Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Additional Material for

Ad Hoc and Sensor Networks

Roger Wattenhofer
wattenhofer@tik.ee.ethz.ch

Fall 2010

Chapter 2

Geo-Routing

2.1 Slide 2/34

Theorem 2.1. *Let the cost of a best route for a given source destination pair be c. Then any deterministic (randomized) geo-routing algorithm has (expected) cost* $\Omega(c^2)$.

Proof. We are given a positive integer k, and define an Euclidean graph G (see figure on slide 2/34): On a circle, we evenly distribute $2k$ nodes such that the distance between two neighboring points is exactly 1; thus the circle has radius $r \approx k/\pi$. For every second node of the circle we construct a chain of $\lceil r/2 \rceil - 1$ nodes. The nodes of such a chain are arranged on a line pointing towards the centre of the circle; the distance between two neighboring nodes of a chain is exactly 1. Node w is one arbitrary circle node with a chain: The chain of w consists of $\lceil r \rceil$ nodes with distance 1. The last node of the chain of w is the center node; note that the edge to the center node does not need to have distance 1.

Please note that the unit distance graph consists of the egdes on the circle and the edges on the chains only. In particular there is no edge between two chains because all chains except the w chain end strictly outside radius $r/2$. Also, the graph is a $\Omega(1)$-graph. Note that the graph has k chains with $\Theta(k)$ each.

We route from an arbitrary node on the circle (the source s) to the center of the circle (the destination t). An optimal route between s and t follows the shortest path on the circle until it hits node w, and then directly follows w's chain to t with link cost $c \leq k + r = O(k)$. An ad-hoc routing algorithm with routing tables at each node will find this best route.

A geometric ad-hoc routing algorithm needs to find the "correct" chain w. Since there is no routing information stored at the nodes, this can only be done by exploring the chains. Any deterministic algorithm needs to explore the chains in a deterministic order until it finds the chain w. Thus, an adversary can always place w such that w's chain will be explored as the last, and therefore explore $\Theta(k^2)$ (instead of only $O(k)$) nodes.

The argument is similar for randomized algorithms. By placing w accordingly (randomly!), an adversary forces the randomized algorithm to explore $\Omega(k)$ chains before chain w, with constant probability. Then, the expected link cost of the algorithm is $\Omega(k^2)$.

Thus the (expected) link (or distance) cost is $\Theta(c^2)$. $\square$