Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

HS 2009 Prof. Dr. Roger Wattenhofer, Philipp Sommer, Johannes Schneider

Ad Hoc And Sensor Networks
Sample Solution to Exercise 9

Assigned: November 16, 2009
Due: November 23, 2009

1 Tree-based Clock Synchronization

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.

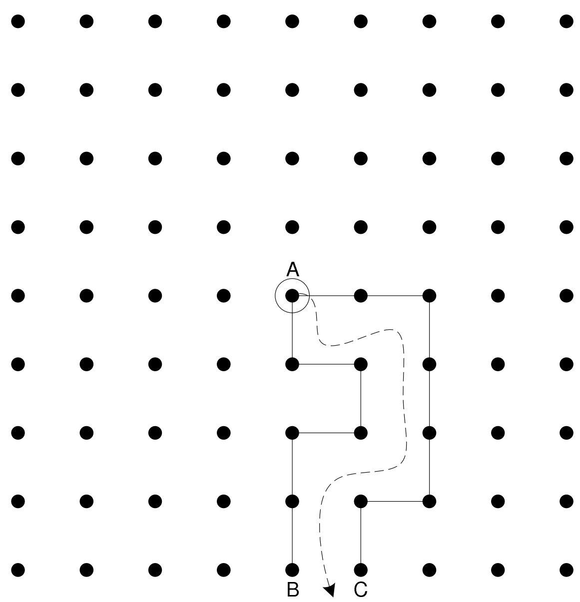

Figure 1: Node A is the root node of the spanning tree (only partially shown). The dashed line is a path between the grid cells from the root node to the outside of the grid.

2 Gradient Clock Synchronization

Blindly adjusting the logical clock rate to the rate of the fastest neighbor is a bad idea. Depending on the ambient temperature the hardware clock rate of a node varies slightly. A small increase in the ambient temperature will cause the hardware clock to run at a slightly higher rate. Thus, also the logical clock of a node runs slightly faster since its rate directly depends on the hardware clock rate. Consequently, other nodes will increase their logical clock rates too when employing the *maximum* algorithm. Since nodes always try to catch up with their fastest neighbors, a node will never decrease its logical clock rate. Due to the small variations in the ambient temperature the logical clock rate of the system will permanently increase and the logical clock runs faster and faster. This can be a severe problem for systems where it is important that the duration of a clock tick corresponds to a fixed time interval, e.g., 1 millisecond.