

$k - k$ Routing, $k - k$ Sorting, and Cut Through Routing on the Mesh¹

Sanguthevar Rajasekaran

Department of Computer and Information Science
Univ. of Pennsylvania, Philadelphia, PA 19104.

Abstract In this paper we present randomized algorithms for $k - k$ routing, $k - k$ sorting, and cut through routing on an $n \times n$ mesh connected computer (referred to simply as the mesh). The stated resource bounds hold with high probability. The algorithm for $k - k$ routing runs in $\frac{k}{2}n + o(kn)$ steps. We also show that $k - k$ sorting can be accomplished within $\frac{k}{2}n + 2n + o(kn)$ steps, and cut through routing can be done in $\frac{kn}{2} + \frac{3}{2}n + o(kn)$ steps.

$\frac{kn}{2}$ is a known lower bound for all the three problems (which is the bisection bound), and hence our algorithms are very nearly optimal. All the above mentioned algorithms have optimal queue length. These algorithms also extend to higher dimensional meshes.

1 Introduction

1.1 Packet Routing

Fixed connection machines are some of the most practical models of parallel computing, as inferred from the parallel computers available today. A fixed connection machine is usually represented as a directed graph whose nodes correspond to processing elements, and whose edges correspond to communication links. The speed of a parallel computer is determined by 1) the computing power of component processors, and 2) the speed of inter-processor communication. Nowadays the computing power of individual processing elements can be made arbitrarily high owing to the decline in hardware costs. Thus the speed of any parallel machine crucially depends on how fast the inter-processor communication is.

A single step of inter-processor communication in a fixed connection network can be thought of as the following task (also called *packet routing*): Each node in the network has

¹Preliminary versions of some of the results in this paper were presented in SPAA 1992 [?].

a packet of information that has to be sent to some other node. The task is to send all the packets to their correct destinations as quickly as possible such that at most one packet passes through any wire at any time.

A special case of the routing problem is called the *partial permutation routing*. In partial permutation routing, each node is the origin of at most one packet and each node is the destination of no more than one packet. A packet routing algorithm is judged by 1) its *run time*, i.e., the time taken by the last packet to reach its destination, and 2) its *queue length*, which is defined as the maximum number of packets any node will have to store during routing. Contentions for edges can be resolved using a *priority scheme*. Furthest destination first, furthest origin first, etc. are examples of priority schemes. We assume that a packet not only contains the message (from one processor to another) but also the origin and destination information of this packet. An algorithm for packet routing is specified by 1) the path to be taken by each packet, and 2) a priority scheme.

1.2 Different Models of Packet Routing and $k - k$ Sorting

How large a packet is (when compared with the channel width of the communication links) will determine whether a single packet can be sent along a wire in one unit of time. If a packet is very large it may have to be split into pieces and sent piece by piece. On this criterion many models of routing can be derived. A packet can be assumed to be either atomic (this model is known as the *store and forward model*), or much larger than the channel width of communication links (thus necessitating splitting).

In the later, if each packet is broken up into k pieces (also called *flits*), where k depends on the width of the channel, the routing problem can be studied under two different approaches. We can consider the k flits to be k distinct packets, which are routed independently. This is known as the *multipacket routing approach* [?]. Each flit will contain information about its origin and destination. The problem of $k - k$ routing is one where $\leq k$ packets originate from any node and $\leq k$ packets are destined for any node under the multipacket model.

Alternatively, one can consider the k flits to form a *snake*. All flits follow the first one, known as the head, to the destination. A snake may never be broken, i.e., at any given time, consecutive flits of a snake are at the same or adjacent processors. Only the head has to contain the origin and destination addresses. This model is called the *cut through routing with partial cuts* or simply the *cut through routing* [?].

The problem of $k - k$ sorting on any fixed connection machine is the problem of sorting where exactly k packets are input at any node.