Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2013 Prof. R. Wattenhofer

Distributed Systems Part II

Solution to Exercise Sheet 9

1 Clock Synchronization: Spanning Tree

We want to show that the stretch of the spanning tree is at least m in every grid of $n = m \times m$ nodes. In other words, there exist two neighboring nodes that are distance m apart in the spanning tree of the grid. We choose an arbitrary Node A as the root node of the spanning tree, see Figure 1. Starting from the root node, we walk between the cells of the grid until we get out of the grid. Since there exists no cycles in a tree, every cell is bounded by at most 3 edges that are part of the spanning tree. Thus, starting from the root node, we can always find a sequence of cells we have to visit to find a way out of the grid. We denote the nodes adjacent to the cell where we leave the grid as Node B and Node C. Consequently, both Node B and Node C have to be neighbors in the grid and both nodes are part of the spanning tree. When travelling from Node B to C (and vice versa) it is necessary to take a detour over the root node since the path between the cells from the root to the outside of the grid never crossed an edge of the spanning tree. Thus, even in the best case where the network diameter is $\frac{m}{2}$, the neighboring nodes B and C have at least a distance of m in the spanning tree. This leads to a stretch of the spanning tree of at least m.

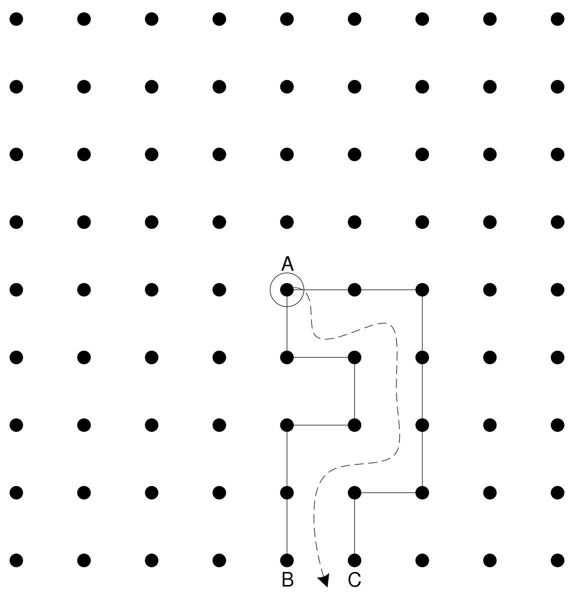

Figure 1: Node A is the root node of the spanning tree (only partially shown). The dashed line is a path between the grid cells from the root node to the outside of the grid.

2 Network Updates

a) v_3 can not change before v_2, but v_2 needs to wait for v_1, requiring three steps in total.

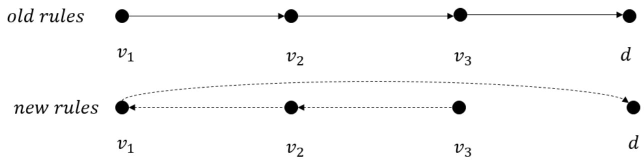

Figure 2: Graph with three rules.

b) Let E' be the set of rules that no longer need to be updated and V' the set of nodes with the property that there exists a path to the destination using only rules from E', with $d \in V'$. Any new rule with the property that it points to a node from V' can not induce a cycle, since all paths from all nodes from V' end at d and d has no outgoing edge. If there are still rules to be updated, then such a rule will always exist, since the set of new rules induces a directed tree with d as its root and all edges in this tree are oriented towards d, meaning at least one new rule will point to a node from V'.

Note: As seen in **c)**, all rules that can be updated might have this property.

c) For a graph with n nodes we use the same concept as in the first item, but with n instead of three vertices v_i. Again, v_i can not change before v_{i-1} for $2 \leq i \leq n$, requiring n steps in total.

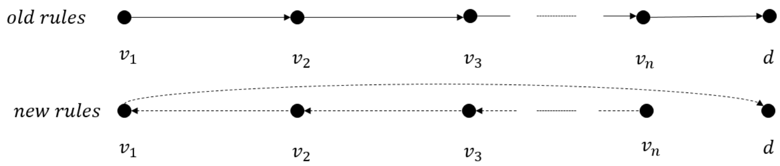

Figure 3: Graph with n rules.

d) Obviously, v_3 can always change in the first step, without any consequences for the other nodes – see **b)**. But what about v_2 and v_1? If v_1 changes in the first step, then updating v_2 would induce a cycle, and vice versa. Therefore two possible ways to migrate the network would be:

- Migrate v_3 and v_1 in the first step. Migrate v_2 in the second step.
- Migrate v_3 and v_2 in the first step. Migrate v_1 in the second step.