Abstract. We discuss recent research evaluating the benefits of certain link labellings in processor networks. Such a labelling, called Sense of Direction or SoD, allows processors to communicate more efficiently with each other, and to exploit topological properties of the network algorithmically.

We shall define sense of direction for several specific classes of network, and show how the election problem on rings can be solved more efficiently if chords and a sense of direction are available. We shall show that elections can be performed with linear complexity in Hypercubes and Cliques if SoD is available, but also that a randomised algorithm can achieve the same complexity without using SoD. Algorithms to compute an SoD in networks were none is given will also be presented.

Some results in this paper were previously unpublished or only presented in technical reports.
1. The group definition and characterisation of Sense of Direction; see Subsection 1.3.
2. Some results (e.g., Theorems 6 and 28, Algorithm 4) are stated more generally than they were before.
3. Improved performance of Attiya’s election algorithm; see Subsection 2.3.
4. The linear chordal ring algorithm with one chord; see Subsection 2.4.

The work reported in Subsections 2.3 and 2.4 was done with Andreas Fabri.
communicate by sharing registers are also common, but in this article we shall assume message based communications.

To study the distributed aspects of these computations, the network is conveniently modelled as a graph, where the processors constitute the nodes and an edge exists between nodes that are connected physically. For example, Inmos produces so-called Transputers, which are powerful processors with four ports to allow connection to four other Transputers. It is possible to connect 16 Transputers in a four-dimensional cube shape as depicted in Figure 1. The system on the left would have no connection to the outside world, so it would not be possible to load programs or inputs, or learn about the outputs of the computation. The processor network must be connected to a host, as depicted on the right, for example by “inserting” the host between two processors that are directly connected in the “idealised” picture on the left. After downloading the programs and inputs, the host computer simply relays all communication between the two processors. We shall ignore the host or other means of communication with the outside world in the rest of the paper.

![Figure 1 Hypercube network of degree four.](image)

Large scale computer networks (such as the Internet) or local area networks often have an arbitrary graph as their topology, because they have evolved gradually from a small network by adding sites on incidental basis. Processor networks, however, are usually constructed with a regular, symmetric topology, and the local names of the connections of a processor are meaningful with respect to the position of the edge in the network. If connection names are meaningful we say the network is equipped with a sense of direction. The goal of this paper is to give insight in how a sense of direction can be exploited, by presenting efficient algorithms that rely critically on sense of direction.

**Overview of the paper.** This section introduces some general notions and problems further discussed in later sections. Subsection 1.1 explains the model used for processor networks. Subsection 1.2 introduces the sense of direction informally, and Subsection 1.3 provides some formal definitions. Subsection 1.4 de-