Sense of direction in *tori*

Szymon Szulc

January 2024

1 Topological structure

Informally *torus* is a two-dimensional (wrap-around) array of n processors. It is a regular graph. Each processor has exactly 4 neighbours. *Torus* of dimensions $a \times b$ has $n = ab$ processors $v_{i,j}(0 \leq i \leq a-1, 0 \leq j \leq b-1)$. Each node $v_{i,j}$ is connected by a bidirectional link with $v_{i,j+1}$, $v_{i,j-1}$, $v_{i+1,j}$, $v_{i-1,j}$, where all the operations on the first index are *modulo* a, while those on the second index are *modulo* b (Figure 1). For the rest of sections, I will assume that, *torus* is a square $a = b = \sqrt{n}$.

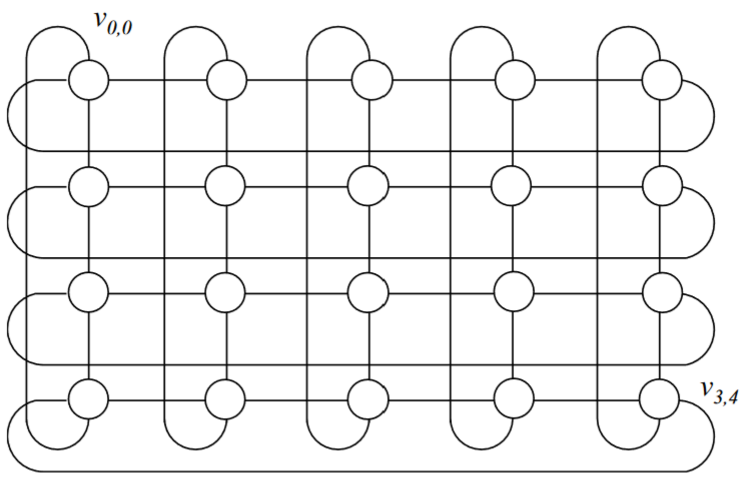

Figure 1: Torus 4×5 [1]

2 Sense of direction in *tori*

Imagine you are looking at the network from a bird's-eye view. Each edge has a label from the set $\{north, south, east, west\}$ assigned in the natural globally consistent way (Figure 2). Having such a labeling, it is trivial to check if 2 paths lead to the same processor. The lack of sense of direction is known to increase the message complexity of the election problem. For example in a *torus* we can choose leader in $\Theta(n)$, instead of $\Theta(n \log n)$ [1], when nodes have the sense of direction.

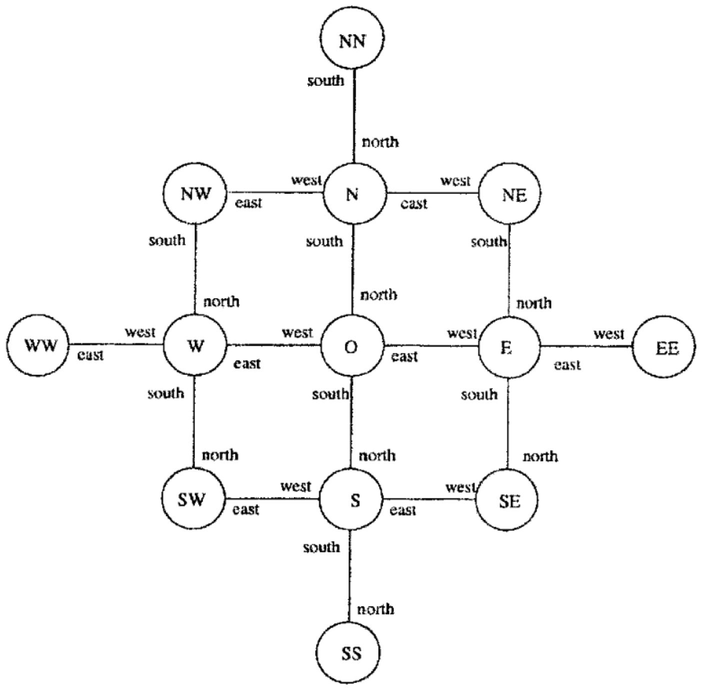

Figure 2: Labels of the edges [1]

3 Outline of leader election algorithm

The main idea behind Peterson's algorithm for square bidirectional *tori* is marking territory. In each stage, each processor tries to mark off the boundary of a square distance d on a side ($d = \alpha^i$ for some constant α). If node closes boundary, it survives to the next stage. If node's message encounters territory with lower *id.* it does not survive, because boundary with lower *id* purges its message. Peterson shows, that $\alpha = 1.1795$ to obtain $\Theta(n)$. Peterson also suggested, that "the algorithm only needs to mark off a square; the orientation of the square is irrelevant."[2]

4 Getting a local sense of direction

4.1 Main idea

We only need the processor to pass message in straight line or make appropriate turn. Is relatively easy to obtain. Each processor has to know the identity and the position of each processor at distance 2.

4.2 Algorithm

In a nutshell, each processor sends its identity to each of its neighbors, which forwards it to its three remaining neighbors.

Algorithm 1 Bernard Mans' algorithm for local sense of direction in *tori*

```
$\forall$ link $r$ initialize $H1_r := \emptyset,\ H2_r := \emptyset$
$\forall$ link $k$ send ONE(myId) on arc k

repeat
    RECEIVE
until received 16 messages

procedure RECEIVE($message(type,\ id,\ link)$)
    if $type = ONE$ then
        $H1_{\text{link}} := \{id\}$
        $\forall$ link $k \neq link$ send TWO(id) on arc k
    end if
    if $type = TWO$ then
        $H2_{\text{link}} := H2_{\text{link}} \cup \{id\}$
    end if
end procedure
```

After execution of this algorithm each node knows:

- its immediate neighbours
- links must be used to reach them
- nodes at distance 2

The amount of information is enough to pass messages straight and make appropriate turns. If $H2_i \cap H2_j = \emptyset$, then links i and j are parallel, else they are perpendicular. (this statement is *true* for $\sqrt{n} > 4$)

4.3 Handrail

We already know how to pass messages in straight lines. But how to make appropriate turns? Initiator node chooses 2 perpendicular links. Using one of them, sends message containing other link immediate neighbour - *handrail*. When node receives such message, it exactly knows, which link (k) to choose to make appropriate turn ($H2_{\text{linkFrom}} \cap H2_k = \textit{handrail}$).

4.4 Complexity

Each processor sent 4 messages to its immediate neighbors and passed 4×3 messages, that is $16n$ messages overall. So this algorithm has message complexity $\Theta(n)$.

References

[1] Bernard Mans. “Optimal Distributed Algorithms in Unlabeled Tori and Chordal Rings”. In: *JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING* (1997).

[2] Gary Lynn Peterson. “Efficient Algorithms for Elections in Meshes and Complete Networks”. In: *TR 140* (1985).