Parallelization of a Centralized Spanning Tree Discovery Algorithm

Contemporary routing protocols predominantly seek routes minimizing hop count. Such a strategy appears misleading, however, under intermittent decline in link quality caused by traffic bursts or other transient factors. The shortest path, in this case, may oblige to transmit over the links with the highest congestion. The issue is crucial, e.g., in multicasting where erroneous route discovery results in sub-optimal spanning trees.

Contemporary approaches optimize multicasting either for energy consumption, improving packet loss ratio or spurring maintenance in ad-hoc networks. In your work you will address an exigent and yet abandoned problem of multicasting optimized for throughput. We have suggested a centralized greedy spanning tree discovery algorithm by relaxation that circumvents links with the lowest bandwidth on demand. This can be compared to the state of the art protocols based on Prim’s and Dijkstra’s methods.

In your work you will:

1. Develop a simulation of the local (IDA) department network (+random networks) in the NS-II simulator based on the suggested algorithm. You will then discuss possible gain/loss in throughput as a function of links’ capacity variations by running the simulation and comparing the results to the ones given by the contemporary techniques.

2. Adapt an existing parallel implementation of the algorithm and test it on a supercomputer located at the National Supercomputer Center. In these experiments you will measure gain/loss in speed compared to the sequential version of the algorithm.

Further information: Mikhail Chalabine, mikch@ida.liu.se; Dr. Christoph Kessler, chrke@ida.liu.se

Prerequisites: Background in Parallel programming, e.g., TDDC78, Concurrent programming, Data structures and algorithms so as good knowledge of C / C++.