Exercise
for Lecture "P2P Systems"

Prof. Dr. David Hausheer
Dipl.-Wirtsch.-Inform. Matthias Wichtlhuber, Leonhard Nobach, M. Sc., Dipl.-Ing. Fabian Kaup, Christian Koch, M. Sc., Dipl.-Wirtsch.-Inform. Jeremias Blendin

TECHNISCHE UNIVERSITÄT DARMSTADT

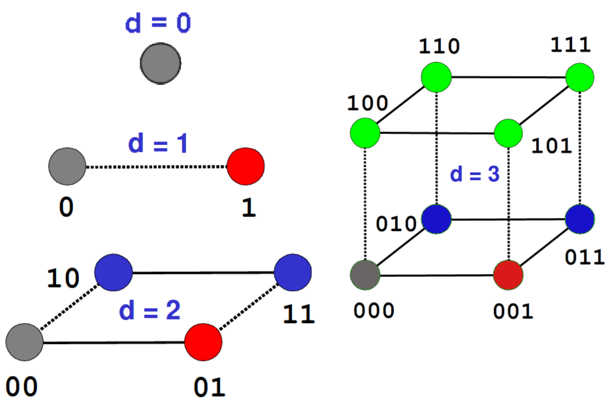
Summer Term 2015
Exercise No. 6
Published at: 26.05.2015, Submission date: 02.06.2015
Submission only via the Moodle platform in PDF, plain text, or JPG/PNG.
Contact: `[mwichtlh|lnobach|fkaup|ckoch|jblendin]@ps.tu-darmstadt.de`
Web: `http://www.ps.tu-darmstadt.de/teaching/p2p/`
– Example Solution –

Problem 6.1 - Hypercube Networks

A) Show graphically the recursive construction of a d-dimensional binary hypercube network, as d varies from 0 to 3. Label each node accordingly.

Solution:

See the following figure. The colors indicate the joining time of new nodes in the following order: grey, red, blue, green



B) Consider a 3-dimensional binary hypercube network. A message should be routed from node 001 to node 110. Show the routing process by evaluating the distance to the final destination, the available next hop options and the routing path. Whenever more than one option is available select any option randomly.

Solution:

Route from 001 to 110: 001 has 3 options (000, 101, 011). Thus, distance = 3.
Select 101 as the next hop (prefix-based routing).
101 has 2 options (100, 111). Thus, distance = 2
Select 111 as next hop.
110 is direct neighbor of 111. Thus, distance = 1

Forward to destination `110`.
The routing path is therefore `001`→`101`→`111`→`110`.

Problem 6.2 - Gnutella 0.6

A) Compare the join process of Gnutella 0.4 and Gnutella 0.6 respectively.

Solution: Connection setup: Gnutella 0.4 => Connect to a bootstrap node that has to be known initially, the bootstrap node forwards ping messages to supply the joining nodes with the addresses of other peers. Gnutella 0.6 => Connect as a leaf to a super peer and announce content, the super peer updates its routing table and is disseminating the information in the network. There is no need to maintain any other connections for the peer, unless it wants to download a file. An election mechanism will determine if a leaf node becomes a superpeer.

B) Compare the dissemination of search queries of Gnutella 0.4 and for *superpeers* of Gnutella 0.6 respectively.

Solution: The principle of query dissemination is the same for both protocols, they use flooding for the query distribution and backward routing to signal query hits. However, superpeers filter search queries for their leaf nodes and only forward relevant ones to them.

C) Given an origin node in a Gnutella system, the result of formula $f(n,t)$ is the maximum number of reachable users from this node, given n, the number of neighbors per node, and t, the used TTL counter.

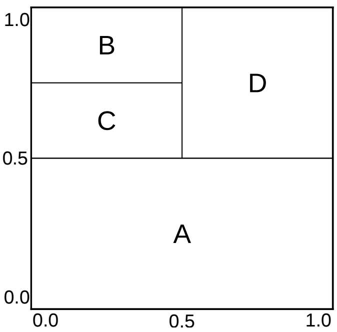
$$f(n,t) = n * \sum_{i=0}^{t-1} (n-1)^i \tag{1}$$

In a Gnutella 0.6 network only a fraction of all peers (super peers) forwards search queries, thus a smaller TTL can be configured. Derive a formula to calculate the percentage of saved routing decisions (not considering leaf nodes) in a Gnutella 0.6 network in comparison to a Gnutella 0.4 network. Use the formula to calculate the percentage of saved routing decisions for $t_{0.4} = 8$ and $t_{0.6} = 5$ and $n = 5$.

Solution: We can build the ratio like this: $f(n, t_{0.4}, t_{0.6}) = [1 - \frac{n*\sum_{i=0}^{t_{0.6}-1}(n-1)^i}{n*\sum_{i=0}^{t_{0.4}-1}(n-1)^i}] * 100$. The result is $[1 - \frac{1705}{109225}] * 100 = 98.43\%$. However, the search complexity is the same for both protocols, thus the better scalability of Gnutella 0.6 is caused by the hybrid P2P concept.

Problem 6.3 - CAN – A Scalable Content Addressable Network

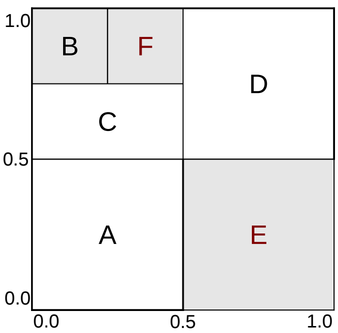
Consider the following topology of a CAN-based P2P system:



In this implementation a zone split is preferentially done horizontally.

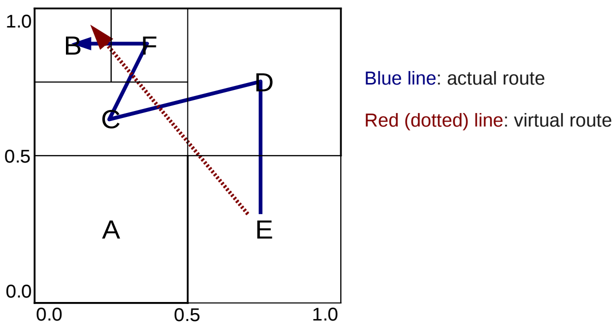
A) Now two nodes join the network: node **E** at (0.9, 0.1) and node **F** at (0.2, 0.8). What will be the resulting network topology after the integration of these new nodes?

Solution:



B) Now node **E** is sending a lookup for the key (0.2, 0.9). What would be the virtual path (also explain why this path will be chosen)? Is this the optimal path from the overlay's perspective?

Solution:

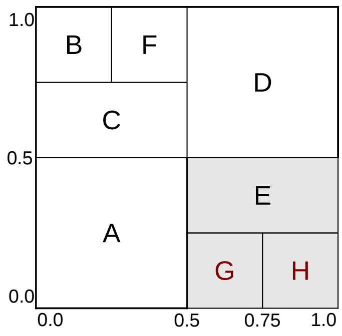


The path is chosen according to the straight line from the requesting node to the destination id ("virtual route"), as explained in the lecture. However, it would be shorter to route $E \rightarrow D \rightarrow F \rightarrow B$. The reason is that the virtual distance between nodes is typically not proportional to the actual transmission time. Therefore, it would be better to choose less overlay hops.

Additionally, if we consider the "wrap-around" property of CAN dimensions (remember that CAN is d-dimensional torus), than we can route $E \rightarrow D \rightarrow B$ or $E \rightarrow A \rightarrow B$ resulting in even less overlay hops.

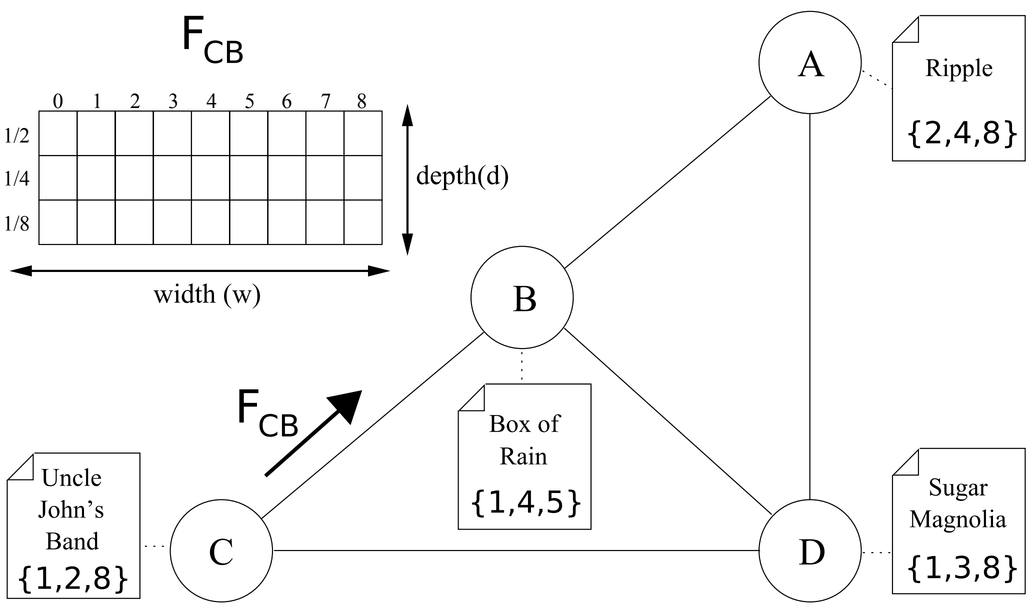
C) Now two more nodes join the network: at first node **G** at (0.6, 0.2) and then (after node **G** aquired its own zone), node **H** at (0.55, 0.1). What will be the resulting network topology after the integration of these new nodes?

Solution: The resulting topology is:



Note that after node **G** aquired its zone being (0.75–1.0, 0.0–0.25), **G**'s exact coordinate is not relevant for splitting. Thus, **G** and **H** may split this zone as shown in the figure.

Problem 6.4 - Bloom Filters



Solution:

A) Consider a Bloom Filter consisting of an array of size m and k hash functions. Initially all the bits of the array are set to zero. Assume that each position of the array can be selected by each hash function with equal probability. Calculate the probability that a certain bit will be set to 1 after inserting n elements.

F_{CB}

	0	1	2	3	4	5	6	7	8
1/2	0	1	0	0	1	1	0	0	0
1/4	0	1	1	1	1	0	0	0	1
1/8	0	1	1	1	1	0	0	0	1

Solution:
(1) The probability that a certain bit is not set to one by a certain hash function during the insertion of an element is $1-1/m$. (2) The probability that it is not set by any of the hash functions is: $(1-1/m)^k$. (3) If we have inserted n elements, the probability that a certain bit is still 0 is $(1-1/m)^{kn}$. (4) Thus, the probability that a certain bit is 1 is: $1-(1-1/m)^{kn}$.

B) Consider a Bloom Filter where $m = 5$ and $k = 2$. Furthermore, $H_1(x) = x \mod 5$ and $H_2(x) = (2 * x + 3) \mod 5$. Initially the bit vector is unset (each element is equal to 0). Insert 9 and 11 in the Bloom Filter. Show the state of each bit before and after each insertion.

Solution:
At first, we calculate the outcome of each hash function for 9 and 11: $H_1(9) = 4$, $H_2(9) = 1$, $H_1(11) = 1$, $H_2(11) = 0$.
Then we set the bits at calculated positions to 1. The following table shows the state of the vector:

	bit 0	bit 1	bit 2	bit 3	bit 4
initially	0	0	0	0	0
after instertion of 9	0	1	0	0	1
after instertion of 11	1	1	0	0	1

C) Assuming the populated Bloom Filter of the previous part, show how the membership operation for the values 15 and 16 is handled by the aforementioned Bloom Filter. Verify the outcome of each operation with respect to false positives and false negatives.

Solution:
(1) Calculate the outcome of each hash function for 15: $H_1(15) = 0$, $H_2(15) = 3$.
Inspect the bits at positions 0 and 3: Since the bit at position 3 is set to 0, 15 is not a member. In this case, the Bloom Filter provides the correct answer.
(2) Calculate the outcome of each hash function for 16: $H_1(16) = 1$, $H_2(16) = 0$.
Both bits are set to 1. In this case, the Bloom Filter provides a false positive since 16 is not an inserted member.
Furthermore, there are no false negatives when using Bloom Filters.

D) Consider the picture of a P2P network above. Denoted are 4 peers, each of them holding a content item with a respective hash. Fill in the attenuated Bloom Filter for node C considering the link F_{CB}.