

### 3. Übungsblatt

### Übung Peer-to-Peer-Systeme und Sicherheit (SS2009)

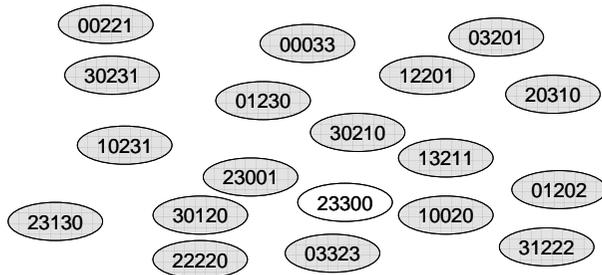
12. Juni 2009  
**Abgabetermin:** Fr. 26.6.  
**Übungstermin:** Do. 2.7.

Dipl.-Inform. Heiko Niedermayer  
 Lehrstuhl für Netzarchitekturen und Netzdienste  
 Technische Universität München

Regeln:  
 Es sind insgesamt 5 Übungsblätter mit je 10 Punkten vorgesehen. Ein Blatt kann von 3 Studenten gemeinsam bearbeitet werden. Für die Übungscredits wird erwartet, dass Aufgaben für 30 Punkte sinnvoll bearbeitet wurden und dass 1x erfolgreich eine Aufgabe vorgerechnet worden ist.

#### Aufgabe 1 (2 Punkte) Pastry

In dieser Aufgabe wollen wir uns Pastry anschauen. In der Abbildung sehen Sie die Knoten mit Ihrer ID an Ihrer Stelle im Underlay. Dies soll verdeutlichen, wer nahe Knoten sind (für Lokalisierungsentscheidung) und welche Knoten existieren. Die dunklen Knoten (= alle bis auf Knoten 23300) seien dabei schon im Pastry-Netzwerk.



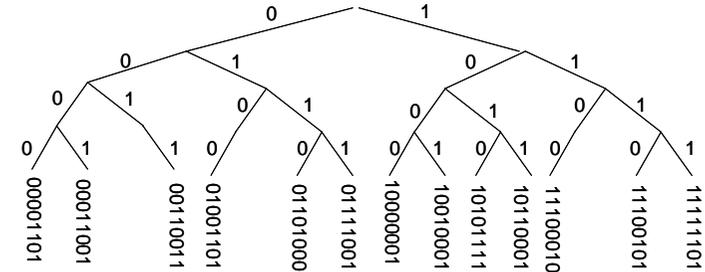
- Der weiße Knoten 23300 will dem Pastry-Netzwerk der dunklen Knoten beitreten und tritt über den ihm bekannten nahen Knoten 10020 bei. Gehen Sie dabei schrittweise den Join-Prozess durch und geben Sie alle relevanten Routing-Informationen der Zwischenknoten an. Aus dieser wird die Routingtabelle von 23300 gebildet und diese stellt das abschließende Ergebnis dar.
- Schicken Sie nun eine Nachricht von 23300 an die ID 03023. Den ersten Schritt des Weges bestimmen Sie aus der Routingtabelle für den Knoten 23300 aus a). Für die restlichen Schritte zum Ziel gehen Sie von jeweils sinnvollen Routingtabellen nach dem Pastry-Verfahren aus ohne diese explizit vorliegen zu haben.

#### Aufgabe 2 (2 Punkte) Ein flexibleres Chord

Beim klassischen Chord zeigen die Finger immer auf den ersten Knoten in den jeweiligen Finger-Intervallen. Da dies keine Auswahl bei der Wahl der Nachbarknoten zulässt, gibt es auch Chord-Varianten, die die Auswahl eines beliebigen Knotens als Finger für ein Intervall zulassen. Damit ist aber nicht mehr sicher gestellt, dass pro Schritt die Distanz zum Ziel halbiert wird, was wir im Beweis für die  $O(\log n)$  Schritte zum Ziel gebraucht haben. Zeigen Sie, dass trotzdem  $O(\log n)$  w.h.p. für die Zahl der Schritte zum Ziel gilt.

#### Aufgabe 3 (2 Punkte) Kademia

Nach Pastry wollen wir hier nun die Vorgehensweise von Kademia durchspielen. Die Bucketgröße sei  $k=2$ . Alpha sei 2. Die IDs seien 8 Bit lang.



- Item 01000001 soll nun gefügt werden. Berechnen Sie seine Distanz (Wert im Dezimalsystem angeben!) nach der XOR-Metrik zu den Knoten 00110011, 01001101 und 10101111 und geben Sie den für das Item verantwortlichen Knoten an.
- Knoten 11010101 soll nun dem Netzwerk hinzugefügt werden. Stellen Sie den zeitlichen Ablauf und das Füllen der Buckets des neuen Knoten dar. Rendezvous-Punkt, über den der neue Knoten den Join-Vorgang startet, sei 00110011.

#### Aufgabe 4 (2 Punkte) Verteilung der Intervallgrößen

In dieser Aufgabe leiten wir die Verteilung der Intervallgrößen in Systemen auf der Basis von Consistent Hashing her (z.B. Chord). Wir gehen dabei von einem reellwertigem ID-Space  $[0,1)$  aus. Die Idee ist, dass wir uns auf Position 0 setzen und dann unser Intervall als Minimum der anderen Knoten vorfinden.

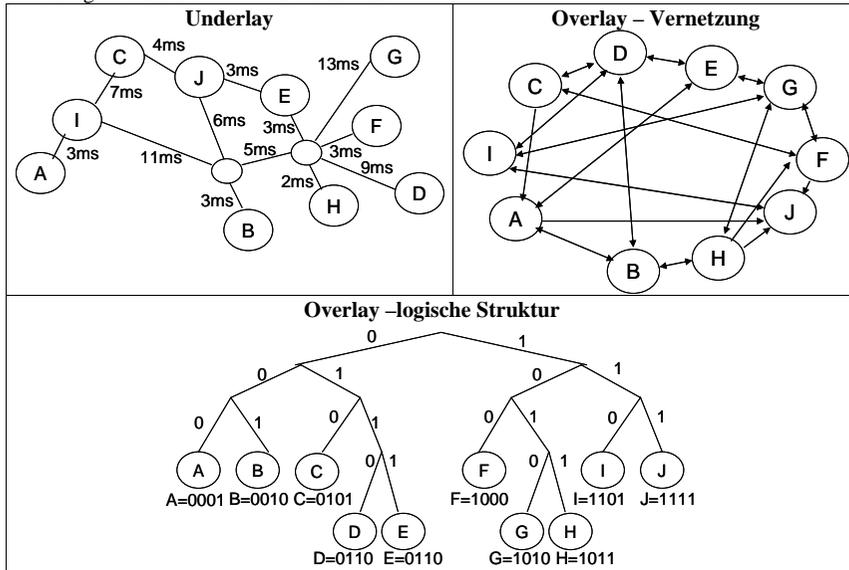
- Die Knoten werden per Zufallszahl gleichmäßig über das Intervall verteilt. Geben Sie die zugehörige Verteilungsfunktion für die Positionierung eines einzelnen Knotens an.
- Berechnen Sie die Verteilungsfunktion für das Minimum über  $n-1$  unabhängige Zufallsvariablen mit der Verteilung aus a). Die Verteilungsfunktion für das Minimum von Zufallsvariablen  $L_1, L_2, L_3, \dots$  ergibt sich über  $L_{\min} = 1 - \prod_i (1 - L_i)$ .
- Leiten Sie nun die Funktion aus b) ab, um die Verteilungsdichtefunktion zu bekommen.

Plotten Sie die Verteilungsdichtefunktion.

--- bitte wenden ---

**Aufgabe 5 (2 Punkte) Stretch und Proximity Neighbor Selection**

Das Bild zeigt einmal das Underlay mit Latenzangaben entlang der Pfade und das Overlay in seiner Vernetzung und seiner logischen Baumstruktur. Das Overlay organisiert sich als Baum analog zur Routingtabelle bei Pastry, aber ohne Neighbor und Leaf Set. Stretch bezieht sich in der Aufgabe immer auf Stretch der Latenz.



- Berechnen Sie den (Latenz-)Stretch für die Anfragen von F nach A und von B nach E.
- Angenommen, jeder Knoten auf dem Weg der Anfragen hätte nun PNS angewendet und dabei für jeden Finger die Latenzoptimalen Knoten aus dem jeweiligen Bereich gewählt. Geben Sie nun den neuen Verlauf der Anfragen aus a) und deren Stretch an.