Packet Routing on the Grid*

Britta Peis, Martin Skutella, and Andreas Wiese

Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany
{peis,skutella,wiese}@math.tu-berlin.de

Abstract. The packet routing problem, i.e., the problem to send a given set of unit-size packets through a network on time, belongs to one of the most fundamental routing problems with important practical applications, e.g., in traffic routing, parallel computing, and the design of communication protocols. The problem involves critical routing and scheduling decisions. One has to determine a suitable (short) origin-destination path for each packet and resolve occurring conflicts between packets whose paths have an edge in common. The overall aim is to find a path for each packet and a routing schedule with minimum makespan.

A significant topology for practical applications are grid graphs. In this paper, we therefore investigate the packet routing problem under the restriction that the underlying graph is a grid. We establish approximation algorithms and complexity results for the general problem on grids, and under various constraints on the start and destination vertices or on the paths of the packets.

1 Introduction

In this paper, we study the packet routing problem on grid graphs. In an instance of this problem we are given a set of unit-size packets with specified source and destination vertices. First, we need to find a path for each packet along which we want to route it. Then we need to find a routing schedule to transfer the packets through the network such that each link in the network can be used by at most one packet at a time. The overall goal is to find a path assignment and routing schedule that minimizes the makespan, i.e., the time when the last packet has reached its destination. We study also the special case that the paths of the packets are given and we only need to find the routing schedule.

The packet routing problem has several applications in practice, e.g., in parallel computing or in cell structured networks. In those settings, packets of information need to be transferred through the network. In order for the network to operate efficiently, it is required that the packets reach their respective destinations as quickly as possible. Therefore, the paths of the packets and the routing schedule need to be computed such that the packets encounter as few delay as possible. One of the most common natural topologies of the routing problem

* This work was partially supported by Berlin Mathematical School, by DFG research center Matheon, and by the DFG Focus Program 1307 within the project "Algorithm Engineering for Real-time Scheduling and Routing".

A. López-Ortiz (Ed.): LATIN 2010, LNCS 6034, pp. 120–130, 2010.

in practical applications are grid graphs, e.g. in parallel computing. Therefore, we take this special structure into account when looking for efficient solution methods.

1.1 Packet Routing Problem

The packet routing problem is defined as follows: Let $G = (V, E)$ be an undirected graph (in our case this will usually be a grid graph). A packet $M_i = (s_i, t_i)$ is a tuple consisting of a start vertex $s_i \in V$ and a destination vertex $t_i \in V$. Let $\mathcal{M} = \left\{M_1, M_2, M_3, ..., M_{|\mathcal{M}|}\right\}$ be a set of packets. Then $(G, \mathcal{M})$ is an instance of the *packet routing problem with variable paths.* The problem has two parts: First, for each packet M_i we need to find a path $P_i = (s_i = v_0, v_1, ...v_{\ell-1}, v_\ell = t_i)$ from s_i to t_i such that $\{v_i, v_{i+1}\} \in E$ for all i with $0 \leq i \leq \ell - 1$. Assuming that it takes one timestep to send a packet along an edge, we need to find a routing schedule for the packets such that each message M_i follows its path P_i from s_i to t_i and each edge is used by at most one packet at a time. We assume that time is discrete and that all packets take their steps simultaneously. The objective is to minimize the makespan, i.e., the time when the last packet has reached its destination vertex. For each packet M_i we define $\bar{D}_i$ to be the length of the shortest path from s_i to t_i, assuming that all edges have unit length. Moreover, the *minimal dilation* $\bar{D}$ is defined by $\bar{D} := \max_i \bar{D}_i$. It holds that $\bar{D}$ is a lower bound for the length of an optimal schedule.

Since there are algorithms known to determine paths for routing the packets (see [4,13,24] or simply take shortest paths) we will also consider the *packet routing problem with fixed paths.* An instance of this problem is a tuple $(G, \mathcal{M}, \mathcal{P})$ such that G is a (grid) graph, $\mathcal{M}$ is a set of packets and $\mathcal{P}$ is a set of predefined paths, one for each packet. Since the paths of the packets are given in advance they do not need to be computed here. The aim is to find a schedule with the properties described above such that the makespan is minimized. For each packet M_i we define D_i to be the length of the path P_i, again assuming that all edges have unit length. Like above we define the *dilation* D by $D := \max_i D_i$. For each edge e we define C_e to be the number of paths that use e. Then we define the *congestion* C by $C := \max_e C_e$. It holds that C and D are lower bounds for the length of an optimal schedule.

We distinguish between grid graphs in which two packets are allowed to use an edge in opposite directions at the same time, or not. The infinite grid graph $G_\# = (V_\#, E_\#)$ is the undirected graph consisting of the vertices $V_\# = \{v_{i,j} | i, j \in \mathbb{Z}\}$ and the edges

$$E_\# = \{\{v_{i,j}, v_{i',j'}\} \mid |i - i'| + |j - j'| = 1\}$$

The directed graph $\overleftrightarrow{G}_\# = \left(V_\#, \overleftrightarrow{E}_\#\right)$ is the bidirected infinite grid graph with $\overleftrightarrow{E}_\# = \{(u, v), (v, u) \mid \{u, v\} \in E_\#\}$. We will consider infinite grid graphs rather than finite grids because we want the borders of a finite grid not to have any