distance-Vector-Routing Verfahren für die in Abbildung 4 gezeigte Netzwerktopologie durch. **Verwenden Sie für alle Kanten das Gewicht 1.** Gehen Sie davon aus, dass die Knoten simultan senden und Nachrichten von den Nachbarknoten auf synchrone Weise erhalten. Bitte benutzen Sie dieselben Symbole wie in der Vorlesung. Füllen Sie die nachfolgenden Tabellen 2 bis 5 aus: Tabelle 2 enthält die Distanztabelle nach der Initialisierung, Tabelle 3 die Distanztabelle nach dem ersten Nachrichtenaustausch usw. Unbenötigte Tabellen streichen Sie durch.

von A	$D_A(\cdot)$	$nh_A(\cdot)$
A		
B		
C		
D		
E		
F		

von B	$D_B(\cdot)$	$nh_B(\cdot)$
A		
B		
C		
D		
E		
F		

von C	$D_C(\cdot)$	$nh_C(\cdot)$
A		
B		
C		
D		
E		
F		

von D	$D_D(\cdot)$	$nh_D(\cdot)$
A		
B		
C		
D		
E		
F		

von E	$D_E(\cdot)$	$nh_E(\cdot)$
A		
B		
C		
D		
E		
F		

von F	$D_F(\cdot)$	$nh_F(\cdot)$
A		
B		
C		
D		
E		
F		

Tabelle 2: Distanztabelle bei der Initialisierung

von A	$D_A(\cdot)$	$nh_A(\cdot)$
A		
B		
C		
D		
E		
F		

von B	$D_B(\cdot)$	$nh_B(\cdot)$
A		
B		
C		
D		
E		
F		

von C	$D_C(\cdot)$	$nh_C(\cdot)$
A		
B		
C		
D		
E		
F		

von D	$D_D(\cdot)$	$nh_D(\cdot)$
A		
B		
C		
D		
E		
F		

von E	$D_E(\cdot)$	$nh_E(\cdot)$
A		
B		
C		
D		
E		
F		

von F	$D_F(\cdot)$	$nh_F(\cdot)$
A		
B		
C		
D		
E		
F		

Tabelle 3: Distanztabelle Schritt 1

von A	$D_A(\cdot)$	$nh_A(\cdot)$
A		
B		
C		
D		
E		
F		

von B	$D_B(\cdot)$	$nh_B(\cdot)$
A		
B		
C		
D		
E		
F		

von C	$D_C(\cdot)$	$nh_C(\cdot)$
A		
B		
C		
D		
E		
F		

von D	$D_D(\cdot)$	$nh_D(\cdot)$
A		
B		
C		
D		
E		
F		

von E	$D_E(\cdot)$	$nh_E(\cdot)$
A		
B		
C		
D		
E		
F		

von F	$D_F(\cdot)$	$nh_F(\cdot)$
A		
B		
C		
D		
E		
F		

Tabelle 4: Distanztabelle Schritt 2

von A	$D_A(\cdot)$	$nh_A(\cdot)$
A		
B		
C		
D		
E		
F		

von B	$D_B(\cdot)$	$nh_B(\cdot)$
A		
B		
C		
D		
E		
F		

von C	$D_C(\cdot)$	$nh_C(\cdot)$
A		
B		
C		
D		
E		
F		

von D	$D_D(\cdot)$	$nh_D(\cdot)$
A		
B		
C		
D		
E		
F		

von E	$D_E(\cdot)$	$nh_E(\cdot)$
A		
B		
C		
D		
E		
F		

von F	$D_F(\cdot)$	$nh_F(\cdot)$
A		
B		
C		
D		
E		
F		

Tabelle 5: Distanztabelle Schritt 3

Zusatzblatt

Zusatzblatt

Arbeitskopie

Aufgabe 3:

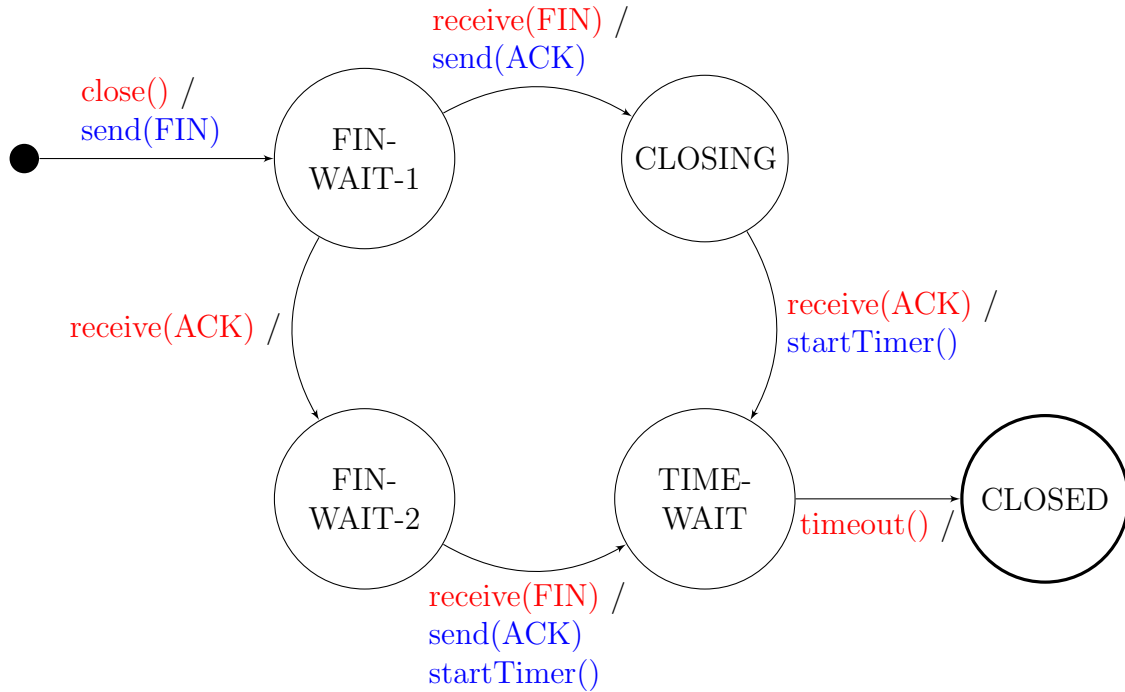


Abbildung 5: TCP-FIN Initiator Verhalten

Aufgabe 4:

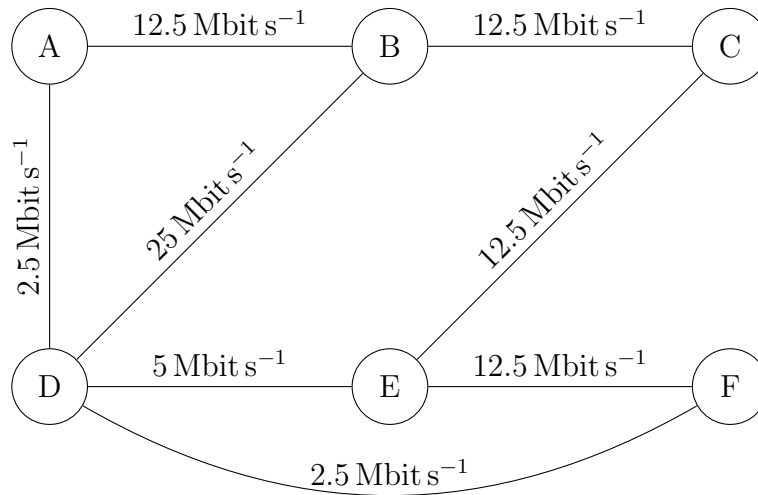


Abbildung 6: Netzwerktopologie