Prof. Dr. Thorsten Strufe
Stefanie Roos
{strufe,roos}@cs.tu-darmstadt.de

Fachgebiet Peer-to-Peer Netzwerke
Fachbereich Informatik
Technische Universität Darmstadt

Exercise 5

for **Peer-to-Peer Networks** - winter term 2012/2013

(20.11.2012)

Deadline for submission: 27.11.2012 1:30 PM

H#5.1 Chord

Consider a Chord network with a 4-bit identifier space. The network currently consists of 5 nodes with the following identifiers 0, 3, 8, 10, and 11.

a) Network structure (3 P.)

Sketch a graph $G = (E,V)$ with $E = E_n \cup E_f$ which corresponds to the networks structure.
Use normal arcs E_n for the successor links (predecessor links should be omitted).
Insert finger links using dashed arcs E_f.
Draw duplicate links only once and use normal arcs if a link is both successor and finger link.

The resulting graph is planar, so draw it planary!

Solution: The list of out-neighbors for each node

- 0: 3, 8
- 3: 8, 11
- 8: 10, 0
- 10:11, 0, 3
- 11: 0, 3

b) Characteristic Path Length (3 P.)

Calculate the characteristic path length of the graph G. Therefore you can use a table similar to Table **??**, to denote the distances from node o to node p first.

Is a chord network always strongly connected?

Solution:

	0	3	8	10	11
0		1	1	2	2
3	2		1	2	1
8	1	2		1	2
10	1	1	2		1
11	1	1	2	3	

The characteristic path length is hence given by $\frac{1*11+2*8+3*1}{5*4} = \frac{30}{20} = 1.5$. Assuming no failures, a chord network is always connected because of the underlying ring topology.

c) Average Routing Length (4 P.)

Now calcute the average routing length, meaning the mean average of hops that is needed to really *route* a package from *o* to node *p*.

The calculation can be done similar to the average length of shortest paths using Table **??** and the formula.

Is the routing in chord done in an optimal way? If not, why can chord not simply be modified to always use the very shortest path? *Solution:*

	0	3	8	10	11
0		1	1	2	3
3	2		1	2	1
8	1	2		1	2
10	1	1	2		1
11	1	1	2	3	

The average routing length is hence given by $\frac{1*11+2*7+3*2}{5*4} = \frac{31}{20} = 1.55$. Hence, routing does not always find the shortest path. This is only possible if every node knows the global structure of the network.

H#5.2 Routing Complexity

a) Routing Complexity (2 P.)

Explain the difference of average case and worst case complexity on the example of routing in Chord.

Solution:
The worst case complexity provides an upper bound on the cost of an algorithm for every possible case. Considering Chord with a b bit ID space, routing terminates in $O(b)$ steps, since the maximal distance is $2^b - 1$, and the distance is at least halved in every step.

The average case complexity provides an upper bound on the expected cost of an algorithm. In case of Chord routing with n nodes, the average case complexity is $O(\log n)$. After dividing 2^b $\log n$ times, there is maximally a distance of $2^b/n$ to the target. Since every node is in average responsible for $2^b/n$ identifier, the target is most likely stored on the successor of the current node. (see Stoica et al.: 'Chord: A scalable peer-to-peer lookup service for internet applications' for formal proof)

b) Unstructured Routing (4 P.)

Consider an unstructured P2P system in which a node queries its neighbors sequentially. Let the node u have k neighbors, of which one has the desired file. In case u queries all its neighbors without repetitions, what is the average number of queries needed, what is the worst-case?
Which complexity class is that?

Assume now due to memory constraints, the node does not keep a list of queried nodes, but chooses a random neighbor each time. What is the number of queries in the worst case and in the average case? complexity class?

Hint: You can use that the expected value of geometrically distributed random variable with parameter p is $\frac{1}{p}$.

Solution:

a. without repetition
   - worst-case: have to ask all k neighbors $O(k)$
   - average-case: with equal probability $1,2,\ldots,k$ neighbors need to be asked, so $k/2$ in average, complexity $O(k)$

b. with

- worst-case: always asked wrong neighbor, algorithm does not terminate
- average-case: independent trials with success probability $1/k$, the time of the first success is modelled as a geometrically distributed random variable with expectation and hence average number of queries k, so $O(k)$

H#5.3 Comparing Distance Metrics (5 P.)

Assume we have an n-bit identifier space.
Compare the following distance metrics for nodes $u, v \in V$ with identifiers $id(u)$ and $id(v)$:

i. $d_{\oplus}(u,v) := id(u) \oplus id(v)$[1]

ii. $d_{\mathscr{P}}(u,v) := min\{|id(v) - id(u)|, 2^n - |id(v) - id(u)|\}$

$d_{\oplus}(u,v)$ is for example used by Kademlia and $d_{\mathscr{P}}$ is used by Pastry.

a) What is the maximal distance in both metrics?

b) Assuming a fixed node u. How many neighbours with distance 1 can u have in the different metrics?

c) Assume $n = 8$. Calculate the distances $d_{\oplus}(u,v)$ and $d_{\mathscr{P}}(u,v)$ for the following pairs of identifiers:

   i. $u = 11\ 111\ 110_2, v = 00\ 000\ 000_2$

   ii. $u = 11\ 011\ 110_2, v = 01\ 011\ 010_2$

d) Can you find an example where $d_{\oplus}(u,v) < d_{\mathscr{P}}(u,v)$?

Solution:

a) XOR: 2^n for IDs differing in each bit; Pastry: 2^{n-1} for nodes on opposite sides of the ring

b) XOR: 1 (only last bit different), Pastry: 2 (each side of the node on the ring)

c) Assume $n = 8$. Calculate the distances $d_{\oplus}(u,v)$ and $d_{\mathscr{P}}(u,v)$ for the following pairs of identifiers:

   i. $d_{\oplus}(u,v) = 11\ 111\ 110_2$, $d_{\mathscr{P}}(u,v) = \min\{254 - 0, 256 - 254\} = 2$

   ii. $d_{\oplus}(u,v) = 10\ 000\ 100_2$, $d_{\mathscr{P}}(u,v) = \min\{222 - 90, 256 - 222 + 90\} = 124$

d) No. If the first k digits of the IDs of v and u agree, but the $k-1$-th digit does not, we have $d_{\oplus}(u,v) \geq 2^{n-k-1}$, but $d_{\mathscr{P}}(u,v) \leq 2^{n-k-1}$ (subtract smaller number from larger, first k digits are 0)

[1] $\oplus$ means bitwise XOR