Deterministic 1-k Routing on Meshes
With Applications to Worm-Hole Routing

Jop F. Sibeyn* Michael Kaufmann†

Abstract

In 1-k routing each of the \( n^2 \) processing units of an \( n \times n \) mesh connected computer initially holds 1 packet which must be routed such that any processor is the destination of at most \( k \) packets. This problem has great practical importance in itself and by its implications for hot-potato worm-hole routing.

We present a near-optimal deterministic algorithm running in \( \sqrt{k} \cdot n/2 + O(n) \) steps, and an algorithm with slightly worse routing time but working queue size three. Nontrivial extensions are given to 1-k routing, and for routing on higher dimensional meshes. We show that under a natural condition 1-k routing can be performed in \( O(n) \) steps. Finally we show that \( k-k \) routing can be performed in \( O(k \cdot n) \) steps with working queue size four. Hereby hot-potato worm-hole routing can be performed in \( O(k^{3/2} \cdot n) \) steps.

Keywords: theory of parallel and distributed computation, meshes, packet routing, hot-potato worm-hole routing.

1 Introduction

Various models for parallel machines have been considered. One of the best studied machines with a fixed interconnection net-

---

*Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. E-mail: jopsi@mpi-sb.mpg.de. This research was partially supported by EC Cooperative Action IC-1000 (Project ALTEC: Algorithms for Future Technologies).

†Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, 72076 Tübingen, Germany. E-mail: mk@informatik.uni-tuebingen.de

work, is the MIMD mesh. In this model the processing units, PUs, form an array of size \( n \times n \) and are connected by a two-dimensional grid of communication links (see Section 2 for details).

The problems concerning the exchange of packets among the PUs are called routing problems. Here the destinations of the packets are known beforehand. The packets must be send to their destination such that at most one packet passes through any wire during a single step. The quality of a routing algorithm is determined by (1) its running time, the maximum time a packet may need to reach its destination, and (2) its queue length, the maximum number of packets any PU may have to store.

A special case of the routing problem is permutation routing. In permutation routing, each PU is the origin of at most one packet and each PU is the destination of at most one packet. Permutation routing has been considered extensively. Optimal algorithms were found [15, 10, 2].

When the size of the packets is so large that they cannot be transferred over a connection in a single step, the packets have to be split into several flits. The routing of these flits is considered in the \( k-k \) routing problem: each PU is assumed to send and receive at most \( k \) packets. If the flits are routed independently of each other we speak of multi-packet routing. Multi-packet routing is also important when the PUs have to route packets to several destinations. Multi-packet routing algorithms [6, 9, 7] solve this task much faster than routing the packets one-by-one. Alternatively, the flits can be routed as a kind of worm such that con-
secutive flits of a packet reside in adjacent PUs during all steps of the routing: cut-through routing [6]. If there is the additional condition that the worms may be expanded and contracted only once, then this variant is called worm-hole routing [5, 12]. Unlike the other more theoretical models, worm-hole routing has direct applications in many parallel machines [1, 13, 18, 3].

We consider an original variant of the routing problem: the routing of 1-k distributions, under which every PU is sending at most one packet, but may be the destination of up to k packets. 1-k routing reflects practical purposes better than the routing of permutations: if the PUs are working independently of each other and generate packets that have to be transferred to other PUs, then it is unrealistic to assume that every PU is the destination of at most one packet. The parameter $k$, $1 \leq k \leq n^2$, need not to be known by the PUs, but is needed for stating the complexity of the problem. The 1-k routing problem also has implications for hot-potato worm-hole routing. Hot-potato routing is a routing paradigm in which packets may never be queued at a PU but have to keep moving at all times until they reach their destination [4]. Like worm-hole routing, this model is used in many systems. In a recent paper of Newman and Schuster [12] it is demonstrated that under a light condition any efficient 1-k routing algorithm with working queue size at most four is useful as a subroutine for the hot-potato worm-hole routing problem. We are among the first to perform a detailed analysis of the 1-k routing problem for meshes. In [11] Makedon and Symvonis independently consider the problem. Their algorithm has running time $O(\sqrt{k}\cdot n)$. Furthermore, for certain expander networks, Peleg and Upfal have studied in [14] the related $l$-$k$ routing problem.

It turns out (Section 5) that under a natural condition $l$-$k$ distributions can be routed in $O(l\cdot n)$ time for $l$ up to $n\cdot k$. Informally, the condition is that the 'density of the destinations increases gradually', excluding large areas in which all PUs receive $k$ packets.

In Section 6 we aim for a working queue size four or less, in order to create a subroutine for the hot-potato worm-hole routing algorithm of [12]. This is achieved with routing time $5\cdot \sqrt{k}\cdot n + o(\sqrt{k}\cdot n)$. In the algorithm the mesh is subdivided into $k$ squares of size $n/\sqrt{k} \times n/\sqrt{k}$ and the packets are redistributed such that every square holds at most one packet for each destination. Then these squares are rotated along a Hamiltonian cycle and after