

Leistungsnachweis in Technische Informatik 4

Sommersemester 2007

10. März 2008

Name :

Matrikelnummer :

Geburtsdatum:

Studienfach:

Fachsemester:

- 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!

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

Unterschrift

Bewertung:

1	2	3	4	5	Σ

Aufgabe 1: Allgemeine Fragen (20 Punkte)

Bitte antworten Sie kurz in Stichworten.

- a) (3 Punkte) Geben Sie Protokollschichten des Internet hierarchisch von oben nach unten an und ordnen Sie ihnen die jeweiligen Bezeichnungen der PDU zu!

Nr.	Protokollschicht	PDU
5		
4		
3		
2		
1		

- b) (2 Punkte) Auf welchen Schichten arbeiten jeweils Bridges, Hubs, Router, Repeater und Switches?

- c) (2 Punkte) Wodurch wird bei Berechnung der Prüfsumme in TCP und UDP das Schichtenprinzip verletzt?

d) (2 Punkte) Was versteht man unter statistischem Multiplexen?

e) (1 Punkt) Protokolle lassen sich unter anderem durch Message Sequence Charts (MSCs) beschreiben? Was stellt ein MSC in diesem Zusammenhang dar?

f) (2 Punkte) Gemäß den Basic Encoding Rules (BER) erhalten Sie folgenden Bytestream: 2222. Welche Information wurde Ihnen gesendet?

g) (2 Punkte) Warum reagiert TCP bei der Überlastkontrolle konservativer auf das Ende eines Timeouts als auf drei duplizierte ACKs?

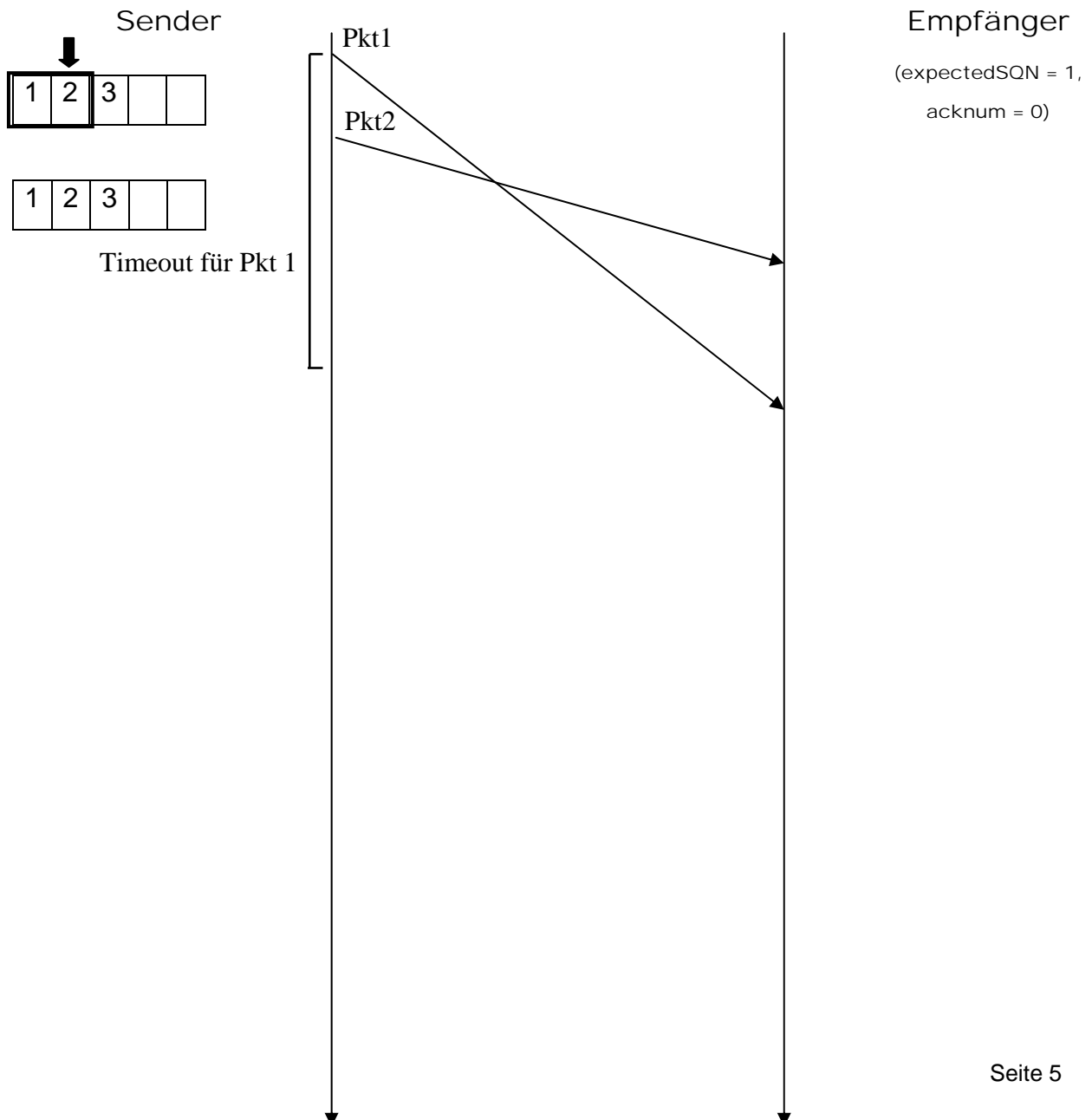
h) (2 Punkte) Geben Sie je ein Beispiel einer Netzwerktechnologie für Datagramm-basierte Paketvermittlung, virtuelle Leitungsvermittlung und klassische Leitungsvermittlung?

i) (2 Punkte) Welches echtzeitfähige Medienzugriffsverfahren kennen Sie? Begründen Sie Ihre Antwort.

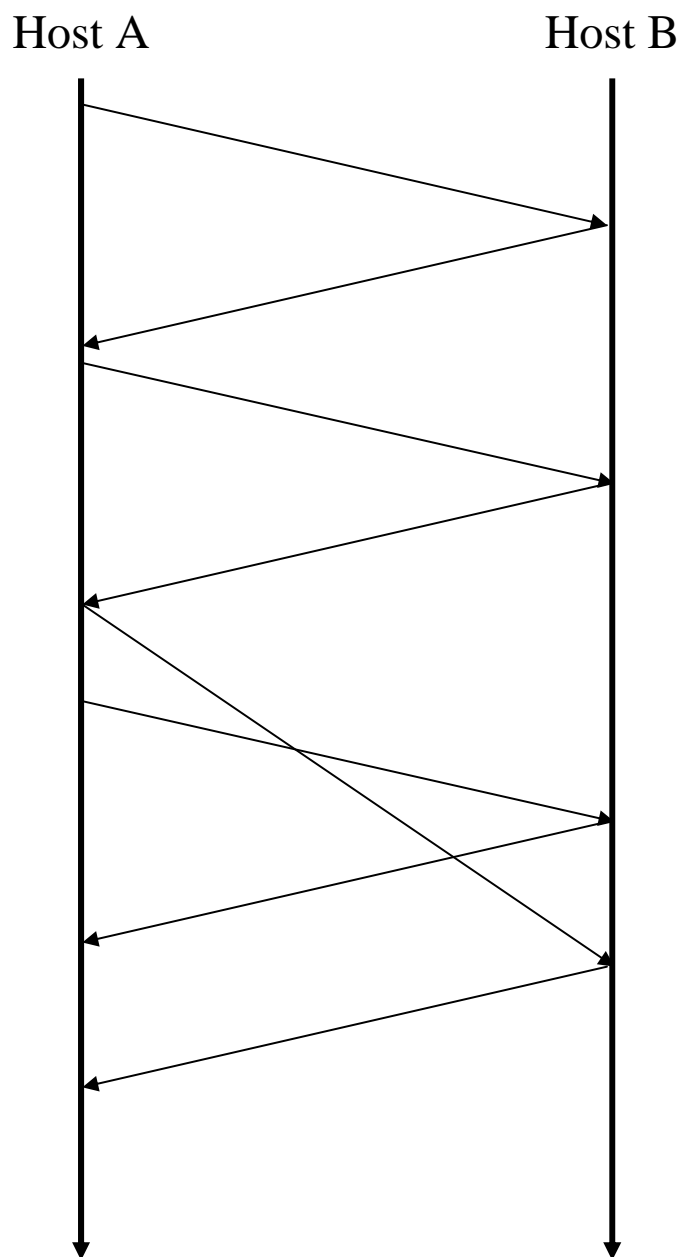
j) (2 Punkte) Wenn man ein analoges Signal mit Frequenzbandbreite 1 kHz periodisch alle 0,4 ms abtastet, kann man es dann (theoretisch) wieder vollständig aus den Abtastwerten zurückgewinnen? Begründen Sie Ihre Antwort.

Aufgabe 2: Transportschicht (25 Punkte)

- a) (8 Punkte) Im Diagramm unten ist ein Ablauf des Schiebefensterprotokolls Go-Back-N wie in der Vorlesung dargestellt. Die Fenstergröße beträgt $W = 2$. Der aktuelle Sendepuffer wird durch das Rechteck gekennzeichnet, der aktuelle Wert von base durch den Pfeil. Dargestellt ist der Zustand nach dem Senden des ersten Pakets. In dem Beispiel sollen drei Pakete verschickt werden. Paket 2 überholt Paket 1 und erreicht den Empfänger vor Paket 1, das nicht mehr innerhalb seines Timeouts (ca. 1.5 RTT) bestätigt werden kann. Führen Sie das Diagramm fort, bis alle drei Pakete beim Sender bestätigt sind. Neben dem Zustand des Sendepuffers nach Senden und Empfangen geben Sie bitte auch an, wann welche Pakete beim Empfänger an die obere Schicht übergeben werden. Es treten keine weiteren Fehler auf, alle weiteren Übertragungszeiten entsprechen der von Paket 2.



b) (8 Punkte) Nehmen Sie nun an, dass Daten über eine TCP-Verbindung verschickt werden, die Host A zu Host B aufbaut. Inclusive des Verbindungsaufbaus könnte die Segmentfolge wie unten dargestellt aussehen, wobei wiederum ein Paket verzögert wird. Wann immer möglich (in dieser Segmentfolge), schickt Host A 500 Bytes an Host B und Host B 300 Bytes an Host A. Host A wählt als Startsequenznummer 0, Host B wählt 1000. Tragen Sie für alle Segmente die Sequenzzummern (SQN), Acknowledgment-Nummern (ACK), die Größe der Nutzdaten sowie eventuell gesetzte TCP-Flags in das Diagramm ein. Es sind keine Pfeile zu ergänzen.



c) (9 Punkte) Ein Objekt der Größe O (in bits), bestehend aus einer geraden Zahl O/S von Segmenten, soll über eine TCP-Verbindung kopiert werden. Unter der Annahme einer dynamischen Fenstergröße, die sich nach dem Slow-Start-Mechanismus entwickelt (wie in der Vorlesung), wird zunächst die erste Hälfte der Segmente erfolgreich übertragen. Die nachfolgenden Segmente gehen verloren, bis der Fehler durch die TCP-Fehlerkontrolle erkannt wird. Danach finden keine Datenverluste mehr statt. Die Übertragungsrate R und die Round Trip Time RTT werden als idealisiert konstant angenommen.

Wie viele Slow-Start-Wartezeiten P entstehen bei der Übertragung des gesamten Objekts?
(Es darf angenommen werden, dass schon die Hälfte des Objekts hinreichend groß ist.)

Berechnen Sie die Latenz für das Kopieren des gesamten Objekts in Abhängigkeit von den gegebenen Größen.

(Veranschaulichen Sie sich das Übertragungsverhalten an Hand einer groben Skizze!)

Aufgabe 3: Programmierung eines SMTP-Clients (25 Punkte)

Gegeben ist das folgende MD-Object:

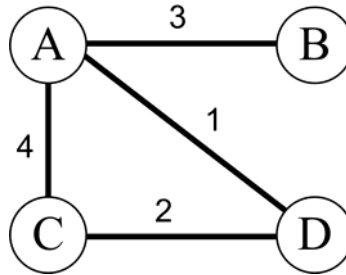
```
public class MD {  
    public final static String SERVER      = "...";  
    public final static String SENDER     = "...";  
    public final static String EMPFAENGER = "...";  
    public final static String BETREFF    = "...";  
    public final static String DATEN      = "...";  
}
```

Geben Sie bitte ein **JAVA**-Programm für einen **SMTP**-Client an, der über **Sockets** eine E-Mail versendet. Die Mail soll über *MD.SERVER* mit dem Absender *MD.SENDER*, dem Empfänger *MD.EMPFAENGER*, dem Betreff *MD.BETREFF* und den Daten *MD.DATEN* versendet werden. Verwenden Sie bitte ausschließlich die Standard-Edition und keine weiteren Bibliotheken. Folgende Punkte sind zu beachten:

- Exceptions müssen berücksichtigt, aber nicht ausgewertet werden.
- Nach dem Verbindungsaufbau zum SMTP-Server soll der Statuscode des SMTP-Servers **einmalig** überprüft werden. Im Fehlerfall soll eine entsprechende Meldung ausgegeben und das Programm beendet werden. Im weiteren Verlauf des Programms können die Antworten des SMTP-Servers als immer passend angesehen und ignoriert werden.
- Das Programm ist **immer** sauber zu beenden (Verbindungsabbau).

Aufgabe 4: Routingverfahren (15 Punkte)

Betrachten Sie folgendes Netzwerk mit vier Knoten:



Simulieren Sie den Distance-Vector-Algorithmus. Wie in Kapitel *Netzwerkschicht* der Vorlesung gehen wir hier 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 aus: Der erste Abschnitt enthält die Distanztabelle nach der Initialisierung, der zweite die Distanztabelle nach dem ersten Nachrichtenaustausch usw. In die letzte Spalte jeder Tabelle soll der nächste Hop eingetragen werden, auf dem das Ziel mit momentan minimalen Kosten erreicht wird.

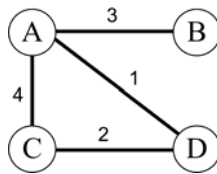
Schritt 1:

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

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

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

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



Schritt 2:

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

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

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

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

Schritt 3:

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

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

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

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

Schritt 4:

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

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

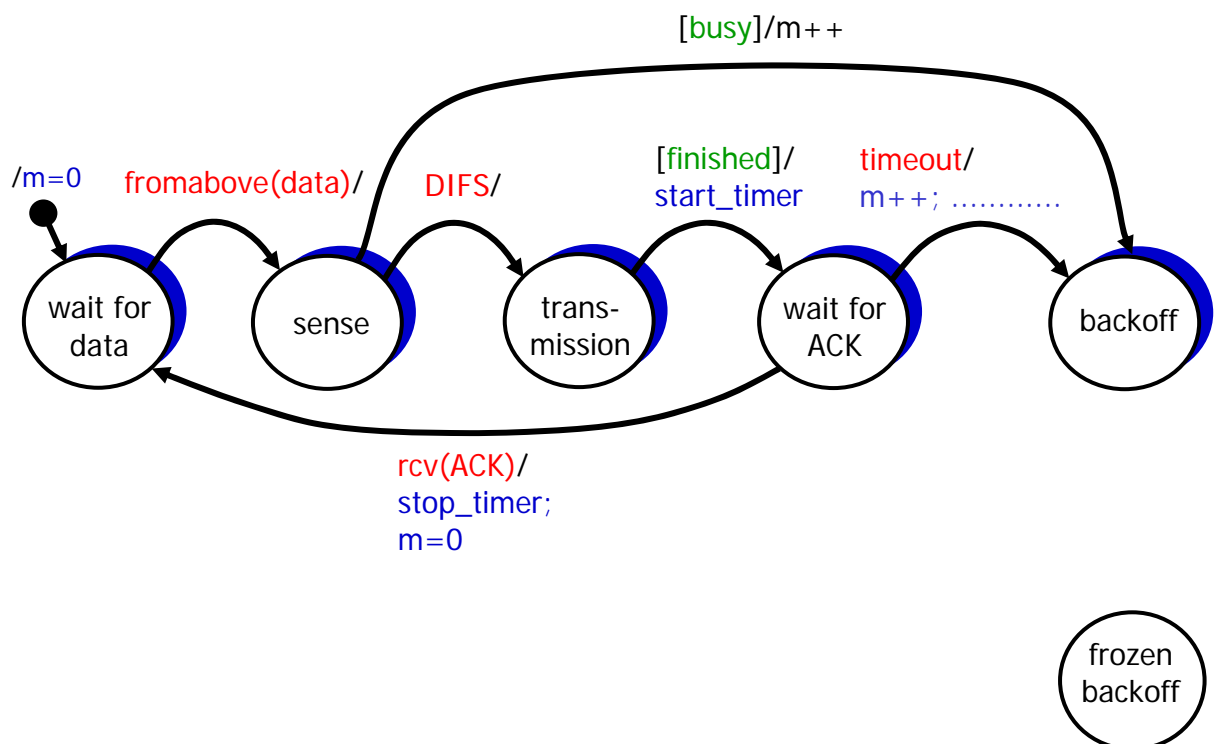
von C zu	$D_C(\cdot)$	$nh_C(\cdot)$
A		
B		
C		
D		

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

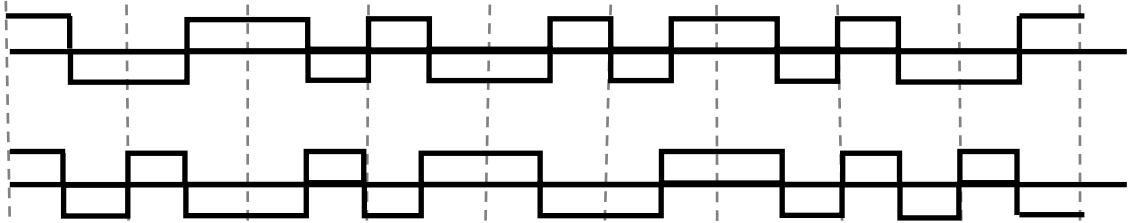
Aufgabe 5: Verbindungs- und physikalische Schicht (15 Punkte)

- a) (10 Punkte) Beim Medienzugriff in drahtlosen LANs nach IEEE 802.11 friert ein Sender seinen Backoff-Timer für die Dauer von anderen Übertragungen ein. D.h., sobald er den Kanal im Backoff-Zustand als belegt erkennt, speichert er seine aktuelle Wartezeit und nimmt den Backoff mit dieser zuletzt gespeicherten Wartezeit wieder auf, wenn der Kanal wieder frei wird. Erweitern Sie das Ihnen aus der Vorlesung bekannte, unten leicht modifizierte Statechart des Senders um den beschriebenen Backoff-Mechanismus. Nutzen Sie den vorgegebenen Zustand "frozen backoff" (weitere Zustände sind nicht nötig) und ergänzen Sie Kanten, Ereignisse, Bedingungen und Aktionen (eventuell auch im Zustandsübergang von "wait for ACK" to "backoff"). Hinweis:

- Die aktuelle Zeit können Sie mit der Methode `time()` abrufen.
- Die erste Wartezeit wird wie bisher zufällig mit `"random(0, ..., 2m - 1) * t"` gewählt, wobei m die Anzahl der Kollisionen und t die feste Dauer eines Zeitschlitzes ist.
- Sie erhalten einen Bonuspunkt, wenn Sie beachten, dass die Wartezeit bei jeder Wiederaufnahme des Backoff auf das nächste Vielfache von t aufgerundet wird.



- b) (5 Punkte) Sie erhalten auf zwei verschiedenen Leitungen jeweils die folgenden kodierten binären Signale:



Das obere Signal wurde mit der Differentiellen Manchester-Kodierung kodiert, das untere mit der (einfachen) Manchester-Kodierung.

Unterscheiden sich die beiden übertragenen Bitfolgen? Wenn ja, an welchen Stellen und welche Bits wurden dort jeweils übertragen?

