ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2015

Prof. R. Wattenhofer
Laura Peer

Principles of Distributed Computing
Exercise 10: Sample Solution

1 Distributed Network Partitioning

For this exercise, we define the neighborhood of a node v as $\mathcal{N}_i(v) := \{w \in V \mid d(v,w) \leq i\}$. Informally, $\mathcal{N}_i(v)$ denotes the set of nodes within distance i to v.

a) The leader starts modified flooding/echo rounds with increasing diameter until the cluster criterion is met. Simultaneously, a BFS tree around the leader is constructed that will be needed to reduce the number of messages sent. Initially, the tree consists of the cluster leader l only, which is considered the root.

In round i the cluster leader initiates a flooding/echo on the BFS tree. A node that joined the BFS tree in round $i-1$ (an $(i-1)$-hop neighbor of l, or l itself if $i = 1$) will notify all its direct neighbors within the whole graph. Once a node v not already in the BFS tree receives such a notification from a node p, it joins the BFS tree with p as parent and informs p about this. If a node v already in the BFS tree receives such a notification from p, it informs p that it will not join the BFS tree. All nodes in the tree accumulate the number of newly joined nodes in their subtree of the BFS tree and return them to their parent. Thus l will be informed about the number $|\mathcal{N}_i(l)|$ of nodes within range i when round i finishes after at most $2i$ time units.

This procedure is started with $i = 1$ and stops when $2|\mathcal{N}_{i-1}(l)| \geq |\mathcal{N}_i(l)|$. Then l informs the nodes in $\mathcal{N}_r(l)$ that they participate in its cluster by a broadcast in the constructed BFS tree. The correctness of this procedure is obvious from the construction. The time complexity is $\sum_{i=1}^{r+1} 2i = (r+1)(r+2)$ for the flooding/echo steps, and additional $r+1$ time units for the final broadcast. In total we get a time complexity of $O(r^2)$. Due to the stop criterion we have $|C| \geq 2^r$. Thus we have $r \leq \log |C|$, implying $O(r^2) \subseteq O(\log^2 |C|) \subset O(|C|)$.

b) An intuitive argument why this property holds goes as follows: A tree has about as many edges as nodes. In each round we sent two messages along the edges of the current tree in order to double the number of nodes in the tree. Hence, the number of messages sent within the tree is in the order of the number of nodes in the final tree, i.e. $|N_r(l)|$, which can be at most $|E'|$ since each node in the tree has at least one edge. Furthermore, we sent two messages over the edges with only one endpoint in the cluster in the last step of the cluster construction and there can be at most $|E'|$ such edges.

Let us now show the claim formally: The i^{th} step of the construction from **a)** will require two messages to be sent over all edges from nodes that joined the BFS tree in the $(i-1)^{\text{th}}$ step, plus two messages per edge in the already constructed BFS tree. For the former, observe that during the (stepwise) construction of the BFS tree we send messages over each edge in E' at most twice, as E' is the set of edges with at least one endpoint in the final cluster. For the latter, note that the number of edges in the tree after the i^{th} step is $|\mathcal{N}_i(l)|-1$. We have the condition $|\mathcal{N}_i(l)| > 2|\mathcal{N}_{i-1}(l)|$ for $i \in \{2,\ldots,r\}$, hence we can estimate $|\mathcal{N}_i(l)| \le 2^{-r+i}|\mathcal{N}_r(l)|$ for $i \in \{1,\ldots,r\}$. Summing over i, we may estimate the total number of messages on the edges of the BFS tree by

$$\sum_{i=1}^{r} 2(|\mathcal{N}_i(l)|-1) < 2|\mathcal{N}_r(l)| \sum_{i=1}^{r} 2^{-r+i} = 2|\mathcal{N}_r(l)| \sum_{i=0}^{r-1} 2^{-i} < 4|\mathcal{N}_r(l)| \le 4|E'|.$$

Thus, in total at most $(4+2)|E'| \in O(|E'|)$ messages are sent during the construction phase of the cluster in the worst case.

c) Consider the tree to be rooted at the first leader. We apply a distributed depth-first search strategy on the tree. After each cluster construction step, the current leader continues the depth-first search for the next leader, which is the next node on the search path that is not assigned to a cluster yet. In the course of the algorithm execution, each edge of the tree is hence traversed twice and thus the time and message complexity is $O(n)$.

d) Let C_i for $i \in I$, denote the vertex sets of the constructed clusters and E'_i the edges "removed" after the cluster construction of the i^{th} cluster. By **b)** the total number of messages due to the cluster construction is bounded by $\sum_i O(|E'_i|) = O(m)$, since the sets E'_i are disjoint. By part **a)** of the exercise we have a worst-case estimate for the (total) time complexity of the cluster construction of $\sum_i O(|C_i|) = O(n)$, using that the vertex sets of the clusters are disjoint. Finally we have to add $O(n)$ messages and $O(n)$ time due to leader election, which was shown in **c)**. Putting everything together, we conclude that the time complexity is $O(n)$ and the message complexity is $O(n+m) = O(m)$.