## Exercise 2

# Exercises Peer-to-Peer-Systems and Security (SS2012)

Thursday 21.5. 2012 **Hand-in**: Thursday 31.5. in lecture or by mail to niedermayer – at – net.in.tum.de **Exercise:** Monday 4.6. in lecture (SS2012) Dr. Heiko Niedermayer Lehrstuhl für Netzarchitekturen und Netzdienste Technische Universität München

<u>Rules:</u> There will be four exercise sheets. You have to hand-in 70 % of the assignments, attend at least 3 exercise courses and present a solution in the exercise course to get the 0.3 bonus. Up to 3 persons may hand in one sheet.

### Task 1 CoolSpots Munich III

This time you should solve the task to organize the CoolSpots Munich network with a Distributed Hash Table. Use  $spot_ID = h(GPS \text{ coordinate of spot})$  to store the items. As item descriptions are rather short, the data is stored on the DHT nodes and not only on the node that contributed the item.

- a) How can you find an item that is directly at your GPS coordinate?
- b) Items at my exact GPS coordinates are not too useful, propose a change to the item storage and lookup mechanism to efficiently find items close to a given GPS coordinate?
- c) Now, assume a multi-dimensional ID space. What is changing?

#### Task 2 Key-based-Routing-API

Please, briefly describe your idea first, before you write pseudocode. Use a pseudocode that avoids unnecessary details. Assume that you have a structure Peer-to-Peer system that implements the Key-based Routing-API. All messages are processed recursively (e.g. no iterative lookup):

- a) *Ping and Pong:* a node A pings (sends "Ping") a node B via the KBR network by sending message to its ID. Node B replies to the ID of node A with a "Pong". Give the code for the send and receive operation.
- b) *Counter:* implement a counter that counts the number of hops (in the overlay) that the ping message takes.

## Task 3 Consistent Hashing – Distribution of Interval sizes

In this task we want to compute the distribution of the interval size in systems on the basis of Consistent Hashing like Chord. Let us assume the ID space to be real-valued in the interval [0,1). Without loss of generality we can put our node on the position 0 in the ID space.

- a) Nodes are positioned randomly on the basis of uniform random numbers. What the cumulative distribution function (CDF) L of the corresponding uniform distribution.
- b) Now calculate the CDF for the minimum of n-1 independent experiments with the distribution from a). Hint: The CDF for the minimum von random variables with CDFs L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>, ... is given by the forumla  $L_{\rm min} = 1 \prod (1 L_i)$ .
- c) Now differentiate L\_min to get the probability density function l\_min.
- d) Plot the probability density function l\_min.

# Task 4 A flexible Chord

The finger table entries in the classic Chord algorithm always point to the first node in the corresponding finger interval. This does not allow the freedom to select a finger among multiple peers. Yet, there are proposals to allow Chord to link to any node in the interval of the finger. If you remember the proof for the complexity of the lookup of O(logn) in Chord, we needed that the distance is halved per step.

Show that despite of that change, Chord still achieves O(logn) hops with high probability.

# Task 5 Distance and links

The more links each node in a DHT has, the shorter the distance. Yet, one can gain more or gain less depending on how good a strategy is.

Assume first, that a node sits on a ring-like ID space which we simplify to the interval [0,1). Each node links to its successor and predecessor. Each node has 100 long distance links. As long-distance links each node links to nodes i=1..100 in the distance i/100.

- a) Does this approach achieve logarithmic distance?
- b) What happens if you lose contact to your successor?

Assume now that you apply a different strategy and add links in distances  $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{3}{4}$ ,  $\frac{13}{16}$ ,  $\frac{14}{16}$ ,  $\frac{15}{16}$ ,  $\frac{61}{64}$ ,  $\frac{62}{64}$ ,  $\frac{63}{64}$ , ... The basic idea is to divide the first interval and then the most distant intervals into quarters.

c) How does this approach affect the distance? (Hint: What reduction is achieved within one hop.)