Call Scheduling in Trees, Rings and Meshes*

T. Erlebach
TU München, Germany
erlebach@informatik.tu-muenchen.de

K. Jansen
Universität Trier, Germany
jansen@dm3.uni-trier.de

Abstract

The problem of establishing and completing a given set of calls as early as possible is studied for bidirectional and directed calls in various classes of networks. Even under the assumption of unit bandwidth requirements and unit call durations, call scheduling is $\mathcal{NP}$–hard for trees with unbounded degree, for rings, and for meshes. Whereas bidirectional calls can be scheduled optimally in polynomial time for trees of constant degree, the problem for directed calls is already $\mathcal{NP}$–hard for binary trees. Approximation algorithms with constant performance ratio are known for many $\mathcal{NP}$–hard variants of call scheduling.

1 Introduction

It has become apparent that high-speed networks of the future will use Asynchronous Transfer Mode (ATM). When a connection is established in an ATM network, the network guarantees that the requested resources will be available to the connection as long as it remains active. Therefore, simultaneously active calls can only use the same link if the sum of their bandwidth requirements does not exceed the link capacity. As a consequence, call admission control (deciding whether to accept or reject a call) and call scheduling are fields of ongoing research, both in the on-line and off-line scenario.

We investigate the problem of off-line call scheduling for different classes of networks, thereby revealing which cases are solvable optimally in polynomial time and which cases are $\mathcal{NP}$–hard and require approximation algorithms. The off-line call scheduling problem may be of interest in various scenarios arising in practice, e.g. in parallel computer systems interconnected by ATM networks where the communication requirements of an application program are known in advance. Other applications include file transfers, movie transmissions or video conferences. In all these applications the throughput of the network may be improved considerably if communication requests are made in advance so that schedules can be calculated off-line.

*This work was partly funded by the German Science Foundation (DFG), Contract: SFB 342, TP A7.

1.1 The Call Scheduling Model

The communication network is represented by a symmetric directed graph $G = (V, E)$, i.e. each bidirectional link of the physical network is represented by two directed edges. A call is specified by giving its endpoints. For the sake of simplicity, we assume that all edges of the network have the same capacity, that every call requires the full capacity of the edges it uses, and that all calls have unit duration. Furthermore, we consider only the cases where all calls are directed or all calls are bidirectional.

An input to the call scheduling algorithm then consists of a graph G and a set R of calls. If the calls in R are directed, the algorithm must assign each call $(u \to v)$ a directed path in G from u to v and a positive integer time step, such that calls using the same edge are assigned different time steps. If the calls are bidirectional, the algorithm must assign each call $(u \leftrightarrow v)$ two directed paths, one from u to v and one from v to u. For tree networks, the paths are uniquely determined by the endpoints of the calls. The length of a schedule S is the maximum of all time steps assigned to a call in R. Our goal is to minimize the schedule length.

1.2 Related Work

An interesting problem that has been studied recently is the wavelength routing problem in optical networks where virtual connections share physical links by wavelength-division multiplexing, see e.g. [4]. It is easy to see that the problem of assigning wavelengths to connections in optical networks with the goal of minimizing the number of different wavelengths is equivalent to the problem of assigning time steps to calls in our call scheduling model. Hence, $\mathcal{NP}$–hardness results and approximation algorithms can easily be transferred between the two models.

On-line versions of call admission control problems have recently been studied very intensively. Whereas in call admission control models every call must either be established immediately or rejected, the call scheduling model does not reject any call but may defer calls when the network is overloaded. On-line call scheduling was studied in [2]. Preliminary results for off-line call scheduling in trees and rings have appeared in [1].

Table 1: Summary of results

network	bidirectional calls	directed calls
linear array	polynomial	polynomial
bounded degree tree	polynomial	$\mathcal{NP}$–hard 7/4–approx.
arbitrary tree	$\mathcal{NP}$–hard 1.1–approx.	$\mathcal{NP}$–hard 7/4–approx.
ring	$\mathcal{NP}$–hard 2–approx.	$\mathcal{NP}$–hard 2–approx.
mesh	$\mathcal{NP}$–hard	$\mathcal{NP}$–hard

2 Results

Due to space limitations, we will only give an overview of our results and sketch some of the algorithms and ideas for the respective proofs. Table 1 summarizes most of the results in compact form.

Trees Scheduling bidirectional calls in tree networks with unbounded degree is $\mathcal{NP}$–hard, because finding an optimal schedule for the calls touching a single node v is equivalent to the $\mathcal{NP}$–hard problem of edge-coloring a multigraph whose number of nodes is equal to the degree of v. On the other hand, individual schedules for the calls touching every single node v can be combined into a schedule for all calls in polynomial time, such that the resulting schedule is no longer than the longest of the individual schedules.

Because of the close relationship to edge-coloring, the 1.1–approximation algorithm for edge-coloring due to [5] yields a corresponding approximation algorithm for call scheduling in arbitrary trees. Besides, we can prove that edge-coloring in multigraphs can be solved optimally in polynomial time, either by integer linear programming with a fixed number of variables or by a dynamic programming approach, if the number of nodes of the multigraph is bounded by a constant. This gives rise to a polynomial-time call scheduling algorithm for trees of bounded degree.

For directed calls, we have proved that optimal call scheduling is $\mathcal{NP}$–hard even for binary trees by a reduction from arc-coloring. (Arc-coloring is the problem of finding an optimum vertex coloring of a circular-arc graph.) Interestingly, directed calls touching a single node can be scheduled optimally in polynomial time even if the degree of the node is not bounded, but combining individual schedules into an optimal schedule for all calls is $\mathcal{NP}$–hard. Our $\mathcal{NP}$–hardness result resolves an open question in [4]. Kaklamanis and Persiano presented a 7/4–approximation algorithm for scheduling directed calls in trees of arbitrary degree [3].

In addition, we have investigated the case that the number of calls touching a single node of the tree network is bounded by a constant. Our remarks above immediately imply that there is a polynomial-time algorithm for optimal scheduling of bidirectional calls under this assumption, and we have been able to show that this holds also for directed calls. The algorithm for the latter case works by trying out all possible schedules for calls touching a single node and combining for each node the information obtained from its children.

Rings Call scheduling in ring networks is closely related to the arc-coloring problem. The former can be proved $\mathcal{NP}$–hard by reduction from the latter, both for bidirectional and for directed calls. The best approximation algorithms for call scheduling in rings known so far simply ignore an arbitrary edge of the ring and schedule all calls on the remaining linear array network. This approach can be shown to yield approximation ratio 2, but better approximation seems surprisingly difficult.

Meshes In mesh networks a scheduling algorithm has many possible alternatives for routing a call. Unfortunately, this freedom does not make the problem easier. We have proved by a reduction from arc-coloring that optimal call scheduling in mesh networks is $\mathcal{NP}$–hard, both for bidirectional and directed calls.

3 Conclusion

We have studied the off-line call scheduling problem in trees, rings and meshes under the assumption that all calls have unit bandwidth requirements and unit durations. If arbitrary bandwidth requirements are allowed, call scheduling is $\mathcal{NP}$–hard even on a single link, and if arbitrary durations are allowed it is $\mathcal{NP}$–hard even for linear arrays. Future work should concentrate on finding and improving approximation algorithms for the unrestricted call scheduling problem in various classes of networks.

References

[1] T. Erlebach and K. Jansen. Scheduling of virtual connections in fast networks. In *Proc. of the 4th Workshop on Parallel Systems and Algorithms PASA '96*, pages 13–32, 1996.

[2] A. Feldmann. On-line call admission for high-speed networks (Ph.D. Thesis). Technical Report CMU-CS-95-201, School of Computer Science, Carnegie Mellon University, 1995.

[3] C. Kaklamanis and P. Persiano. Efficient wavelength routing on directed fiber trees. In *Proc. of the 4th Ann. European Symp. on Algorithms ESA '96*, 1996.

[4] M. Mihail, C. Kaklamanis, and S. Rao. Efficient access to optical bandwidth. In *Proc. of the 36th IEEE Symp. on Foundations of Computer Science FOCS '95*, pages 548–557, 1995.

[5] T. Nishizeki and K. Kashiwagi. On the 1.1 edge-coloring of multigraphs. *SIAM J. Disc. Math.*, 3(3):391–410, 1990.