MST Construction in O(log log n) Communication Rounds

Thomas Locher <lochert@student.ethz.ch>
Seminar of Distributed Computing WS 04/95

February 3, 2005

1 Introduction

A minimum-weight spanning tree (MST) is a useful distributed construct, as it minimizes the cost associated with global operations such as broadcast. In the paper [1], a synchronous algorithm for the construction of the MST in a complete weighted undirected graph in which all nodes have IDs of $O(log\, n)$ bits is presented. In each communication round, only $O(log\, n)$ bits can be sent over each edge. This restriction implies that only a constant number of node IDs and edge weights can be packed into a single message. Prior to this work, the best known algorithm in this model had a time complexity of O($log\, n$) rounds. The algorithm presented improves this bound, as it requires $O(log\, log\, n)$ rounds in the worst case. What is more, at most $O(n^2\, log\, n)$ bits have to be sent in total, which is optimal. The authors believe that the algorithm may be useful in some overlay networks where reliable point-to-point communication between nodes is provided.

2 Evaluation

The algorithm consists of a few well-defined steps, which makes it simple both to understand how the algorithm works in detail and to verify its correctness. By cleverly sharing the workload among nodes that already belong to the same cluster[1], more information can be spread in a single phase of the algorithm, thereby achieving a time complexity of $O(log\, log\, n)$ rounds in the worst case. Another strength of the algorithm is that it can be easily extended to larger messages. They show that the MST can be constructed in $O(log\, \frac{1}{\epsilon})$ rounds, thus in a constant number of rounds, when messages may contain $O(n^{\epsilon}\, log\, n)$ bits for $\epsilon > 0$. However, for any polylogarithmic message size, $\Theta(log\, log\, n)$ rounds are required.

[1] In each phase of the algorithm, some edges are chosen as a part of the MST. Nodes that are directly or indirectly connected using chosen edges only belong to the same cluster.

A very interesting open problem is whether the time complexity of this algorithm is optimal or whether there is a faster way to construct the MST. In fact, it is not even known if the MST can be constructed in a constant number of rounds in this model. The authors insinuate that such an algorithm does not exist by stating that nodes probably cannot learn the entire topology of the constructed logical graph in a constant number of steps, since such information exchange seems to require too much time. However, they do not elaborate on it. What is more, the authors neither provide any ideas on how to further improve the time complexity, nor reason about the unknown lower bound of the time complexity in this model.

The model itself is highly interesting. As opposed to graphs with a potentially high diameter where routing issues may be the major concern, the complete graph allows for the direct communication between any two nodes in the system. It would therefore be interesting to analyze how hard it is to solve various other problems in this model and possibly derive some lower bounds. More generally, since the diameter of the graph is one, it would be nice to find out what exactly constitutes the problem of solving problems in this model, e.g. if or when symmetry breaking has something to do with it.

3 Conclusions

The paper presents the first algorithm that manages to construct the MST in the given model in sublogarithmic time, breaking the $O(log\, n)$ barrier. Its clear-cut steps enable a straightforward analysis and verification. While the paper offers some new results, many questions are left unanswered, the most obvious being whether the algorithm can be improved or whether there is an inherent lower bound of $\Omega(log\, log\, n)$ rounds required for the construction of the MST in this model.

References

[1] Z. Lotker, E. Pavlov, B. Patt-Shamir, D. Peleg. MST Construction in O(log log n) Communication Rounds. *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, 2003.*