Homogeneous Routing for Homogeneous Traffic Patterns on Meshes

L.D. Aronson

Abstract—The performance analysis of dynamic routing algorithms in interconnection networks of parallel computers has thus far predominantly been done by simulation studies. A limitation of simulation studies is that they usually only hold for specific combinations of network, routing algorithm, and traffic pattern. In this paper, we derive saturation point results for the class of homogeneous traffic patterns and a large class of routing functions on meshes. We show that the best possible saturation point on a mesh is half the best possible saturation point on a torus. We also show that, if we restrict ourselves to homogeneous routing functions, the worst possible saturation point on a mesh is again half the best possible saturation point. Finally, we present a class of homogeneous routing functions, containing the well-known e-cube routing function, which are all optimal for all homogeneous traffic patterns.

Index Terms—Routing, mesh, torus, homogeneous, automorpshism, parallel communication, parallel computer, performance analysis, saturation, traffic pattern.

1 Introduction

An important goal of designing algorithms for routing messages in parallel architectures is that they behave efficiently for frequently occuring communication patterns. However, verifying that they indeed have this desired property turns out to be very difficult. The dominant analysis technique used so far has been the simulation of specific combinations of network architectures, routing algorithms, and communication patterns. While simulation provides some insight, general properties of routing algorithms, such as scalability, cannot be derived.

Also, most papers concerning dynamic routing in parallel computers suffer from all, or nearly all, of the following problems (e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]):

1. The routing algorithms are not precisely specified, therefore, the exact routing algorithm is not known.
2. The traffic patterns are not precisely specified, therefore, the exact pattern is not known.
3. It is not clear what the exact performance measure is.
4. The only knowledge about the performance is obtained by simulation.
5. The performance is only known in one, or a few, specific situations.
6. The influence of many parameters (e.g., packet length) is not known.

Papers like [17] and [18] are exceptions, since they present a precise model and precise results. The first paper, analyzing the e-cube routing function under a uniform traffic pattern on meshes, can be related to our work, although the main performance measure is the average delay, whereas we are interested in saturation points. The second paper considers hypercubes, an architecture we will not explicitly consider.

In this paper, a new analysis model is presented which alleviates the aforementioned deficiencies. The main characteristics of the model and the derived results are the following:

1. Formal definitions of the notions *routing function* and *traffic patterns* are given (note that we use routing functions instead of routing algorithms).
2. The concept of *saturation point* is used as the basic performance measure.
3. All derived theoretical results are proven formally.
4. All results hold for large classes of routing functions and traffic patterns.
5. Our model has a high level of abstraction, which allows us to ignore a number of parameters, such as the switching technique.

The saturation point is our performance measure. It is the injection rate at which the average delay grows without bound. Injection rates are denoted by λ. To compute the saturation point, we compare the capacity of a link to the amount of traffic using that link per time unit. We assume that all links have the same capacity c. The amount of traffic using a link l relative to the injection rate is called the *load* of link l and denoted as $\alpha(l)$. We assume that all nodes have an infinite number of buffers available. Limiting the number of buffers will in reality lead to a decrease of the saturation point, so the results presented in this paper yield upper bounds for saturation points in situations with a limited number of buffers. Furthermore, we consider the effect of the routing function in its purest form without having to consider details which do not influence the saturation point. Since saturation occurs at the link which is most heavily loaded, we also define the maximum load α_{max}. Using the

• *The author is with Information Technology and Systems, Delft University of Technology, The Netherlands. E-mail: L.D.Aronson@kabelfoon.nl.*

Manuscript received 12 Sept. 1997; revised 4 Oct. 1999; accepted 11 Apr. 2000.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 105656.

TABLE 1
Maxium Loads for Links in Dimension i for Regular Routing Functions and Regular Traffic Pattern T

torus	mesh (best)	mesh (worst)
$\frac{1}{2}\Lambda_i(T)$	$\Lambda_i(T)$	$2\Lambda_i(T)$

above, we derive that the saturation point λ_{sat} is c divided by α_{max}.

In this paper, we consider "homogeneous" traffic patterns and routing functions. What we mean by this term is explained in Section 3. The main analysis results of this paper concern tori and meshes. The homogeneity of tori is used to derive the maximum load in each dimension for regular routing functions and traffic patterns. This maximum load will be expressed in a function Λ on the traffic pattern T. On tori, the derived maximum load in dimension i is $\frac{1}{2}\Lambda_i(T)$ (see Table 1). Then, we show that the maximum load for homogeneous traffic patterns on meshes is $\Lambda_i(T)$, twice the best possible maximum load for tori. Meshes are less regular than tori, but their regularities can still be used to derive performance results. We show that any homogeneous routing function on a mesh has, at most, a maximum load of $2\Lambda_i(T)$. Finally, we give a class of routing functions on meshes which are optimal for the considered traffic patterns. In other words, all routing functions in the given class have a maximum load of $\Lambda_i(T)$ for any homogeneous traffic pattern T.

To get to these results, we first introduce, in Section 2, our model of graphs, traffic patterns, and routing functions, and our basic method of analysis, which enables us to compute saturation points. Then in Section 3, to be able to handle the regularities in tori and meshes, we introduce the concept of automorphism and the concept of partial automorphism. Where automorphisms are quite usual in graph theory, we extend the notion from graphs to traffic patterns and routing functions. As will be shown, the use of automorphisms greatly simplifies the analysis of routing algorithms, without sacrificing the possibility of optimality. We also introduce the concept of homogeneity, which is an extreme form of regularity. Homogeneity plays an important role in our performance results.

Section 4 contains our results. First, homogeneous routing under homogeneous traffic for tori is analyzed. Then, we show that, for meshes, the optimum is at least twice as bad as for tori, when considering the same homogeneous traffic pattern. The use of minimal homogeneous routing functions limits the worst-case performance by another factor of two. Our final result shows that the routing functions in a given subclass of the minimal homogeneous routing functions perform optimally under all traffic patterns under consideration.

2 Model

In this section, we present our model and notation. The three main parameters are interconnection graph, traffic pattern, and routing function. We also present a method to compute the saturation point of a routing function under a certain traffic pattern. An overview of the notation used in this paper is given is given in Table 2.

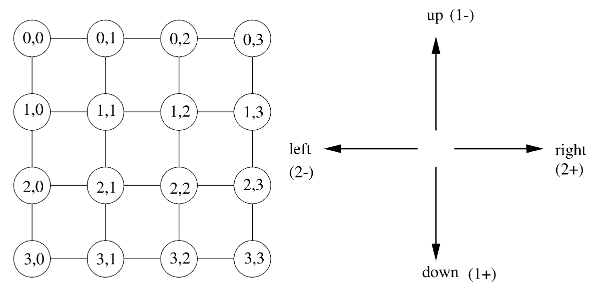

Fig. 1. Directions in a mesh.

2.1 Interconnection Graphs

We consider a network of processors represented by a graph $G = (V, E)$. V is the set of vertices, also called nodes or processors. E is the set of edges, also called links. Each edge is a unidirectional link. In most practical cases, links appear in pairs; one for each direction. The meaning of the symbols used are defined in Table 2.

Definition 2.1. *A graph G is a tuple (V, E), where V is a set of nodes (Vertices) and E is a set of links (Edges).*

In this paper, we are mainly concerned with tori and meshes. The following example introduces the notation involved with these graphs.

Definition 2.2. *Let $N \geq 3$ be an integer. We define N-meshes and N-tori. Let $V = \{(x_1, x_2) \bmod 0 \leq x_1, x_2 \leq N-1\}$. An N-mesh is a graph $G = (V, E)$ such that:*

$$E = \{\langle \mathbf{x}, \mathbf{y}\rangle \mid \mathbf{x}, \mathbf{y} \in V \wedge ((x_1 = y_1 \wedge x_2 = y_2 \pm 1) \vee (x_1 = y_1 \pm 1 \wedge x_2 = y_2))\}.$$

An N-torus has a similar definition, but now using mod N:

$$E = \{\langle \mathbf{x}, \mathbf{y}\rangle \mid (x_1 = y_1 \wedge x_2 = (y_2 \pm 1) \bmod N) \vee (x_1 = (y_1 \pm 1) \bmod N \wedge x_2 = y_2)\}.$$

The vertical links are denoted by E_1, the horizontal links by E_2. This is illustrated by Fig. 1. Also, we denote the vertical links in the positive direction by E_{1+} and those in the negative direction by E_{1-}. Analogously we will use E_{2+} and E_{2-}.

The dimensions of meshes and tori will play an important role. Therefore, we define a seperate distance measure for each dimension. We will denote *source nodes* by $\mathbf{s}$ and *destination nodes* by $\mathbf{d}$.

Definition 2.3. *The ith dimension torus distance between $\mathbf{s}$ and $\mathbf{d}$, denoted as $\mathrm{dis}_{i,\mathbf{sd}}$, is defined as:*

$$\mathrm{dis}_{i,\mathbf{sd}} = \min(|s_i - d_i|, N - |s_i - d_i|).$$

2.2 Traffic Patterns

Given a graph $G = (V, E)$, we characterize a traffic pattern T by a $|V| \times |V|$ matrix (τ_{sd}), where τ_{sd} specifies the number

TABLE 2
The Meaning of Used Symbols

symbol	meaning
$\mathbf{s}$	the source node of a packet
$\mathbf{d}$	the destination node of a packet
$\mathbf{x}$	the current node of a packet
$\mathbf{y}$	the next node of a packet
$\mathrm{dis}_{i,\mathbf{xy}}$	the distance in dimension i between node $\mathbf{x}$ node $\mathbf{y}$
$\mathbb{P}$	set of all packets
P	packet
$\mathrm{src}(P)$	source of a packet P
$\mathrm{dst}(P)$	destination of a packet P
$\mathrm{mark}(P)$	mark of a packet P
$l = \langle \mathbf{x}, \mathbf{y} \rangle \in E$	a link of the network
E_i	all links in dimension i
E_{i+}	all links in dimension i in positive direction
E_{i-}	all links in dimension i in negative direction
c	the capacity of the links
T	a traffic pattern
τ	the traffic pattern matrix
R	a routing function
$\rho(l)$	the routing matrix of a link
$\alpha(l)$	the load of a link under P and R
$\alpha(T, R)$	the load of a traffic pattern and a routing function

of packets $\mathbf{s}$ sends to $\mathbf{d}$ relative to the total number of messages sent by $\mathbf{s}$. The following must hold:

$$\sum_{d \in V} \tau_{\mathrm{sd}} = 1,$$

for all $\mathbf{s} \in V$.

Example 2.4. Consider an N-mesh or N-torus. Suppose that we have a traffic pattern such that each node $\mathbf{s}$ sends an equal amount of its packets to each of the nodes which is one horizontal step, one vertical step, or both, away from that node $\mathbf{s}$ (Fig. 2). We do this modulo N, so e.g., node $(0, 0)$ sends $\frac{1}{8}$ of its packets to node $(N-1, N-1)$. We will call this traffic pattern the local traffic pattern, or T_{loc}. We can model this as follows:

$$\tau_{sd} = \begin{cases} \frac{1}{8}, & \text{if } \mathrm{dis}_{1,\mathbf{sd}} \leq 1,\ \mathrm{dis}_{2,\mathbf{sd}} \leq 1, \text{and } \mathbf{s} \neq \mathbf{d}; \\ 0, & \text{otherwise.} \end{cases}$$

2.3 Routing Functions

We introduce the concept of a routing function. Routing functions are applied in a node, given a packet and the state of the network. In this paper, we refrain from considering the network state, because that would add a lot of complexity without adding much to the results of this paper. However, a part of the results can be extended to adaptive routing functions (i.e., routing functions which effectively use state information), see [19].

Packets. In this paper, only the routing information stored in the packets is relevant; this information is usually stored in the header. We denote the set of all possible packet headers by the packet set $\mathbb{P}$. Also, when we say "packet," we in fact mean the header. We use the symbol P to denote packets. All kinds of routing information can be stored in the packet header, but at least the destination of the packet should be available. For this paper it is convenient to store three kinds of information in a packet P: the source of a packet, denoted by $\mathrm{src}(\mathrm{P})$, the destination of a packet, denoted by $\mathrm{dst}(\mathrm{P})$, and a mark, which is denoted by $\mathrm{mark}(\mathrm{P})$. The mark is an element of $\mathbb{N}$ and can be used to pass information from one router to the next.

Routing functions will be denoted by R. Routing functions take the current node $\mathbf{x}$ and the current packet P as input and give a probability distribution on pairs $(\mathbf{y}, P')$ as output. This means that if, for example, $R(\mathbf{x}, P)(\mathbf{y}, P') = \frac{1}{4}$, there is a 25 percent probability that packet P entering node $\mathbf{x}$ is forwarded as P' to node $\mathbf{y}$. A routing function has to comply with some restrictions. First, if a packet is at its destination, then it should not be forwarded anymore. Otherwise, the sum of the probabilities on all output pairs $(\mathbf{y}, P')$ should be 1. Second, a routing

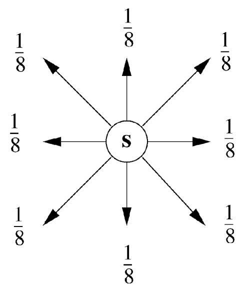

Fig. 2. The local traffic pattern $T_{\rm loc}$.

function can only send a packet residing at node $\mathbf{x}$ to a node $\mathbf{y}$ if there exists a link between $\mathbf{x}$ and $\mathbf{y}$. Third, the output packet P' should have the same source and destination as the input packet P. Putting this together, we get the following definition.

Definition 2.5. *Let $G = (V, E)$ be a directed graph, let $\mathbf{x} \in V$, and let $\mathbb{P}$ be a packet set. A routing function is a function $R : V \times \mathbb{P} \to [0,1]^{V\times\mathbb{P}}$[1] such that:*

1. *For all $\mathbf{x} \in V$ and $P \in \mathbb{P}$:*

$$\sum_{\mathbf{y}\in V}\sum_{P'\in\mathbb{P}} R(\mathbf{x}, P)(\mathbf{y}, P') = \begin{cases} 0, & \text{if } \mathbf{x} = \mathrm{dst(P)}, \\ 1, & \text{otherwise;} \end{cases}$$

2. *For all $\mathbf{x}, \mathbf{y} \in V$ and $P, P' \in \mathbb{P}$:*

$$R(\mathbf{x}, P)(\mathbf{y}, P') > 0 \;\Rightarrow\; \langle \mathbf{x}, \mathbf{y}\rangle \in E;$$

3. *For all $\mathbf{x}, \mathbf{y} \in V$ and $P, P' \in \mathbb{P}$:*

$$R(\mathbf{x}, P)(\mathbf{y}, P') > 0 \;\Rightarrow\; \mathrm{src(P)} = \mathrm{src(P')} \text{ and } \mathrm{dst(P)} = \mathrm{dst(P')}.$$

The following example is to illustrate both the application of the above definition to formalize routing strategies and to introduce an example routing function which will be used throughout this paper.

Example 2.6. Consider an N-mesh. Let us introduce the following routing strategy: If a packet is already at its destination row or column, then route it along that row or column to its destination. Otherwise, if the packet has mark 0, then route it horizontally towards its destination and mark the packet either 0 or 1 with 50 percent probability. If the packet is marked 1, route it vertically towards its destination and mark it 0. Initially, the packet has mark 0.

We can now write this strategy down by means of a routing function $R_{\rm ex}$. Let $P = (\mathbf{s}, \mathbf{d}, k)$. We only give the specification for $x_1 \le d_1$ and $x_2 \le d_2$; the other cases can be specified analogously.

1. Let $x_1 = d_1$ and $x_2 < d_2$. Then $R_{\rm ex}(\mathbf{x}, P)((x_1, x_2 + 1), P) = 1$.
2. Let $x_2 = d_2$ and $x_1 < d_1$. Then $R_{\rm ex}(\mathbf{x}, P)((x_1 + 1, x_2), P) = 1$.

1. $[0,1]^{V\times\mathbb{P}}$ is the set of all functions $f : V \times \mathbb{P} \to [0,1]$, so all probability functions on all node-packet pairs.

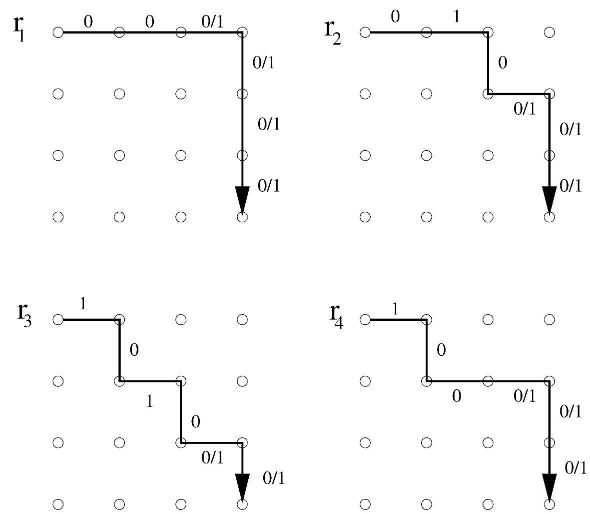

Fig. 3. Possible routes between (0,0) and (3,3) for $R_{\rm ex}$. The numbers at the routes are the values of mark(*P*) at the corresponding links.

3. Let $x_1 < d_1$, $x_2 < d_2$, and $k = 0$. Then

$$R_{\rm ex}(\mathbf{x}, P)((x_1, x_2 + 1), P')) = \begin{cases} \frac{1}{2}, & if\ P' = (\mathbf{s}, \mathbf{d}, 0); \\ \frac{1}{2}, & if P' = (\mathbf{s}, \mathbf{d}, 1). \end{cases}$$

4. Let $x_1 < d_1$, $x_2 < d_2$, and $k = 1$. Then $R_{\rm ex}(\mathbf{x}, P)((x_1 + 1, x_2), P') = 1$, where $P' = (\mathbf{s}, \mathbf{d}, 0)$.

2.4 Analysis of Routing Functions

To analyze the performance of a routing function for a specific traffic pattern, we need to compute how often each route is used. A *route* is a sequence of nodes such that between each pair of subsequent nodes there exists a link. For example, the route $\mathbf{r}$, starting in s, and going through $\mathbf{u}$ and $\mathbf{v}$ to end in $\mathbf{d}$, will be written as:

$$r = [\mathbf{s}, \mathbf{u}, \mathbf{v}, \mathbf{d}].$$

This is a valid route if and only if $\langle \mathbf{s}, \mathbf{u}\rangle \in E$, $\langle \mathbf{u}, \mathbf{v}\rangle \in E$, and $\langle \mathbf{v}, \mathbf{d}\rangle \in E$. We denote *the set of all routes* between two nodes $\mathbf{s}$ and $\mathbf{d}$ by $\mathrm{routes}_{G,\mathrm{sd}}$. Now we can assign to each route $\mathbf{r}$ between $\mathbf{s}$ and $\mathbf{d}$ the probability that $\mathbf{r}$ will be used. We denote this probability by $\rho(r)_{\rm sd}$. The following example shows how we can compute these probabilities.

Example 2.7. Consider the routing function $R_{\rm ex}$ (see Example 2.6) on a 4-mesh. Let $\mathbf{s} = (0,0)$ and $\mathbf{d} = (3,3)$. We start with packet $P_0 = (\mathbf{s}, \mathbf{d}, 0)$. Consider the route r_1, as depicted in Fig. 3. This route is used if the first two random choices result in 0s. The third random choice does not have any influence on the route that is used. The probability that both choices are 0 is $\frac{1}{4}$. Thus, we have $\rho(r_1)_{\rm sd} = \frac{1}{4}$. Analogous results hold for the routes r_2, r_3, and r_4; all these routes have a probability of $\frac{1}{4}$ to be chosen.

We can also determine which sequences of packets yield a given route r. The set of all these packet sequences is denoted by $\mathbf{packets}(r)$.

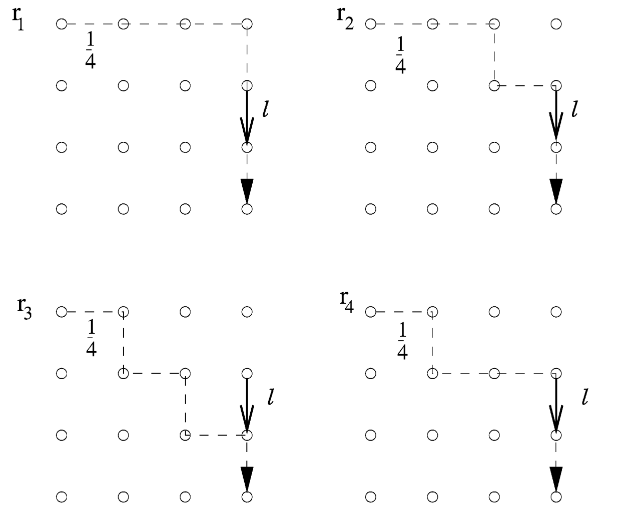

Fig. 4. The relative route use $\rho(r)_{\mathrm{sd}}$ and the relative link use $\rho(l)_{\mathrm{sd}}$ for the routing function R_{ex}.

Example 2.8. Consider again the routing function R_{ex} and Fig. 3. For convenience, we denote packets just by their marks. We can see that the sequence $[0,0,0,0,0,0]$ causes route r_1 to be used, but that the sequence $[0,0,0,1,1,1]$ also causes r_1. Thus, we obtain $\mathbf{packets}(r_1) = \{[0,0,0,0,0,0],[0,0,0,1,1,1]\}$. In an analogous manner, we get

$\mathbf{packets}(r_2) = \{[0,1,0,0,0,0],[0,1,0,1,1,1]\}$,
$\mathbf{packets}(r_3) = \{[1,0,1,0,0,0],[1,0,1,0,1,1]\}$, and
$\mathbf{packets}(r_4) = \{[1,0,0,0,0,0],[1,0,0,1,1,1]\}$.

Now we can compute how often a particular link is used on average by packets traveling between a given pair of nodes. This number is denoted by $\rho(l)_{\mathrm{sd}}$, where l is the link, $\mathbf{s}$ is the source node, and $\mathbf{d}$ the destination node. This value is usually, at most, one, but in general a route can pass the same link more than once. Therefore, we say that a link $l = \langle \mathbf{x}, \mathbf{y} \rangle$ *occurs* in a sequence of nodes $[\mathbf{x_0}, \ldots, \mathbf{x_r}]$ if there is a $0 \leq i \leq r-1$ such that $\mathbf{x} = \mathbf{x_i}$ and $\mathbf{y} = \mathbf{x_{i+1}}$. The function $\mathbf{occurs}(l, r)$ is defined as the number of times l occurs in r.

Definition 2.9. *Let $G = (V, E)$ be a directed graph and let R be a routing function. Let $l \in E$ be a link and let $\mathbf{s}, \mathbf{d} \in V$ be nodes. The function $\rho(l)_{\mathrm{sd}}$ is defined as follows:*

$$\rho(l)_{\mathrm{sd}} = \sum_{r \in \, \mathrm{routes}_{G,\mathrm{sd}}} \mathrm{occurs}(l, r) \cdot \rho(r)_{\mathrm{sd}}.$$

Example 2.10. Consider Fig. 4. The routes used by R_{ex} from $\mathbf{s} = (0,0)$ to $\mathbf{d} = (3,3)$ are again shown. For these routes hold $\rho(r_i)_{\mathrm{sd}} = \frac{1}{4}$, $i = 1, \ldots, 4$. Also, the link $l = \langle (1,3), (2,3) \rangle$ is depicted. As can be seen, all routes shown in the figure, except r_3, use l. Therefore, $\frac{3}{4}$ of the packets from $\mathbf{s}$ to $\mathbf{d}$ use link l. Formally:

$$\begin{aligned}\rho(l)_{\mathrm{sd}} &= 1 \cdot \rho(r_1)_{\mathrm{sd}} + 1 \cdot \rho(r_2)_{\mathrm{sd}} + 0 \cdot \rho(r_3)_{\mathrm{sd}} + 1 \cdot \rho(r_4)_{\mathrm{sd}} \\ &= \frac{1}{4} + 0 + \frac{1}{4} + \frac{1}{4} = \frac{3}{4}.\end{aligned}$$

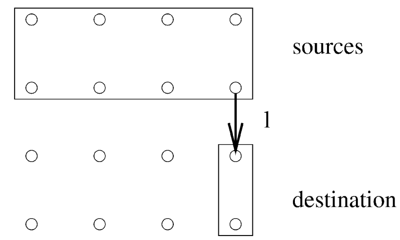

Fig. 5. Relevant source-destinations pairs for computation of $\alpha(l)$.

2.5 Load and Saturation

By using the knowledge we have about the traffic pattern and the routing function, we can count the average number of packets that cross a link per time unit and, thus, compute the saturation point. By summing weighted ρ's for all source-destination pairs, we can compute a measure for the average number of packets crossing a link relative to the injection rate. This amount is called the *link load* and is denoted by $\alpha(l)$.

Definition 2.11. *Let $G = (V, E)$ be a directed graph, let T be a traffic pattern, and let R be a routing function. Let $l \in E$ be a link. The link load $\alpha(l)$ is defined as follows:*

$$\alpha(l) = \sum_{\mathbf{s}, \mathbf{d} \in V} \tau_{\mathbf{sd}} \cdot \rho(l)_{\mathbf{sd}}.$$

Example 2.12. Consider again the traffic pattern T_{loc} and the routing function R_{ex}. We are going to compute the link load for link $l = \langle (1,3), (2,3) \rangle$.

First, we determine the relevant source-destination pairs. Consider Fig. 5. Only the sources in the upper rectangle and the destinations in the lower rectangle are relevant with respect to the use of link l, because R_{ex} is a minimal routing function.

Now, we can compute $\rho(l)_{\mathrm{sd}}$ for all relevant source-destination pairs. To save work, however, we consider all τ_{sd} simultaneously. Thus, if $\tau_{\mathrm{sd}} = 0$ for some $\mathbf{s}$ and $\mathbf{d}$, we do not have to compute $\rho(l)_{\mathrm{sd}}$ since the contribution to the link load is 0 anyway. The results of those computations are given in Table 3. Each item in the table correpsonds to a source-destination pair $\mathbf{s}$, $\mathbf{d}$. The first number of each item is τ_{sd}, the second is $\rho(l)_{\mathrm{sd}}$. If $\tau_{\mathrm{sd}} = 0$, however, then $\rho(l)_{\mathrm{sd}}$ is not relevant so we have written a "?" in those places. To compute the link load $\alpha(l)$, we have to multiply all τ_{sd}s with $\rho(l)_{\mathrm{sd}}$s and then add them all. Therefore,

$$\alpha(l) = \frac{5}{8}.$$

Since the link with the highest load in a network saturates at the lowest injection rate, we also define the maximum load $\alpha_{\max}(T, R)$ as follows.

Definition 2.13. *Let $G = (V, E)$ be a graph, T a traffic pattern, and R a routing function. The maximum load $\alpha_{\max}(T, R)$:*

$$\alpha_{\max}(T, R) = \max_{l \in E} \alpha(l).$$

TABLE 3
$\tau_{\rm sd}/\rho(l)_{\rm sd}$ in Example 2.12 for the Relevant Source-Destination Pairs S-D

$\mathbf{d}\backslash\mathbf{s}$	$(0,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$(1,0)$	$(1,1)$	$(1,2)$	$(1,3)$
$(2,3)$	$0/?$	$0/?$	$0/?$	$0/?$	$\frac{1}{8}/\frac{1}{4}$	$0/?$	$\frac{1}{8}/1$	$\frac{1}{8}/1$
$(3,3)$	$\frac{1}{8}/\frac{3}{4}$	$0/?$	$\frac{1}{8}/1$	$\frac{1}{8}/1$	$0/?$	$0/?$	$0/?$	$0/?$

The *input rate* λ is defined as the average number of packets each node injects into the network per time unit. The following lemma states at which input rate a network saturates: the *saturation point*, $\lambda_{\rm sat}$.

Lemma 2.14. *Let $G = (V, E)$ be a graph, c the link capacity, T a traffic pattern, and R a routing function. Then,*

$$\lambda_{\rm sat}(T, R) = \frac{c}{\alpha_{\max}(T, R)}.$$

A high saturation point is an important goal for the designer of routing functions. Therefore, we will call a routing function optimal for a particular traffic pattern if it has the highest possible saturation point, i.e., the lowest possible maximum load.

Definition 2.15. *Let $G = (V, E)$ be a graph and T a traffic pattern. A routing function R is called optimal for T if for all routing functions R' it holds that $\alpha_{\max}(T, R) \le \alpha_{\max}(T, R')$.*

3 Regularities

In this section, we model regularities and study their influence on the performance of routing functions. First, we introduce the concept of automorphism to model regularities in graphs. Then we extend the concept to include traffic patterns and routing functions. We show that if the same regularity occurs for graph, traffic pattern, and routing function, then the load exhibits this regularity, too. Also, we prove that, given a graph and a traffic pattern exhibiting a certain regularity, there also is an optimal routing function exhibiting that regularity.

Then we introduce an extreme form of regularity, namely homogeneity. We show that tori are homogeneous. Next, we want to do a similar thing for meshes. However, the automorphism concept is too strict to meaningfully define homogeneous routing functions for meshes. Therefore, we introduce the concept of a "partial automorphism."

3.1 Automorphisms

To be able to consider regularities of graphs, traffic patterns, and routing algorithms, we use maps under which graphs, traffic patterns, and routing algorithms are invariant: automorphisms. This concept is often used in graph theory.

Let $G = (V, E)$ be a graph. Our basic map is a function $\varphi : V \to V$. We extend the domain of φ to links, routes, and packets, in a straightforward way. If $l = \langle \mathbf{x}, \mathbf{y} \rangle$ is a link, then $\varphi(l) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle$; if $r = [\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_k}]$ is a route, then $\varphi(r) = [\varphi(\mathbf{x}_0), \varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_k)]$; and if $P = (\mathbf{s}, \mathbf{d}, k)$ is a packet, then $\varphi(P) = (\varphi(\mathbf{s}), \varphi(\mathbf{d}), k)$.

Such a map φ is called an *automorphism for graph G* if it is bijective and for all $l \in E$ holds that $\varphi(l) \in E$. An example of an automorphism is a mirror operation.

Definition 3.1. *Let $\mathbf{x} = (x_1, x_2)$ be a node such that $0 \le x_1 \le N-1$ and $0 \le x_2 \le N-1$, for some $N \ge 2$. The mirror operation in dimension 1 is denoted by $_{[1]}$, and defined as:*

$$(x_1, x_2)_{[1]} = (N - 1 - x_1, x_2).$$

The mirror operation in dimension 2 is defined analogously.

Example 3.2. Let $G = (V, E)$ be an N-mesh. We show that $_{[1]}$ is an automorphism for G. Consider a vertical link in the negative direction $l = \langle (x_1, x_2), (x_1 - 1, x_2) \rangle$ (Fig. 6). We have

$$l_{[1]} = \langle (N - 1 - x_1, x_2), (N - x_1, x_2) \rangle,$$

which is a vertical link in the positive direction. Consider a horizontal link in the positive direction $l = \langle (x_1, x_2), (x_1, x_2 + 1) \rangle$. We have

$$l_{[1]} = \langle (N - 1 - x_1, x_2), (N - 1 - x_1, x_2 + 1) \rangle,$$

which is also a horizontal link in the positive direction. The other links can be shown to comply analogously. Also, in a smimilar manner it can be shown that $_{[2]}$ is an automorphism for G.

The same kind of regularities can be defined for traffic patterns and routing functions. Instead of respecting the existence of links in graphs, these regularities should respect the values in the traffic pattern and in the routing function, respectively. So, if T is a traffic pattern, then φ is called an *automorphism for traffic pattern T* if for all $\mathbf{s}, \mathbf{d} \in V$ holds

$$\tau_{\rm sd} = \tau_{\varphi({\rm s})\varphi({\rm d})}. \tag{1}$$

Example 3.3. Consider the traffic pattern $T_{\rm loc}$ and the mirror operation $_{[1]}$. We have that the distance in dimension one

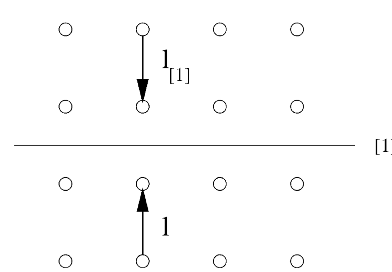

Fig. 6. A mirror operation: $l = \langle (3,1), (2,1) \rangle$ and $l_{[1]} = \langle (0,1), (1,1) \rangle$.

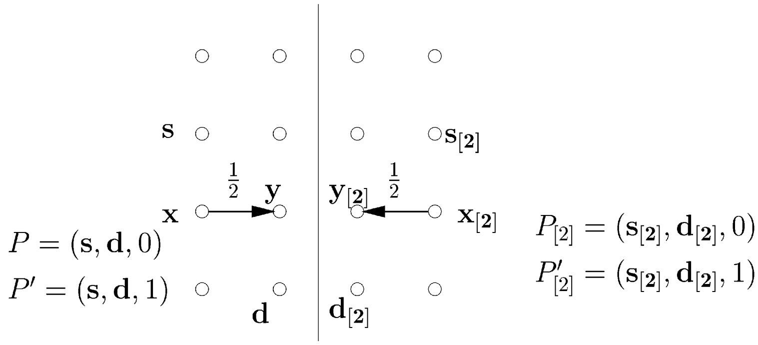

Fig 7. Routing function automorphism: on the left, P has a probability $\frac{1}{2}$ to be changed into P' and routed along $\langle \mathbf{x}, \mathbf{y} \rangle$; on the right, $P_{[2]}$ has a probability $\frac{1}{2}$ to be changed into $P'_{[2]}$ and routed along $\langle \mathbf{x}_{[2]}, \mathbf{y}_{[2]} \rangle$.

between $\mathbf{s}$ and $\mathbf{d}$ is equal to the distance in dimension one between $\mathbf{s}_{[1]}$ and $\mathbf{d}_{[1]}$:

$$\begin{aligned} \mathrm{dis}_{1,\mathrm{s}_{[1]}\mathrm{d}_{[1]}} &= \min(|N-1-s_1-(N-1-d_1)|, \\ &\qquad N-|N-1-s_1-(N-1-d_1)|) \\ &= \min(|d_1-s_1|, N-|d_1-s_1|) = \mathrm{dis}_{1,\mathrm{sd}}. \end{aligned}$$

The same holds for the distance in dimension two. Therefore, if $\tau_{\mathrm{sd}} = \frac{1}{8}$, then both dimension distances are, at most, one and $\mathbf{s} \neq \mathbf{d}$. Thus, both dimension distances between $\mathbf{s}_{[1]}$ and $\mathbf{d}_{[1]}$ are at most one and $\mathbf{s}_{[1]} \neq \mathbf{d}_{[1]}$, so $\tau_{\mathrm{s}_{[1]}\mathrm{d}_{[1]}} = \frac{1}{8}$. Otherwise, both values are zero. We conclude that $_{[1]}$ is an automorphism for the traffic pattern T_{loc}. Analogously, we can show that $_{[2]}$ is an automorphism for the traffic pattern T_{loc}. □

We also extend the concept of automorphism to routing functions. Now the function φ should respect the probabilities of the routing function. Let $G = (V, E)$ be a graph. We get that φ is called an *automorphism for routing function* R if φ is an automorphism for graph G and for all $\mathbf{x}, \mathbf{y} \in V$ and $P, P' \in \mathbb{P}$ holds:

$$R(\mathbf{x}, P)(\mathbf{y}, P') = R(\varphi(\mathbf{x}), \varphi(P))(\varphi(\mathbf{y}), \varphi(P')). \qquad (2)$$

Example 3.4. Consider the routing function R_{ex} and the mirror operation $_{[2]}$. Earlier (Example 3.2) it has been shown that $_{[2]}$ is an automorphism for N-meshes. So, to show that $_{[2]}$ is an automorphism for R_{ex}, it suffices to proof that R_{ex} satisfies (2) for all nodes and packets where $\varphi =_{[2]}$. Consider, for example, a packet $P = (\mathbf{s}, \mathbf{d}, 0)$ and a node $\mathbf{x}$, such that $x_1 < d_1$ and $x_2 < d_2$, as shown in Fig. 7 (Case 3 of Example 2.6). We have $P_{[2]} = (\mathbf{s}_{[2]}, \mathbf{d}_{[2]}, 0)$. We have that $\mathbf{x}_{[2]} = (x_1, N-1-x_2)$ and $\mathbf{d}_{[2]} = (d_1, N-1-d_2)$, so the mirrored packet $P_{[2]}$ must be routed the the left and downwards from $\mathbf{x}_{[2]}$. Now let $P' = (\mathbf{s}, \mathbf{d}, 1)$, so $P'_{[1]} = (\mathbf{s}_{[2]}, \mathbf{d}_{[2]}, 1)$. We have

$$R_{\mathrm{ex}}(\mathbf{x}, P)((x_1, x_2+1), P') = \frac{1}{2}$$

and we also have

$$R_{\mathrm{ex}}(\mathbf{x}_{[2]}, P_{[2]})((N-x_1-1, N-x_2), P'_{[2]}) = \frac{1}{2}.$$

Therefore, Condition (2) is satisfied in this case. If we check all the other cases, too, we find that $_{[2]}$ is indeed an automorphism for the routing function R_{ex}. A similar thing can be shown for $_{[1]}$.

The set of all automorphisms of a graph G is called the *automorphism group of graph* G, denoted by $Aut(G)$. Analogously, we define the automorphism groups of a traffic pattern T and a routing function R; these are denoted by $\mathrm{Aut(T)}$ and $\mathrm{Aut(R)}$, respectively.

We can now derive two highly intuitive results. First, we show that if a traffic pattern and a routing function exhibit the same regularity (i.e., automorphism), then the link load α exhibits that regularity, too. Second, we show that if a graph G and a traffic pattern T exhibit the same regularity, then there exists an optimal routing function exhibiting that regularity, too.

Theorem 3.5. *Let $G = (V, E)$ be a directed graph, T a traffic pattern, and R a routing function. If $\varphi \in \mathrm{Aut(T)} \cap \mathrm{Aut(R)}$, then for all $l \in E$ holds*

$$\alpha(\varphi(l)) = \alpha(l).$$

Proof. Suppose $\varphi \in \mathrm{Aut(T)} \cap \mathrm{Aut(R)}$ for some function φ. Since φ is an automorphism for R, we have for all $\mathbf{s}, \mathbf{d} \in V$ and $r \in \mathtt{routes}_{G,\mathbf{sd}}$ that $\rho(r)_{\mathrm{sd}} = \rho(\varphi(r))_{\varphi(\mathrm{s})\varphi(\mathrm{d})}$. Furthermore, it is clear that the function φ also respects occurs; that is, $\mathrm{occurs}(l, r) = \mathrm{occurs}(\varphi(l), \varphi(r))$. Therefore, we can derive

$$\begin{aligned} \rho(l)_{\mathrm{sd}} &= \sum_{r \in \mathrm{routes}_{G,\mathrm{sd}}} \mathtt{occurs}(l, r) \cdot \rho(r)_{\mathrm{sd}} \\ &= \sum_{r \in \mathrm{routes}_{G,\mathrm{sd}}} \mathtt{occurs}(\varphi(l), \varphi(r)) \cdot \rho(\varphi(r))_{\varphi(\mathrm{s})\varphi(\mathrm{d})} \\ &= \sum_{r \in \mathrm{routes}_{G,\varphi(\mathrm{s})\varphi(\mathrm{d})}} \mathtt{occurs}(\varphi(l), r) \cdot \rho(r)_{\varphi(\mathrm{s})\varphi(\mathrm{d})} \\ &= \rho(\varphi(l))_{\varphi(\mathrm{s})\varphi(\mathrm{d})}. \end{aligned}$$

Now, we can also derive

$$\begin{aligned} \alpha(l) &= \sum_{\mathrm{s,d} \in V} \tau_{\mathrm{sd}} \cdot \rho(l)_{\mathrm{sd}} = \sum_{\mathrm{s,d} \in V} \tau_{\varphi(s)\varphi(d)} \cdot \rho(\varphi(l))_{\varphi(\mathrm{s})\varphi(\mathrm{d})} \\ &= \sum_{\mathrm{s,d} \in V} \tau_{\mathrm{sd}} \cdot \rho(\varphi(l))_{\mathrm{sd}} = \alpha(\varphi(l)). \end{aligned}$$

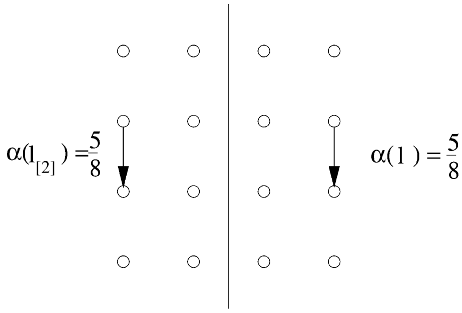

Fig 8. Symmetry of the link load $\alpha(l)$ under traffic pattern T_{loc} and routing function R_{ex}.

Example 3.6. Consider again traffic pattern T_{loc} and routing function R_{ex}. Earlier (Example 2.12) we have computed for $l = \langle(1,3),(2,3)\rangle$ that $\alpha(l) = \frac{5}{8}$. Using the facts that $_{[2]} \in \mathrm{Aut}(\mathrm{T_{loc}}) \cap \mathrm{Aut}(\mathrm{R_{ex}})$, we can apply Theorem 3.5 and derive that $\alpha(l) = \alpha(l_{[2]})$. Therefore, $\alpha(l_{[2]}) = \frac{5}{8}$, where $l_{[2]} = \langle(1,0),(2,0)\rangle$ (see also Fig. 8).

We now show that, when designing routing functions, we can respect given regularities of traffic patterns without sacrificing the possibility of optimality.

Theorem 3.7. *Let $G = (V, E)$ be a graph and T a traffic pattern. Let $\varphi \in \mathrm{Aut(G)} \cap \mathrm{Aut(T)}$. There exists a routing function R which is optimal for T such that $\varphi \in \mathrm{Aut(R)}$.*

Proof. If φ is an automorphism for graph G, it has a finite order, say q. This means that q is the smallest positive integer such that φ^q is the unity function. Now consider an optimal routing function R_{opt} for T. We use φ to construct q routing functions $R_0, \ldots, R_{q-1}$, such that for all $\mathbf{x}, \mathbf{y} \in V$ and $P, P' \in \mathbb{P}$:

$$R_i(\mathbf{x}, P)(\mathbf{y}, P') = R_{opt}(\varphi^{\mathrm{i}}(\mathbf{x}), \varphi^{\mathrm{i}}(P))(\varphi^{\mathrm{i}}(\mathbf{y}), \varphi^{\mathrm{i}}(P)).$$

Note that $R_0 = R_{opt}$. Now, when a packet is injected, one of the routing functions R_i is chosen randomly, so each routing function R_i has a probability of $\frac{1}{q}$ to be chosen. This packet is now routed to its destination by the selected R_i. We can implement this by marking the packet with i, such that all nodes the packet encounters know that R_i should be applied.

We call this new routing strategy R'. We have $\varphi \in \mathrm{Aut(R')}$. This can be seen as follows: R' chooses R_i with probability $\frac{1}{q}$. Also, it chooses $R_{(i+1) \bmod q}$ with the same probability. Thus, (2) is satisfied.

From the new routing strategy R' follows that for all links, l holds that

$$\alpha_{R'}(l) = \frac{1}{q}\alpha_{R_0}(l) + \ldots + \frac{1}{q}\alpha_{R_{q-1}}(l),$$

where α_R denotes the link load under routing function R. Since all link loads for R_i, $0 \leq i \leq q-1$, are smaller than the maximum link load under R_{opt}, we get that

$$\begin{aligned}\alpha_{R'}(l) &\leq \frac{1}{q}\alpha_{\max}(T, R_{opt}) + \ldots + \frac{1}{q}\alpha_{\max}(T, R_{opt}) \\ &= \alpha_{\max}(T, R_{opt}),\end{aligned}$$

which implies that R' is optimal for T. □

3.2 Homogeneity

We introduce the concept of *homogeneity*. In our context, a graph, traffic pattern, or routing function is homogeneous if it has the same structure at every node. Formally, for each pair $\mathbf{x}, \mathbf{y} \in V$ of nodes there must be an automorphism φ such that $\varphi(\mathbf{x}) = y$. To explore the concept of homogeneity, we need a translation operator for nodes.

Definition 3.8. *Let $G = (V, E)$ be an N-mesh or N-torus. Let $\mathbf{x}, \mathbf{a} \in V$ be nodes. The node addition operator $\oplus$ is defined as follows:*

$$\mathbf{x} \oplus \mathbf{a} = ((x_1 + a_1) \bmod N, (x_2 + a_2) \bmod N).$$

Tori are homogeneous graphs. This can be described using the node addition operator; for all $\mathbf{a} \in V$ holds that $(\oplus\, \mathbf{a}) \in \mathtt{Aut(G)}$, where $G = (V, E)$ is an N-torus. Therefore, for all $\mathbf{x}, \mathbf{y} \in V$ there exists an automorphism φ such that $\varphi(\mathbf{x}) = \mathbf{y}$. Such a graph is called *homogeneous*. Now suppose we have a traffic pattern T and a routing function R such that $(\oplus\, \mathbf{a}) \in \mathrm{Aut(T)} \cap \mathrm{Aut(R)}$ for all $\mathbf{a} \in V$. We call T and R *homogeneous*, too. Furthermore, suppose that T and R are also symmetric: $_{[1]}, _{[2]} \in \mathrm{Aut(T)} \cap \mathrm{Aut(R)}$. Then we obtain that all vertical links have the same load and also all horizontal links have the same load. On top of that, if the routing function is also minimal, then it is optimal.

Theorem 3.9. *Let $G = (V, E)$ be an N-torus. Let T be a symmetrical homogeneous traffic pattern and let R be a symmetrical homogeneous routing function. Then for $i = 1, 2$ and for all $l_1, l_2 \in E_i$ holds:*

$$\alpha(l_1) = \alpha(l_2).$$

The proof can be found in [19].

The notion of automorphism with respect to routing functions is a little too strict to be useful on the mesh, because a routing function can only be homogeneous if the graph is. Thus, since meshes are not homogeneous, there are no homogeneous routing functions for meshes. More concretely, consider an N-mesh $G = (V, E)$. For most links $l = \langle\mathbf{x}, \mathbf{y}\rangle \in E$ holds that $l \oplus (0,1) \in E$, except for the links at the right border of the mesh. Therefore, $(\oplus(0,1))$ is not an automorphism for G. However, we would like to use the regularity of meshes.

We thus introduce the concept of a "partial automorphism" for routing functions by dropping the condition that φ should be an automorphism for the graph it acts on and demand only that the routing function respects φ where the graph allows it.

Definition 3.10. *Let R be a routing algorithm for a graph $G = (V, E)$. A bijective graph map $\varphi : G \to G'$ is a partial automorphism for R if for all $\mathbf{s} = \mathbf{x_0}, \mathbf{d} = \mathbf{x_k} \in V$ and all $r = [\mathbf{x_0}, \ldots, \mathbf{x_k}] \in \mathtt{routes}_{G,\mathrm{sd}}$ holds: if $\varphi(r) \in \mathrm{routes}_{G,\varphi(\mathrm{s})\varphi(\mathrm{d})}$, then for all $[P_{0,\ldots,}P_k] \in \mathtt{packets}(r)$ and all $0 \leq i < k$:*

$$R(\mathbf{x_i}, P_i)(\mathbf{x_{i+1}}, P_{i+1}) = R(\varphi(\mathbf{x_i}), \varphi(P_i))(\varphi(\mathbf{x_{i+1}}), \varphi(P_{i+1})).$$

The set of all partial automorphisms for R is called the *partial automorphism set of* R, denoted by $\mathrm{PAut(R)}$. Note that $\mathrm{Aut(R)} \subseteq \mathrm{PAut(R)}$. Note also that $\mathrm{PAut(R)}$ is not

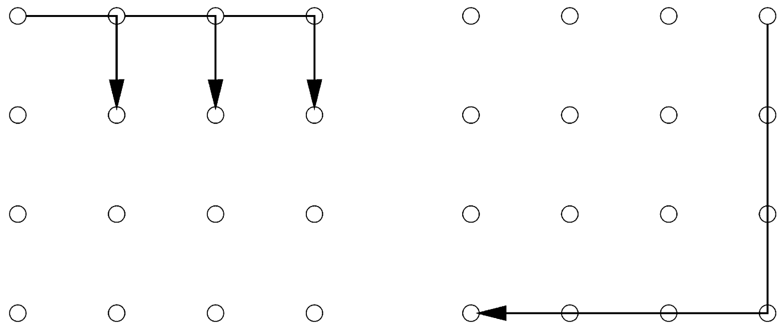

Fig. 9. $\oplus(0,1)$ as partial automorphism for a routing function.

necessarily a group, i.e., it is not necessarily closed with respect to composition.

Example 3.11. Consider a 4-mesh $G=(V,E)$ and the mapping $\varphi=\oplus(0,1)$, which is a translation of one to the right. Now, φ is not an automorphism for G. This can be seen as follows: Consider link $l=\langle(0,2),(0,3)\rangle$. We get $\varphi(l)=\langle(0,3),(0,1)\rangle$, which is a nonexistent link in G.

Next, consider a routing function R, source node $\mathbf{s}=(0,0)$ and destination node $\mathbf{d}=(1,1)$. Suppose R routes all packets from $\mathbf{s}$ to $\mathbf{d}$ using route $r=[(0,0),(0,1),(1,1)]$ (Fig. 9). Suppose $\varphi=\oplus(0,1)$ is a partial automorphism for R. If we apply φ on r once, we see that we again get a route in G. So, to route from $\varphi(\mathbf{s})=(0,1)$ to $\varphi(\mathbf{d})=(2,2)$, R uses $\varphi(r)=[(0,1),(0,2),(1,2)]$. We can do the same for the route from $\varphi^2(\mathbf{s})$ to $\varphi^2(\mathbf{d})$. However, if we apply φ a third time, we see that $\varphi^3(r)$ is not an existing route in G. Therefore, R uses an other route from $(0,3)$ to $(3,0)$. This "exception" causes that φ is not an automorphism for R, but since we use the regularity of the mesh where possible, we call it a partial automorphism for R.

Lemma 3.12. *Let $G=(V,E)$ be a directed graph, T be a traffic pattern, and R a routing algorithm. If $\varphi\in\mathrm{PAut(R)}$, then for all $\mathbf{s},\mathbf{d}\in V$ and $r\in\texttt{routes}_{G,\mathrm{sd}}$: if $\varphi(r)\in\texttt{routes}_{G,\varphi(\mathrm{s})\varphi(\mathrm{d})}$ then*

$$\rho(r)_{\mathrm{sd}}=\rho(\varphi(r))_{\varphi(\mathrm{s})\varphi(\mathrm{d})}.$$

The proof can be found in [19].

Similar to homogeneous routing functions, we call a routing function *partially homogeneous* if for all $\mathbf{a}\in V$ holds that $(\oplus\,\mathbf{a})\in\mathrm{PAut(R)}$.

4 Homogeneous Traffic and Homogeneous Routing on Tori and Meshes

In this section, we present our main results. We define the traffic load in dimension i, which will be denoted by $\Lambda_i(T)$, where T is the traffic pattern. The first result concerns homogeneous traffic patterns and routing functions on tori. We show that these routing functions have maximum load $\frac{1}{2}\Lambda_i(T)$ in each dimension and that this is the optimum. Then we turn to homogeneous traffic patterns on meshes and derive a lower bound on the maximum load, which is twice as bad as the optimum for tori. Next, we show that if we use minimal homogeneous routing functions, the maximum load is at most twice the best-case bound. Finally, we present a class of minimal homogeneous routing functions of which the performance matches the best-case bound, which leads to the conclusion that all those routing functions are optimal.

4.1 Tori

First, we define a measure to characterize traffic patterns, Λ. This measure will be used to express our results. The traffic pattern measure Λ indicates what load the traffic pattern puts in total on the network.

Definition 4.1. *Let $G=(V,E)$ be a directed graph and let T be a traffic pattern for G. The traffic load in dimension i, denoted $\Lambda_i(P)$, is defined as:*

$$\Lambda_i(T)=\sum_{d\in V}\tau_{0\mathrm{d}}\mathrm{dis}_{i,0\mathrm{d}},$$

where $0=(0,\ldots,0)$.

Corollary 4.2. *Let $G=(V,E)$ be an N-torus, T a traffic pattern, and R a minimal routing function, such that $\oplus a\in\mathrm{Aut(T)}\cap\mathrm{Aut(R)}$ for all $\mathbf{a}\in V$ and ${}_{[1],[2]}\in\mathrm{Aut(T)}\cap$ Aut(T). For all $i=1,2$, and $l\in E_i$ holds*

$$\alpha(l)=\frac{1}{2}\Lambda_i(T).$$

Furthermore, R is optimal for T.

Proof. Since R is minimal, we have that

$$\sum_{l\in E_i}\alpha(l)=\sum_{\mathrm{s,d}\in V}\tau_{\mathrm{sd}}\cdot\mathrm{dis}_{i,\mathrm{sd}}=\sum_{\mathrm{s,d}\in V}\tau_{0(\mathrm{d}\ominus\mathrm{s})}\cdot\mathrm{dis}_{i,0(\mathrm{d}\ominus\mathrm{s})}=|V|\cdot\Lambda_i(T).\tag{3}$$

Furthermore, according to Theorem 3.9, all links in E_i have the same load, so we get for an arbitrary link $l'\in E_i$:

$$\sum_{l\in E_i}\alpha(l)=|E_i|\cdot\alpha(l')=2\cdot|V|\cdot\alpha(l').\tag{4}$$

Combining (3) and (4) yields the link loads. To see that R is optimal, reflect that for any routing function holds that

$$\sum_{l\in E_i}\alpha(l)\geq|V|\cdot\Lambda_i(T),$$

so there is always a link in E_i having at least load $\frac{1}{2}\Lambda_i(T)$. □

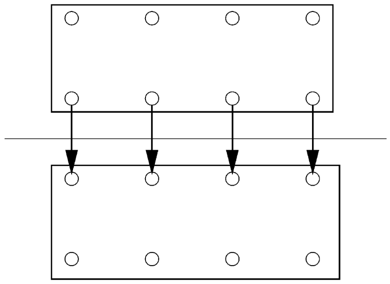

Fig. 10. Bisection in dimension one: The traffic from the upper part to the lower part has to use at least one of the links in the bisection.

Example 4.3. Consider an N-torus and, again, the traffic pattern T_{loc}. Let R be the routing function which routes randomly in any profitable direction (that is, it can only route from $\mathbf{x}$ to $\mathbf{y}$ if $\mathrm{dis}_{i,\mathrm{yd}} < \mathrm{dis}_{i,\mathrm{xd}}$ for $i = 1$ or $i = 2$, where $\mathbf{d}$ is the destination).

To compute $\Lambda_1(T_{\mathrm{loc}})$, consider Fig. 2. There are eight kinds of packets, all having weight $\frac{1}{8}$. Six of those kinds of packets travel a vertical distance of one, two kinds of packets travel a vertical distance of zero. Therefore, we have

$$\Lambda_1(T_{\mathrm{loc}}) = 6 \cdot \frac{1}{8} \cdot 1 + 2 \cdot \frac{1}{8} \cdot 0 = \frac{3}{4}.$$

The same holds for $\Lambda_2(T_{\mathrm{loc}})$.

Applying Corollary 4.2 yields that

$$\alpha_{\max}(T_{\mathrm{loc}}, R) = \frac{1}{2} \cdot \frac{3}{4} = \frac{3}{8}$$

and that R is optimal for T_{loc}.

4.2 Best Case Behavior on Meshes

We derive a lower bound on the maximum load for homogeneous traffic patterns using the bisection bound. Later it will turn out that this bound is sharp since there are routing functions achieving the bound.

Theorem 4.4. *Let $G = (V, E)$ be an N-mesh. Let T be a traffic pattern such that ${}_{[1],[2]} \in \mathrm{Aut(T)}$ and $(\oplus\mathbf{a}) \in \mathrm{Aut(T)}$ for all $\mathbf{a} \in V$. Let R be a routing function. For $i = 1, 2$, there is an $l \in E_i$ such that*

$$\alpha(l) \geq \Lambda_i(T).$$

Proof. Assume that $i = 1$; the case $i = 2$ is analogous. Consider the "bisection" containing links of E_1: These are the links $\langle \mathbf{x}, \mathbf{y} \rangle$ such that $x_1 = \lfloor \frac{N}{2} \rfloor - 1$ and $y_1 = \lfloor \frac{N}{2} \rfloor$ (see Fig. 10). The amount of traffic which has to pass this bisection can be computed as follows:

$$\sum_{s_1=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{s_2=0}^{N-1} \sum_{d_1=\lfloor \frac{N}{2} \rfloor}^{N-1} \sum_{d_2=N-1} \tau_{\mathrm{sd}} = \sum_{s_1=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{s_2=0}^{N-1} \sum_{d_1=\lfloor \frac{N}{2} \rfloor}^{N-1} \sum_{d_2=N-1} \tau_{0,\mathrm{d}\ominus\mathrm{s}}$$
$$= \sum_{s_1=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{s_2=0}^{N-1} \sum_{\Delta_1=\lfloor \frac{N}{2} \rfloor - s_1}^{N-1-s_1} \sum_{\Delta_2=0}^{N-1} \tau_{0\Delta}.$$

In [19] this is shown to be equal to $N\Lambda_1(T)$.

Since the "bisection" contains N links, we obtain that there is at least one link l such that:

$$\alpha(l) \geq \frac{N\Lambda_1(T)}{N} = \Lambda_1(T). \qquad \square$$

4.3 Worst Case Behavior on Meshes

In this section, we derive an upper bound for the maximum load of partial node homogeneous routing functions on meshes, for the class of node homogeneous traffic patterns. This bound is twice as large as the above derived best case lower bound . It can even be slightly improved by seperately considering the traffic which travels only in one dimension (see [19]), but for the sake of clarity, we have left out that part.

To bound the maximum load, we first derive bounds for every choice of Δ. For such a Δ, we divide the mesh into columns of sources (Fig. 11). Then each column is divided into two parts: the sources from which we have to route down (picture on the left) and the sources from which we have to route up (picture on the right). Then we show that each link is used by at most one route of both parts. This is expressed by the following lemma.

Lemma 4.5. *Let $G = (V, E)$ be an N-mesh. Let R be a minimal routing function such that $(\oplus\mathbf{a}) \in \mathrm{PAut(R)}$ for all $\mathbf{a} \in V$. Let $0 \leq x_2, s_2, \Delta_2 \leq N - 1$ be such that $s_2 \leq x_2 < s_2 + \Delta_2 \leq N - 1$. Let $l = \langle \mathbf{x}, \mathbf{y} \rangle \in E_{2+}$. Let $0 \leq \Delta_1 \leq N - 1$. Then*

1. $$\sum_{\mathrm{s}_1=0}^{N-1-\Delta_1} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} \leq 1,$$

2. $$\sum_{\mathrm{s}_1=N-\Delta_1}^{N-1} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} \leq 1.$$

A proof can be found in [19]. We illustrate the lemma by means of an example.

Example 4.6. Consider a 6-mesh (Fig. 11). Let $s_2 = 1$ and $\Delta = (2, 4)$. Consider all links $l = \langle \mathbf{x}, \mathbf{y} \rangle \in E_{2+}$ such that $x_2 = 2$ (as shown by the dashed line).

1. Consider $0 \leq s_1 \leq N - 1 - \Delta_1 = 3$. Now, whatever route we take from $\mathbf{s}$ to $\mathbf{s} \oplus \Delta$, the regularity of the routing function causes that each of the considered links is used by at most one of the considered source-destination pairs.
2. Consider $N - \Delta_1 = 4 \leq s_1 \leq N - 1 = 5$. Again, whatever route we take from $\mathbf{s}$ to $\mathbf{s} \oplus \Delta$, each of the links under consideration is used by at most one of these source-destination pairs.

Next, we add the bounds for all columns, where we account for the fact that a link $l = \langle \mathbf{x}, \mathbf{y} \rangle \in E_{2+}$ can only be used by a route from s to $\mathbf{s} \oplus \Delta$ if $s_2 \leq x_2 < (s_2 + \Delta_2) \mathrm{mod} N$.

Lemma 4.7. *Let $G = (V, E)$ be an N-mesh. Let R be a minimal routing function such that $(\oplus a) \in \mathrm{PAut(R)}$ for all $\mathbf{a} \in V$. For $i = 1, 2$, all $l \in E_i$, and $\Delta \in V$ holds*

$$\sum_{s \in V} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} \leq 2 \cdot \min(\Delta_i, N - \Delta_i).$$

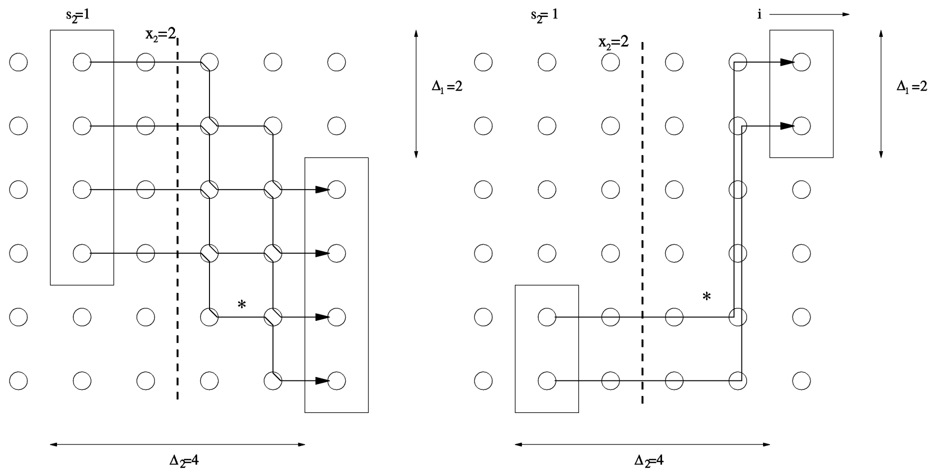

Fig. 11. Illustration of division of load for partially node homogenous routing functions; each link crossed by the dashed line is used by at most one of the depicted source-destination pairs.

Proof. Assume that $l = \langle \mathbf{x}, \mathbf{y} \rangle \in E_{2+}$. The proofs for the other links are analogous. We get

$$\sum_{\mathrm{s} \in V} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} = \sum_{s_1=0}^{N-1} \sum_{s_2=\max(0,x_2-\Delta_2+1)}^{\min(x_2,N-\Delta_2-1)} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta}.$$

Appplying Lemma 4.5 yields that this is, at most,

$$2 \cdot (\max(0, x_2 - \Delta_2 + 1) - \min(x_2, N - \Delta_2 - 1)).$$

In [19] it is proven, by an exhaustive case analysis, that this term is, at most,

$$2 \cdot \min(\Delta_2, N - \Delta_2). \qquad \square$$

The last step is to combine the result above with the homogeneity of the traffic pattern. This results in an upper bound for the maximum load which is twice as large as the lower bound we presented in the preceding section.

Theorem 4.8. *Let $G = (V, E)$ be an N-mesh. Let T be a traffic pattern and R a minimal routing function such that $_{[1]},_{[2]} \in \mathrm{Aut(T)}$ and $\oplus\mathbf{a} \in \mathrm{Aut(T)} \cap \mathrm{PAut(R)}$ for all $\mathbf{a} \in V$. For all $l = \langle \mathbf{x}, \mathbf{y} \rangle \in E_i$ holds, for $i = 1, 2$,*

$$\alpha(l) \leq 2\Lambda_i(T).$$

Proof. Applying Definition 2.11 and the homogeneity of T yields

$$\alpha(l) = \sum_{\mathrm{s},\mathrm{d} \in V} \tau_{\mathrm{sd}} \cdot \rho(l)_{\mathrm{sd}} = \sum_{\mathrm{s},\Delta \in V} \tau_{\mathrm{s},\mathrm{s}\oplus\Delta} \cdot \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta}$$
$$= \sum_{\Delta \in V} \tau_{0\Delta} \sum_{s \in V} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta}.$$

When we apply Lemma 4.7 we get that this is, at most,

$$2 \sum_{\Delta \in V} \tau_{0\Delta} \min(\Delta_i, N - \Delta_i) = 2\Lambda_i(T). \qquad \square$$

4.4 A Class of Optimal E-Cube-Like Routing Functions

In this section, we present a class of e-cube-like routing functions for meshes which are optimal (matching the already presented lower bound) for all traffic patterns which are both homogeneous and symmetrical in both dimensions. These routing functions are partially node homogeneous. The class contains the popular e-cube routing function. All routing functions in the class produce only horizontal-first and vertical-first routes.

To obtain a maximum load equal to the best bound (Theorem 4.4), we try to find partial node homogeneous routing functions such that the sum of 1. and 2. of Lemma 4.5 is at most one, so for all $l \in \langle \mathbf{x}, \mathbf{y} \rangle$, $0 \leq s_2 \leq x_2 < s_2 + \Delta_2 \leq N - 1$, $0 \leq \Delta_1 \leq N - 1$:

$$\sum_{s_1=0}^{N-1-\Delta_1} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} + \sum_{s_1=N-\Delta_1}^{N-1} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} = \sum_{s_1=0}^{N-1} \rho(l)_{\mathrm{s},\mathrm{s}\oplus\Delta} \leq 1.$$

We will achieve this by using only "vertical-first" and "horizontal-first" routes. To see why we need those routes, consider Fig. 11 again. In the left part of the figure, a kind of "zigzag"-routes are used. This causes that link $\langle (4,3), (4,4) \rangle$, denoted by "*", is used in both the left part and the right part of the figure. Now, we can avoid using any horizontal link 2 times by choosing other routes in the right part, but then the routing function would not be partially node homogeneous anymore. In fact, it is impossible to find two "congruent" routes such that none of the horizontal links is used twice. However, this becomes possible if we use "vertical-first" or "horizontal-first" routes in the left part of Fig. 11. This yields the picture in Fig. 12.

Definition 4.9. *The class* `ec_routes` *is defined as the of all routing functions satisfying the following conditions:*

1. *All routes are either "vertical-first" routes or "horizontal-first" routes. For a given source-destination pair $\mathbf{s}, \mathbf{d}$ $r_{1,\mathbf{sd}}$ will denote the vertical-first route and $r_{2,\mathbf{sd}}$ will denote the horizontal-first route (see Fig. 13).*
2. *For all $\Delta, \mathbf{s}, \mathbf{s}' \in V$:*

$$\rho(r_{1,\mathrm{s},\mathrm{s}\oplus\Delta})_{\mathrm{s},\mathrm{s}\oplus\Delta} = \rho(r_{1,\mathrm{s}',\mathrm{s}'\oplus\Delta})_{\mathrm{s}',\mathrm{s}'\oplus\Delta}.$$

Note that 1. and 2. together imply that 2. also holds for $r_{2,\mathrm{s},\mathrm{s}\oplus\Delta}$. Also note that all routing functions in `ec_routes` are partially homogeneous.

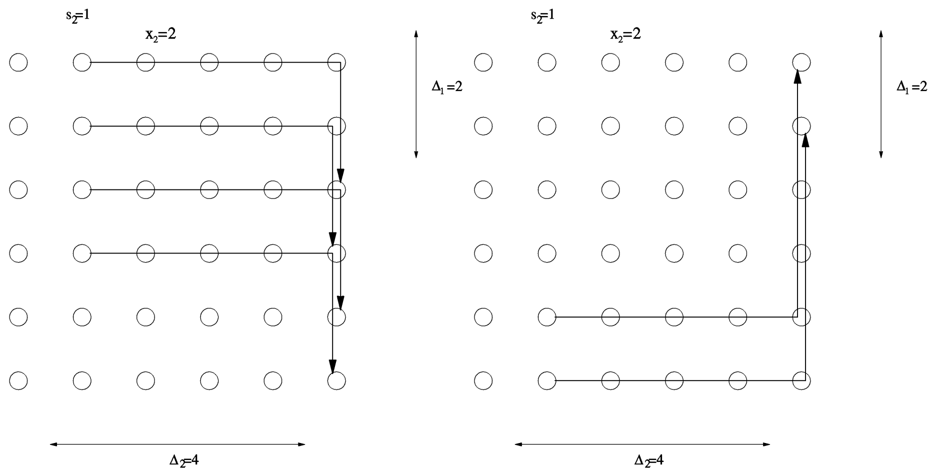

Fig 12. Routes which, at most, use each horizontal link once.

This definition actually means that for each "distance" Δ between source and destination, there is a fraction of the packets routed vertical-first and a fraction routed horizontal-first. For all source-destination pairs having "difference" vector Δ, these fractions should be equal. For different Δ, however, the fractions can be different.

Theorem 4.10. *Let $G = (V, E)$ be an N-mesh. Let T be a traffic pattern such that* $_{[1],[2]} \in \mathrm{Aut(T)}$ *and* $\oplus\mathbf{a} \in \mathrm{Aut(T)}$ *for all* $\mathbf{a} \in V$. *Let* $R \in$ `ec_routes`, $l \in E_i$

$$\alpha_{TR}(l) \leq \Lambda_i(T),$$

for all $l = \langle \mathbf{x}, \mathbf{y} \rangle \in E_i$, *for* $i = 1, 2$. *So R is optimal.*

Proof. We give a sketch of the proof here; the complete proof can be found in [19]. Let $q(\Delta)$ be the fraction of the packets from s to s $\oplus\, \Delta$ that is routed horizontal-first, for all $s \in V$. As can be seen from Fig. 12, for each column of source nodes holds that all horizontal links are used at most once. The same holds for the $1 - q(\Delta)$ part of the packets routed vertical-first. Therefore, we have for all $l \in E_2$ and $0 \leq s_2 \leq N - 1$ that

$$\sum_{s_1=0}^{N-1} \rho(l)_{\mathbf{s},\mathbf{s}\oplus\Delta} \leq q(\Delta) + 1 - q(\Delta) = 1.$$

Adding this for all s_2 yields for all $l \in E_2$:

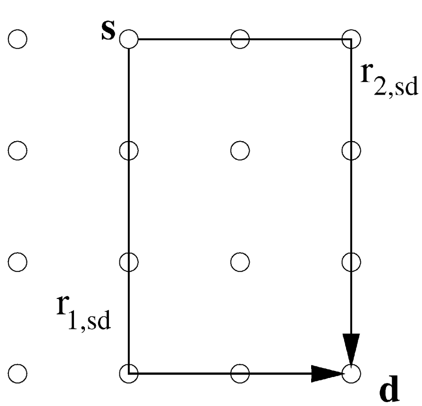

Fig 13. Vertical-first route $r_{1,\mathrm{s},\mathrm{d}}$and horizontal-first route $r_{2,\mathrm{s},\mathrm{d}}$ for source s and destination d.

$$\sum_{\mathrm{s} \in V} \rho(l)_{\mathrm{s},\oplus\boldsymbol{\Delta}} \leq \min(\Delta_2, N - \Delta_2).$$

Adding for all Δ and accounting for the traffic pattern yields for all $l \in E_2$:

$$\alpha(l) \leq \Lambda_2(T).$$

For $l \in E_1$, the result can be derived analogously.

This, in combination with Theorem 4.4, leads to the conclusion that R is optimal. □

5 Conclusions

We believe that our results are relevant in practice, in the first place because our saturation point analysis provides the designer of routing algorithms with an upper bound on the practical performance. Also, the computed link loads represent the actual link loads in practice.

It would still be nice to see how accurate the saturation point analysis is in practical situations, where the number of buffers is finite so additional parameters like injection distribution and switching technique may play a role. However, a thorough comparison to simulation results published before cannot be performed, for the following reasons: In the first place, very few papers consider the same combinations of graph, traffic pattern, and routing function. Second, the performance of routing functions is usually presented in throughput/latency plots. The problem with these plots is that both axes represent *measured*, hence *dependent*, variables. It is not possible to get an accurate estimate of the saturation point from such plots, because the measured throughput may be higher than the saturation point. This is due to the following phenomenon: While some parts of the network are already saturated, other parts may still be quite low on traffic so an increased injection rate can increase the load in those parts of the network. This increases the average throughput, but not the maximum injection rate at which saturation occurs. In the third place, the throughput is often normalized while it is not always clear with respect to what value (see e.g., [9]). Therefore, unnormalized injection rate/throughput and injection rate/latency plots would be far more suited for comparisons.

The only case in which we are actually able to compare our results to earlier published results is in the case of e-cube routing under uniform traffic on meshes ([8]) and tori ([3]). In both cases, about 70-80 percent of our computed saturation point is achieved. Since the mentioned papers consider wormhole routing, it can be expected that the number of available buffers is low, which probably causes the fact that the networks saturate at 70-80 percent of the saturation points presented by us.

We have presented results showing how minimal partially node homogeneous routing algorithms perform on a two-dimensional mesh for node homogeneous traffic patterns. Many traffic patterns found in literature belong to this class: For example, the uniform traffic pattern, the local traffic pattern, (see [7]) and a nameless traffic pattern using the Hamming distance in the hypercube (see [18]). The minimal partial node homogeneous routing algorithms always have a load between the optimum and the optimum plus a term which depends on the traffic pattern. We have also presented a class of routing algorithms which is not just optimal for one traffic pattern, but for a whole class of traffic patterns.

Of course, there is much work left to be done. The assumption of infinite buffer capacity could be dropped to obtain more accurate results than the upper bounds presented in this paper. We could also consider other performance measures. Also, our theory can be extended to other classes of traffic patterns, routing algorithms, and graphs. We think that the theory can be extended to higher-dimensional meshes and tori, because most of the individual steps taken in this paper either apply, or can be extended to apply, to an arbitrary number of dimensions. The framework presented in this paper provides a systematic method for further research.

ACKNOWLEDGMENTS

The author would like to thank H. Sips, R. Sommerhalder, and the referees for their constructive comments, which have led to a much clearer structure and better presentation of this paper.

REFERENCES

[1] K. Bolding and L. Snyder, "Mesh and Torus Chaotic Routing," *Proc. Advanced Research in VLSI and Parallel Systems Conference,* pp. 333–347, Mar. 1992.

[2] K. Bolding and W. Yost, "Design of a Router for Fault-Tolerant Networks," *Proc. Parallel Computer Routing and Comm.,* vol. 853, pp. 226–240, May 1994.

[3] R.V. Boppana and S. Chalasani, "A Comparison of Adaptive Wormhole Routing Algorithms," *Proc. Int'l Symp. Computer Architecture,* pp. 351–360, May 1993.

[4] Y.M. Boura and C.R. Das, "A Class of Partially Adaptive Routing Algorithms for n_Dimensional Meshes," *Int'l Conf. Parallel Processing,* vol. 3, pp. 175–182, 1993.

[5] W. Chou, A.W. Bragg, and A.A. Nilsson, "The Need for Adaptive Routing in the Chaotic and Unbalanced Traffic Environment," *IEEE Trans. Communications,* vol. 29, no. 4, pp. 481–490, Apr. 1981.

[6] J. Duato, "Deadlock-Free Adaptive Routing Algorithms for Multicomputers: Evaluation of a New Algorithm," *Proc. Third IEEE Symp. Parallel and Distributed Processing,* pp. 840–847, 1991.

[7] J. Duato and P. López, "Performance Evaluation of Adaptive Routing Algorithms for k-Ary n-Cubes," *Proc. Parallel Computer Routing and Comm.,* vol. 853, pp. 45–59, May 1994.

[8] S.A. Felperin, L. Gravano, G.D. Pifarré, and J.L.C. Sanz, "Fully-Adaptive Routing: Packet Switching Performance and Wormhole Algorithms," *Proc. Supercomputing '91,* Nov. 1991.

[9] P.T. Gaughan and S. Yalamanchili, "Adaptive Routing Protocols for Hypercube Interconnection Protocols," *Computer,* vol. 26, no. 5, pp. 12–23, May 1993.

[10] C.J. Glass and L.M. Ni, "Adaptive Routing in Mesh-Connected Networks," *Proc. 12th Int'l Conf. Distributed Computing Systems,* pp. 12–19, June 1992.

[11] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," *Proc. 19th Int'l Symp. Computer Architecture,* pp. 278–287, 1992.

[12] J.Y. Ngai and C.L. Seitz, "A Framework for Adaptive Routing in Multicomputer Networks," *Proc. ACM Symp. Parallel Algorithms and Architectures,* pp. 1–9, 1989.

[13] T.D. Nguyen and L. Snyder, "Performance Analysis of a Minimal Adaptive Router," *Proc. Parallel Computer Routing and Comm.,* vol. 853, pp. 31–44, May 1994.

[14] G.D. Pifarré, L. Gravano, S.A. Felperin, and J.L.C. Sanz, "Fully Adaptive Minimal Deadlock-Free Packet Routing in Hypercubes, Meshes, and Other Networks: Algorithms and Simulations," *IEEE Trans. Parallel and Distributed Systems,* vol. 5, no. 3, pp. 247–263, Mar. 1994.

[15] G.D. Pifarré, L. Gravano, G. Denicolay, and J.L.C. Sanz, "Adaptive Deadlock- and Livelock-Free Routing in the Hypercube Network," *Trans. Parallel and Distributed Systems,* vol. 5, no. 11, pp. 1,121–1,139, Nov. 1994.

[16] S.D. Young and S. Yalamanchili, "Adaptive Routing in Generalized Hypercube Architectures," *Proc. Third IEEE Symp. Parallel and Distributed Processing,* pp. 564–571, 1991.

[17] M. Mitzenmacher, "Bounds on the Greedy Routing Algorithm for Array Networks," *J. Computer and Systems Sciences,* vol. 53, pp. 317–327, 1996.

[18] G.D. Stamoulis and J.N. Tsitsiklis, "The Efficiency of Greedy Routing in Hypercubes and Butterflies," *Proc. Second Ann. ACM Symp. Parallel Algorithms and Architectures,* pp. 248–259, 1991.

[19] L.D. Aronson, "A Theory of Routing Functions from Concept to Analysis," doctoral thesis, Delft University of Technology, 1999.

[20] L.D. Aronson, "Saturation Point Analysis for Nonadaptive Routing Algorithms," Technical Report 96-143, Faculty of Math. and Informatics, Delft Univ. of Technology, 1996. http://www.pds.twi.tudelft.nl/pubs/tech_rep/1996/htm.

Leon Aronson graduated in 1993 from the Delft University of Technology. In the years following, he has performed research in the field of routing in parallel computers. In 1999, he received his PhD for his thesis about routing, also at Delft.