

Algorithmen für Ad-hoc- und Sensornetze

VL 06 – Routing

Dr. rer. nat. Bastian Katz

Lehrstuhl für Algorithmik I
Institut für theoretische Informatik
Universität Karlsruhe (TH)
Karlsruher Institut für Technologie

3. Juni 2009
(Version 2 vom 5. Juni 2009)

Heute

- » Greedy-Einbettungen in metrische Räume
 - » Wenn Greedy Routing auf echten Koordinaten fehlschlagen kann, gibt es Koordinaten, auf denen es funktioniert?
 - » zwei exotische Exemplare
- » Pseudo-Geometrisches Routing
 - » Greedy Routing ohne *metrische* Einbettung: Einfache Koordinaten, Routing ohne Tabellen
 - » Theoretisches: Schranken
 - » Praktisches: Beacon-Vector-Routing
- » Compact Routing
 - » Routing mit Routingtabellen
 - » Ein neues Modell: Bounded Doubling Dimension

Greedy-Einbettungen

Müssen Koordinaten immer plausibel sein? Reicht nicht *zweckdienlich*?

Greedy Routing

Weiterleiten der Nachricht an den Nachbarn, der dem Ziel am nächsten liegt (Greedy Routing) ist immer erfolgreich genau dann, zu jedem Knotenpaar s, t ein Nachbar p von s existiert mit $d(p, t) < d(s, t)$.

Greedy-Einbettungen

Eine Einbettung in einen metrischen Raum, in der zu jedem Knotenpaar s, t ein Nachbar p von s existiert mit $d(p, t) < d(s, t)$ heißt *Greedy-Einbettung*. (Erstmal: metrisch=euklidisch)

Wann gibt es Greedy-Einbettungen?

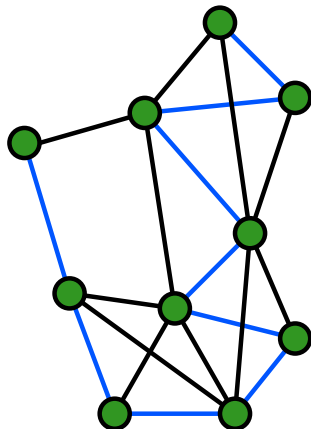
Satz

Enthält ein Graph einen hamiltonschen Pfad, gibt es eine Greedy-Einbettung in die Ebene.

» Beweis: Sei v_1, v_2, \dots ein hamiltonscher Pfad, dann ordne den Pfad auf einer Geraden an:

$$p(v_i) := (i, 0)$$

» schon Greedy-Einbettung in \mathbb{R}



Wann gibt es Greedy-Einbettungen?

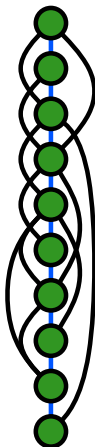
Satz

Enthält ein Graph einen hamiltonschen Pfad, gibt es eine Greedy-Einbettung in die Ebene.

- Beweis: Sei v_1, v_2, \dots ein hamiltonscher Pfad, dann ordne den Pfad auf einer Geraden an:

$$p(v_i) := (i, 0)$$

- schon Greedy-Einbettung in \mathbb{R}

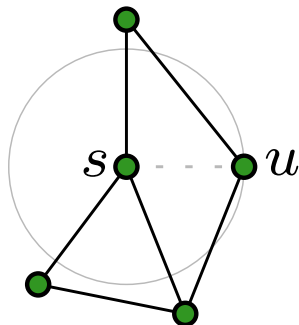


Wann gibt es keine Greedy-Einbettungen?

Lemma

In Greedy-Einbettungen ist jeder Knoten mit dem dichtesten Knoten verbunden.

- » Beweis:
 - » Sei ein Knoten s nicht mit dem dichtesten Knoten u verbunden.
 - » Knoten u hat keinen Nachbarn, der dichter an s ist als er selbst.

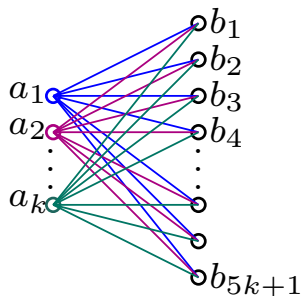


Wann gibt es keine Greedy-Einbettungen?

Satz

Die Graphen $K_{k,5k+1}$ haben keine Greedy-Einbettungen in der Ebene.

- » Angenommen, ein $K_{k,5k+1}$ wäre so eingebettet:
 - » Ordne jedes b_i dem dichtesten a_j zu
 - » Einem a^* sind ≥ 6 b_i zugeordnet und
 - » ein Winkel zwischen zwei b_i ist $< \frac{\pi}{6}$

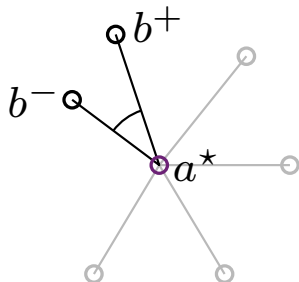


Wann gibt es keine Greedy-Einbettungen?

Satz

Die Graphen $K_{k,5k+1}$ haben keine Greedy-Einbettungen in der Ebene.

- » Angenommen, ein $K_{k,5k+1}$ wäre so eingebettet:
 - » Ordne jedes b_i dem dichtesten a_j zu
 - » Einem a^* sind ≥ 6 b_i zugeordnet und
 - » ein Winkel zwischen zwei b_i ist $\leq 60^\circ$
 - » nenne diese b^- (dichter) und b^+ (weiter weg)
 - » b^+ ist dichter an b^- als an a^* (und damit an allen a_j)
 - » dichtester Knoten an b^+ ist ein b_i
 - » Widerspruch!



Dreifach zusammenhängende planare Graphen

Vermutung [Papadimitriou et. al. 04] (inzwischen gelöst)

Enthält ein Graph einen dreifach zusammenhängenden planaren Subgraphen, lässt er sich greedy in die Ebene einbetten.

- » kein Widerspruch zur Vermutung:
- » $K_{1,6}$ und $K_{2,11}$ sind nicht dreifach zusammenhängend
- » $K_{3,16}, \dots$ sind nicht mehr planar.

Offene Frage

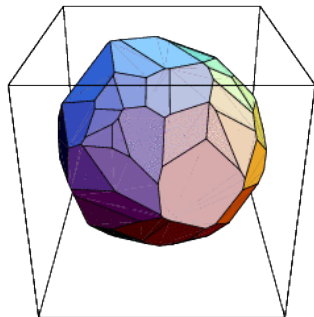
Selbst wenn das so ist: Gibt es eine bessere Charakterisierung von Graphen, die sich greedy in \mathbb{R}^2 einbetten lassen?

Exot I: Einbettung in \mathbb{R}^3

Satz

Jeder dreifach zusammenhängender planare Graph läßt sich so in den \mathbb{R}^3 einbetten, daß Greedy Routing mit einer geeigneten Distanzfunktion immer erfolgreich ist.

- » Beweis:
 - » Jeder dreifach zshg. planare Graph ist Kantengraph eines konvexen Polytops
 - » sogar so, dass alle Kanten eine gemeinsamen Kugel (um 0) berühren
 - » setze $d(t, v) = \vec{t} \cdot \vec{v}$
- » keine Metrik, schwer zu verteilen

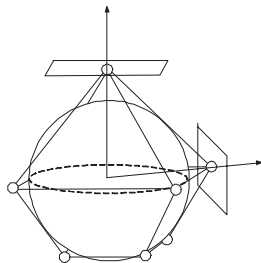


Exot I: Einbettung in \mathbb{R}^3

Satz

Jeder dreifach zusammenhängender planare Graph läßt sich so in den \mathbb{R}^3 einbetten, daß Greedy Routing mit einer geeigneten Distanzfunktion immer erfolgreich ist.

- » Beweis:
 - » Jeder dreifach zshg. planare Graph ist Kantengraph eines konvexen Polytops
 - » sogar so, dass alle Kanten eine gemeinsamen Kugel (um 0) berühren
 - » setze $d(t, v) = \vec{t} \cdot \vec{v}$
- » keine Metrik, schwer zu verteilen

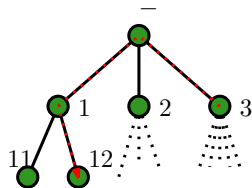


Exot II: Routing in der Hyperbolischen Ebene

Tree-Routing

- » Vorbereitung
 - » Berechne aufspannenden Baum
 - » Wähle Wurzel w beliebig
 - » Identifiziere jeden Knoten mit Abstiegen von w
- » Routing
 - » ist t Nachfolger, steige in entsprechenden Teilbaum ab
 - » sonst steige auf

- » Leicht zu verteilen
- » schlechter Worst-Case-Stretch: $\Theta(n)$
- » nicht *greedy* in \mathbb{R}^2 möglich ($K_{1,6}$)

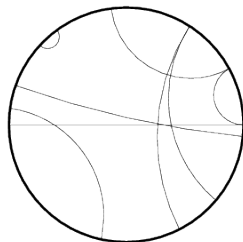


Tree-Routing in der Hyperbolischen Ebene

Poincaré-Modell der Hyperbolischen Ebene

$$H = \{x \in \mathbb{R}^2 : |x| < 1\}$$
$$d(u, v) = \operatorname{arccosh} \left(1 + \frac{2\|u - v\|}{(1 - \|u\|)(1 - \|v\|)} \right)$$

- Punkte im Einheitskreis
- Geraden: Kreise, die Rand orthogonal schneiden

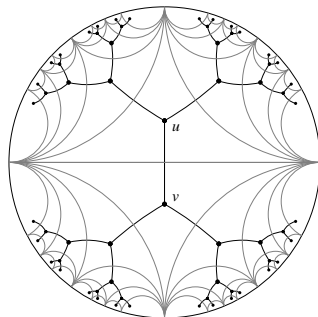


Tree-Routing in der Hyperbolischen Ebene

Satz (nur zum Staunen!)

d -reguläre Bäume haben *greedy embeddings* in der Hyperbolischen Ebene.

- » Damit haben alle zshg. Graphen *greedy embeddings*:
 - » Jeder Baum ist Teil eines d -regulären Baumes
 - » Jeder Graph ist ein Baum mit zusätzlichen Kanten



Pseudo-geometrisches Routing

Die einfachsten Koordinaten, mit denen wir einen Knoten identifizieren können, sind Hop-Abstände zu ausgewählten Knoten!

- » Wähle Ankerknoten a_1, a_2, \dots, a_k ,
- » Jeder Ankerknoten flutet das Netz zur Abstandsmessung
- » Knoten u hat Koordinaten $(d_G(u, a_1), \dots, d_G(u, a_k)) \in \mathbb{Z}^k$

Gute Ankerknoten

Eindeutige Namen

Ankerknoten müssen garantieren, dass jeder Knoten *eindeutige* Koordinaten $(d_G(u, a_1), \dots, d_G(u, a_k))$ zugewiesen bekommt.

Routing

Ankerknoten sollen eine lokale Routing-Regel ermöglichen.

- Jeder Knoten entscheidet auf Basis seiner Koordinaten und der Koordinaten seiner Nachbarn (wie beim Georouting).

Warm-up: Pfade

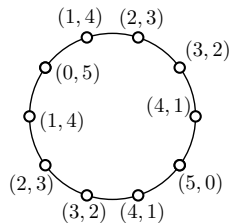
- Bei gezielter Wahl reicht 1 Ankerknoten für eindeutige Namen
 - Wähle Knoten am Ende des Pfades!
 - Routing dann klar (oder?)
- Bei zufälliger Wahl reichen 2 Ankerknoten immer
 - Wenn zwei Knoten u, v dieselbe Koordinate $d_G(u, a_k) = d_G(v, a_k)$ haben, dann liegt a_k genau in der Mitte (und da liegt maximal ein a_k).
 - lokales Routing auf kürzestem Weg möglich
 - aus lokaler Information lassen sich Positionen der Anker ableiten
 - dann ist auch klar, in welche Richtung geroutet werden muss

(2, 4) (1, 3) (0, 2) (1, 1) (2, 0) (3, 1) (4, 2) (5, 3) (6, 4)



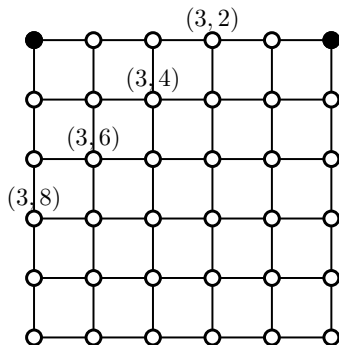
Zum Üben: Ringe

- » Es reichen bei gezielter Wahl 2, bei zufälliger 3 Knoten
 - » (fast) dasselbe Argument: Haben zwei Knoten dieselben Koordinaten, dann liegt der entsprechende Anker gleich weit von beiden entfernt
 - » so ein Problem entsteht bei einem Knoten immer
 - » bei zwei Knoten nur dann, wenn die Anker sich gegenüberliegen!
 - » lokales Routing wieder auf kürzestem Weg möglich!
 - » (Alles andere wäre auch traurig!)



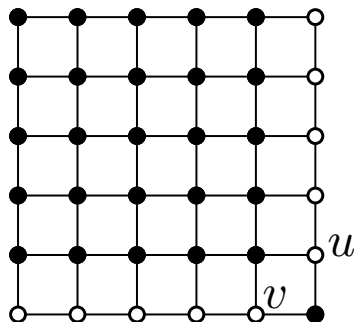
Quizfrage: Eindeutige Namen im Gitter

- » Bei gezielter Wahl reichen 2 Anker
 - » Knoten in den beiden oberen Ecken benennen eindeutig
 - » lokales Routing wieder einfach!
- » erst $\Theta(n)$ beliebige Knoten lösen das Problem immer!
 - » Knoten u, v haben identische Abstände zu allen Ankern!
 - » (Das Routingproblem klammern wir mal aus)



Quizfrage: Eindeutige Namen im Gitter

- » Bei gezielter Wahl reichen 2 Anker
 - » Knoten in den beiden oberen Ecken benennen eindeutig
 - » lokales Routing wieder einfach!
- » erst $\Theta(n)$ beliebige Knoten lösen das Problem immer!
 - » Knoten u, v haben identische Abstände zu allen Ankern!
 - » (Das Routingproblem klammern wir mal aus)

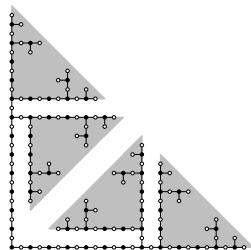


Jackpot: Wieviele Knoten braucht man in UDG?

Unit-Dist-Trees

Rekursiv definierte Bäume auf dem Gitter mit $\Theta(n)$ Blättern. (ohne Beweis)

- » Zwei benachbarte Blätter sind nur unterscheidbar, wenn einer der beiden Anker ist!
- ⇒ selbst bei gezielter Wahl braucht man die Hälfte der Blätter, also $\Theta(n)$ Anker!
- » bei zufälliger Wahl schlimmstenfalls alle!
- » (dazu reicht schon ein kleiner Baum am Rand eines Netzes!)



Kurze Bilanz

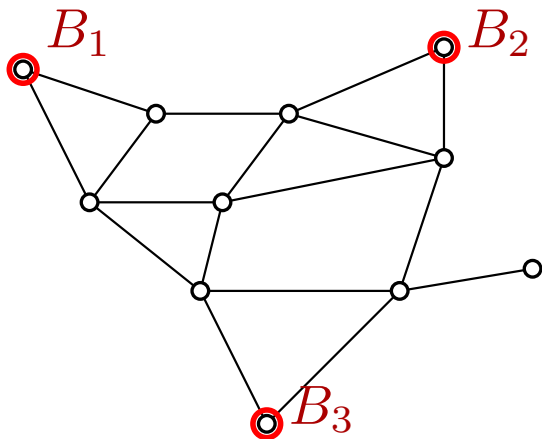
- » Gute Anker zu garantieren ist sehr schwer
 - » bei zufälliger Wahl sind im schlechtesten Fall alle Knoten nötig, sobald die Instanz einen kleinen Baum enthält
 - » UDT geben sogar Schranke bei gezielter Auswahl
- » Widerspricht der Praxis!
 - » Netze sind keine UDT
 - » (allerdings auch keine Gitter)
 - » schon verhältnismäßig wenige Anker reichen meist aus, um fast alle Knoten eindeutig zu benennen!

Beacon Vector Routing

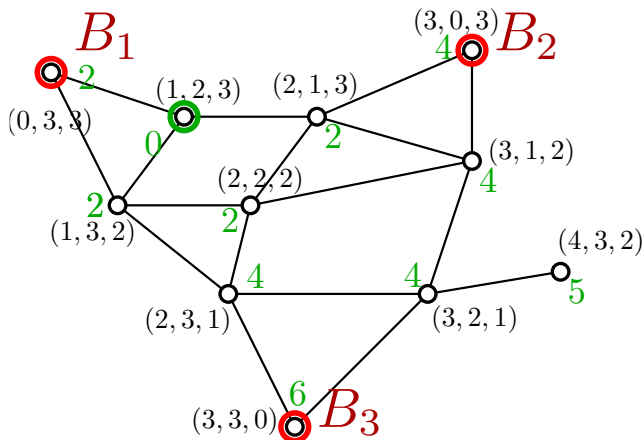
Beacon Vector Routing

- 1 r Anker a_i fluten das Netz (zufällig oder heuristisch gewählt)
 - » Knoten *kennen* kompletten Abstandsvektor $(d_G(u, a_1), \dots, d_G(u, a_k))$ von sich und Nachbarn
 - » Adressen bestehen aus *auseindeutiger ID* des Knotens und Abständen zu den $k < r$ dichtesten Ankern (und deren IDs)
- 2 Knoten wählen Nachbarn, der metrische Abstandsfunktion $d : \mathbb{Z}^k \rightarrow \mathbb{R}^+$ verbessert, wenn möglich
- 3 wenn keine Verbesserung möglich, route in Richtung des dem Ziel dichtesten Ankers
 - » Paket enthält bisher minimalen Abstand δ^-
 - » wird der unterschritten, wird wieder greedy geroutet
- 4 wenn das Paket den Anker erreicht, wird lokal geflutet
 - » Radius ist bekannt!

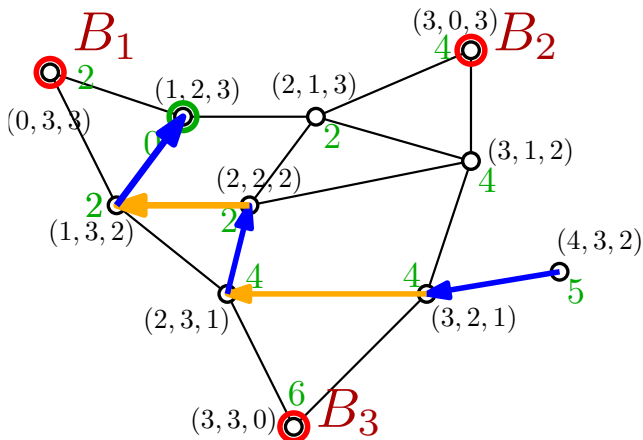
Ein einfaches Beispiel (L_1 -Metrik)



Ein einfaches Beispiel (L_1 -Metrik)



Ein einfaches Beispiel (L_1 -Metrik)



Alternative Metriken

Beobachtung

Die Information von Landmarken, denen man näherkommen will, ist wertvoller als die Information von Landmarken, von denen man sich entfernen muss.

- Intuition: Es gibt weniger Wege zu einem Anker als von einem Anker weg

Alternative Metrik

$$d(v, t) = \sum_{i=1}^k C \max(v_i - t_i, 0) + \max(t_i - v_i, 0) \quad C > 1$$

Compact Routing

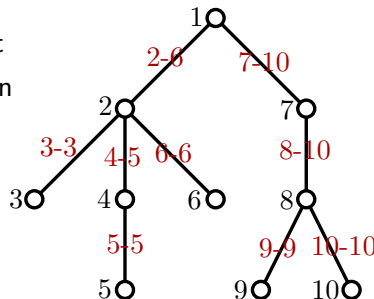
Als *kompakt* bezeichnet man alle Routingverfahren, die garantiert mit Routingtabellen mit $o(n)$ Bits lokalem Speicher auskommen.

- Knoten entscheiden auf Basis von Zieladresse und Information im Speicher, an welchen Nachbarn sie ein Paket schicken
- Unterscheidung: labeled vs name-independent
 - labeled: Knoten können IDs wählen
 - name-independent: Knoten müssen mit ursprünglichen IDs adressiert werden

Tradeoff lokaler Speicher / Routing Stretch!

Einfaches Beispiel: Interval Routing

- beliebige Wurzel spannt Baum auf
- Knoten werden in Pre-order numeriert
- jeder Knoten merkt sich, welche IDs in welchem Teilbaum liegen
- alle anderen IDs werden Richtung Wurzel geroutet
- $O(\deg(v) \log n)$ Bits reichen aus!
- aber: Wege können sich um Faktor $\Theta(n)$ verlängern! (Beispiel?)



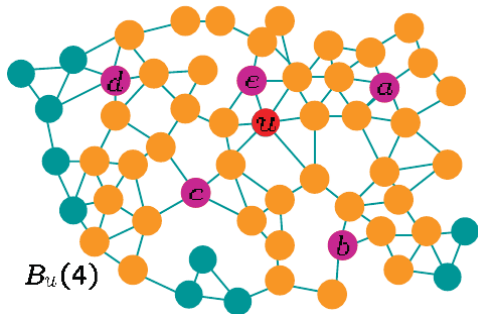
Klassisches Resultat (ohne Beweis)

In allgemeinen Graphen ist mit $o(n)$ Bits kein Stretch < 3 für alle Knotenpaare möglich!

Doubling Dimension

Definition

Ein Graph hat „Doubling Dimension“ α , wenn sich die k -hop-Nachbarschaft jedes Knotens mit maximal 2^α $k/2$ -hop-Nachbarschaften von anderen Knoten abdecken lässt.



Ab jetzt gehen wir von einer konstant beschränkten DD aus!

Knobelaufgabe

Gibt es UDG, die keine konstant beschränkte DD haben?

Bild: Roger Wattenhofer

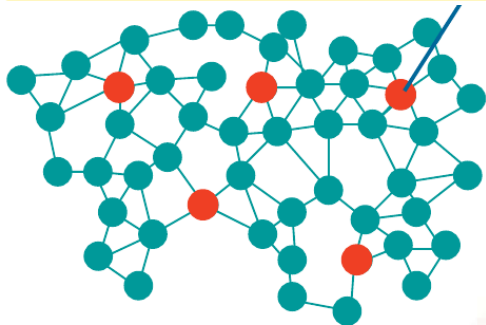
Bastian Katz – Algorithmen für Ad-hoc- und Sensornetze

ρ -Netze

Definition

Ein ρ -Netz ist eine Knotenteilmenge $U \subset V$ (Zentren) mit

- » für jedes $v \in V$ gibt es ein $u \in U$ mit $d_G(u, v) \leq \rho$
- » alle $u_1, u_2 \in U$ haben Abstand $d_G(u_1, u_2) > \rho$



Zentral leicht zu berechnen,

- » Entferne sukzessive Knoten u mitsamt ρ -Nachbarschaft aus G .

verteilte Verfahren kommen noch..

Bild: Roger Wattenhofer

Bastian Katz – Algorithmen für Ad-hoc- und Sensornetze

ρ -Netze

Wähle ρ -Netze für $\rho = 1, 2, 4, 8, 2^{\lceil \log D \rceil}$

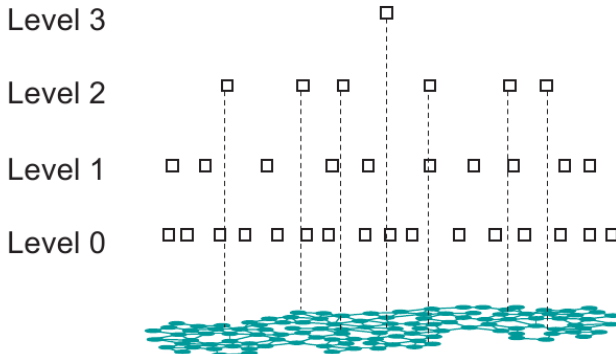


Bild: Roger Wattenhofer

Bastian Katz – Algorithmen für Ad-hoc- und Sensornetze

Adressen und Baumstruktur

- Jeder Knoten aus einem ρ -Netz flutet seine 2ρ -Nachbarschaft
- Knoten merken sich alle ankommenden Nachrichten
 - Das sind nur konstant viele pro Ebene (nicht ganz trivial)
 - ⇒ Routingtabelle mit $O(\log \deg(v) \cdot \log D)$ reicht aus, um an alle bekannten Zentren zu erreichen

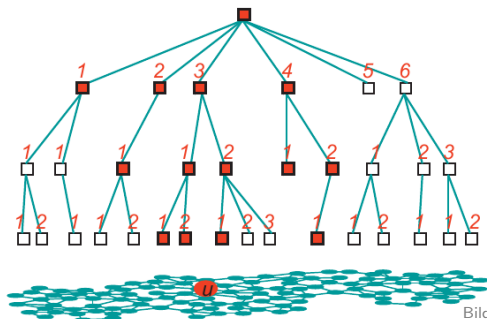


Bild: Roger Wattenhofer

Bastian Katz – Algorithmen für Ad-hoc- und Sensornetze

Adressen und Baumstruktur

- Jeder Knoten aus ρ -Netz wählt dichtesten Knoten aus 2ρ -Netz als Vorgänger
- Jeder Knoten aus ρ -Netz nummeriert seine Nachfolger
 - Damit hat jeder Knoten eine ID aus $O(\log \Delta \cdot \log D)$ Bits

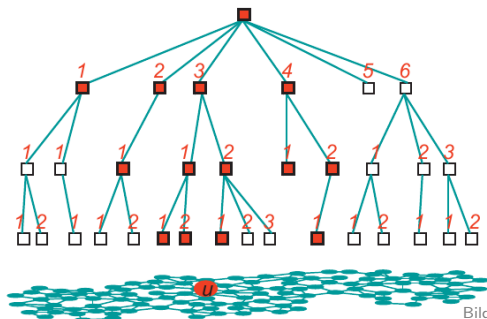


Bild: Roger Wattenhofer

Bastian Katz – Algorithmen für Ad-hoc- und Sensornetze

Routing

Um ein Paket zu einem Knoten zu routen, wird es zum niedrigsten bekannten Zentrum geschickt, das in der Adresse des Zielknotens auftaucht.

- Bsp: u schickt ein Paket an $R : 2 : 1 : 1$ direkt an $R : 2 : 1$
- Jedes Level- i -Zentrum kennt alle adressierbaren Level- $i - 1$ -Zentren
- Stretch ist konstant, kann bis auf $1 - \epsilon$ reduziert werden (o.B.)

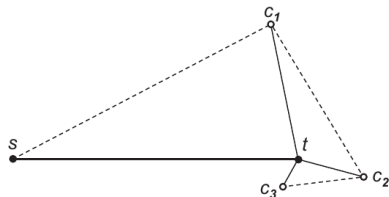
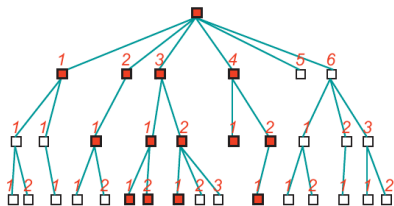


Bild: Roger Wattenhofer

Bastian Katz – Algorithmen für Ad-hoc- und Sensornetze



Zum Mitnehmen

- » Greedy-Einbettungen
 - » Wann kann man einen Graphen so einbetten, dass klassisches Greedy-Routing immer funktioniert?
 - » Vor allem theoretisch interessantes Problem
- » Pseudo-Geometrisches Routing
 - » Theorie: Auf Basis von Anker-Anständen zu routen ist nicht so leicht umzusetzen (viele Anker, Routing-Funktion?)
 - » Praxis: Beacon-Vector-Routing sehr einfach und erfolgreich
- » Compact Routing
 - » klassische Untersuchung des Tradeoffs zwischen Stretch und lokalem Speicher
 - » Neue Graphenmodellen lassen bessere Lösungen zu!

Literatur

- 1 C. H. Papadimitriou, D. Ratajczak: *On a conjecture related to geometric routing*. In: Theor. Comput. Sci., 344(1):3–14, 2005
- 2 M. Wattenhofer, R. Wattenhofer, P. Widmayer: *Geometric Routing without Geometry*. In: 12th Colloquium on Structural Information and Communication Complexity (SIROCCO), 2005
- 3 R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, I. Stoica: *Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensor networks*. In: Proceedings of the Second USENIX Symposium on Networked Systems Design and Implementation (NSDI), 329–342, 2005
- 4 R. Flury, R. Wattenhofer: *Routing, Anycast, and Multicast for Mesh and Sensor Networks*. In: 26th Annual IEEE Conference on Computer Communications (INFOCOM), 2007