

---

Leistungsnachweis in  
**Rechnerkommunikation**

Wintersemester 2012/2013

28. März 2013

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 17 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 17 Blätter der Klausur erhalten:

\_\_\_\_\_  
Unterschrift

<b>Bewertung:</b>	1	2	3	4	$\Sigma$

---

# 1 Allgemeine Fragen (25 Punkte gesamt)

## 1.1 UDP vs. TCP (2 + 2 + 2 Punkte)

**a)** Nennen Sie eine klassische Anwendung die üblicherweise mittels UDP kommuniziert. Was ist der Vorteil im Gegensatz zu TCP? Begründen Sie kurz Ihre Antwort.

**b)** Nennen Sie eine klassische Anwendung die üblicherweise mittels TCP kommuniziert. Was ist hier der Vorteil im Gegensatz zu UDP? Begründen Sie kurz Ihre Antwort.

**c)** Könnten theoretisch die Anwendungen aus den Teilaufgaben a) und b) auch das jeweils andere Übertragungsprotokoll verwenden? Was wäre ein Nachteil bzw. was müsste beachtet werden? Bitte erläutern Sie beide Fälle.

---

## 1.2 FTP (2 + 4 Punkte)

**a)** Erklären Sie den Begriff *Out-of-Band-Control* anhand des FTP-Protokolls. Was ist der entscheidende Vorteil?

**b)** Erklären Sie kurz die Funktionsweise des *Passive Modes* im FTP-Protokoll. Nennen Sie den wesentlichen Vorteil dieser Datenübertragung im Vergleich zum *Active Mode*?

## 1.3 Latenzzeit (4 + 6 + 3 Punkte)

Ein Objekt der Größe  $O = 1 \text{ MB}$  soll über ein Netzwerk mit 5 dazwischen liegenden *Store-and-Forward*-Routern ( $E = 6$  Links) übertragen werden. Die Paketgröße beträgt  $L = 1 \text{ kB}$ . Die Übertragungsrate jedes Links beträgt  $R = 8 \text{ Mbit s}^{-1}$ . (Vereinfachung:  $1 \text{ MB} = 10^3 \text{ kB} = 10^6 \text{ B}$ . Es gibt keine Fehler und auch keine Ausbreitungsverzögerung)

**a)** Wie lange dauert es ( $d_1$ ) das Objekt  $O$  zu übertragen, wenn keine Übertragungsprotokoll zugrunde liegt (keine ACKs)?

---

**b)** Gehen Sie jetzt davon aus, dass *Go-Back-N* mit einer festen Fenstergröße von  $W = 5$  für die Übertragung verwendet wird. ACKs haben eine idealisierte Größe von 0. Wie lange dauert es ( $d_2$ ) bis das Objekt  $O$  vollständig auf dem Zielhost verfügbar ist? Wie viele Wartezeiten  $P$  werden benötigt und wie lange dauert eine Wartezeit  $d_p$ . (Hinweis: Es gelten weiterhin die unter Abschnitt 1.3 genannten Größen.  $E = 6$  Links, ...)

**c)** Stellen Sie *Go-Back-N* und *Selective-Repeat* gegenüber, indem Sie je einen Vor- und einen Nachteil des Verfahrens nennen. Wie lässt sich TCP **mit** SACK und TCP **ohne** SACK in diese Verfahren einordnen. Begründen Sie Ihre Antwort.

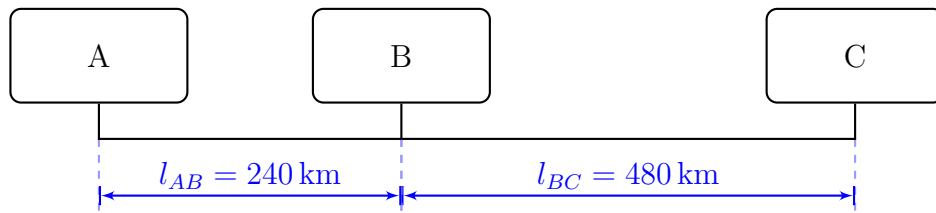


Abbildung 1: Netzwerkkonfiguration

## 2 Medienzugriff (25 Punkte gesamt)

Folgendes gilt: Drei Hosts  $A$ ,  $B$  und  $C$  seien über einen Netzwerk-Bus gemäß Abbildung 1 miteinander verbunden. Die Bitrate beträgt  $R = 125 \text{ kbit s}^{-1}$ .

### 2.1 Latenzzeit (5 Punkte)

Die Ausbreitungsgeschwindigkeit beträgt  $v = 2 \times 10^8 \text{ m s}^{-1}$ . Die Abstände entnehmen Sie Abbildung 1. Bestimmen Sie die Ausbreitungsverzögerungen  $d_{AB}$ ,  $d_{BC}$  und  $d_{AC}$  zwischen den einzelnen Hosts in Bitzeiten. (Hinweis: Berechnen Sie zunächst die Ausbreitungsverzögerung in der üblichen Einheit Sekunden. Wie aus der Übung bekannt entspricht eine Bitzeit der Zeit, die benötigt wird um ein Bit auf die Leitung zu legen. Bei der hier gegebenen Bitrate  $R$  gilt also:  $1 \text{ Bitzeit} = 8 \mu\text{s}$ )

---

Rahmensendedauer ( $L = 1000$ bit)	$d_{trans} = 1000$ Bitzeiten
Ausbreitungsverzögerung	$d_{AB} = 150$ Bitzeiten $d_{BC} = 300$ Bitzeiten $d_{AC} = 450$ Bitzeiten
Dauer, die der Kanal vor dem tatsächlichen Senden frei sein muss	$t_{sense} = 50$ Bitzeiten
Jam-Signal-Sendedauer	$t_{jam} = 50$ Bitzeiten
Backoff-Konstanten	$K_A = 3$ $K_B = 2$ $K_C = 1$
Backoff-Zeitslot	$\tau = 700$ Bitzeiten

---

Tabelle 1: Netzwerktopologie

## 2.2 Carrier Sense Multiple Access (3 + 14 + 3 Punkte)

Nehmen Sie an, dass die drei Hosts aus Abbildung 1 das 1-persistente CSMA/CD-Verfahren für den Medienzugriff verwenden. Weiterhin gelten die Werte aus Tabelle 1.

**a)** Stellt die gewählte Rahmengröße von  $L = 1000$  bit in dem hier gezeigten Netzwerk sicher, dass Kollisionen zuverlässig detektiert werden können? Wie groß darf die maximale Ausbreitungsverzögerung (in Bitzeiten) für die hier gegebenen Parameter zwischen 2 Hosts sein. Geben Sie auch den formalen Zusammenhang an.

---

**b)** Host  $A$  und Host  $B$  wollen zum Zeitpunkt  $t_0 = 0$  Bitzeiten gleichzeitig einen Rahmen versenden. Dazu lauschen beide gleichzeitig auf dem Medium und entscheiden sich zum Zeitpunkt  $t_1 = 50$  Bitzeiten den Rahmen zu senden. Host  $C$  möchte zum Zeitpunkt  $t_2 = 320$  Bitzeiten einen Rahmen versenden und beginnt somit zu lauschen. Zeichnen Sie in die Abbildungen 2 und 3 den Rahmenaustausch maßstabsgetreu ein bis alle 3 Pakete erfolgreich versendet wurden. Nehmen Sie (wie in der Übung) vereinfachend an, dass für alle Hosts konstante Backoff-Faktoren gemäß Tabelle 1 gelten.

**c)** Nehmen Sie **allgemein** an, dass die Hosts  $A$  und  $B$  zum Zeitpunkt  $t_0$  bereits  $(m - 1)$  fehlgeschlagene Sendeveruche unternommen haben, d.h. in Abbildung 2 wird der  $m$ -te Fehlversuch detektiert. Geben Sie ausgehend von dem Zeitpunkt  $t_k$ , der erkannten Kollision zwischen Host  $A$  und Host  $B$ , eine Formel an, zu welchem Zeitpunkt  $t_{r,min}$  sich Host  $A$  frühestens wieder im Lausch-Zustand befindet. Zu welchem Zeitpunkt  $t_{r,max}$  beginnt Host  $A$  im ungünstigsten Fall wieder zu lauschen?

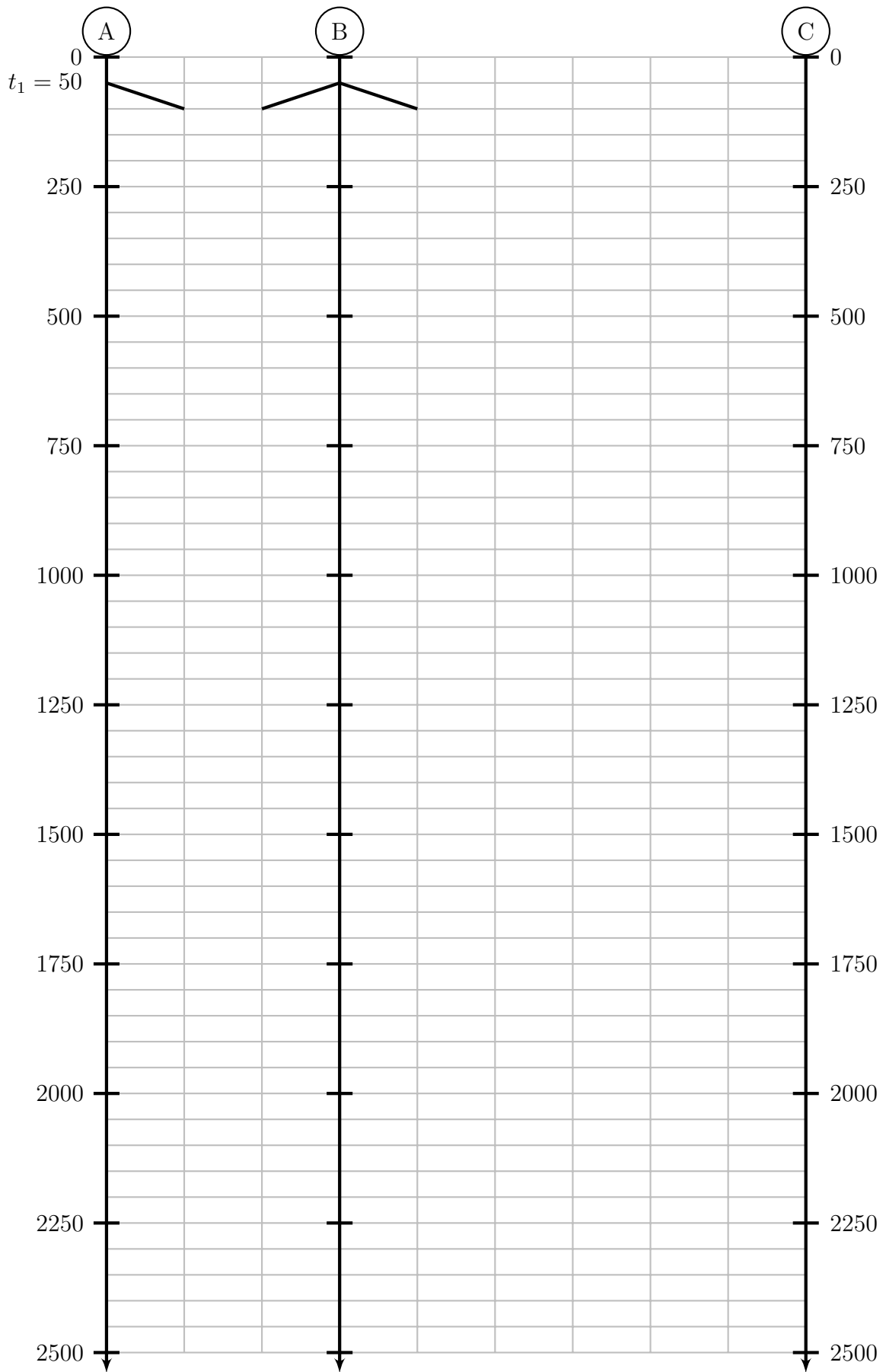


Abbildung 2: Ablaufdiagramm für CSMA/CD



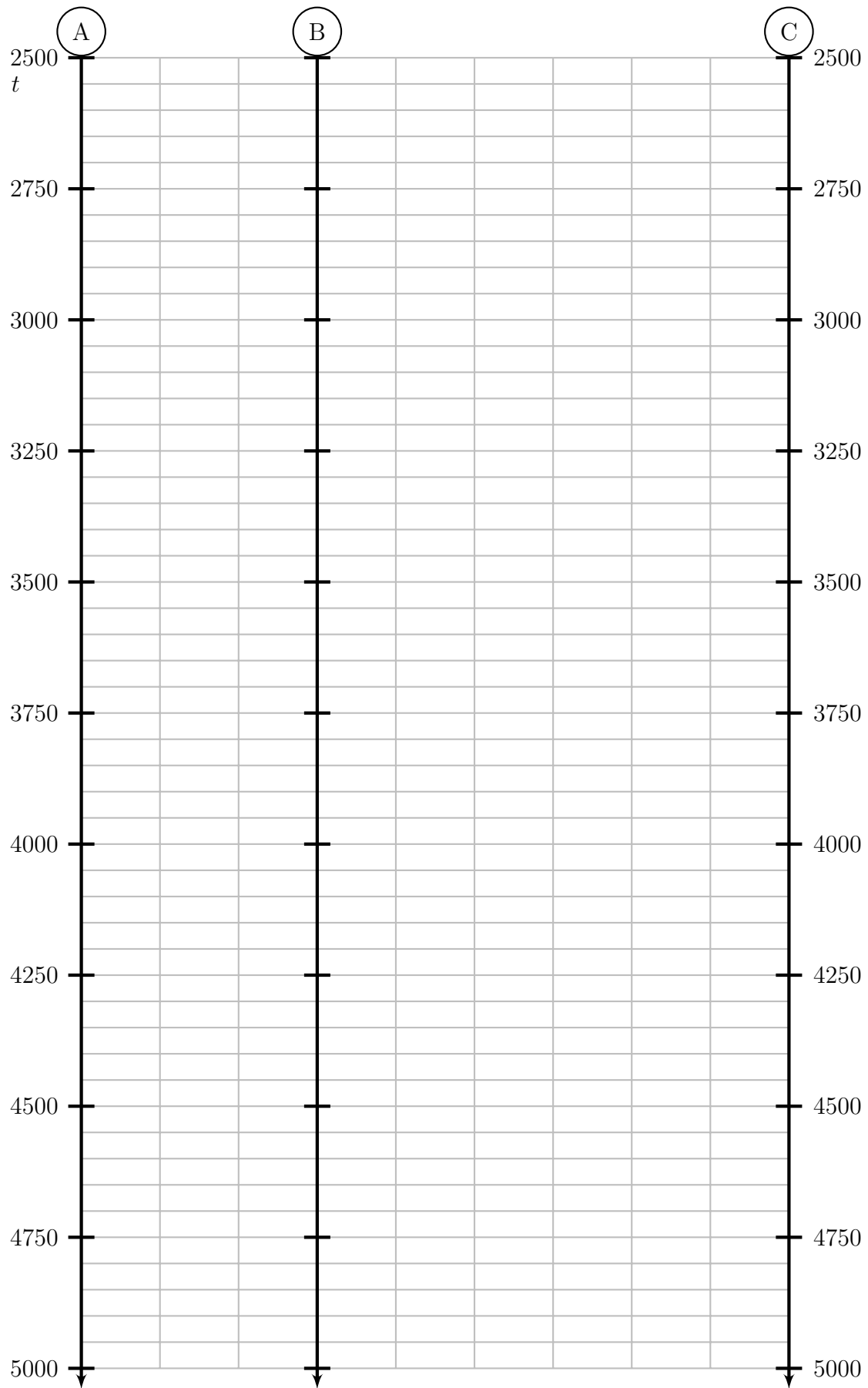


Abbildung 3: Ablaufdiagramm für CSMA/CD (fortgesetzt)

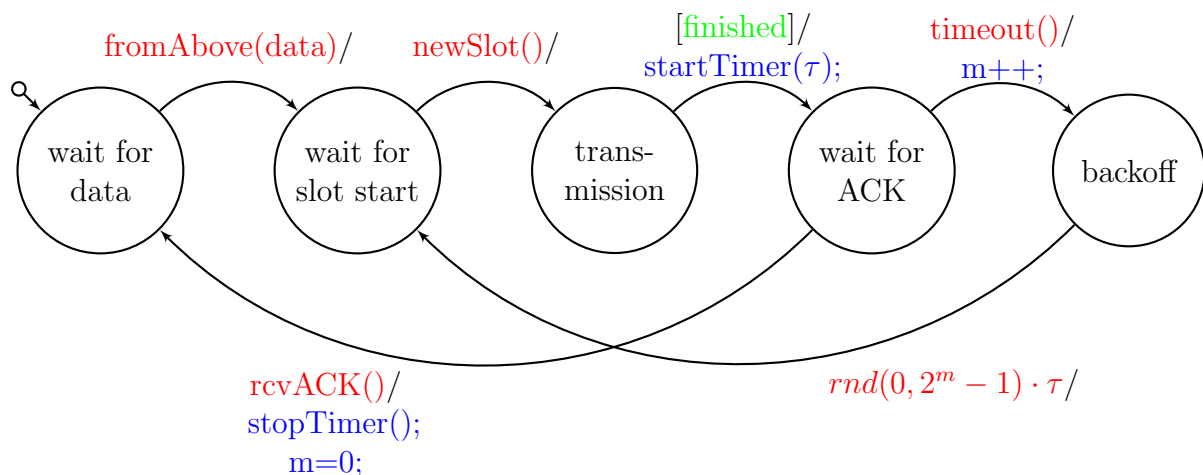


Abbildung 4: *Slotted ALOHA* - Zustandsdiagramm

### 3 Slotted ALOHA (25 Punkte)

Abbildung 4 zeigt die Senderseite einer *Slotted-Aloha*-basierten Übertragung. Realisieren Sie das gezeigte Verhalten blockierungsfrei in der Klasse *Saloha*. Leiten Sie die Klasse *Saloha* von *Base* ab und verwenden Sie die in *Base* bereitgestellten Methoden. Führen Sie eine Zustandsvariable ein, die klar die Zustände aus Abbildung 4 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 Methode *fromAbove()* soll eine Exception werfen wenn sie aufgerufen wird ohne, dass das Zustandsdiagramm dies vorsieht. 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
6     class Packet {...}
7     final int TAU = 10; //Slotzeit
8
9     /* Exception fuer einen falschen Systemzustand */
10    class WrongStateException extends Exception {...}
11
12    /* Sendet sofort ein Paket zum Kommunikationspartner */
13    void send(Packet pkt) throws IOException {...}
14
15    /* Startet den Timer der in s Millisekunden ablaeuft */
16    void startTimer(int s) {...}
17
18    /* Stopt den Timer */
19    void stopTimer() {...}
20
21    /* Liefert eine Zufallszahl im Bereich [b,e) zurueck */
22    int rnd(int b, int e) {...}
23
24
25    /**
26     * Folgendes muss in Saloha implementiert werden
27     */
28
29    /* Wird aufgerufen wenn Daten verschickt werden sollen*/
30    abstract void fromAbove(Packet pkt) throws
31        WrongStateException;
32
33    /* Wird nach Ablauf des Timers aufgerufen */
34    abstract void timeout();
35
36    /* Wird ZYKLISCH zu JEDEM Slotbegin aufgerufen */
37    abstract void newSlot() throws IOException;
38
39    /* Wird aufgerufen wenn ein ACK empfangen wird */
40    abstract void revACK();
41 }

```

---

```
public class Saloha
```

---

## 4 Routingverfahren (25 Punkte gesamt)

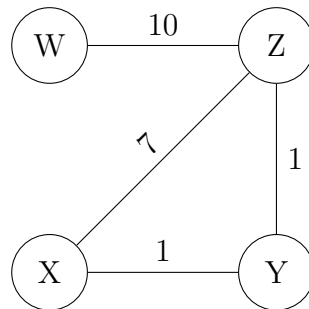


Abbildung 5: Netzwerktopologie

### 4.1 Distance-Vector-Routing (13 Punkte)

Führen Sie das Distance-Vector-Routing-Verfahren für die in Abbildung 5 gezeigte Netzwerktopologie durch. 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.

### 4.2 Poisoned Reverse (10 + 2 Punkte)

Poisoned Reverse ist ein Verfahren zur Vermeidung von Routing Schleifen im Distanzvektor-Routing-Protokollen.

**a)** Tabelle 6 zeigt die konvergierten Routing-Informationen für das Netzwerk in Abbildung 5. Nehmen Sie an, dass nun die Kosten der Kante zwischen den Knoten  $Z$  und  $Y$  auf  $c(Z, Y) = 10$  erhöht werden. Im Folgenden soll ausschliesslich der kürzeste Weg zu Knoten  $W$  betrachtet werden. Zeigen Sie, wie sich die Routing-Informationen **ohne** (in Tabelle 7) und **mit** (in Tabelle 8) Poisoned Reverse an den beteiligten Knoten ändern bis erneute Konvergenz erreicht wird.

Die Tabellen stellen einen zeitlichen Verlauf dar. Beispiel: Im ersten Schritt aktualisiert der Knoten  $Y$  seinen Routing-Information zu Knoten  $W$ , wie in Tabelle 7 dargestellt, auf die Kosten 13 über Knoten  $X$ .

**b)** Nehmen Sie unabhängig von der Teilaufgabe a) nun an, dass die Kosten der Kante zwischen den Knoten  $Z$  und  $W$  in Abbildung 5 auf  $c(W, Z) = 30$  erhöht worden sind ( $c(Z, Y) = 1$ ). Kann Poisoned Reverse auch in diesem Fall zur schnelleren Schleifenauflösung beitragen? Begründen Sie kurz Ihre Antwort?

von W zu	$D_W(\cdot)$	$nh_W(\cdot)$
W		
X		
Y		
Z		

von Z zu	$D_Z(\cdot)$	$nh_Z(\cdot)$
W		
X		
Y		
Z		

von X zu	$D_X(\cdot)$	$nh_X(\cdot)$
W		
X		
Y		
Z		

von Y zu	$D_Y(\cdot)$	$nh_Y(\cdot)$
W		
X		
Y		
Z		

Tabelle 2: Distanztabelle bei der Initialisierung

von W zu	$D_W(\cdot)$	$nh_W(\cdot)$
W		
X		
Y		
Z		

von Z zu	$D_Z(\cdot)$	$nh_Z(\cdot)$
W		
X		
Y		
Z		

von X zu	$D_X(\cdot)$	$nh_X(\cdot)$
W		
X		
Y		
Z		

von Y zu	$D_Y(\cdot)$	$nh_Y(\cdot)$
W		
X		
Y		
Z		

Tabelle 3: Distanztabelle Schritt 1

von W zu	$D_W(\cdot)$	$nh_W(\cdot)$
W		
X		
Y		
Z		

von Z zu	$D_Z(\cdot)$	$nh_Z(\cdot)$
W		
X		
Y		
Z		

von X zu	$D_X(\cdot)$	$nh_X(\cdot)$
W		
X		
Y		
Z		

von Y zu	$D_Y(\cdot)$	$nh_Y(\cdot)$
W		
X		
Y		
Z		

Tabelle 4: Distanztabelle Schritt 2

von W zu	$D_W(\cdot)$	$nh_W(\cdot)$
W		
X		
Y		
Z		

von Z zu	$D_Z(\cdot)$	$nh_Z(\cdot)$
W		
X		
Y		
Z		

von X zu	$D_X(\cdot)$	$nh_X(\cdot)$
W		
X		
Y		
Z		

von Y zu	$D_Y(\cdot)$	$nh_Y(\cdot)$
W		
X		
Y		
Z		

Tabelle 5: Distanztabelle Schritt 3

von Z zu	$D_Z(\cdot)$	$nh_Z(\cdot)$
W	10	W

von X zu	$D_X(\cdot)$	$nh_X(\cdot)$
W	12	Y

von Y zu	$D_Y(\cdot)$	$nh_Y(\cdot)$
W	11	Z

Tabelle 6: Routing Informationen für den kürzesten Weg zu  $W$  in Abbildung 5 **vor** der Änderung der Kosten ( $c(Z, Y) = 1$ )

Die Änderung der Routing Informationen über die Zeit für den kürzesten Weg zu  $W$ :  
(mit  $c(Z, Y) = 10$ )

Zeit	Von ... zu $W$	$D_{...}(W)$	$nh_{...}(W)$
$t_0$	Y	13	X
$t_1$			
$t_2$			
$t_3$			
$t_4$			
$t_5$			
$t_6$			
$t_7$			
$t_8$			
$t_9$			

Tabelle 7: Ohne Poisoned Reverse

Time	Von ... zu $W$	$D_{...}(W)$	$nh_{...}(W)$
$t_0$			
$t_1$			
$t_2$			
$t_3$			
$t_4$			
$t_5$			
$t_6$			
$t_7$			
$t_8$			
$t_9$			

Tabelle 8: Mit Poisoned Reverse

---

Zusatzblatt



# Arbeitskopie

## Aufgabe 3:

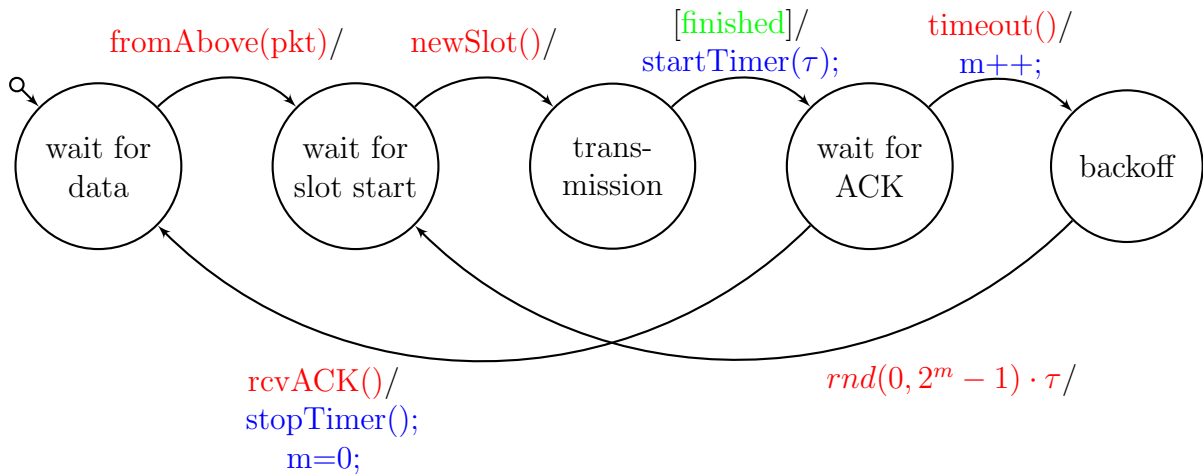


Abbildung 6: *Slotted ALOHA* - Zustandsdiagramm

Listing 2: Abstrakte Klasse Base

```

1 import java.io.IOException;
2 public abstract class Base throws Exception {
3     class Packet {...}; final int TAU = 10;
4     class WrongStateException extends Exception {...}
5     void send(Packet pkt) throws IOException {...}
6     void startTimer(int s) {...}
7     void stopTimer() {...}
8     int rnd(int b, int e) {...}
9     // Folgendes muss in Saloha implementiert werden
10    abstract void fromAbove(Packet pkt) throws
11        WrongStateException;
12    abstract void timeout();
13    abstract void newSlot() throws IOException;
14    abstract void rcvACK();
15 }
  
```

## Aufgabe 4:

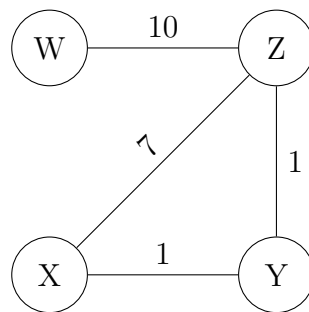


Abbildung 7: Netzwerktopologie