Exercise 2: Gradients are Hard

Task 1: No Uncertainty, still no Knowing the Time

In some networks, frequent communication is too expensive, for instance in terms of energy. This is especially true in wireless sensor networks, where successfully sending a message is a challenge in itself. In such networks, a fairly complicated scheduler may be in place. We model this by taking away the control over the message schedule from our algorithm designer: An adversary decides when messages are sent, with the only condition that a node broadcasts its current logical clock value at least every T time. The uncertainty and even message transmission times may be small in comparison to the influence of this issue; for simplicity, assume that messages arrive instantaneously, i.e., $d = u = 0$.

a) Show that you can choose message sending times and hardware clock rates such that you can introduce a skew of $\Omega((\vartheta - 1)Td(v, w))$ between the hardware clocks of nodes v and w before they notice (assuming that $\vartheta \in \mathcal{O}(1)$). Aim for a statement that can replace Lemma 1.5! You are free to show the claim for bipartite graphs only.

b) Use this to prove a theorem analogous to Theorem 2.5 in this setting! (Discussing the changes that have to be made to the proof of Theorem 2.5 suffices.)

c) Show that one can do *much* better if the message schedule is under one's control! (Nodes can send only $O(1)$ messages per T time unit and you still need to satisfy that $\forall v \in V(G), t \in \mathbb{R}^+ : l_v(t) \leq \vartheta(1 + \mu)$. Aim for a skew of $O((\vartheta - 1)T)$.)

Solution

a) Just as in Lemma 1.5, we set $H_x(0) := 0$ for all $x \in V$ and both executions, and in $\mathcal{E}_1$ all hardware clock rates are 1. As the graph is bipartite, it can be 2-colored. All nodes of one color transmit messages at times kT, $k \in \mathbb{N}_0$, while all nodes of the other color transmit messages at times $(k + 1/2)T$, $k \in \mathbb{N}_0$.

   Now choose $v, w \in V$. We modify $\mathcal{E}_1$ into $\mathcal{E}_v$ by changing the hardware clock rates of all nodes $x \in V$ with $d(x, v) < d(x, w)$ to ϑ. Messages are transmitted at the same local times. Trivially, the executions are indistinguishable at all nodes at time 0. We keep track of the nodes that can distinguish the two executions. We claim that node $x \in V$ cannot distinguish the two executions prior to local time $T/2 \cdot (d(v, w)/2 - \min\{d(x, v), d(x, w)\})$. Note that, as all messages are sent at the same local times in both executions, the reception times are identical everywhere but for nodes at the "fault line," i.e., for messages between nodes with different clock rates. Such nodes will be able to distinguish the executions no earlier than local time $T/2$, meaning that all other nodes are oblivious of the difference between the executions until this local time. This anchors the induction. For the step, note that the knowledge cannot be passed on until the next message is sent, which always takes $T/2$ local time.

   We conclude that v and w cannot distinguish the two executions prior to local time $t_0 := Td(v, w)/(4\vartheta) \in \Omega(Td(v, w))$. As skew is built up at rate $\vartheta - 1$, the claim follows.

b) Instead of using Lemma 1.5 in the construction, we now use a). This means that, in each iteration, we have that $t_{i+1} = t_i + Td(v, w)/(4\vartheta)$ and the skew added is $\Omega((\vartheta - 1)Td(v, w))$. We choose b such that only half of the average skew that is added is lost due to w catching up, i.e., $b \in \Theta(\sigma)$, where $\sigma := \mu/(\vartheta - 1)$ as usual. We get that $\mathcal{L} \in \Omega((\vartheta - 1)T \log_\sigma D)$.

c) TODO

Task 2: No Fast Edge Insertion

Suppose that new edges may be added to the graph at arbitary times. The endpoints of the edges will know that the edge is new, and seek to ensure a small local skew also on such edges.

a) Show a worst-case lower bound of $(\mathcal{G} - \mathcal{L})/\mu$ on the time required to achieve a skew of at most $\mathcal{L}$ on a new edge. (Assume that there was no prior edge insertion.)

b) Show that, in the worst case and even with unbouded logical clock rates, $\Omega((d-u)uD/\mathcal{L})$ time is required to achieve a skew of at most $\mathcal{L}$ on a new edge, if the skew on the edges that were fully integrated must not exceed $\mathcal{L} \ll uD$. Here, D denotes the diameter of the graph before the (single) additional edge is inserted. (Hint: Consider an execution with skew $\Omega(uD)$ between $v, w \in V$ at some time $t \geq (d-u)uD/\mathcal{L}$ and insert $\{v, w\}$ at time $t - \Theta((d-u)uD/\mathcal{L})$.)

Solution

a) For any $\varepsilon > 0$, there is an execution in which a skew of $\mathcal{G} - \varepsilon$ is reached between two nodes $v, w \in V$ at some time t. Pick such an execution and set all hardware clock rates to 1 at time t. If $\{v, w\} \in E$, then $\mathcal{L} \geq \mathcal{G} - \varepsilon$. If this happens for arbitrarily small ε, then $\mathcal{L} = \mathcal{G}$ and the statement is trivial. So, in the following, assume w.l.o.g. that $\{v, w\} \notin E$. Hence, we can insert the edge $\{v, w\}$ at time t. As all hardware clock rates are 1, the skew on the edge cannot be reduced faster than rate μ. Thus, it takes at least $(\mathcal{G} - \varepsilon - \mathcal{L})/\mu$ time to achieve a skew of at most $\mathcal{L}$ on the edge. As this is true for arbitrarily small $\varepsilon > 0$, the claim follows.

b) We follow the hint; the existence of such executions was shown in the first lecture. Abbreviate $g := |L_v(t) - L_w(t)| \in \Omega(uD)$. We cannot have that $\{v, w\} \in E$, as otherwise $\mathcal{L} \geq g$. We insert $\{v, w\}$ at time $t' := t - (g - 2\mathcal{L})/(4\mathcal{L})$ for some sufficiently small constant $c > 0$. As messages travel for at least $d - u$ time, nodes in distance $(g - 2\mathcal{L})/(4\mathcal{L})$ or more from both v and w cannot distinguish the new execution (with the additional edge) from the previous one prior to time t. As hardware clocks are the same in both executions, their logical clock values remain the same up to this time. In order to achieve $L_x(t'') - L_y(t'') \leq \mathcal{L}$ for all $\{x, y\} \in E \cup \{v, w\}$ at a time $t'' \in [t', t)$, the algorithm must thus change the logical clock values of nodes in distance smaller than $(g - 2\mathcal{L})/(4\mathcal{L})$ from v or w so that these constraints are satisfied. However, if the logical clock value of x does not change, the value of its neigbhor cannot differ by more than $2\mathcal{L}$ without violating the skew constraint in one of the two executions. Its neighbor thus cannot adapt its clock by more than $4\mathcal{L}$, and so on. By induction over decreasing distance to v and w, it follows that v and w each cannot change their clock values by more than $(g - 2\mathcal{L})/2$ between the two executions. Thus, at least $g - (g - 2\mathcal{L}) = 2\mathcal{L}$ skew must remain on the edge prior to time t.