Journal of Algorithms, Vol 31(1), 1999

Shortest Path Routing in Arbitrary Networks*

Friedhelm Meyer auf der Heide and Berthold Vöcking†
Department of Mathematics and Computer Science
and Heinz Nixdorf Institute, University of Paderborn
33095 Paderborn, Germany

Abstract

We introduce an on-line protocol which routes any set of N packets along shortest paths with congestion C and dilation D through an arbitrary network in $O(C+D+\log N)$ steps, with high probability. This time bound is optimal up to the additive $\log N$, and it has previously only been reached for bounded-degree leveled networks.

Further, we show that the above bound holds also for random routing problems with C denoting the maximum expected congestion over all links. Based on this result, we give applications for random routing in Cayley networks, general node symmetric networks, edge symmetric networks, and de Bruijn networks.

Finally, we examine the problems arising when our approach is applied to routing along non-shortest paths, deterministic routing, or routing with bounded buffers.

1 Introduction

Communication among the processors of a parallel computer usually requires a large portion of runtime of a parallel algorithm. These computers are often realized as relatively sparse networks of a large number of processors such that each processor can directly communicate with a few neighbors only. Thus, most of the communication must proceed through intermediate processors. One of the basic problems in this context is to route simultaneously many message packets through the network. Whereas most previous theoretical research on packet routing concentrates on special classes of networks as, e.g., leveled networks, we are interested in *universal* routing algorithms that can be used in any network.

Assume that we are given an arbitrary processor network. A *packet routing problem of size N* on this network is defined by a set of N packets each of which has a *source* and a *destination* node. The goal is to route each packet from its source to its destination. A routing problem in which every node is the source of h packets and the destination of h packets is called an *h-to-h-routing problem*, and a routing problem in which every node sends h packets to random destinations chosen independently and uniformly from the set of nodes is called a *random h-routing problem*.

*A preliminary version was presented at the 12th STACS, 1995, see [9].

†email: {fmadh,voecking}@uni-paderborn.de. Supported in part by DFG-Sonderforschungsbereich 376 "Massive Parallelität: Algorithmen, Entwurfsmethoden, Anwendungen", by EU ESPRIT Long Term Research Project 20244 (ALCOM-IT), and by DFG Leibniz Grant Me872/6-1.

Our investigations are based on the *store-and-forward* model. In this model, the packets are viewed as atomic objects, and it is assumed that the routing proceeds in synchronized steps such that a packet can cross at most one link in a step. In the *multi-port* model, each link can forward at most one packet a step, whereas in the *single-port* model, each processor can forward at most one packet a step. We assume the multi-port model, since it has become most common for store-and-forward routing in recent years. But note that all our techniques and results can be easily adapted to the single port model.

At the beginning of the first step, each packet is stored in an *initial buffer* at its source node. During the routing, it moves forward step by step, and at each link on its path, it is stored in a *link buffer* at the end of the link until it is allowed to move forward along the next link. Upon traversing the last link on its path, the packet is removed from the link buffer and placed in a *final buffer* at its destination. In the following, any bound on the *buffer size* required by a routing protocol refers only to the link buffers, because the size of the initial and final buffers are determined by the particular routing problem. The path traversed by a packet from its source to its destination is called the *routing path* of the packet.

A *routing protocol* describes the rules for moving the packets to their destinations. We aim to construct routing protocols that minimize the total number of steps required to deliver all packets. We break this problem into two parts: the problem of selecting the routing paths and the problem of scheduling the movements of the packets along these paths.

The *path selection problem* is defined as follows. We are given the sources and the destinations of the packets, and we have to determine the routing paths. This can be done by a *path system* $\mathcal{W}$ which is a set of paths through the network. It includes a path $w(u, v) = (u \rightarrow \cdots \rightarrow v)$ for every pair u and v of nodes. If all paths in $\mathcal{W}$ are shortest paths, then we call $\mathcal{W}$ a *shortest path system*. For every packet with source u and destination v, we choose the path $w(u, v)$ as its routing path. Instead of determining the routing paths by a path system beforehand, the routing paths can be selected during the routing, i.e., each node chooses the link for transmitting a selected packet just before the packet is passed on. For instance, the next link on a packet's routing path can be chosen uniformly and randomly from the set of outgoing links which belong to a shortest path to the destination of the packet. But note that we always assume that the path selection process is completely independent from the scheduling process. Thus, the routing paths can be viewed as input for the scheduling process.

An *oblivious routing problem* is defined by a set of N packets with preset routing paths. The term "oblivious" means that the packets are not allowed to leave their preset routing paths. An oblivious routing problem is called *shortest paths routing problem* if all routing paths are shortest paths. Since the paths have already been specified, a routing protocol for oblivious routing problems, which we call *oblivious routing protocol*, has only to determine which packets are allowed to move forward in a step and which have to wait. If we allow a global controller to precompute this schedule, we talk about *off-line* routing. If the schedule is produced while the packets are routed through the network, this is called *on-line* routing. We are interested in the construction of on-line routing protocols.

The following parameters greatly influence the routing time for oblivious routing problems:

- the *congestion* C, i.e., the maximum number of routing paths that pass through the same link, and
- the *dilation* D, i.e., the maximum length of the routing paths in the problem.

Clearly, $\max\{C, D\} = \Omega(C + D)$ is a lower bound on the routing time for any protocol on any oblivious routing problem with congestion C and dilation D, because at least one link

must be traversed by C packets, and at least one packet has to traverse D links. An oblivious protocol is said to be *greedy* if packets only have to wait at a link because they are delayed by another packet which moves along this link or because the link buffer at the end of the link is full. $C \cdot D$ is an upper bound on the routing time of greedy protocols on networks with unbounded buffers, because each packet has to wait at most $C - 1$ steps on every link of its routing path. An oblivious routing protocol is said to be *nonpredictive* if contention is resolved by a deterministic algorithm that is based only on the history of the contending packets' travels through the network and on information carried with the packets that is independent of their destination [4]. For instance, the first-in first-out protocol is nonpredictive. Finally, we call an oblivious routing protocol *strongly on-line* if we assume that the processors do not use any information about the parameters of a routing problem, i.e., the congestion, the dilation, or the size of the problem.

In the following, we represent the underlying processor network by a digraph $\mathcal{G} = (V, E)$, where V is the set of nodes or processors, and $E \subseteq V \times V$ is the set of directed edges or links. Of course, any network description which is based on undirected graphs can be represented in the digraph model just by replacing each undirected edge by two directed edges in opposite direction. We denote the diameter of the network by $\mathrm{diam}(\mathcal{G})$.

1.1 Known Results

All results in this section relate the routing times and buffer sizes required by oblivious routing protocols to the size, the congestion, and the dilation of the underlying oblivious routing problem. The size is denoted by N, the congestion by C, and the dilation by D.

Leighton, Maggs, and Rao [6] show that any oblivious routing problem can be routed in time $O(C + D)$ with constant-size link buffers, thereby achieving the naive lower bound. Their proof is based on the Lovász Local Lemma and shows only the existence of the optimal schedule. In [5], Leigthon, Maggs, and Richa present an algorithm for computing this schedule. But since the runtime of this algorithm is polynomial in the number of packets and links, it can not be applied to turn the above off-line protocol into an efficient on-line protocol.

Rabani and Tardos [11] modify the off-line protocol of Leighton, Maggs, and Rao such that they can replace the Lovász Local Lemma by a Chernoff bound in the analysis. This modification allows to calculate the schedule in a distributed way and on-line. The protocol guarantees routing time $O(C)+(\log^* N)^{O(\log^* N)} \cdot D+\mathrm{polylog}(N)$, w.h.p.[1]. Recently, Ostrovsky and Rabani [10] have improved on this result. They achieve routing time $O(C + D + \log^{1+\epsilon} N)$, w.h.p., for arbitrary $\epsilon > 0$. Both protocols are suitable for arbitrary simple routing paths, that is paths without cycles. They require buffers of size C.

Besides their off-line result, Leighton, Maggs, and Rao present in [6] a simple on-line scheduling protocol which completes the routing in time $O(C+D\cdot\log(DN))$, w.h.p., and requires buffers of size $\log(DN)$. A similar routing time but with much smaller buffer size is achieved in [3]: a protocol is given requiring routing time $O(C + D \cdot \log N)$, w.h.p., using buffers of size $\log D$.

Better results are known for special classes of networks. For instance, Ranade [12] proposes a probabilistic on-line routing protocol for butterfly networks. The proof is based on the delay sequence technique developed by Aleliunas [1] and Upfal [14]. The protocol can be easily extended to the class of bounded-degree leveled networks ([4], [8]). In a *leveled network*, the nodes can be partitioned into levels $0, \ldots, L$ such that each link in the network leads from some node on level i to some node on level $i + 1$, for $0 \leq i \leq L - 1$. Mostly, it is assumed that

[1] Throughout this paper, *w.h.p.* (with high probability) means: with probability at least $1 - N^{-\alpha}$ for any fixed constant α with N denoting the number of packets.

packets are routed only from level 0 to level L. Ranade's protocol completes the routing in time $O(C + L + \log N)$, w.h.p., using buffers of constant size. Note that all of the preceding protocols delay packets even if the next edge on their path is free. Thus, none of them are greedy.

Leighton [4] introduces a simple probabilistic greedy protocol for butterfly networks. It is called the *random-rank protocol*. This protocol is a simplified version of Ranade's protocol. Initially, each packet is assigned a random rank. The ranks are used to determine which packets move forward and which have to wait in a step. Applied to leveled networks, the protocol achieves asymptotically the same performance as Ranade's protocol, but it requires buffers of size C.

A detailed survey about all these routing protocols, including also most of the results presented in this work, is given in a book of Scheideler [13].

1.2 Overview – New Results

In Section 2, we introduce a new probabilistic on-line routing protocol which we call *growing-rank protocol*. We show that the growing-rank protocol routes any shortest paths routing problem of size N with congestion C and dilation D in $O(C + D + \log N)$ steps, w.h.p. Thus, we obtain the same bound for arbitrary networks as previously known only for bounded-degree leveled networks. Our protocol is greedy and very simple. The main difference to Leighton's random-rank protocol is that the packets' ranks are increased whenever the packets move forward. We present three versions of the growing-rank protocol. The first requires that estimations of C, D, and N are distributed among the processors, the second requires only that an upper bound on N is known by all processors, and the third makes no use of any of these parameters. Therefore, the above result is strongly on-line.

The drawback of the growing-rank protocol is that it requires shortest paths. This condition can be slightly weakened: A collection of paths $\mathcal{P}$ on a network $\mathcal{G} = (V, E)$ is said to be *shortcut-free*, if there is a subnetwork $\mathcal{G}' = (V, E')$ with $E' \subseteq E$ such that the paths in $\mathcal{P}$ are shortest paths in $\mathcal{G}'$. Of course, every set of shortest paths is shortcut-free.

Further, we investigate the behavior of the growing-rank protocol on random routing problems. We show that the above time bound holds even if C denotes the maximum expected congestion over all links. This value can be calculated very easily and exactly for many randomized path selection strategies. This is illustrated by several applications in Section 3.

We start with calculating the maximum expected congestion for random routing problems on *Cayley networks*. This class includes many important standard networks, e.g., all tori, the cube-connected cycles, and the butterfly networks. We give a simple scheme for the construction of *symmetric shortest path systems* in these networks. If the packets of a random h-routing problem on a network $\mathcal{G}$ are sent along the paths in this system, then the maximum expected congestion is at most $h \cdot \mathrm{diam}(\mathcal{G})$. Hence, the growing-rank protocol routes random h-routing problems on any Cayley network $\mathcal{G}$ of size n in time $O(h \cdot \mathrm{diam}(\mathcal{G}) + \log n)$, w.h.p.

Further, we investigate node and edge symmetric networks. Intuitively, a network is node (edge) symmetric, if it looks the same viewed from any node (edge) of the network. For instance, every Cayley network is node symmetric, and all equal-sided tori are edge symmetric. We give a very simple randomized path selection strategy which generates the routing paths to the destinations during the routing. This strategy achieves optimal maximum expected congestion for random routing problems on networks in both classes. Given any node symmetric network $\mathcal{G}$ of size n, we show that the maximum expected congestion for random h-routing problems is at most $h \cdot \mathrm{diam}(\mathcal{G})$. This implies routing time $O(h \cdot \mathrm{diam}(\mathcal{G}) + \log n)$, w.h.p. Given any

edge symmetric network $\mathcal{G}$ of size n and degree Δ, we show that the maximum expected congestion for random h-routing problems is at most $h/\Delta \cdot \mathrm{diam}(\mathcal{G})$. This implies routing time $O((h/\Delta + 1) \cdot \mathrm{diam}(\mathcal{G}) + \log n)$, w.h.p.

Our last application is a simple routing scheme for de Bruijn networks. We show that the maximum expected congestion is at most $O(h \cdot \log n)$ if the packets of a random h-routing problem are routed along shortest paths in the n-node de Bruijn network. This gives optimal routing time $O(h \cdot \log n)$, w.h.p.

By applying Valiant's paradigm *first routing to a random destination* [15], all application results for random h-routing problems hold also for arbitrary h-to-h-routing problems.

In Section 4, we examine the limits of our approach to design efficient oblivious routing protocols and ask:

- Is the restriction to shortest routing paths necessary?
- Do we need randomization?
- What happens if the buffer size is bounded?

We answer these questions by three examples.

The first example shows that the growing-rank protocol performs poorly on some routing problems with non-shortest paths. For instance, we describe an oblivious routing problem of size N with congestion $C = \log N / \log\log N$ and dilation $D = \log N$ for which the expected routing time of the growing-rank protocol is $\Omega(C \cdot D)$.

The second example illustrates that randomization is necessary. We show that given any nonpredictive protocol, there is a shortest paths routing problem with congestion C and dilation D that takes time $\Theta(C \cdot D)$. The result holds for any C and D. (A similar example with time bound $\Omega(C \cdot D / \log C)$ can be found in [7].) Interestingly, the underlying network is the butterfly network. Note that Ranade's protocol, Leighton's random-rank protocol, and the growing-rank protocol are nonpredictive for any fixed choice of the initial ranks. As a consequence, all of the three protocols perform poorly for routing on the butterfly in a deterministic setting.

The last example illustrates that routing with bounded buffers is a much more challenging task than routing with unbounded buffers, i.e., buffers of size C. In particular, the example shows that, in case of bounded buffers, a packet p can be delayed by packets whose routing paths do not overlap with the routing path of p. This can lead to a routing time much worse than $C \cdot D$ steps which is the upper bound for greedy routing with bounded buffers. Ranade's protocol [12] uses ghost packets to deal with this problem. But this technique is suitable only for leveled networks. Another difficulty arises if we consider non-leveled networks: the *deadlock* problem. Suppose there are m links $e_0, \ldots, e_{m-1}$ with full packet buffers, and every link e_i holds only packets that wait for moving forward along $e_{(i+1) \bmod m}$, for $0 \leq i \leq m-1$. Then all links are blocked, i.e., a deadlock occurs. We believe that avoiding deadlocks is the major problem to be solved in order to generalize our results to networks with bounded buffers.

2 The Growing-Rank Protocol

Now we introduce the growing-rank protocol. Suppose we are given a shortest paths routing problem of dilation D, congestion C, and size N on an arbitrary network $\mathcal{G}$. Let Q_e denote the set of packets that wait for moving forward over an outgoing link e in a step. Because the routing paths have already been determined, the protocol only has to specify which of the packets in Q_e is allowed to move forward and which packets have to wait. The protocol

forwards packets whenever possible, i.e., if Q_e is not empty, then one of the packets in Q_e is moved forward along e. The priorities among the packets are determined by *random ranks*.

Suppose R and $m := R/D$ are suitably large integers. (The exact value of R will be specified later.) Initially, each packet is assigned an integer rank chosen randomly, independently, and uniformly from the set $\{0, 1, \ldots, R-1\}$. Whenever a packet traverses a link, its rank is increased by m. If two or more packets are contending to move forward along a link, then one of those with minimum rank is chosen. Thus, for each outgoing link e with $|Q_e| \geq 1$, a step looks like this:

1. choose a packet $p \in Q_e$ with minimum rank,
2. increase the rank of p by m, and
3. move p forward along e.

In order to break ties among packets with same rank, we assume that each packet p has a unique *ident-number* denoted by $\mathrm{id}(p)$. If there are several packets with the same minimum rank, then one with smallest ident-number is chosen. These ident-numbers can be easily generated. For example, the ith packet starting at the jth processor gets the ident-number $i \cdot n + j$ with n denoting the total number of processors.

In the following, we denote the rank of p while waiting for moving forward along link e by $\mathrm{rank}^e(p)$. Further, we define the *ident-rank* of p at e as $\text{id-rank}^e(p) := \mathrm{rank}^e(p) + \mathrm{id}(p)/(\max(id) + 1)$ with $\max(id)$ denoting the maximum ident-number. Note that, at each link, the ident-ranks of all packets are distinct. The protocol ensures that, whenever a packet p delays a packet p' at a link e, then $\text{id-rank}^e(p) < \text{id-rank}^e(p')$.

2.1 Analysis of the Protocol

We will show that the growing-rank protocol completes the routing of any shortest paths routing problem of size N with congestion C and dilation D in $O(C + D + \log N)$ steps, w.h.p. Our analysis is based on a delay sequence argument similar to that in [4], [8], and [12].

Definition 2.1 ((s, ℓ, r)-delay sequence) *An (s, ℓ, r)-delay sequence consists of*

- *s* delay packets *$p_1, \ldots p_s$;*
- *s not necessarily distinct links $e_1, \ldots, e_s$ such that e_1 is the last link on the routing path of p_1, and for $2 \leq i \leq s$, e_i is a link on the routing path of p_{i-1} and p_i;*
- *$s-1$ integers $\ell_1, \ldots, \ell_{s-1} \geq 0$ with $\sum_{i=1}^{s-1} \ell_i \leq \ell$ such that for $1 \leq i \leq s-1$, ℓ_i is the number of links on the routing path of p_i from e_{i+1} to e_i (excluding e_{i+1}, and including e_i); and*
- *s integers $r_1, \ldots, r_s$ with $0 \leq r_s \leq r_{s-1} \leq \ldots \leq r_1 \leq r - 1$.*

We call s the length *of the delay sequence, and we say a delay sequence is* active, *if $\mathrm{rank}^{e_i}(p_i) = r_i$ for $1 \leq i \leq s$.*

Lemma 2.2 *Suppose the routing takes $T \geq R/m + D$ or more steps. Then a $(T - R/m - D, R/m + D, R + D \cdot m)$-delay sequence is active.*

Proof. We give a construction scheme for a delay sequence. Let p_1 be a packet that moves forward in step T or later along the last link on its routing path. Call this link e_1. We follow p_1's routing path backwards to the last link on this path where it was delayed. Call this link e_2 and the packet that caused the delay p_2. We now follow the path of p_2 backwards until we reach a link e_3 at which p_2 was forced to wait, because the packet p_3 was preferred. We change the packet again and follow the path of p_3 backwards. We can continue this construction until we reach a packet p_s which was not delayed in a step before. Thus, we have determined the delay packets and the links of a delay sequence of length s.

For $1 \leq i \leq s$, we set $r_i := \text{rank}^{e_i}(p_i)$. Since the growing-rank protocol prefers packets with smaller rank and since the maximum rank occurring during the routing is smaller than $R' := R + D \cdot m$, we have $0 \leq r_s \leq r_{s-1} \leq \ldots \leq r_1 \leq R'$.

The path from the source of p_s to the destination of p_1 recorded by the above process in reversed order is called *delay path*. It consists of contiguous parts of the delay packets' routing paths. We define the ℓ_i's to be the lengths of these parts as described in the definition of the delay sequence. Let ℓ denote the number of links on the delay path. Since the ranks in our sequence are increased by m at each of these links, it follows $\ell \cdot m \leq R'$. Consequently, we have $\sum_{i=1}^{s-1} \ell_i \leq \ell \leq R'/m \leq R/m + D$.

Our construction covers up at least T steps and consists of ℓ moves and s delays. Consequently, we have $s \geq T - \ell \geq T - R'/m = T - R/m - D$. Thus, if we stop the above construction at packet $p_{T-R/m-D}$, then we have built an active $(T - R/m - D, R/m + D, R + D \cdot m)$-delay sequence. ∎

Lemma 2.3 *If the routing paths of the packets are shortest paths, then the delay packets in an active delay sequence are pairwise distinct.*

Proof. Suppose, in contrast to our claim, that there is some packet p appearing twice in the delay sequence. Then there are i and j with $1 \leq i < j \leq s$ and $p = p_i = p_j$. Thus, the routing path of p crosses the delay path at the collision links e_j and e_i in that order.

Let κ denote the number of links on the routing path of p from e_j to e_i. Then the rank of p is increased κ times by m on this part of the routing path, and consequently,

$$\text{id-rank}^{e_i}(p) \quad = \quad \text{id-rank}^{e_j}(p) + \kappa \cdot m \ . \tag{1}$$

On the other hand, each packet p_k with $i + 1 \leq k \leq j$ delays the packet p_{k-1} at link e_k. Thus, $\text{id-rank}^{e_k}(p_{k-1}) > \text{id-rank}^{e_k}(p_k)$. Further, the ranks are increased by m on every link on the delay path between e_j and e_i. The number of these links is $\sum_{k=i}^{j-1} \ell_k$. This gives

$$\begin{aligned} \text{id-rank}^{e_i}(p) \quad &> \quad \text{id-rank}^{e_j}(p) + \sum_{k=i}^{j-1} \ell_k \cdot m \\ &\geq \quad \text{id-rank}^{e_j}(p) + \kappa \cdot m \ . \end{aligned} \tag{2}$$

Note that $\sum_{k=i}^{j-1} \ell_k \geq \kappa$, because the routing path of p is a shortest path.

Clearly, (2) contradicts (1). Consequently, there is no packet that appears twice in the delay sequence. ∎

Lemma 2.4 *The number of different (s, ℓ, r)-delay sequences is at most*

$$\text{ds}(s, \ell, r) := N \cdot 2^{\ell} \cdot \left(\frac{2\text{e}C \cdot (s + r)}{s} \right)^{s} \ .$$

Proof. We count the number of possible choices for each component:

- There are N possibilities to choose p_1. Of course, this fixes e_1 as well.
- Further, there are $\binom{(s-1)+\ell}{s-1} \leq \binom{s+\ell}{s}$ ways to choose the ℓ_i's, because $\sum_{i=1}^{s-1} \ell_i \leq \ell$.
- Now suppose p_{i-1}, e_{i-1}, and ℓ_{i-1} for $2 \leq i \leq s$ are fixed. Then e_i is that link on the routing path of p_{i-1} which has distance ℓ_{i-1} to e_{i-1}. Thus, e_i is fixed as well, and hence, we have at most C possibilities to choose p_i. Therefore, the number of possibilities to fix $p_2, \ldots, p_s$ and $e_2, \ldots, \mathrm{e}_s$ is at most $C^{s-1} \leq C^s$.
- Finally, there are $\binom{s+r}{s}$ possibilities to choose the r_i's such that $0 \leq r_s \leq \ldots \leq r_1 \leq r-1$.

Altogether, we find that the number of (s, ℓ, r)-delay sequences is at most

$$N \cdot \binom{s+\ell}{s} \cdot C^s \cdot \binom{s+r}{s} \ .$$

Applying the inequalities $\binom{a}{b} \leq 2^a$ and $\binom{a}{b} \leq \left(\frac{\mathrm{e}a}{b}\right)^b$ completes the proof. ∎

Theorem 2.5 *Suppose we are given a shortest paths routing problem of size N with congestion C and dilation D on an arbitrary network $\mathcal{G}$. Then the growing-rank protocol completes the routing in $O(C + D + \log N)$ steps, w.h.p.*

Proof. Lemma 2.2 and Lemma 2.3 show that the probability $\mathrm{prob}(T)$ that the routing takes $T = s + R/m + D$ or more steps is bounded by the probability that an $(s, R/m + D, R + D \cdot m)$-delay sequence with distinct delay packets is active. The probability that a fixed delay sequence with s distinct packets is active is R^{-s} because the ranks of all packets have to match the ranks in the sequence. Combining this with the bound on the number of delay sequences in Lemma 2.4 gives

$$\begin{aligned} \mathrm{prob}(T) &\leq \mathrm{ds}\left(s, \frac{R}{m} + D, R + D \cdot m\right) \cdot R^{-s} \\ &\leq N \cdot 2^{R/m+D} \cdot \left(\frac{2\mathrm{e}C \cdot (s + R + D \cdot m)}{s}\right)^s \cdot R^{-s} \ . \end{aligned}$$

Assuming $R \geq s$ gives

$$\begin{aligned} \mathrm{prob}(T) &\leq N \cdot 2^{R/m+D} \cdot \left(\frac{2\mathrm{e}C \cdot (2R + D \cdot m)}{s}\right)^s \cdot R^{-s} \\ &\leq N \cdot 2^{R/m+D} \cdot \left(\frac{2\mathrm{e}C \cdot (2 + D \cdot m/R)}{s}\right)^s \end{aligned}$$

which is at most $N^{-\alpha}$ for $s = \max\{4\mathrm{e}C \cdot (2 + D \cdot m/R), R/m + D + (\alpha+1)\log N\}$. Hence, the routing takes

$$T = s + \frac{R}{m} + D = \max\{4\mathrm{e}C \cdot (2 + D \cdot \frac{m}{R}), \frac{R}{m} + D + (\alpha+1)\log N\} + \frac{R}{m} + D \qquad (3)$$

or more steps with probability $N^{-\alpha}$. Finally, applying $m = R/D$ yields that the routing is completed in $O(C + D + \log N)$ steps, w.h.p. ∎

2.2 Becoming Strongly On-Line

The drawback of the protocol presented above is that each processor has to know estimations of the congestion C, the dilation D, and the size of the routing problem N. This is because we have assumed that the range of the ranks is sufficiently large, i.e., $R \geq \max\{4eC \cdot (2 + D \cdot m/R), 2D + (\alpha + 1)\log N\}$, and that the packets' ranks are increased by $m = R/D$ whenever the packets move forward. It is easy to check that the result on the routing time holds for every choice of R and m that satisfy $R = \Omega(C + D + \log N)$, $m = \Omega(R/(C + D + \log N))$, and $m = O(R/D)$.

In particular, it seems to be difficult to compute the congestion of the routing problem. Fortunately, we need only an upper bound on this value, e.g., $N \cdot D$ or $N \cdot \mathrm{diam}(\mathcal{G})$. Of course, D and N can be computed and distributed among the processors in $O(\mathrm{diam}(\mathcal{G}))$ steps. Alternatively, we can use upper bounds on D and N instead of exact values. But note that whereas the quality of the upper bound on N influences only the range of the ranks, the quality of the dilation bound influences the routing time. I.e., if we choose $m = R/D^*$ with $D^* \geq D$ then we get routing time $O(C + D^* + \log N)$. For instance, bounding the dilation by the diameter $\mathrm{diam}(\mathcal{G})$ gives routing time $O(C + \mathrm{diam}(\mathcal{G}) + \log N)$, even if $D << \mathrm{diam}(\mathcal{G})$.

The following variation of the protocol achieves routing time $O(C + D + \log N)$ without assuming that the dilation or the congestion are known by the processors. The processors only have to know upper bounds on the diameter of the network and the size of the routing problem. For $k \geq 0$, define *time interval* k to begin at step 2^k and to end at step $2^{k+1} - 1$. Hence, each interval k has length 2^k. Define k^* to be the smallest integer satisfying $2^{k^*+1} - 1 \geq N \cdot \mathrm{diam}(\mathcal{G})$. Then the routing is completed at the end of interval k^* surely. Choose $R = 2^{k^*-2}$. The value of m varies during the routing, i.e., the ranks of forwarded packets are increased by $m_k := R/2^{k-2}$ in interval k. Note that m_k is an integer for every $k \leq k^*$.

Now let k' denote the smallest integer satisfying $2^{k'-1} \geq \max\{12eC, D + (\alpha+1)\log N\} + D$, and suppose some packets have not reached their destination at the beginning of interval k'. We want to estimate the probability that the routing is not completed during interval k'. Analogously to the argument in the proof of Theorem 2.5, the probability that the routing is not completed during the next

$$
\begin{aligned}
T \quad &\overset{(3)}{=} \quad \max\{4eC \cdot (2 + D \cdot \frac{m_{k'}}{R}), \frac{R}{m_{k'}} + D + (\alpha+1)\log N\} + \frac{R}{m_{k'}} + D \\
&\overset{m_{k'}=R/2^{k'-2}}{\leq} \quad \max\{4eC \cdot (2 + D/2^{k'-2}), D + (\alpha+1)\log N\} + D + 2^{k'-1} \\
&\leq \quad \max\{12eC, D + (\alpha+1)\log N\} + D + 2^{k'-1} \\
&\leq \quad 2^{k'-1} + 2^{k'-1} = 2^{k'}
\end{aligned}
$$

steps is at most $N^{-\alpha}$. Consequently, all packets reach their destination in interval k', w.h.p., and thus, the routing takes at most $2^{k'+1} - 1 = O(C + D + \log N)$ steps, w.h.p.

Now we assume that the packets do not know any information about the routing problem, neither the congestion, the dilation, nor the size of the routing problem, nor any estimation of these values. Then the following variation of the growing-rank protocol achieves routing time $O(C + D + \log N)$, w.h.p. As above, define interval k to begin at step 2^k and to end at step $2^{k+1} - 1$, for $k \geq 0$. At the beginning of interval k, each packet is assigned a new random rank from the interval $R_k := 2^k$. (Note that it is sufficient to append a new random bit at the least significant position to each rank instead of assigning completely new ranks which simplifies the protocol slightly.) The ranks of the packets are increased by $m := 4$ when they are forwarded.

Let k' denote the smallest integer satisfying $2^{k'-1} \geq \max\{12eC, D + (\alpha+1)\log N\} + D$, and suppose the routing is not completed at the beginning of interval k'. Then the probability that some packets have not reached their destination in

$$\begin{aligned} T &\overset{(3)}{=} \max\{4eC \cdot (2 + D \cdot \frac{m}{R_{k'}}), \frac{R_{k'}}{m} + D + (\alpha+1)\log N\} + \frac{R_{k'}}{m} + D \\ &\overset{R_{k'}/m = 2^{k'-2}}{\leq} \max\{4eC \cdot (2 + D/2^{k'-2}), D + (\alpha+1)\log N\} + D + 2^{k'-1} \\ &\leq \max\{12eC, D + (\alpha+1)\log N\} + D + 2^{k'-1} \\ &\leq 2^{k'-1} + 2^{k'-1} = 2^{k'} \end{aligned}$$

steps is at most $N^{-\alpha}$. Therefore, the routing takes at most $2^{k'+1} - 1 = O(C + D + \log N)$ steps, w.h.p, which gives the following corollary.

Corollary 2.6 *Any shortest paths routing problem with congestion C, dilation D, and size N can be routed strongly on-line in time $O(C + D + \log N)$, w.h.p.*

2.3 Analysis for Random Routing Problems

Suppose N packets should be routed along randomly chosen shortest routing paths from their source node to a random destination in a network $\mathcal{G} = (V, E)$. Let $\mathcal{P}$ denote the set of packets, and $\mathcal{M}$ the set of all shortest paths in $\mathcal{G}$. We model the selection of the random destination for a packet $p \in \mathcal{P}$ and the selection of the routing path to this destination together, i.e., by a random choice of a path m from $\mathcal{M}$. For $m \in \mathcal{M}$ and $p \in \mathcal{P}$, we denote the probability that m is the randomly chosen routing path for p by $\text{prob}(p, m)$. Note that we do not demand that the routing paths are chosen uniformly from the set of all paths starting at the source of p. Further, we do not demand that all nodes have the same probability to become the random destination of p. However, in all of our applications they have. The only restriction we place on the path selection process is that the routing path for a packet is chosen independently from the routing paths of other packets and from the scheduling process. For any N-tuple of paths $M \in \mathcal{M}^p$ we say M *describes the result of the random path selection*, if $M = \times_{p\in\mathcal{P}} m_p$ with m_p denoting the randomly chosen routing path for packet p. Finally, the probability that M describes the result of the random path selection is denoted by $\text{prob}(M)$. Then for any $M = \times_{p\in\mathcal{P}} m_p \in \mathcal{M}^p$, we have $\text{prob}(M) = \prod_{p\in\mathcal{P}} \text{prob}(p, m_p)$ because the routing paths are chosen independently from each other.

The following example shows how random routing problems in which the routing paths are determined by a shortest paths system $\mathcal{W}$, which includes exactly one path $w(u, v) = (u \rightarrow \cdots \rightarrow v)$ for every pair u and v of nodes, can be represented in the above model. We assume that the random destination for each packet $p \in \mathcal{P}$ is chosen randomly, independently, and uniformly from the set V of nodes. Let $\text{source}(p)$ denote the source node of p and $\text{dest}(p)$ the randomly selected destination of p. Then we choose the path $w(\text{source}(p), \text{dest}(p)) \in \mathcal{W}$ as p's routing path. This strategy can be easily expressed in terms of the above model by simply specifying the probabilities that a path $m \in \mathcal{M}$ is the routing path of a packet $p \in \mathcal{P}$, i.e., we set

$$\text{prob}(p, m) := \begin{cases} \frac{1}{|V|} & \text{if } m \in \mathcal{W}_{\text{source}(p)} \\ 0 & \text{otherwise} \end{cases}$$

with $\mathcal{W}_v \subset \mathcal{W}$ denoting the set of paths starting at processor $v \in V$.

The following theorem bounds the routing time of the growing-rank protocol on random routing problems as described above. We assume that the packets' ranks are increased by $m := R/\mathrm{diam}(\mathcal{G})$ when the packets move forward. The given routing time depends on the total number of packets N, the diameter $\mathrm{diam}(\mathcal{G})$, and the maximum expected congestion $C_{\exp} := \max\{E(C_e)|e \in E\}$ with $E(C_e)$ denoting the expected number of packets traversing link e. We will see later that $C_{\exp}$ can be calculated very easily and exactly for random h-routing problems on several classes of networks.

Theorem 2.7 *Suppose we are given an arbitrary network $\mathcal{G}$ in which N packets should be routed along random routing paths. Suppose all routing paths are shortest paths which are chosen independently from each other. Let $C_{\exp}$ denote the maximum expected congestion. Then the growing-rank protocol completes the routing in $O(C_{\exp} + \mathrm{diam}(\mathcal{G}) + \log N)$ steps, w.h.p.*

Proof. Because of Lemma 2.2 and 2.3, we can bound the probability $\mathrm{prob}(T)$ that the routing takes $T = s + R/m + \mathrm{diam}(\mathcal{G})$ or more steps by the probability that an $(s, R/m + \mathrm{diam}(\mathcal{G}), R + \mathrm{diam}(\mathcal{G}) \cdot m)$-delay sequence with distinct packets is active. For any N-tuple of paths $M \in \mathcal{M}^P$ let $\mathrm{ds}(s, \ell, r, M)$ denote the number of possible (s, ℓ, r)-delay sequences with distinct delay packets under the assumption that M describes the result of the random paths selection. Then

$$\mathrm{prob}(T) \leq \sum_{M \in \mathcal{M}^p} \mathrm{prob}(M) \cdot \mathrm{ds}\left(s, \frac{R}{m} + \mathrm{diam}(\mathcal{G}), R + \mathrm{diam}(\mathcal{G}) \cdot m, M\right) \cdot R^{-s} \ . \tag{4}$$

Note that

$$\mathrm{ds}(s, \ell, r) := \sum_{M \in \mathcal{M}^p} \mathrm{prob}(M) \cdot \mathrm{ds}(s, \ell, r, M)$$

is equal to the expected number of possible (s, ℓ, r)-delay sequences for randomly chosen M. We can count this number as follows. There are at most $N \cdot \binom{s+\ell}{s} \cdot \binom{s+r}{s}$ ways to choose p_1, e_1, the ℓ_i's, and the r_i's. Now suppose p_{i-1}, e_{i-1}, and ℓ_{i-1} are fixed, for $2 \leq i \leq s$. Then e_i is that link on the routing path of p_{i-1} which has distance ℓ_{i-1} to e_{i-1}. Thus, e_i is fixed as well. What is the expected number of candidates for p_i under the assumption that e_i and $p_1, \ldots, p_{i-1}$ are already fixed? - The routing path of p_i must traverse e_i, and p_i must be distinct from $p_1, \ldots, p_{i-1}$. Let $\mathcal{M}_{e_i} \subseteq \mathcal{M}$ be the set of all paths in $\mathcal{M}$ that cross e_i. Then, the expected number of possibilities to choose p_i is at most

$$\sum_{p \in \mathcal{P} \setminus \{p_1, \ldots, p_{i-1}\}} \sum_{m \in \mathcal{M}_{e_i}} \mathrm{prob}(p, m) \ \leq \ \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}_{e_i}} \mathrm{prob}(p, m) \ = \ E(C_{e_i}) \ \leq \ C_{\exp} \ .$$

Because this bound is independent from the choices for the delay packets $p_1, \ldots p_{i-1}$, the expected number of choices for $p_2, \ldots, p_s$ and $e_2, \ldots, \mathrm{e}_s$ is at most $C_{\exp}^{s-1}$. Putting all the pieces together, we get

$$\begin{aligned} \mathrm{ds}(s, \ell, r) &\leq N \cdot \binom{s+\ell}{s} \cdot C_{\exp}^{s-1} \cdot \binom{s+r}{s} \\ &\leq N \cdot 2^{\ell} \cdot \left(\frac{2\mathrm{e}C_{\exp} \cdot (s+r)}{s}\right)^{s} \ . \end{aligned}$$

Applying this to equation (4) yields

$$\mathrm{prob}(T) \qquad \leq \qquad N \cdot 2^{R/m + \mathrm{diam}(\mathcal{G})} \cdot \left(\frac{2\mathrm{e}C_{\exp} \cdot (s + R + \mathrm{diam}(\mathcal{G}) \cdot m)}{s}\right)^{s} \cdot R^{-s}$$

$$\stackrel{m=R/\mathrm{diam}(\mathcal{G})}{\leq} \quad N \cdot 2^{2\cdot\mathrm{diam}(\mathcal{G})} \cdot \left(\frac{2\mathrm{e}C_{\exp}\cdot(s+2R)}{s}\right)^{s} \cdot R^{-s} \ .$$

We set $s := \max\{12\mathrm{e}C_{\exp}, 2\cdot\mathrm{diam}(\mathcal{G})+(\alpha+1)\log N\}$, for constant α. Thus $T = s+2\cdot\mathrm{diam}(\mathcal{G}) = O(C_{\exp}+\mathrm{diam}(\mathcal{G}))+(\alpha+1)\log N$. Further, we choose $R \geq s$. Then

$$\mathrm{prob}(T) \leq N \cdot 2^{2\cdot\mathrm{diam}(\mathcal{G})} \cdot \left(\frac{2\mathrm{e}C_{\exp}\cdot 3R}{12\mathrm{e}C_{\exp}\cdot R}\right)^{(\alpha+1)\log N+2\cdot\mathrm{diam}(\mathcal{G})} = N^{-\alpha} \ .$$

This completes the proof of Theorem 2.7 ∎

3 Applications

Now we give several applications for the growing-rank protocol. We investigate random routing problems on node symmetric networks, edge symmetric networks, and de Bruijn networks. All results in this section are consequences of Theorem 2.7.

3.1 Node Symmetric Networks

An *automorphism* of a network $\mathcal{G} = (V, E)$ is a permutation $\phi : V \longrightarrow V$ with the property that $(u, v) \in E \Leftrightarrow (\phi(u), \phi(v)) \in E$. The automorphisms of $\mathcal{G}$ form an algebraic group under the operation of composition. This group is denoted by $Aut(\mathcal{G})$. An automorphism group $U \subseteq Aut(\mathcal{G})$ is said to be *transitive* on $\mathcal{G}$ if, given any two nodes u and v, there is an automorphism $\phi \in U$ such that $\phi(u) = v$, and a network $\mathcal{G}$ is called *node symmetric* if $Aut(\mathcal{G})$ is transitive on it. Intuitively, a node symmetric network looks the same, if viewed from any node of the network.

The class of Cayley networks is an important subclass of node symmetric networks. Many standard networks belong to this class, e.g., all tori, the cube-connected-cycles, and the wrapped butterfly networks. Cayley networks are defined as follows. Let Γ be a finite algebraic group with identity 1, and suppose Σ is a set of generators of Γ with $1 \notin \Sigma$. Then the Cayley network $\mathcal{G}_{\Gamma,\Sigma} = (V, E)$ is defined by $V = \Gamma$ and $E = \{(a, b) \mid a^{-1}b \in \Sigma\}$. Figure 1 shows an example for a Cayley network.

Suppose $\mathcal{W}$ is a path system on a network $\mathcal{G} = (V, E)$ that includes a shortest path $w(u, v) = (u \to \cdots \to v)$ for every pair u and v of nodes. We call $\mathcal{W}$ *symmetric* if given any two nodes u and v there is a permutation $\psi : V \longrightarrow V$ such that for every path $(w_0 \to w_1 \to \cdots \to w_\ell) \in \mathcal{W}$ with $w_i = u$ there is a path $(\psi(w_0) \to \psi(w_1) \to \cdots \to \psi(w_\ell)) \in \mathcal{W}$ with $\psi(w_i) = v$, for $0 \leq i \leq \ell$. Roughly speaking, a *symmetric path system* has the property that it looks the same viewed from any node of the network.

Lemma 3.1 *For every Cayley network, there is a symmetric shortest path system.*

Proof. Let $\mathcal{G}_{\Gamma,\Sigma} = (V, E)$ be a Cayley network. Then there is a transitive automorphism group U of size $|V|$ [2]. We denote by ϕ_u^v the automorphism of U which maps the node u onto the node v, for $u, v \in V$. Thus, $U = \{\phi_u^v \mid v \in V\}$, for any $u \in V$.

Suppose $w = (w_0 \to w_1 \to \cdots \to w_\ell)$ is a path in $\mathcal{G}$ and ϕ is an automorphism of $\mathcal{G}$. Then we define $\phi(w) := (\phi(w_0) \to \phi(w_1) \to \cdots \to \phi(w_\ell))$. Since ϕ is an automorphism, $\phi(w)$ is a shortest path in $\mathcal{G}$ if and only if w is a shortest path in $\mathcal{G}$.

We construct a symmetric shortest path system in two steps. (For simplicity of notation, we assume $V = \{0, 1 \ldots, n-1\}$.)

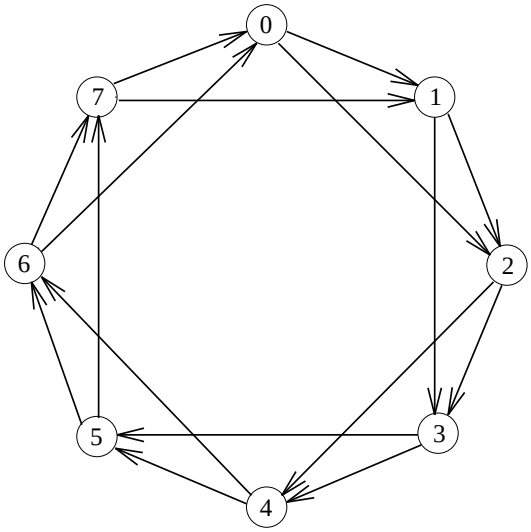

Figure 1: The Cayley network $\mathcal{G}_{\Gamma,\Sigma}$ with $\Gamma = (\mathbb{Z}_8, +)$ and $\Sigma = \{1, 2\}$. Note that the identity in Γ is 0 rather than 1 as Γ is an additive group.

Step 1: choose arbitrarily a shortest path $w(0, v)$ from the node 0 to every node $v \in V$.

Step 2: for every $u \in V \setminus \{0\}$ and every $v \in V$, define the path $w(u, v)$ from u to v by $w(u, v) := \phi_0^u(w(0, \phi_u^0(v)))$.

In the first step we have chosen n *prototype paths* (including the trivial one from 0 to 0). In the second step we have made $n - 1$ copies of each prototype path. Thus, every automorphism of U, except for the identity, has been used once for copying each prototype path.

Let u and v be two nodes of $\mathcal{G}$. We have to show that there is a permutation ψ which maps every path $w = (w_0 \to w_1 \to \cdots \to w_\ell) \in \mathcal{W}$ with $u = w_i$ onto a path $w' = (w'_0 \to w'_1 \to \cdots \to w'_\ell) \in \mathcal{W}$ with $v = w'_i$ for $0 \le i \le \ell$. For ψ we choose the automorphism $\phi_u^v \in U$. Clearly, ϕ_u^v maps u onto v. Thus, it remains only to prove that $w' = \phi_u^v(w)$ is a member of the path system $\mathcal{W}$.

From the construction scheme, we know that w is a copy of a prototype path w'' or a prototype path w'' itself. We claim

$$w' = \phi_u^v(w) \overset{(a)}{=} \phi_{w_0}^{w'_0}(w) \overset{(b)}{=} \phi_0^{w'_0} \circ \phi_{w_0}^0(w) \overset{(c)}{=} \phi_0^{w'_0}(w'') \ .$$

This can be proved as follows:

a) There is exactly one automorphism in U that maps w_0 onto w'_0. Consequently, $\phi_u^v = \phi_{w_0}^{w'_0}$.

b) Since U is a group, $\phi_0^{w'_0} \circ \phi_{w_0}^0$ is an element of U, and since there is only one automorphism in U that maps w_0 onto w'_0, it follows $\phi_0^{w'_0} \circ \phi_{w_0}^0 = \phi_{w_0}^{w'_0}$

c) It is $w = \phi_0^{w_0}(w'')$, and consequently, $w'' = \phi_{w_0}^0(w)$.

Hence, the automorphism $\phi_0^{w'_0} \in U$ maps w'' onto w'. If $\phi_0^{w'_0}$ is the identity, then w' is generated in Step 1. Otherwise, w' is generated as a copy of w'' in Step 2. ∎

Theorem 3.2 *Let $\mathcal{G} = (V, E)$ be a Cayley network with symmetric shortest path system $\mathcal{W}$. Suppose each processor sends h packets to randomly and uniformly chosen destinations along the paths described in $\mathcal{W}$. Then the growing-rank protocol completes the routing in time $O(h \cdot \mathrm{diam}(\mathcal{G}) + \log |V|)$, w.h.p.*

Proof. For every path $w \in \mathcal{W}$, the expected number of packets that traverse w is $h/|V|$. Further, for symmetry reasons, the number of paths in $\mathcal{W}$ passing through a node v is the same for all nodes $v \in V$, namely at most $\mathrm{diam}(\mathcal{G}) \cdot |V|$. Hence, the expected number of packets that pass through a node is at most

$$\frac{h}{|V|} \cdot \mathrm{diam}(\mathcal{G}) \cdot |V| = h \cdot \mathrm{diam}(\mathcal{G}) \ ,$$

and therefore $C_{\exp} \leq h \cdot \mathrm{diam}(\mathcal{G})$. Finally, applying Theorem 2.7 yields that the routing time of the growing-rank protocol is $O(h \cdot \mathrm{diam}(\mathcal{G}) + \log(h \cdot |V|)) = O(h \cdot \mathrm{diam}(\mathcal{G}) + \log |V|)$, w.h.p. ∎

For bounding $C_{\exp}$ in the proof of the above theorem, we used the symmetry properties of the path system $\mathcal{W}$. As seen, symmetric path systems can be easily constructed for Cayley networks. For non-Cayley node symmetric networks, like for example the Petersen graph [16], the construction in the proof of Lemma 3.1 fails. Here we have to choose another path selection strategy. Suppose the destinations for the packets are specified. Then we select the routing paths randomly during the routing instead of beforehand by a path system. We assume that each processor chooses randomly the link for transmitting a selected packet just before the packet is passed on. This link is chosen randomly and uniformly from the set of outgoing links which belong to a shortest path to the destination of the packet.

Theorem 3.3 *Let $\mathcal{G} = (V, E)$ be a node symmetric network. Suppose each processor sends h packets to randomly and uniformly chosen destinations. Further, suppose that the routing paths are selected randomly during the routing as described above. Then the growing-rank protocol completes the routing in time $O(h \cdot \mathrm{diam}(\mathcal{G}) + \log |V|)$, w.h.p.*

Proof. Let C_v denote the number of packets that traverse a node $v \in V$. $E(C_v)$ is the same for all nodes $v \in V$ for symmetry reasons. Therefore,

$$|V| \cdot C_{\exp} \leq \sum_{v \in V} E(C_v) \leq h \cdot |V| \cdot \mathrm{diam}(\mathcal{G})$$

which gives $C_{\exp} \leq h \cdot \mathrm{diam}(\mathcal{G})$. Finally, it follows from Theorem 2.7 that the routing time of the growing-rank protocol is $O(h \cdot \mathrm{diam}(\mathcal{G}) + \log |V|)$, w.h.p. ∎

3.2 Edge Symmetric Networks

We say that a network $\mathcal{G} = (V, E)$ is *edge symmetric*, if given any pair of edges (u, v) and (u', v') there is an automorphism $\phi \in Aut(\mathcal{G})$ such that $\phi(u) = u'$ and $\phi(v) = v'$. Thus, each edge in an edge symmetric network can be mapped by an automorphism onto any other edge. Intuitively, all edges in an edge symmetric network look the same. All equal-sided tori, for example, are edge symmetric. For these networks we suggest the same path selection strategy as for the general node symmetric networks. The following result improves the one for node symmetric networks slightly.

Theorem 3.4 *Let $\mathcal{G} = (V, E)$ be a edge symmetric network of degree Δ. Suppose each processor sends h packets to randomly and uniformly chosen destinations. Further, suppose that the routing paths are selected randomly during the routing as described above. Then the growing-rank protocol completes the routing in time $O((h/\Delta + 1) \cdot \mathrm{diam}(\mathcal{G}) + \log |V|)$, w.h.p.*

Proof. Let C_e denote the number of packets that traverse an edge $e \in E$. For symmetry reasons, $E(C_e)$ is the same for all $e \in E$, namely C_{exp}. Hence,

$$|E| \cdot C_{\mathrm{exp}} = \sum_{e \in E} E(C_e) \leq h \cdot |V| \cdot \mathrm{diam}(\mathcal{G})$$

which gives

$$C_{\mathrm{exp}} \leq \frac{h \cdot |V| \cdot \mathrm{diam}(\mathcal{G})}{|E|} = \frac{h \cdot \mathrm{diam}(\mathcal{G})}{\Delta} \quad .$$

Now applying Theorem 2.7 yields that the routing time of the growing-rank protocol is $O(h/\Delta \cdot \mathrm{diam}(\mathcal{G}) + \mathrm{diam}(\mathcal{G}) + \log |V|)$, w.h.p. ∎

3.3 De Bruijn Networks

The k-dimensional *de Bruijn network* has $n = 2^k$ nodes. These nodes are represented by k-bit binary strings, and each node $u_1 u_2 \dots u_k$ has a link to the node $u_2 \dots u_k 0$ and to the node $u_2 \dots u_k 1$. The diameter of the network is $k = \log n$. Figure 2 gives an example.

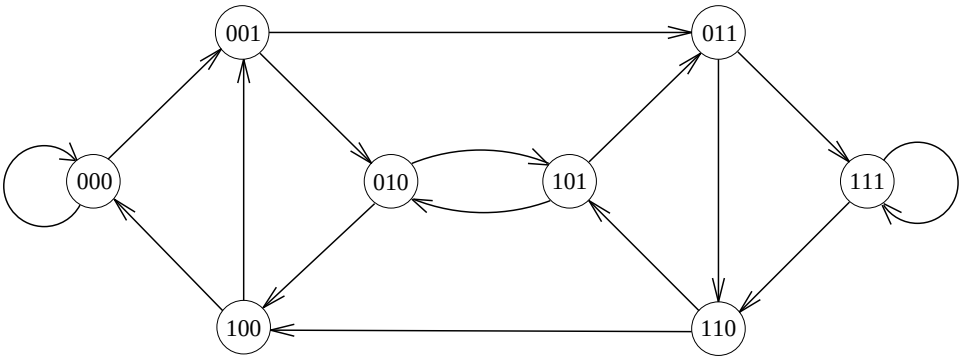

Figure 2: The 3-dimensional de Bruijn network.

For two nodes $u = u_1 \dots u_k$ and $v = v_1 \dots v_k$, we define $\kappa = \kappa(u, v) \leq k$ to be the largest integer satisfying $u_{k-\kappa+1} \dots u_k = v_1 \dots v_\kappa$. For instance, $\kappa(0000101, 1010000) = 3$. Let $\mathcal{W}$ be the path system in which the path $w(u, v) \in \mathcal{W}$ from a node u to a node v is defined by

$$\begin{aligned} w(u,v) = (u = u_1 \dots u_k = u_1 \dots u_{k-\kappa} v_1 \dots v_\kappa \quad &\rightarrow \quad u_2 \dots u_{k-\kappa} v_1 \dots v_{\kappa+1} \\ &\rightarrow \quad u_3 \dots u_{k-\kappa} v_1 \dots v_{\kappa+2} \\ &\vdots \\ &\rightarrow \quad v_1 \dots v_k = v) \quad . \end{aligned}$$

Obviously, the length of this path is $k - \kappa$, and since this is equal the distance between u and v, the path is a shortest path.

Theorem 3.5 *Suppose each processor in the de Bruijn Network of size n sends h packets to randomly and uniformly chosen destinations along the paths in $\mathcal{W}$. Then the growing-rank protocol completes the routing in time $\Theta(h \cdot \log n)$, w.h.p.*

Proof. Define $k := \log n$. First, we show that there are at most $k \cdot n$ paths in $\mathcal{W}$ that pass through an arbitrary link (u, u'). Define M_i to be the set of all nodes v such that the distance from v to u is i, and define M'_i to be the set of all nodes v such that the distance from u' to v is i, for $0 \leq i \leq k$. Obviously, $|M_i| \leq 2^i$ and $|M'_i| \leq 2^i$.

Suppose $w(v, v')$ is a path from $\mathcal{W}$ of length ℓ such that (u, u') is the ith link on this path for $1 \leq i \leq \ell$. Then $v \in M_{i-1}$ and $v' \in M'_{\ell-i}$. Thus, the number of paths that pass through (u, u') is at most

$$\left| \bigcup_{\ell=1}^{k} \bigcup_{i=1}^{\ell} M_{i-1} \times M'_{\ell-i} \right| \leq \sum_{\ell=1}^{k} \sum_{i=1}^{\ell} 2^{i-1} \cdot 2^{\ell-i} \leq (k-1) \cdot 2^k + 1 \leq k \cdot n \ .$$

As a consequence, the expected number of packets that traverse through (u, u') is at most $(h \cdot k \cdot n)/n = h \cdot k$, and hence, $C_{\text{exp}} \leq h \cdot k$. Now our theorem follows by applying Theorem 2.7. ∎

4 Limits of our Approach

In this section, we try to illustrate which additional problems occur for routing along non-shortest paths, for deterministic routing, and for routing with bounded buffers.

4.1 The Growing-Rank Protocol on Non-Shortest Paths

Here we investigate the behavior of the growing-rank protocol on non-shortest paths. We give an oblivious routing problem with congestion C and dilation D where the protocol behaves poorly, e.g., takes expected time $\Theta(C \cdot D)$ for $D = \log N$ and $C = \log N / \log\log N$. The routing paths in this example are non-shortest but *simple*, i.e., each node appears at most once in the path.

Theorem 4.1 *Suppose N, D, and C satisfy $\log N / \log\log N \leq C \leq N^\epsilon$ with $\epsilon < 1$ and $C \geq D / \log\log N$. Then there is an oblivious routing problem of size N, dilation D, and congestion C such that the expected routing time of the growing-rank protocol on this problem is $\Omega(C + D \cdot \log N / \log\log N)$.*

Remark 4.2 *We assume that the packets' ranks are increased by $m = R/D$ when the packets move forward. It is easy to check that the result holds also for any m with $m = \Omega(R/(C + D + \log N))$ and $m = O(R/D)$ if C satisfies $C \geq (R/m)/\log\log N$ instead of $C \geq D/\log\log N$. Note that this gives examples with routing time $\Theta(C \cdot D)$ for every m fulfilling the conditions described in Section 2.2.*

Proof. Consider the *zip network* in Figure 3. For simplicity, we assume that C is even and

Figure 3: The zip network and the routing path for the packets in $\mathcal{A}$. (Sorry, I lost the picture, see Journal of Algorithms.)

$D = 2d - 1$ for some d as given in this picture. Suppose we are given two sets $\mathcal{A}$ and $\mathcal{B}$ each of $C/2$ packets with source node u_1 and v_1 respectively. These packets should be routed with the growing-rank protocol. The routing path of the packets in $\mathcal{A}$ is

$$u_1 \to u_2 \to v_1 \to v_2 \to v_3 \to v_4 \to u_3 \to u_4 \to u_5 \to u_6 \to v_5 \to \cdots \\ \cdots \to v_{d-3} \to v_{d-2} \to v_{d-1} \to v_d \to u_{d-1} \to u_d$$

as shown in the picture, and the routing path of the packets in $\mathcal{B}$ is

$$v_1 \to v_2 \to u_1 \to u_2 \to u_3 \to u_4 \to v_3 \to v_4 \to v_5 \to v_6 \to u_5 \to \cdots \\ \cdots \to u_{d-3} \to u_{d-2} \to u_{d-1} \to u_d \to v_{d-1} \to v_d \ .$$

Define $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{B}$ to be the sets of packets with initial ranks smaller than $2m$. Suppose $|\mathcal{A}'| = |\mathcal{B}'| = k$. Then the ranks of the packets in $\mathcal{A}'$ are bigger than the ranks of the packets in $\mathcal{B}'$ at node v_1, because they have been increased twice by m on their way from u_1 to v_1. Consequently, these packets are delayed by the packets of $\mathcal{B}'$ for $k-2$ steps at this node. By the same arguments, the packets in $\mathcal{B}'$ are delayed for $k-2$ times at node u_1. Further, suppose all other packets have ranks not smaller than $4m$. Then the packets in $\mathcal{A}'$ and $\mathcal{B}'$ are not affected by these packets, and the above event recurs at the nodes u_3, v_3; u_5, v_5; and so on. As a consequence, the first packet reaches its destination after $(k-2) \cdot d/2 + (2d-1)$ steps, and thus, the routing time is at least $(k-2) \cdot d/2 + (2d-1) + (c-1) \geq k \cdot D/4 + C/2$.

Now assume that we have a routing network which includes $\lfloor N/C \rfloor$ disjoint copies of the zip network each of which with the above described routing problem. This gives an oblivious routing problem of size (at most) N, dilation D, and congestion C. We will show that the expected routing time for this problem is $\Omega(C + D \cdot k)$ for suitable $k = \Omega(\log N / \log\log N)$.

This is trivially true for $C = \Omega(k \cdot D)$ since any protocol requires at least C steps. Further, it is true for $D = O(1)$ since $C \geq \log N / \log\log N$. Therefore, we assume that $C \leq k \cdot D/2$ and $D \geq 8$. Then the probability that $k \leq C/2 = \Omega(\log N / \log\log N)$ packets from the set $\mathcal{A}$ in a fixed copy have ranks smaller than $2m$ and $C/2 - k$ packets have ranks of at least $4m$ is

$$\begin{aligned} \binom{C/2}{k} \cdot \left(\frac{2m}{R}\right)^k \cdot \left(1 - \frac{4m}{R}\right)^{C/2-k} &\overset{(a)}{\geq} \binom{C/2}{k} \cdot \left(\frac{2m}{R}\right)^k \cdot 4^{\left(-\frac{4m \cdot (C/2-k)}{R}\right)} \\ &\overset{(b)}{\geq} \binom{C/2}{k} \cdot \left(\frac{2m}{R}\right)^k \cdot 4^{-k} \\ &\geq \left(\frac{C \cdot m}{4k \cdot R}\right)^k \ , \end{aligned}$$

where equation a) holds because $4m/R \leq 4/D \leq 1/2$, and equation b) holds because $C \leq k \cdot D/2 \leq k \cdot R/2m$. As the same bound holds for the packets in $\mathcal{B}$, the probability that the event described above happens in none of the at least $\lfloor N/C \rfloor \geq N/2C \geq N^{1-\epsilon}/2$ copies is at

most

$$\begin{aligned}\left(1-\left(\frac{C\cdot m}{4k\cdot R}\right)^{2k}\right)^{N^{1-\epsilon}/2} &\leq \exp\left(-\frac{N^{1-\epsilon}}{2}\cdot\left(\frac{C\cdot m}{4k\cdot R}\right)^{2k}\right)\\ &\overset{(c)}{\leq} \exp\left(-\frac{N^{1-\epsilon}}{2}\cdot\left(\frac{1}{4k\cdot\log\log N}\right)^{2k}\right)\\ &\overset{(d)}{\leq} \exp\left(-\frac{N^{1-\epsilon}}{2}\cdot(\log N)^{-(1-\epsilon)\log N/2\log\log N}\right)\\ &\leq \exp\left(-\frac{N^{(1-\epsilon)/2}}{2}\right)=o(1)\ ,\end{aligned}$$

where Equation (c) holds because $C \geq D/\log\log N = R/(m\cdot\log\log N)$, and Equation (d) holds because $k \leq (1-\epsilon)\log N/4\log\log N$. As a consequence, the expected routing time is at least $(1-o(1))\cdot(C+D/4\cdot k) = \Omega(C+D\cdot\log N/\log\log N)$. ∎

4.2 Deterministic Routing

Now we consider deterministic routing. We investigate the behavior of nonpredictive routing protocols in which all scheduling decisions have to be independent from the future routing paths of the packets. Note that the growing-rank protocol is not deterministic, and hence not nonpredictive. However, for any fixed setting of the initial ranks it is nonpredictive. The same holds for Leighton's random-rank protocol [4] and for Ranade's protocol [12]. The following example shows that all these protocols perform poorly in a deterministic setting even on leveled networks. (A similar example yielding a lower bound of $\Omega(C\cdot D/\log C)$ rather than $\Omega(C\cdot D)$ is presented in [6].)

Theorem 4.3 *Suppose we are given any deterministic non-predictive routing protocol $\mathcal{Q}$ for routing on the D-dimensional butterfly network. Then, for any C, there is a routing problem with congestion C for which $\mathcal{Q}$ takes time $\Omega(C\cdot D)$.*

Proof. Fix an arbitrary output node v on level D of the butterfly. This node is the root of a complete binary tree T of height D whose leaves are the 2^D input nodes on level 0.

We assume that each input node wants to send out C packets. For the first edge on the routing path of each packet we choose the edge to the parent node of the source node in the tree T. The following edges are specified inductively such that each edge in our tree T is passed by C routing paths. Suppose u is a node on level ℓ with $1 \leq \ell \leq D-1$ which belongs to the tree T. Then u is crossed by $2C$ routing paths. We assume that these paths are determined already up to level ℓ, and we have to continue the paths up to the next level. This we do depending on the behavior of protocol $\mathcal{Q}$ up to level ℓ. We choose the paths of those C packets that would arrive first at node u to leave the tree and the paths of the other C packets to stay in the tree, i.e., to cross the parent node of u. This defines the routing paths inside the trees. For the paths outside the tree we only demand that they have congestion C.

Now we calculate the time which is taken by $\mathcal{Q}$ for routing the above defined problem. A node on level $1 \leq \ell \leq D$ receives its first packet at time $(\ell-1)\cdot C/2+\ell$ or later. (For simplicity, we assume that C is even.) Hence, the root receives its first packet at time $(D-1)\cdot C/2+D$ or later. As a consequence, the routing takes at least $C/2\cdot(D-1)+D+(C-1) = \Omega(C\cdot D)$ steps. ∎

4.3 Routing with Bounded Buffers

Suppose the packets that are transmitted along a link are stored in an *link buffer* at the end of the link until they are forwarded along the next link on their paths. If a link buffer is full, then the respective link cannot transmit packets until one of the packets leaves the buffer.

As seen in the introduction, $C \cdot D$ is an upper bound on the routing time of greedy protocols on networks with unbounded buffers. The following example shows that this bound does not hold for networks with bounded buffers.

Theorem 4.4 *For every C, D, and B there exists a greedy routing protocol that requires time $\Omega((C/2-B) \cdot D^2)$ for a routing problem with congestion C and dilation D on a bounded-degree leveled network with buffer size B.*

Proof. For $C/2 \leq B$ our theorem is trivially true. Therefore, we assume $C/2 > B$. Further, we assume for simplicity that C is even.

Figure 4 defines the *railway network of depth D*. Suppose we have $D+1$ sets $A_0, \ldots, A_D$

Figure 4: The railway network of depth D. (Sorry, I lost the picture, see Journal of Algorithms.)

each of which including $C/2$ packets. The packets in A_i, for $0 \leq i \leq D$, should be routed from node u_i to node v_i. The scheduling rules are defined by a simple rule: packets in A_j are prefered against packets in A_i, for $0 \leq i < j \leq D$. How long does it take until the first packet of A_0 reaches v_0?

After $D+B-1$ steps, B packets from each set A_i with $1 \leq i \leq D$ have traversed link e_i. Furthermore, the buffers at the end of each link e_i, for $1 \leq i \leq D-1$, are filled with B packets from A_i at this time. In the following $C/2-B$ time steps, the link e_D is traversed by packets of A_D. As a consequence, all packets stored in the buffers of $e_1, \ldots, e_{D-1}$ are blocked. In general, the packets in A_{D-i} with $0 \leq i \leq D-1$ traverse along link e_{D-i} from time $D+B+i \cdot (C/2-B)$ to $D+B+(i+1) \cdot (C/2-B)-1$. Hence, the packets of $A_0, \ldots, A_{D-i-1}$ are blocked during this time. Consequently, the first packet of A_0 reaches v_0 after time step $D+B+D \cdot (C/2-B) = B+D \cdot (C/2-B+1)$.

Now suppose the packets of A_0 together with the A_0-packets of $D-1$ other railway networks of same depth, are used to build a new routing problem of the above described structure on a further railway network of depth $D-1$. Note that this does not increase the depth of the network since the packets in A_0 have traversed only one edge so far. In the added network there will be a set of C packets which takes $B+(D-1) \cdot (C/2-B+1)$ further steps until the first of these packets traverses link e_1 in this network. Again, this set of packets can be used as input for a railway network of depth $D-2$. If we continue this construction until the added networks have depth 1, then we get a routing problem with congestion C on a leveled network of depth D. The routing problem constructed in this way takes time $\sum_{i=1}^{D}(i \cdot (C-B+1)+B)+(C-1) = \Omega(C \cdot D^2)$. ∎

5 Acknowledgement

We would like to thank Christian Scheideler and Rolf Wanka for helpful discussions.

References

[1] M. Aleliunas. Randomized parallel communication. In *Proceedings of the Symposium on Principles of Distributed Computing*, pp. 60-72, 1982.

[2] N.L. Biggs. *Algebraic graph theory*, Second Edition, Cambridge University Press (Cambridge 1993).

[3] R. Cypher, F. Meyer auf der Heide, C. Scheideler, and B. Vöcking. Universal Algorithms for Store-and-Forward and Wormhole Routing. In *Proceedings of the 28th Symposium on Theory of Computing*, pp. 356-365, 1996.

[4] F.T. Leighton. *Introduction to parallel algorithms and architectures: arrays · trees · hypercubes*. Morgan Kaufmann Publishers (San Mateo, CA 1992).

[5] F.T. Leighton, B.M. Maggs, and A.W. Richa. Fast algorithms for finding O(congestion + dilation) packet routing schedules. Technical Report CMU-CS-96-152, School of Computer Science, Carnegie-Mellon University, 1996. *Combinatorica*, to appear.

[6] F.T. Leighton, B.M. Maggs, and S.B. Rao. Universal packet routing algorithms (Extended Abstract). In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pp. 256-271, 1988.

[7] F.T. Leighton, B.M. Maggs, and S.B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica* 14 (2), pp. 167-186, 1994.

[8] F.T. Leighton, B.M. Maggs, A.G. Ranade, and S.B. Rao. Randomized routing and sorting on fixed-connection networks. In *Journal of Algorithms* 17, pp. 157-205, 1994.

[9] F. Meyer auf der Heide and B. Vöcking. A Packet Routing Protocol for Arbitrary Networks. In *Proceedings of the 12th Symposium on Theoretical Aspects of Computer Science*, pp. 291-302, 1995.

[10] R. Ostrovsky and Y. Rabani. Universal $O(\text{congestion} + \text{dilation} + \log^{1+\epsilon} N)$ local control packet switching algorithms. In *Proceedings of the 29th Annual Syposium on Theory of Computing*, pp. 644-653, 1997.

[11] Y. Rabani and É. Tardos. Distributed Packet Switching in Arbitrary Networks. In *Proceedings of the 28th Annual Syposium on Theory of Computing*, pp. 366-375, 1996.

[12] A.G. Ranade. How to emulate shared memory. In *Journal of Computer and System Sciences* 42, pp. 307-326, 1991.

[13] C. Scheideler. *Universal Routing Strategies for Interconnection Networks*. Lecture Notes in Computer Science 1390. Springer-Verlag (Berlin, Heidelberg 1998).

[14] E. Upfal. Efficient schemes for parallel communication. In *Journal of the Association for Computing Machinery* Vol. 31, No. 3, pp. 507-517, 1984.

[15] L.G. Valiant. A scheme for fast parallel communication. In *SIAM Journal on Computing* 11/2, pp. 350-361, 1982.

[16] H.P. Yap. *Some topics in graph theory*, Cambridge University Press (Cambridge 1986).