Exercise 3

Exercises Peer-to-Peer-Systems and Security (SS2010)

Monday 31.5 2010
Hand-in: Monday 14.6. 2010 in lecture or per mail
Exercise: to be announced

Dipl.-Inform. Heiko Niedermayer
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München

Rules: There will be five exercise sheets with each 10 points. You have to achieve 50 % of the points and present a solution in the exercise course to get the 0.3 bonus.

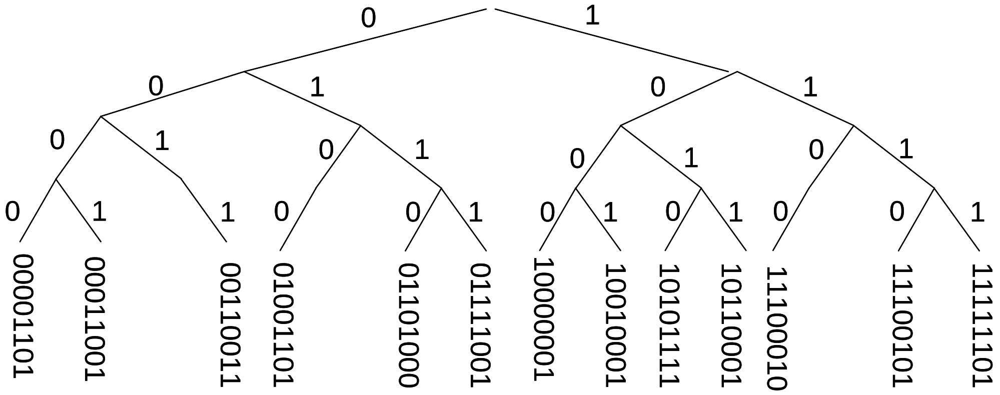
Task 1 (2 Points) A flexible Chord

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 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 2 (2 Points) Kademlia

Now, we simulate the operation of Kademlia. Bucket size is k=2. Alpha is 2. The IDs are 8 bit long.



a) You want to store item 01000001. Calculate the distance (in bits and decimal notation) according to the XOR metric to the nodes 00110011, 01001101 and 10101111. Which node is responsible for the item?

b) Node 11010101 joins the network. Node 00110011 is the rendez-vous peer that node 11010101 uses to join. Describe the operations of the join operations over time including the filling of the buckets.

Task 3 (4 Points) CoolSpots – specific structure

Propose a structured network that is not a DHT for the CoolSpots network. It should be possible to

- efficiently look-up / store a GPS coordinate
- efficiently support range queries like all spots close to the coordinates on the way from Garching Forschungszentrum to Munich Marienplatz. It should be efficient in the sense that the cost is dominated by either the look-up (O(logn)) or the number of responsible nodes for the area (all nodes holding relevant data for the range query).
- the state for the routing should be at most O(logn)
- the required state for replication of a spot or similar measures should be below a small constant k per spot (e.g. k = 5).

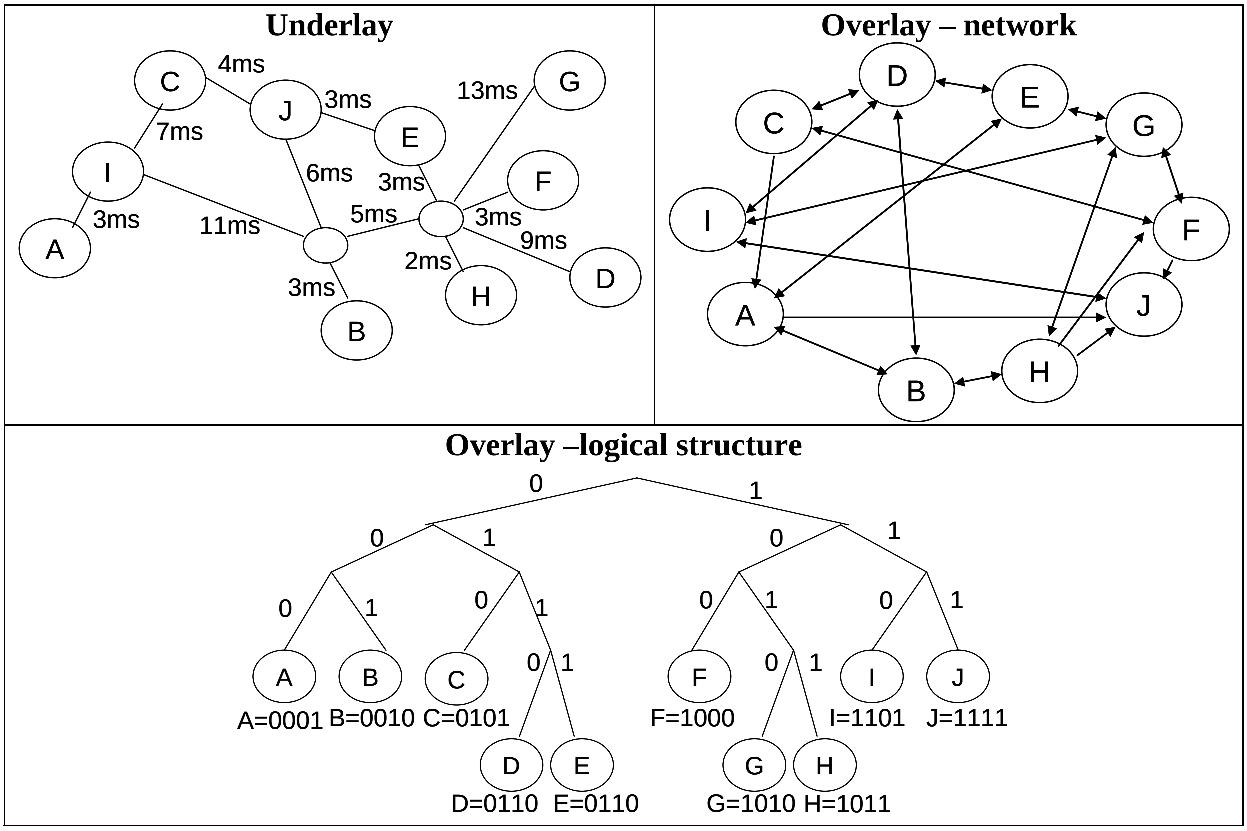
Describe

a) the structure and its maintenance
b) the look-up operation and the storage operation
c) this example range query for GPS coordinates from Garching to Munich Marienplatz

Task 4 (2 Points) Stretch and Proximity Neighbor Selection

The Figure below shows the underlay including the latencies along different paths. The second image is the structure of the overlay (nodes that link to each other) and the image on the bottom is the logical structure if we assume that our network uses a routing table like in Pastry, but without neighbor and leaf set.

In this task, stretch always refers to latency. Always state what path is used when you calculate the stretch.



a) What is the stretch (with respect to latency) for queries from F to A and from B to E?

b) Assume that each node on the way now applies PNS for its routing table. This means that it selected the latency-optimal node in each subtree. What is now the stretch for the queries from a)?