Leistungsnachweis in

Rechnerkommunikation

Wintersemester 2009 / 2010

1. April 2010

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 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!

Ich habe die Hinweise auf dieser Seite zur Kenntnis genommen und alle 17 Blätter der Klausur empfangen:

_____

Unterschrift

Bewertung:

1	2	3	4	Σ

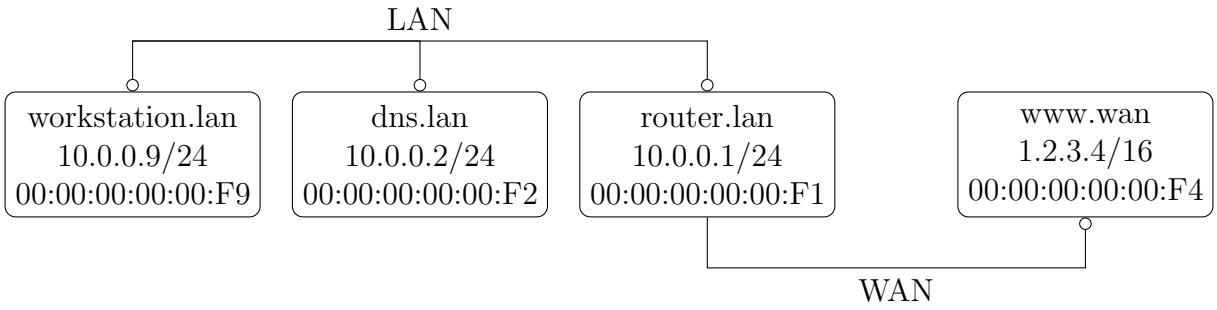

Abbildung 1: Netzwerkkonfiguration

1 Verbindungsaufbau (18 Punkte)

Gegeben sei eine Netzwerkkonfiguration gemäß Abbildung 1. Sie besteht aus einem *Local Area Network* (LAN) und aus einem *Wide Area Network* (WAN). Beide Netzwerke sind mit einem IP-Router verbunden. Jedes Rechteck bildet einen Teilnehmer ab. In der ersten Zeile steht der Name des Teilnehmers. In der zweiten Zeile seine IP und in der dritten Zeile seine *Medium Access Control* (MAC) Adresse. Die WAN-Informationen des Router Interfaces sind nicht gelistet, da sie für diese Aufgabe keine Relevanz haben. Die IP-Adressen werden statisch zum Systemstart vergeben. Die Workstation *workstation.lan* kennt außerdem die IP und den Namen des *Domain Name Servers* (DNS), sowie des IP-Routers.

Kurz nach dem Systemstart will nun der Benutzer der Workstation *workstation.lan* auf folgende Webseite zugreifen: *http://www.wan/test.html*. Tragen sie alle Pakete, die die Workstation hierfür versendet und empfängt, in Tabelle 1 ein. Die Abkürzung "Src" steht für die Quell- und die Abkürzung "Dest" für die Ziel-Adresse. Sie können die IPund MAC-Adressen jeweils mit dem letzten Byte abkürzen (z.B. für die MAC-Adresse der Workstation: "F9"). Broadcast-Adressen kürzen sie mit "FF" ab. Wird ein Feld nicht benötigt so streichen Sie es aus. Der Paketverlauf soll bis einschließlich dem HTTP-Request dargestellt werden. In der Spalte Protokoll nennen Sie das höchstwertige Protokoll. In die letzte Spalte tragen Sie die wichtigen Optionen und/oder den Payload ein. Eine Pseudonotation ist ausreichend.

Src MAC	Dest MAC	Protokoll	Src IP	Dest IP	Optionen/Bemerkung/Payload

Tabelle 1: Packetverlauf

2 TCP (34 Punkte insgesamt)

Host A baut zu Host B eine TCP-Verbindung auf. Ohne eine konkrete Anfrage beginnt Host B so bald wie möglich mit dem Senden der Daten. Während des Sendens geht das komplette zweite Datenfenster verloren (siehe Abbildung 2). Als Überlastkontrolle wird nur Slow-Start und Timeout verwendet.

Host A beginnt das Senden mit der Sequenznummer $Seq = 10$ und Host B beginnt das Senden mit der Sequenznummer $Seq = 40$. Weiterhin gelten die Konstanten gemäß Tabelle 2.

Ausbreitungsverzögerung:	$d_{prop} = 45\,\text{ms}$
Raten:	$R = 10\ \frac{Mbit}{s}$
Paketlänge:	$L = 1250\ Byte\ (= MSS)$
Timeout:	$\tau = 150\,\text{ms}$

Tabelle 2: Konstanten

2.1 Round Trip Time (2 Punkte)

Die gemessene mittlere Round Trip Time (RTT) für die Verbindung zwischen Host A und Host B beträgt $2 \cdot d_{prop} = 90ms$. Diese Messung wurde durchgeführt als alle beteiligten Systeme ohne Last liefen (sofortiges Bearbeiten der Pakete). Da die Last bei Host B durch andere Dienste, die unabhängig vom Netzwerk sind, angestiegen ist, benötigt Host B nun im Mittel 10ms um Pakete zu bearbeiten. Ändert sich dadurch etwas an der RTT?

$RTT =$

2.2 TCP Verlauf (16 Punkte)

Tragen Sie in Abbildung 2 den weiteren Ablauf für die nächsten zwei Fenster (inklusive der ACKs) ein. Berücksichtigen Sie, dass keine weiteren Fehler auftreten. Die Abkürzung CW in Abbildung 2 steht für *Congestion Window* und wird mit 1 *Maximum Segment Size* (MSS) initialisiert (CW = 1 MSS). Setzen Sie für jedes gesendete Paket die passenden Flags und die richtigen Sequenz- und ACK-Nummern. Zeigen Sie an, ob Daten im Paket transportiert werden. Nennen Sie weiterhin den Wert des CW nach jedem Paket, das bei Host B eingetroffen ist. Markieren Sie in Abbildung 2 den Zeitpunkt t_1, der angibt, dass 5000 Byte auf Host A verfügbar sind. Gehen sie davon aus, dass das *Advertized Window* von Host A immer groß genug ist.

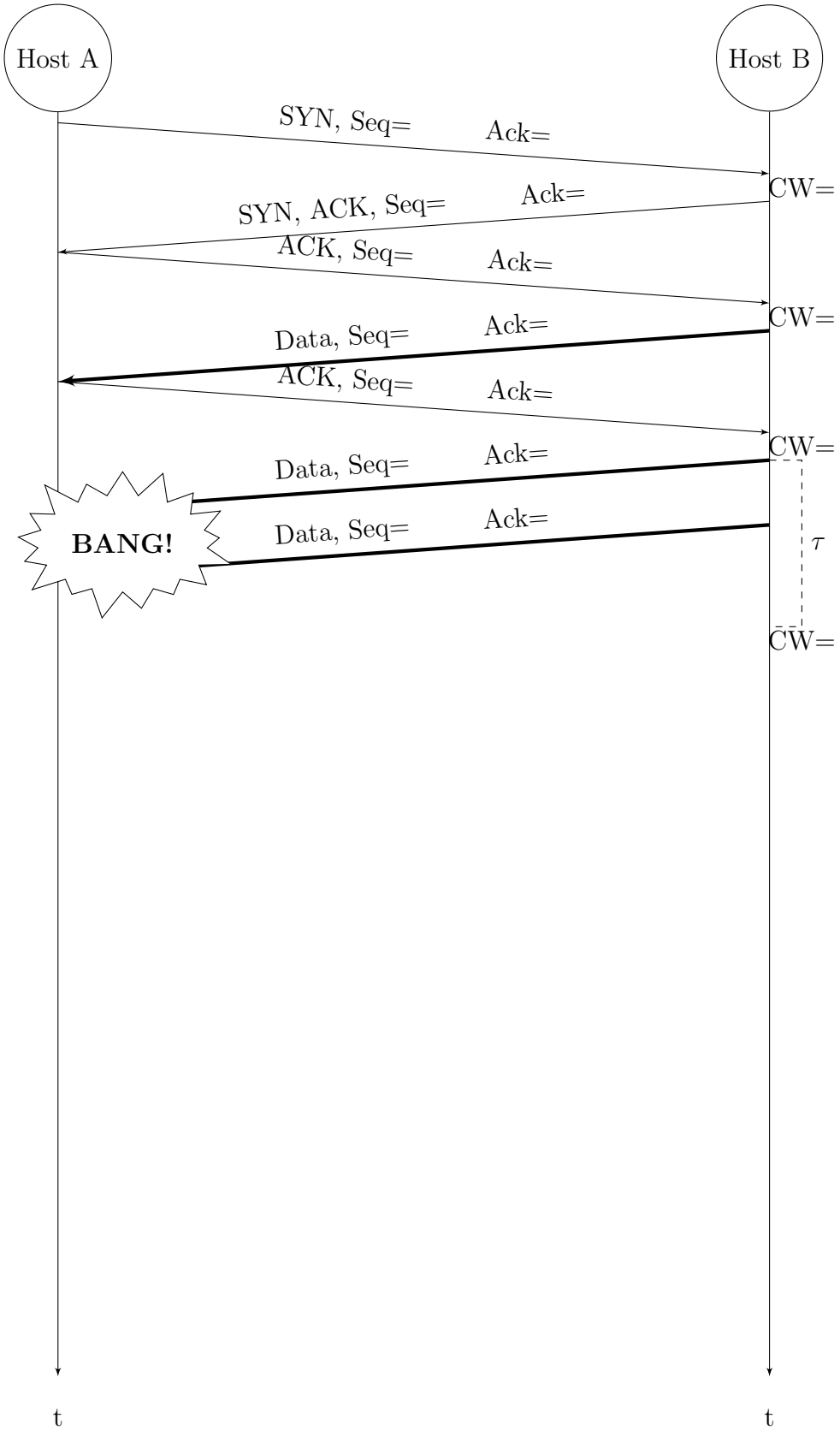

Abbildung 2: Ablaufdiagramm für eine TCP-Verbindung

2.3 Leistungsanalyse (16 Punkte)

Modifizieren Sie die aus der Vorlesung bekannte Formel für die TCP-Latenzberechnung mit Slow Start (1) so, dass immer genau das n-te Fenster komplett verloren geht und sonst keine weiteren Fehler auftreten. Die für die Berechnung benötigten Wartezeiten $P_i = min(Q_i, K_i - 1)$ müssen nicht vollständig hergeleitet werden. Der Ansatz ist ausreichend. Bitte beachten Sie den unrealistischen Ansatz, dass das *Advertized Window* als unendlich groß angenommen wird. Auch nach Fehlern wird mit Slow Start begonnen. Achten Sie darauf, für welchen Host die angegebenen/berechneten Zeiten gelten.

Berechnen Sie damit die Latenzzeit, die benötigt wird, um ein Objekt $O = 5000\,\text{Byte}$ zu übertragen (Zeit bis die Daten verfügbar sind), wenn das zweite Fenster komplett verloren geht. Die benötigten Konstanten entnehmen Sie bitte Tabelle 2 sowie der Aufgabe 2.1.

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

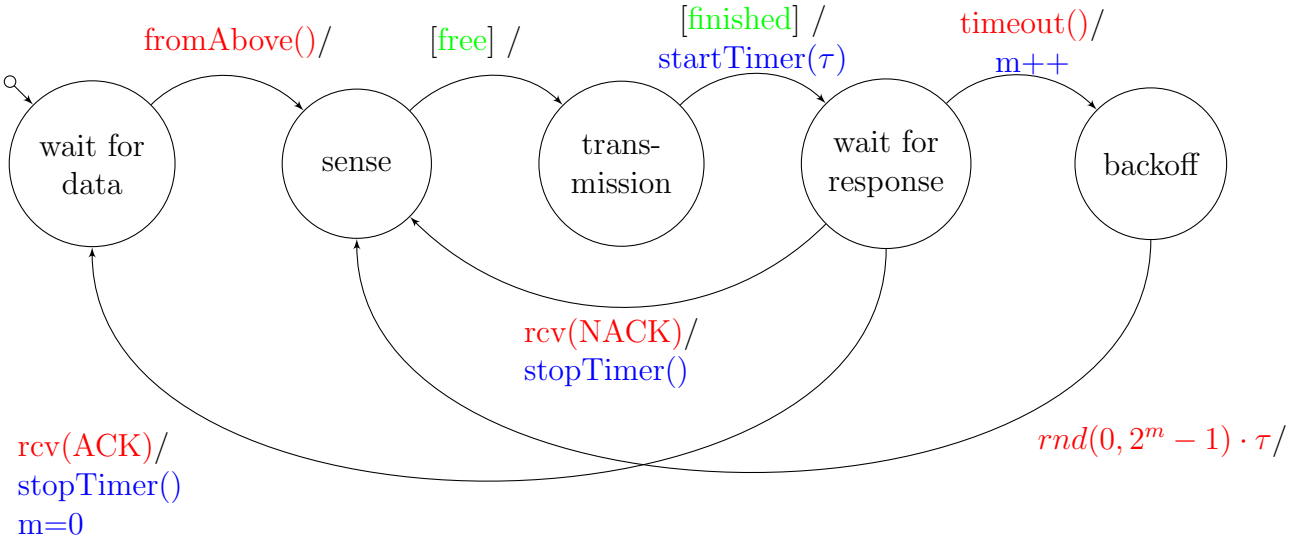

Abbildung 3: Sender-Statechart

3 Sicherungsschicht (35 Punkte insgesamt)

Abbildung 3 zeigt einen leicht veränderten Statechart für einen 1-persistenten *Carrier Sense Multiple Access* (CSMA) Sender. Um den Durchsatz zu erhöhen, wurden sogenannte Not-ACKnowledgement (NACK) eingeführt. Diese werden vom Empfänger versendet, wenn ein Paket nicht korrekt empfangen wurde. Zwischen dem Empfang eines Pakets und dem Antworten (ACK oder NACK) wartet der Empfänger eine Zeit τ. Danach sendet er ohne den Kanal zu prüfen.

3.1 Empfängerchart (4 Punkte)

Vervollständigen Sie den in Abbildung 4 gegebenen Statechart, damit er die oben genannten Spezifikationen erfüllt. Wenn ein Paket richtig empfangen wurde, soll zusätzlich die Methode *toAbove()* aufgerufen werden.

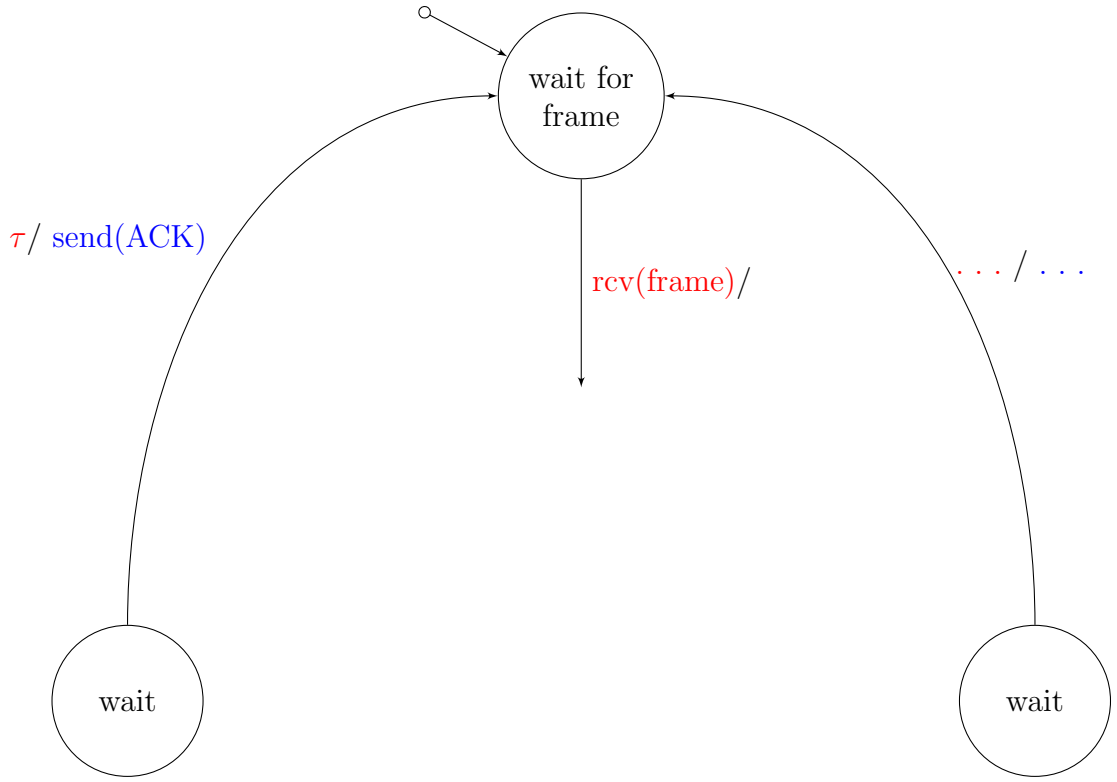

Abbildung 4: Empfänger-Statechart

3.2 Senderprogramm (27 Punkte)

Realisieren Sie das in Abbildung 3 gezeigte Verhalten eines CSMA Senders in JAVA. Leiten Sie dazu die Klasse BehaviorImpl von der Klasse Behavior (Listing 1) ab und implementieren Sie unter Berücksichtigung folgender Aspekte das Verhalten. (Die abstract-Methoden müssen implementiert werden.)

- Die Methode *startTimer(ms)* startet einen Timer, der, wenn die Methode *stopTimer()* nicht aufgerufen wird, nach *ms* Millisekunden die Methode *timeout()* aufruft.
- Die Methode *transmit()* stellt den eigentlich Sendevorgang eines Pakets dar. Sie gibt die Kontrolle zurück, wenn das Senden beendet ist.
- Die Methode *isChannelFree()* gibt *true* zurück, wenn der Kanal frei ist, sonst *false*.
- Die Methode *fromAbove()* wird aufgerufen, wenn die darüberliegende Schicht Daten versenden will.
- Die Methode *rcv(type)* wird aufgerufen, wenn eine Bestätigung vom Empfänger vorliegt. Die Variable *type* ist entweder ACK oder NACK.
- Sie dürfen (wenn benötigt) andere Methode oder Variablen hinzufügen, um das gewünschte Verhalten zu erzielen.
- Zufallszahlen zwischen 0 (inklusive) und i(exklusive) können mit *rnd.nextInt(i)* generiert werden.
- Implementieren sie den Statechart blockierungsfrei. Einzig auf das Freiwerden des Kanals darf aktiv gewartet werden.
- Sollte die Methode *fromAbove()* von der darüberliegenden Schicht aufgerufen werden, während ein anderer Sendevorgang noch nicht abgeschlossen ist, so soll eine Exception zurückgeliefert werden. (*throw new NotReadyException*)

Listing 1: Abstrakte Klasse Behavior

```
import java.*;

public abstract class Behavior
{
        final static int ACK = 1;
        final static int NACK = 2;
        final int        TAU = 100;       //Millisekunden

        int m = 0;                        //Anzahl Fehlversuche
        Random rnd = new Random();

        public class NotReadyException extends Exception {...}
        void startTimer(int ms) {...}     //startet den Timer
        void stopTimer() {...}            //stopt den Timer
        void transmit(String data) {...} //sendet die Daten
        boolean isChannelFree() {...}     //Kanal frei?

        abstract void fromAbove(String data) throws
           NotReadyException;
        abstract void rcv(int type);
        abstract void timeout();

}
```

```
public class BehaviorImpl extends Behavior {
```

3.3 Leistungsanalyse (4 Punkte)

Gehen sie im weiteren von einem CSMA/CD (CSMA mit Kollisionserkennung) Übertragungsverfahren aus. Kann der normierte maximale Durchsatz S_A von dem einfachen ALOHA Protokoll höher sein als der normierte maximale Durchsatz S_C von CSMA/CD? Wenn ja, unter welcher Bedingung? Begründen sie ihre Antwort mit einer Rechnung. Näherungswerte sind hierfür ausreichend.

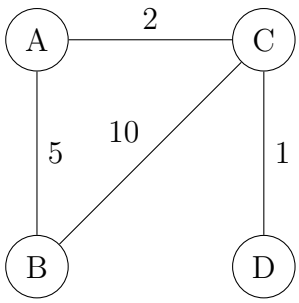

Abbildung 5: Netzwerk

4 Routingverfahren (13 Punkte)

Abbildung 5 zeigt ein Netzwerk mit vier Knoten. Simulieren Sie den Distance-Vector-Algorithmus, um minimale Pfade zwischen den Knoten zu erhalten. 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 3 bis 6 aus: Tabelle 3 enthält die Distanztabellen nach der Initialisierung, Tabelle 4 die Distanztabellen 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. Unbenötigte Tabellen streichen Sie durch.

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

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

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

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

Tabelle 3: Distanztabelle bei der Initialisierung

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

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

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

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

Tabelle 4: Distanztabelle Schritt 1

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

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

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

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

Tabelle 5: Distanztabelle Schritt 2

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

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

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

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

Tabelle 6: Distanztabelle Schritt 3

Zusatzblatt

Zusatzblatt