Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Systems@ETH Zürich

Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems

— Solution to Assignment 11 —

1 Clock Synchronization

Quiz

1.1 Clock Synchronization & Topology

a) We can observe, that in a balanced binary tree the furthest distance nodes have from the root is $\log_2(n)$. The network diameter is limited by $D \leq 2 \cdot \log_2(n)$, and therefore global skew is at most $2(1+\varepsilon)\log_2(n)T + \frac{2\varepsilon}{1-\varepsilon}$.

b) In a system where the **linear real-time envelope** condition holds, each clocks time is in the interval $[(1-\varepsilon)t, (1+\varepsilon)t]$ If we now consider the case where in one partition all clocks run with $(1-\varepsilon)t$ and in the other all clocks run with $(1+\varepsilon)t$ the maximal skew they achieve is $2\varepsilon \cdot t$.

Advanced

1.2 Clock Synchronization: Spanning Tree

The grid is composed of cells and nodes. The nodes are shown as black dots in Figure 1 and the cells are the areas between four (neighboring) nodes. Now we look at an arbitrary cell in the grid. For any tree we can draw on this grid, there has to be a way to walk out of the grid without crossing the edges of the tree as shown in Figure 1 because there are no loops in a tree. This holds for every cell in the grid, especially for the cell in the middle (or adjacent to the center node) of the grid. Let us assume we leave the grid between two nodes B and C. These two nodes are neighbors on the grid, since otherwise the nodes in between would not be connected to the spanning tree. If m is even, then the distance between Node A (adjacent to the middle cell, opposite the exit side) and Node B or C, respectively, is at least $\frac{m}{2}$. If m is odd, either of the two routes can be $\frac{m-1}{2}$ but the other is at least $\frac{m+1}{2}$, since the distance from A to the edge of the grid is $\frac{m-1}{2}$, but for one of the nodes one needs to make at least one step in the perpendicular direction.

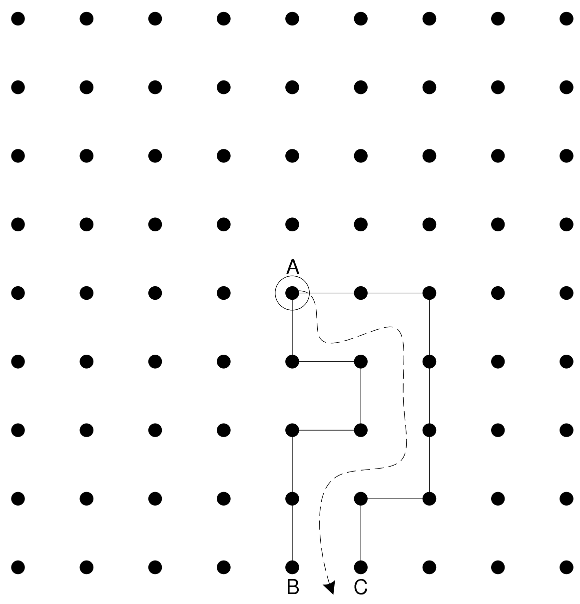

Figure 1: Node A is the center node of the spanning tree (only partially shown). The dashed line is a path through grid cells from the center Node A to the outside of the grid.

2 Distributed Storage

Quiz

2.1 Hypercubic Networks

The hypergraphs are drawn as shown in Figure 2. The 4-dimensional hypercube $M(2,4)$ can also be drawn as two cubes side-by-side with connected corners.

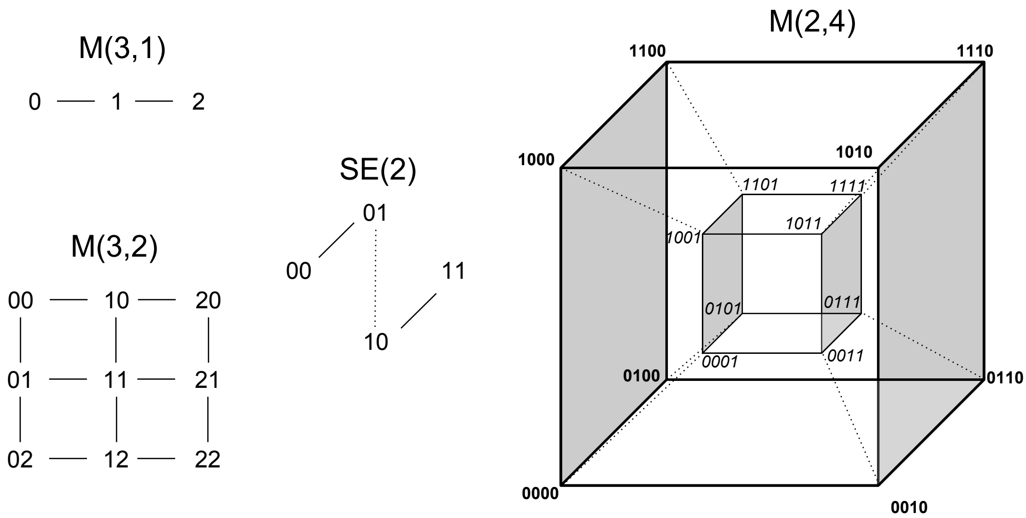

Figure 2: Drawings of $M(3,1)$, $M(3,2)$, $SE(2)$ and $M(2,4)$. Note that we have used dashed lines and solid lines for $SE(2)$ the other way around compared to the lecture.

Basic

2.2 Iterative vs. Recursive Lookup

a) In the recursive lookup there is no difference between a request originating at the node and a request being forwarded. Only once the lookup is finished do we need to care about forwarding the result to the previous node or returning it to the caller. This allows the same lookup logic to be reused for both. Furthermore, if the response is returned through the same path as the request was sent through, then the intermediate nodes can cache the result, potentially speeding up future lookups and distributing the load of a popular item.

b) Recursive lookups can easily be misused to mount Denial-of-Service attacks, since a single request message from an originating node is forwarded over multiple hops. Each hop multiplies the impact this message has on the network. Thus the attacker's bandwidth is potentially multiplied by the number of hops the request is routed through. Furthermore, if the result is returned over the same path as the request, then the attacker is hidden behind a number of hops and the victim only sees traffic originating from the last hop.

2.3 Building a set of Hash functions

The salted hashing function derivation allows random access to any of the derived hashing functions without having to recompute the intermediate functions, as is the case in the iterative hashing function derivation scheme. Furthermore, iterative hashing allows a user which knows the first hash value to deduce all the others by repeatedly applying h. This information can be used by an attacker to target specific nodes storing a movie.

Advanced

2.4 Multiple Skiplists

a) Figure 3 shows the structure of a multi-skiplist with 8 nodes and 3 levels. Notice that each of the lists would wrap around at the ends.

b) Unlike in the single skiplist each node now has a constant degree of $2 \cdot d$, i.e., on each level it has a right and a left neighbor, including the full list at $l = 0$.

c) The number of hops is still $O(\log n)$, just like for the simple skiplist.

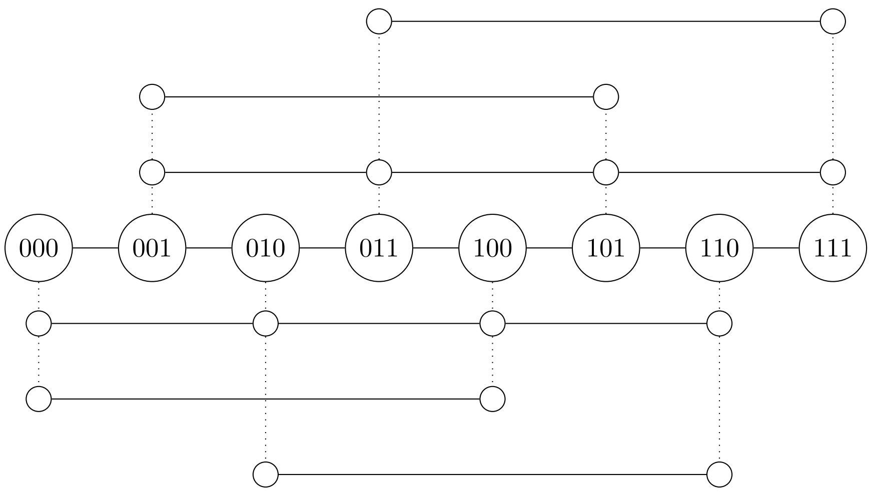

Figure 3: A multi-skiplist with 3 levels and 8 nodes.