# Exercise 3

Peer-to-Peer Networks, Spring 2008

## Guidelines

- Group work allowed, groups of up to 3 people allowed
- Due date: 7.3. at 12:00
- Exercise is worth 18 points
- What to return: One zip- or tar-file which contains 4 files. One of the 4 files is a text file (plain text) with answers to the written questions below. The other 3 files contain the object-to-node mappings for each of the three parts *in the format specified below.*
- Don't forget to write all the names and student numbers on the answer!
- Return by email to

## General Information

In this exercise, you will take a look at how different DHTs map objects to nodes, and compare them. We will look at 3 different DHTs, Chord, CAN, and Tapestry. For each one of them, you need to report the distribution of objects to nodes, and compare them in terms of how well they distribute the objects.

On the web site you will find a zip-file exercise3.zip which contains files for this exercise, two for each of the parts. For each part, one of the two files will contain node IDs and the other will contain the object IDs.

## File Formats

The files on the web site contain simply a column of numbers (either node or object IDs). The files with the word "node" in the name are node IDs and the others, with the word "objects" in the name are object IDs. For CAN, the numbers give 2-dimensional coordinates separated by a comma (e.g., 23756,37696 means the point with x-coordinate 23756 and y-coordinate 37696).

For your answers, use the following file format. Put one line per node, start the line with the node ID, followed by a colon (:), and then all the objects that were mapped to that node. Use plain text. For example: node1: o1 o2 o3 o4. The answer file should have one line for each node. Keep the leading zeros in the node and object identifiers. The order of nodes in the file and objects in the line of a node do not matter.

**If your returned files do not follow the above format, you will not get any points for those parts of the exercise**.

*Name your files in such a way that the filename tells which DHT is in that file.* For example, call them chord.txt, can.txt, and tapestry.txt.

## Question 1: Chord, 2 points

The files for Chord are called chord-node.txt and chord-objects.txt. The IDs are distributed between 0 and 99999. Map the objects to the correct nodes and provide your answer in the above format.

## Question 2: CAN, 3 points

The files for CAN are can-node.txt and can-objects.txt. We use a 2-dimensional CAN, so each line contains two coordinates separated by a comma. Assume that each node is the center of the zone for which that node is responsible. Use the standard Euclidean distance as a metric and map the objects to their closest nodes. Nodes are inserted consecutively from top to bottom of the .txt file. If a zone is square shaped and needs to be split, the zone will be divided vertically. If a zone is a rectangle, it will be split horizontally and the result will be two squares.

## Question 3: Tapestry, 5 points

The files for Tapestry are called tapestry-node.txt and tapestry-objects.txt. The IDs in these files are all 5-digit numbers, i.e., distributed between 00000 and 99999. This means that your Tapestry network uses base 10 numbers and has 5 digits in the namespace, i.e., 10 columns and 5 levels in the neighbor map.

You need to accomplish the following steps:

- Create neighbor maps for all of the nodes. When several nodes are possible candidates for an entry in the neighbor table, pick the node with the smallest ID among them.
- Map each object to its responsible node. For each object, pick a starting node (at random) and route the object according to the neighbor maps all the way to its responsible node.

When there is no entry in the neighbor map for the next hop, pick the closest entry in the neighbor map for that level which has an entry (can be the node itself). In case of ties, pick the smaller of the two.

Return the object-to-node mapping in the above format.

## Written Questions, 8 points

Answer each of the following questions. Questions 1, 2, 3, and 5 are worth 1 point each. Question 4 is worth 4 points.

1. A 1-dimensional CAN forms a ring, just like Chord. However, there is one fundamental difference in how Chord and 1-dimensional CAN distribute objects to nodes. What is it?

2. Which of the three DHTs provides the best load balancing? As a metric for evaluating the load balance, use the standard deviation of the number of objects stored on each node.

3. What is the neighbor table for node 28614 in the Tapestry exercise? Is it unique? If yes, why? If not, what would be needed to make it unique?

4. What is wrong with the following Tapestry neighbor table? Use base 10 numbers and assume that no other nodes exist besides the ones in the table. Pick out the entries which are incorrect, justify why they are incorrect, and give a correct entry (some cases may have several alternatives for the correct entry; only one needs to be given). Surrogates are assumed to be correct unless you have proof to the contrary.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 00000 | 17366 | 25572 | 03483 | 45768 | 50008 | 68326 | 75572 | 84622 | 29245 |
| 02008 | 21632 | 25572 | 23018 | 24486 | 25623 | 26473 | 27567 | 28390 | 29245 |
| 25012 | 25103 | 25245 | 25319 | 25409 | 25535 | 25637 | 25782 | 25818 | 25991 |
| 25502 | 25513 | 25529 | 25529 | 25547 | 25557 | 25557 | 25572 | 25583 | 25583 |
| 25570 | 25570 | 25572 | 25572 | 25572 | 25572 | 25572 | 25572 | 25587 | 25578 |

5. Which node owns the above neighbor table? Justify your answer.