Universal Algorithms for Store-and-Forward and Wormhole Routing

Robert Cypher*
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA

Friedhelm Meyer auf der Heide, Christian Scheideler, Berthold Vöcking†
Department of Mathematics and Computer Science
and Heinz Nixdorf Institute, University of Paderborn
33102 Paderborn, Germany

Abstract

In this paper we present routing algorithms that are universal in the sense that they route messages along arbitrary (simple) paths in arbitrary networks. The algorithms are analyzed in terms of the number of messages being routed, the maximum number of messages that must cross any edge in the network (edge congestion), the maximum number of edges that a message must cross (dilation), the buffer size, and the bandwidth of the links. We present two main results, both of which have applications to universal store-and-forward routing and universal wormhole routing. Our results yield significant performance improvements over all previously known universal routing algorithms for a wide range of parameters, and they even improve many time bounds for standard networks. In addition, we present adaptations of our main results for routing along shortest paths in arbitrary networks, and for routing in leveled networks, node-symmetric networks, edge-symmetric networks, expanders, butterflies, and meshes.

1 Introduction

A fundamental problem in any parallel or distributed system is the efficient communication of data between processors. While it is possible to design a specific routing algorithm for each possible interconnection network, a much more general approach is to create a single *universal* routing algorithm that can be used in any network [LMR94, GO93]. More precisely, a universal routing algorithm specifies a protocol for sending n messages along any n (simple) paths in any network. In addition to providing a unified approach to routing in standard networks, universal routing algorithms are ideally suited to routing in irregular networks that are used in wide-area networks and that arise when standard networks develop faults. Furthermore, universal routing places no restrictions on the pattern of communication that is being implemented (such as requiring that it form a permutation).

Most parallel and distributed systems utilize either *store-and-forward* or *wormhole* routing. In store-and-forward routing, each message can cross a single link in unit time. The store-and-forward model is a standard model which has been widely used to study routing and other problems in parallel computers (see, e.g., [L92]). In wormhole routing, messages are sent as *worms*, each of which consists of a sequence of fixed size units called *flits*. The *length* of a worm is the number of flits that it contains. The first flit is called the *head* and the remaining flits are called the *body* of the worm. During routing, a worm occupies a contiguous sequence of edges along its path, one flit per edge*. Wormhole routing is an extremely popular strategy for data movement in parallel computers and is used in a variety of machines including the Intel Paragon, CRAY T3D, MIT J-Machine, and Stanford DASH.

In this paper we will present on-line universal algorithms for both store-and-forward and wormhole routing. The interconnection network will be modeled as an undirected graph $G = (V, E)$ where each edge in E consists of two *links*, one in each direction. The routing problem will be defined by specifying a path collection $\mathcal{P}$, which is a multiset of paths in G. A path collection is *simple* if no path contains the same edge more than once, and it is a *shortest path collection* if all paths are shortest paths in G. The routing problem consists of routing one message along each of the paths in $\mathcal{P}$. Each node in V contains an *injection buffer* and a *delivery buffer*. Initially, each message is stored in the injection buffer of its source. Once a message reaches its destination, it is stored in the destination's delivery buffer. The routing algorithm operates in discrete, synchronous time steps.

When the store-and-forward routing model is used, it will

* email:cypher@cs.jhu.edu. Part of this research was performed while the author was visiting the University of Paderborn, supported by the Ministry of Research of Northrhine-Westfalia, and part was supported by NSF grant MIP-9525887.

† email: {fmadh,chrsch,voecking}@uni-paderborn.de, Fax: +49-5251-606482. Supported in part by DFG-Sonderforschungsbereich 376 "Massive Parallelität: Algorithmen, Entwurfsmethoden, Anwendungen", and by DFG Leibniz Grant Me872/6-1.

*It should be noted that circuit-switched routing can be viewed as a special case of wormhole routing in which the length of each worm is at least as large as the distance between the worm's source and its destination.

be assumed that each link has a *link buffer* of size A. Each packet must be stored in the buffer associated with a link prior to crossing the link. At most one packet can cross a link in unit time. When the wormhole routing model is used, it will be assumed that every worm is of length L. At most one flit from each of B different worms can cross a link in unit time. While many papers on wormhole routing assume that the parameter B (which we call the *link bandwidth*) equals one, most parallel machines that support wormhole routing multiplex several worms over each link in order to prevent deadlocks and/or improve performance [DS87, D90]. Furthermore, the use of optical links with extremely high bandwidths (see [SM93]) and their ability to support ATM-based B-ISDN networks with multiple virtual channels per link indicate that link bandwidths greater than one will become increasingly common in the near future. None of our wormhole routing algorithms ever delays worms that encounter congestion. As a result, our algorithms can operate without any buffers for delayed flits (which is a significant advantage when optical links are used, as the flits do not have to be converted to and from electronic form for buffering).

Regardless of the switching mode being used, the following parameters will be used to analyze the performance of the routing algorithms. The parameter n will denote the number of messages being sent. The parameter C, called the *congestion*, is defined as the maximum number of paths in $\mathcal{P}$ that cross any link. The parameter D, called the *distance* or *dilation*, is defined as the length of the longest path in $\mathcal{P}$.

1.1 Previous Results

In terms of universal store-and-forward routing, the best time bounds were obtained by Leighton, Maggs and Rao [LMR94]. They presented an optimal $O(C+D)$ time *off-line* algorithm for arbitrary simple path collections that uses constant size link buffers [LMR94]. Unfortunately, there is a significant gap between this algorithm and the best known *on-line* algorithm, which requires buffers of size $O(\log(nD))$ and takes $O(C+D\log n)$ time steps, w.h.p.[†], for the same problem [LMR94]. Better results are known for shortest path collections. Meyer auf der Heide and Vöcking gave an on-line algorithm for routing along arbitrary shortest paths that takes time $O(C+D+\log n)$, w.h.p., which is optimal for routing in bounded degree networks [MV95]. However, their algorithm requires buffers of size C.

Many results have been obtained for store-and-forward routing in specific networks or classes of networks. Valiant showed that any permutation can be routed in $O(\log N)$ time, w.h.p., in an N-node hypercube [V82]. Pippenger presented an algorithm for routing any permutation in a variant of the $\log N$-dimensional butterfly in $O(\log N)$ time, w.h.p., using constant size buffers [P84]. Leighton, Maggs, Ranade and Rao presented an algorithm for routing in an arbitrary leveled network [LMRR94] with an arbitrary path collection in time $O(C+D+\log n)$, w.h.p., using constant size buffers, and they created optimal algorithms for permutation routing in meshes, hypercubes, butterflies, and shuffle-exchange networks [LMRR94]. Hot-potato routing algorithms, in which packets are never buffered due to congestion, have been presented for two-dimensional tori and hypercubes [FR92], higher-dimensional tori [MW95], and arbitrary node-symmetric networks [MS95]. Simulations of hot-potato routing on various networks have also been performed in [AS92, GH92, M89].

In the area of universal wormhole routing, Greenberg and Oh created a randomized algorithm for arbitrary simple path collections in networks with link bandwidth 1 that requires time $O(C\cdot D\cdot \ell + C\cdot L\cdot \ell \log n)$, w.h.p., where $\ell = \min\{D, L\}$ [GO93]. This algorithm does require the ability to buffer blocked worms.

A number of results are also known for wormhole routing on specific networks. In [BRST93] Bar-Noy *et al.* present a randomized wormhole routing protocol for the m by m mesh that routes any permutation in $O(L\cdot m)$ steps, w.h.p., without buffering. Felperin, Raghavan and Upfal gave an algorithm for routing N worms, one per processor, from the top row to random destinations in the bottom row of a $\log N$-dimensional butterfly network in $O(L\log N\cdot \min\{L, \log N\})$ time, w.h.p., [FRU92]. Ranade, Schleimer and Wilkerson improved this result by showing that $q\cdot N$ worms of length $\log N$, q per processor, can be routed from the top row to random destinations in the bottom row in $O(q\log^2 N\cdot \log\log N)$ steps [RSW94]. Furthermore, they showed that $\Omega(q\log^2 N/(\log\log N)^2)$ steps are necessary for this task. Both of these results for wormhole routing on butterflies require the ability to buffer blocked worms [FRU92, RSW94].

1.2 New Results

In this section we present two theorems that have many consequences for both store-and-forward and wormhole routing.

Main Theorem 1: *Given any simple path collection in a network with bandwidth $\Theta(\log(C\cdot D)/\log\log(C\cdot D))$, there exists an on-line packet routing algorithm that requires*

$$O\left(D\log\log n + C + \frac{\log n\cdot\log\log n}{\log\log(C\cdot D)}\right)$$

time, w.h.p., without buffering.

Main Theorem 2: *Given any simple path collection in a network with bandwidth $B \le \log n$, there exists an on-line wormhole routing algorithm that requires*

$$O\left(\frac{L\cdot C\cdot D^{1/B} + (D+L)\log n}{B}\right)$$

time for worms of length L, w.h.p., without buffering.

It should be noted that, in case of routing to random destinations in some given network, the proof of Main Theorem 2 can easily be modified to show that it suffices to choose C to be the *expected congestion*, i.e. the maximum over all edges of the expected number of worms that pass an edge (rather than an upper bound on the congestion that holds w.h.p.).

In the following we present applications of the main theorems to various path collections and networks, including node-symmetric networks, edge-symmetric networks, and arbitrary bounded degree expanders.

Definition 1.1 *A graph $G=(V,E)$ is* node-symmetric *if for any pair u, v of vertices in G there exists an automorphism $\varphi : V \to V$ mapping u to v such that for the graph*

[†]Throughout the paper, the terms "*with high probability*" and "*w.h.p.*" mean "with probability at least $1-n^{-\alpha}$" where $\alpha > 0$ is an arbitrary constant.

$G_\varphi = (V, E_\varphi)$ *with* $E_\varphi = \{\{\varphi(x), \varphi(y)\} \mid \{x, y\} \in E\}$ *it holds that* $G_\varphi = G$.

Definition 1.2 *A graph* $G = (V, E)$ *is* edge-symmetric *if for any pair* e, e' *of edges in* G *there exists an automorphism* $\varphi : E \to E$ *mapping* e *to* e' *such that for the graph* $G_\varphi = (V, E_\varphi)$ *with* $E_\varphi = \{\varphi(e) \mid e \in E\}$ *it holds that* $G_\varphi = G$.

Definition 1.3 *A graph* $G = (V, E)$ *is an* expander *if for any subset* $U \subset V$ *with* $|U| \leq \alpha|V|$ *it holds that* $|\{v \in V \mid \exists u \in U : \{u, v\} \in E\}| \geq (1 + \beta)|U|$ *for some constants* $\alpha, \beta > 0$.

1.2.1 Applications to Store-and-Forward Routing

The techniques used for Main Theorems 1 and 2 yield the following results for store-and-forward routing. The algorithms are analyzed in terms of n, the number of packets being routed, C, the congestion of the path collection, D, the dilation of the path collection, and A, the buffer size. The following result is a corollary of Main Theorem 2 that is obtained by setting $L = 1$ and simulating link bandwidth with buffers.

Corollary 1.4 *Given any simple path collection, there exists an on-line store-and-forward routing algorithm that requires* $O(C \cdot D^{1/A} + D \log n)$ *time, w.h.p.*

This result is optimal if $C \geq D \log n$ and $A \geq \log D$, while the previous best bound was optimal only if $C \geq D \log n$ and $A \geq \log(nD)$ [LMR94]. Further improvements can be obtained for $C \leq D \log n \log\log(C \cdot D)/\log(C \cdot D)$ by using Main Theorem 1.

Corollary 1.5 *Given any simple path collection in a network, there exists an on-line store-and-forward routing algorithm that requires*

$$O\left(\left(D \log\log n + C + \frac{\log n \cdot \log\log n}{\log\log(C \cdot D)}\right) \cdot \frac{\log(C \cdot D)}{\log\log(C \cdot D)}\right)$$

time, w.h.p., and uses buffers of size $O(\log(C \cdot D)/\log\log(C \cdot D))$.

This result significantly improves both the buffer requirements and the running time for universal store-and-forward routing over a wide range of parameters. For example, if $C = D = \log n$, Corollary 1.5 yields a routing time of $O(\log n(\log\log n)^2/\log\log\log n)$, w.h.p., using buffers of size $O(\log\log n/\log\log\log n)$. In contrast, the fastest previously known algorithm for this problem requires $O(\log^2 n)$ time, w.h.p., and uses buffers of size $O(\log n)$ [LMR94]. Applying random walk techniques created by Broder *et al.* [BFU92], this result immediately yields the following corollary.

Corollary 1.6 *For any bounded degree expander with* n *processors,* n *packets, one per processor, can be routed on-line to random destinations in*

$$O\left(\frac{\log n(\log\log n)^2}{\log\log\log n}\right)$$

time, w.h.p., using buffers of size $O(\log\log n/\log\log\log n)$.

The following two theorems provide bounds for store-and-forward routing in arbitrary shortest path collections. In the case of shortest path collections, Theorem 1.7 provides the first nontrivial tradeoff between routing time and buffer size, and Theorem 1.8 yields stronger bounds than those given by Corollary 1.5.

Theorem 1.7 *Given any shortest path collection, there exists an on-line store-and-forward routing algorithm that requires*

$$O\left(\frac{C \cdot D^{1/A} + \log n}{A} \cdot (C + D + \log n)\right)$$

time, w.h.p.

Theorem 1.8 *Given any shortest path collection in a network, there exists an on-line store-and-forward routing algorithm that requires*

$$O\left((C + D) \cdot \frac{\log(C \cdot D)}{\log\log(C \cdot D)} + (D + \log n)\log\log n\right)$$

time, w.h.p., and uses buffers of size $O(\log(C \cdot D)/\log\log(C \cdot D))$.

If every processor in a node-symmetric network sends a packet to a random destination, then the expected congestion is bounded by the diameter [MV95]. Thus, applying Theorem 1.8 to these networks yields the following result.

Corollary 1.9 *Given any node-symmetric network of size* n *and diameter* $D = \Omega(\log n)$*, there exists an on-line store-and-forward routing algorithm that routes* n *packets, one per processor, to random destinations in*

$$O\left(D \cdot \left(\frac{\log D}{\log\log D} + \log\log n\right)\right)$$

time, w.h.p., and uses buffers of size $O(\log D/\log\log D)$.

1.2.2 Applications to Wormhole Routing

Any algorithm for routing worms of length L in a network with link bandwidth B, congestion C, and dilation D, requires at least $\Omega(L \cdot C/B + D + L)$ time. This is because there is at least one worm of length L which has to travel distance D, and there is at least one edge of bandwidth B which has to transmit $L \cdot C$ flits. Thus, the wormhole routing result of Main Theorem 2 is optimal for $B \geq \log D$ and $L \cdot C \geq (D + L)\log n$ or for $B \geq \log n$. Even for bandwidth 1, Main Theorem 2 improves upon previous best upper bounds for a wide range of parameters. For example, if $C = D = \log n$ and $B = 1$, Main Theorem 2 requires $O(\log^3 n)$ time, w.h.p., while the previous best algorithm for this problem required $O(\log^4 n)$ time, w.h.p. [GO93].

Applying Main Theorem 2 to node- and edge-symmetric networks gives the following results.

Corollary 1.10 *Given any node-symmetric network of size* n *and diameter* D*, there exists an on-line wormhole routing algorithm that routes* n *worms, one from each processor, to random destinations in*

$$O\left(\frac{L \cdot D^{1+1/B} + (D + L)\log n}{B}\right)$$

time, w.h.p., without buffering.

Corollary 1.11 *Given any edge-symmetric network of size n, diameter D, and degree Δ, there exists an on-line wormhole routing algorithm that routes n worms, one from each processor, to random destinations in*

$$O\left(\frac{L \cdot D^{1+1/B}}{\Delta B} + \frac{(D+L)\log n}{B}\right)$$

time, w.h.p., without buffering.

This is because the expected congestion in the above corollaries can be bounded by D and D/Δ, respectively [MV95, MS95]. Applying random walk techniques introduced in [BFU92] yields the following corollary.

Corollary 1.12 *Given any bounded degree expander of size n, there exists an on-line wormhole routing algorithm that routes n worms, one from each processor, to random destinations in*

$$O\left(\frac{L\log^{1+1/B} n + \log^2 n}{B}\right)$$

time, w.h.p., without buffering.

Applying Main Theorem 2 to the d-dimensional mesh with side length m (that is, with $N = m^d$ processors) yields $O(\frac{1}{B}(q \cdot L \cdot m \cdot (m \cdot d)^{1/B} + (m \cdot d + L) \cdot d \cdot \log m))$ time, w.h.p., for routing $q \cdot N$ worms, one from each processor, to random destinations. This is because the expected congestion is at most $q \cdot m$. A more careful analysis, tailored to meshes, gives the following improvement.

Theorem 1.13 *Given a d-dimensional mesh of side length m with $N = m^d$ nodes, there exists an on-line wormhole routing algorithm that routes $q \cdot N$ worms, q from each processor, to random destinations in*

$$O\left(\frac{q \cdot L \cdot m \cdot d^{1/B} + (m \cdot d + L) \cdot d \log m}{B}\right)$$

time, w.h.p., without buffering.

Theorem 1.13 is optimal for $B \geq \log d$, $q \geq d \cdot \log m/m$, and $q \cdot L \geq d^2 \log m$. Thus, this result generalizes the results obtained by Bar-Noy *et al.* for the 2-dimensional mesh [BRST93].

The next theorem presents upper and lower bounds for wormhole routing on the d-dimensional butterfly network. We consider the problem of routing worms from the inputs on level 0 to the outputs on level d such that each of the $N = 2^d$ inputs has to send q worms to randomly selected outputs. Each worm has to follow the unique shortest path from its input to its output node.

The upper bound is the same as we would obtain by applying Main Theorem 2 and the fact that the expected congestion is at most q. But whereas the algorithm used for proving Main Theorem 2 uses random ranks for priorities and assigns random delays (as in, e.g., [RSW94]), here we require neither random ranks nor delays. In particular, we give fixed, explicitly defined priorities to the worms. For bandwidth $B = 1$, the upper bound improves that given in [RSW94] by a factor of $\log d$, although our algorithm does not buffer the worms during the routing. For bandwidth $B \geq \log\log N$, $q \geq \log N$ and $q \cdot L \geq \log^2 N$, our algorithm is optimal.

The lower bound holds for any algorithm that moves the worms from the inputs to the outputs along their unique shortest paths without delaying them, even if off-line routing is allowed. It extends the lower bound from [RSW94] to arbitrary bandwidth, and shows that our upper bound is nearly optimal.

Theorem 1.14 *Let $N = 2^d$. Given a d-dimensional butterfly network without buffers, routing qN worms, q from each input, to random outputs*

a) can be done by a deterministic on-line algorithm in

$$O\left(\frac{q \cdot L \cdot \log^{1/B} N + (L + \log N) \cdot \log N}{B}\right)$$

time, w.h.p., and

b) needs

$$\Omega\left(\frac{q \cdot L \cdot \log^{1/B} N \cdot (\log\log N)^{-2/B}}{B}\right)$$

time, w.h.p., for $q \leq \log^\alpha N$ for any fixed $\alpha > 0$.

Finally, we present a result which is well suited for wormhole routing in leveled networks with short worms. In leveled networks it can easily be guaranteed that worms collide only with their heads. As a consequence, the result of Main Theorem 1 can also be applied to worms of length $L \geq 1$, which gives the following corollary.

Corollary 1.15 *Given any leveled network with depth D, there exists an on-line wormhole routing protocol requiring bandwidth $B = \Theta(\log(C \cdot D)/\log\log(C \cdot D))$, that routes n worms from the highest to the lowest level in*

$$O\left(L\left(D\log\log n + C + \frac{\log n \cdot \log\log n}{\log\log(C \cdot D)}\right)\right)$$

time, w.h.p., without buffering.

For example, when $D \geq \log n$ and $C \geq D\log\log n$, Corollary 1.15 yields a routing time of $O(L \cdot C)$, which is only a factor of $\log C/\log\log C$ away from the lower bound.

1.3 Organization of the Paper

The algorithms that yield Main Theorems 1 and 2 are presented and analyzed in Sections 2 and 3, respectively. Proofs of the remaining theorems are given in Section 4.

2 Proof of Main Theorem 1

2.1 Routing Algorithm 1

Routing Algorithm 1 routes packets (worms of length $L = 1$) using bandwidth $B = \Theta(\log(C \cdot D)/\log\log(C \cdot D))$. The idea behind this algorithm is that each packet chooses a random delay independently from the other packets. After waiting this amount of time, a packet tries to route along its prescribed path to its destination without waiting. We define the following rule in case of collisions between packets:

Contention resolution rule:
If more than B packets attempt to use the same link during the same time step, then *all* of them are discarded.

The routing algorithm consists of $k+1$ rounds (the parameters k, C_r and B will be determined later).

Routing Algorithm 1:
all n packets are declared active
for $r = 0$ through k do:

- **forward pass:** for each active packet, send 2^r copies of it with random startup delays in $[C_r] = \{0, 1, \ldots, C_r - 1\}$, route the copies according to the contention resolution rule above
- **backward pass:** successful packets become inactive via acknowledgments

Clearly, the forward pass of round r requires $O(C_r + D)$ steps. The backward pass can be organized in such a way that acknowledgments of successful copies are not delayed or discarded by running the forward pass in reverse order. A packet becomes inactive if it receives an acknowledgment of at least one of its copies.

2.2 Analysis of Algorithm 1

Definitions: A packet's *route* consists of the (at most D) edges that it must cross. A *site* is an ordered pair (e, s) where e is an edge and s is a step in the forward pass (of the round currently being considered). A *packet's sites* consist of the sites (e, s) where e is an edge on the packet's route. Given any packet p, *p's conflicting packets* consist of the packets with routes which intersect p's route (including p itself). A *packet's region* consists of the sites of its conflicting packets. A packet p *aims for* a site (e, s) if p selects a random delay such that it will be present in edge e at step s of the forward pass, provided that it is not discarded before reaching edge e. Note that whether or not a packet aims for a site is strictly a function of its starting time and its path, and is independent of the starting times of other packets. A packet p *is discarded at a site* (e, s) if p attempts to enter edge e at step s but is discarded because of edge contention. Note that if y packets are discarded at a site (e, s), then at least $y > B$ packets aim for site (e, s).

We will begin by analyzing the forward pass of round 0. We will set $C_0 = C$ and $B = 10 \ln(C \cdot D)/\ln\ln(C \cdot D)$. Consider any packet p and mark any m sites in p's region. Let the random variable X denote the number of distinct packets that aim for any of the marked sites. Note that the expected number of packets which aim for any one site is at most one (because at most C packets can have the possibility of aiming for the site, and the probability that such a packet does aim for the site is $1/C$). Therefore, $E(X) \leq m$.

Because X is the sum of independent Bernoulli trials, we can use a Hoeffding-Chernoff bound to bound the probability that X is larger than $B \cdot m$:

$$\text{Prob}(X > B \cdot m) \leq (e/B)^{B \cdot m} = e^{-m \cdot B(\ln B - 1)}.$$

Note that there are at most

$$\binom{C \cdot D^2(C+D)}{m} \leq \left(\frac{eC \cdot D^2(C+D)}{m}\right)^m \leq e^{m(3\ln(C \cdot D)+1)}$$

different ways of marking m sites in p's region. Therefore, the probability that more than $B \cdot m$ distinct packets aim for *any* set of m sites in p's region is at most

$$e^{m(3\ln(C \cdot D)+1-B(\ln B-1))}.$$

Let $m = (\alpha+2)\ln n/2\ln(C \cdot D)$. Because

$$\begin{aligned} \ln B - 1 &= \ln(10/e) + \ln\ln(C \cdot D) - \ln\ln\ln(C \cdot D) \\ &\geq (1/2)\ln\ln(C \cdot D) + 1 \ , \end{aligned}$$

we have

$$e^{m(3\ln(C \cdot D)+1-B(\ln B-1))} \leq n^{-(\alpha+2)}.$$

Note that if more than $B \cdot m$ packets are discarded at sites in p's region, then more than $B \cdot m$ distinct packets must aim for some set of m sites in p's region. (This can be argued as follows: Let z be the number of sites in p's region at which packets were discarded. If $z \leq m$, let S be a set of any m sites including these z sites. If $z > m$, let S be a set of any m of these z sites and note that more than B packets were discarded at each of them. Therefore, $|S| = m$, more than $B \cdot m$ packets were discarded at sites in S, each of these discarded packets aimed for the site at which it was discarded, and these discarded packets must be distinct because a packet can only be discarded once.) As a result, the probability that more than $B \cdot m$ packets are discarded at sites in *any* packet's region is at most $n^{-(\alpha+1)}$. We will say that round 0 succeeded iff for every packet p, at most $B \cdot m = 5(\alpha+2)\ln n/\ln\ln(C \cdot D)$ of p's conflicting packets were not successfully delivered to their destinations.

Now consider round 1. It is identical to round 0 in terms of the number of (copies of) packets that are discarded, except it has congestion at most $2B \cdot m$ (assuming that round 0 succeeded) because two copies are made of each of the at most $B \cdot m$ packets that were discarded at sites in each packet's region, and because the paths are simple, each of these packet copies can contribute at most one to the congestion of an edge. Therefore, we will set $C_1 = 2B \cdot m$ and an analysis identical to the analysis of round 0 shows that the probability that more than $B \cdot m$ packet copies are discarded at sites in *any* packet's region during round 1 is at most $n^{-(\alpha+1)}$. Because both copies of a packet must be discarded in order for the packet to fail to be delivered, it follows that the probability that there exists a packet p such that more than $B \cdot m/2$ of p's conflicting packets were not successfully delivered after round 1 is at most $n^{-(\alpha+1)}$.

In general, for any round i, $1 \leq i \leq k$, we will set $C_i = 2B \cdot m$. We will say that round i succeeded iff for every packet p, at most $B \cdot m/2^i$ of p's conflicting packets were not successfully delivered to their destinations during the first i rounds. Using an analysis identical to that given for round 1, it follows that for any round i, $1 \leq i \leq k$, if all rounds $j < i$ succeeded then the probability that round i fails is at most $n^{-(\alpha+1)}$. Setting $k = \log(B \cdot m) + 1 = O(\log\log n)$, it follows that if all $k+1$ rounds $0 \ldots k$ succeed, then every packet has been successfully delivered, and this happens with probability at least $1 - k/n^{\alpha+1} \geq 1 - n^{-\alpha}$.

Finally, note that the algorithm requires link bandwidth of $B = O(\log(C \cdot D)/\log\log(C \cdot D))$ and takes $2(C+D) + 2k(C_1 + D) = O(D\log\log n + C + \log n \log\log n/\log\log(C \cdot D))$ time, w.h.p. This completes the proof of Main Theorem 1.

3 Proof of Main Theorem 2

3.1 Routing Algorithm 2

In the following we prove Main Theorem 2. The idea of our protocol here is that each worm ω_i chooses uniformly and independently from the other worms a random rank $r_i \in [R] = \{0, \ldots, R-1\}$ (R will be specified later) and a random delay $d_i \in [D]$. We define the following rule:

> **Contention resolution rule:**
> If more than B worms attempt to use the same link during the same time step, then those B with lowest rank win. Ties in rank are broken according to an arbitrary total order on the worms. If a worm loses with its k-th flit at a link, the flits from k to L are discarded when reaching this link. The flits 1 to $k-1$ continue the routing.

Our protocol then works as follows.

> **Routing Algorithm 2:**
> all n worms are declared active and choose random $r_i \in [R]$ and $d_i \in [D]$
> **repeat**
>
> - **forward pass**: Each active worm ω_i waits for d_i steps. Then it is routed along its path, obeying the contention resolution rule above.
> - **backward pass**: For each worm that completely reached its destination during the forward pass, an acknowledgment is sent back to the source. Upon receipt of the acknowledgment, the source declares the worm inactive.
>
> **until** no worm is active

Clearly, the forward pass only needs $2D + L$ steps, since after $2D + L$ steps all worms that did not fail have completely reached their destinations. These worms are called *successful*. Whether a worm was successful can easily be detected at its destination by counting the number of flits of the worm that arrive.

In the backward pass, the forward pass is run in reverse order, except that only the heads of successful worms participate. Therefore, no collisions can occur in the backward pass, and $2D$ steps suffice to send all acknowledgments back.

3.2 Analysis of Algorithm 2

In the following we prove an upper bound for the number of rounds required by the protocol that holds w.h.p. For this, we use a delay sequence argument.

Definition 3.1 *An* (s, B)-delay sequence *consists of*

(1) $s \cdot B + 1$ *delay worms* $\omega_1, \ldots, \omega_{s \cdot B+1}$ *such that during the forward pass of round* $i \geq 1$, *worm* $\omega_{i \cdot B+1}$ *is prevented from using a link by worms* $\omega_{(i-1)B+1}, \ldots, \omega_{i \cdot B}$;

(2) $s \cdot B + 1$ *integer keys* $r_1, \ldots, r_{s \cdot B+1}$ *such that* $0 \leq r_1 \leq \ldots \leq r_{s \cdot B+1} \leq R - 1$.

We say that a delay sequence is *active* if $\text{rank}(\omega_i) = r_i$ for $1 \leq i \leq s \cdot B + 1$.

Lemma 3.2 *If the routing takes more than* s *rounds, there exists an active* (s, B)*-delay sequence.*

Proof. In the following we will give a construction for such a delay sequence. If the routing takes more than s rounds then there is a worm $\omega_{s \cdot B+1}$ that has not completely reached its destination in round s. In such a case there must have been worms $\omega_{(s-1)B+1}, \ldots, \omega_{s \cdot B}$ that prevented $\omega_{s \cdot B+1}$ from using some link in round s. According to the routing protocol, $\omega_{(s-1)B+1}, \ldots, \omega_{s \cdot B}$ can be chosen such that $\text{rank}(\omega_{(s-1)B+1}) \leq \ldots \leq \text{rank}(\omega_{s \cdot B}) \leq \text{rank}(\omega_{s \cdot B+1})$. But if $\omega_{(s-1)B+1}$ was still active in round s, there must have been worms $\omega_{(s-2)B+1}, \ldots, \omega_{(s-1)B}$ in round $s-1$ that prevented $\omega_{(s-1)B+1}$ from using some link, where $\text{rank}(\omega_{(s-2)B+1}) \leq \ldots \leq \text{rank}(\omega_{(s-1)B}) \leq \text{rank}(\omega_{(s-1)B+1})$. Continuing this argument back to round 1 yields the lemma with $r_i := \text{rank}(\omega_i)$, $1 \leq i \leq s \cdot B + 1$. ∎

Lemma 3.3 *The number of different active* (s, B)*-delay sequences is at most*

$$n \cdot (D \cdot C^B)^s \binom{s \cdot B + R}{s \cdot B + 1}$$

Proof. We count the number of possible choices for the components:

- There are at most $n \cdot (D \cdot C^B)^s$ choices for the delay worms. This is because there are at most n possibilities to determine the worm $\omega_{s \cdot B+1}$, and if $\omega_{i \cdot B+1}$ is fixed, there are at most $D \cdot C^B$ choices for the worms $\omega_{(i-1)B+1}, \ldots, \omega_{i \cdot B}$ for $1 \leq i \leq s$.
- There are $\binom{(s \cdot B+1)+R-1}{s \cdot B+1}$ possibilities to choose the r_i's such that $0 \leq r_1 \leq \cdots \leq r_{s \cdot B+1} \leq R - 1$.

Multiplying these terms yields the lemma. ∎

Lemma 3.4 *Let* $\ell = \min\{2L, D\}$, $B \leq \log n$, *and* $R \geq \max\{4e\ell \cdot C \cdot D^{-1+1/B}, (\alpha+1)\log n\}$. *Then the above routing protocol completes the routing of all* n *worms in*

$$\max\left\{\frac{4e\ell \cdot C \cdot D^{1/B}}{D \cdot B}, \frac{(\alpha+1)\log n}{B}\right\}$$

rounds with probability $1 - n^{-\alpha}$ *for every* $\alpha > 0$.

Proof. Since each worm determines a rank and chooses a random delay, and the contention resolution scheme of the above routing protocol requires that in any active delay sequence the worms have to be pairwise distinct, the probability that a particular (s, B)-delay sequence is active is at most $R^{-(s \cdot B+1)}(\ell/D)^{s \cdot B}$. Therefore, if we choose

$$T \geq \max\left\{\frac{4e\ell \cdot C \cdot D^{1/B}}{B \cdot D}, \frac{(\alpha+1)\log n}{B}\right\}$$

and $R = T \cdot B$ then

$$\begin{aligned}
&\text{Prob(the routing takes more than } T \text{ rounds)} \\
&\overset{\text{Lemma 3.2}}{\leq} \text{Prob (a } (T, B)\text{-delay sequence is active)} \\
&\overset{\text{Lemma 3.3}}{\leq} n(D \cdot C^B)^T \binom{T \cdot B + R}{T \cdot B + 1} R^{-(T \cdot B+1)} \left(\frac{\ell}{D}\right)^{T \cdot B} \\
&\leq n \left(\frac{\ell \cdot C \cdot D^{1/B}}{D}\right)^{T \cdot B} \left(\frac{e(T \cdot B + R)}{R(T \cdot B + 1)}\right)^{T \cdot B+1}
\end{aligned}$$

$$\leq \quad n\left(\frac{e\ell \cdot C \cdot D^{1/B}(T \cdot B + R)}{T \cdot B \cdot D \cdot R}\right)^{T \cdot B}$$

$$\overset{R=T\cdot B}{=} \quad n\left(\frac{2e\ell \cdot C \cdot D^{1/B}}{T \cdot B \cdot D}\right)^{T \cdot B}$$

$$\overset{T \geq \frac{4e\ell \cdot C \cdot D^{1/B}}{B \cdot D}}{\leq} \quad n2^{-T \cdot B} \quad \overset{T \geq \frac{(\alpha+1)\log n}{B}}{\leq} \quad n^{-\alpha} \ .$$

■

This yields Main Theorem 2.

4 Proofs of Theorems in Section 1

4.1 Proof of Theorem 1.7

Consider routing n packets along a shortest path collection with congestion C and dilation D in a network with buffer size A. Each packet P_i gets two random ranks: the first rank r_i^d is chosen as in Section 3, and the second rank r_i^g is chosen as in the growing rank protocol in [MV95]. The new protocol then works as follows

for each packet P_i, choose random ranks r_i^d and r_i^g,
all n packets are declared active
repeat

- **forward pass:** route packets according to the growing rank protocol in [MV95], in case of overfull buffers use the contention resolution rule of Section 3
- **backward pass:** successful packets become inactive via acknowledgments

until no packet is active

¿From [MV95] we know that the growing rank protocol requires at most $O(C + D + \log n)$ time, w.h.p. Using the proof of Main Theorem 2 it can be shown that $O(\frac{1}{B}(C \cdot D^{1/A} + \log n))$ rounds suffice to route all packets, w.h.p. This completes the proof of Theorem 1.7.

4.2 Proof of Theorem 1.8

Routing Algorithm 1 can be modified to yield improved performance for arbitrary shortest path collections. First, it is adapted to create a store-and-forward routing algorithm that satisfies the conditions of Corollary 1.5. In particular, each step of the wormhole routing algorithm with bandwidth B now requires B time units and uses buffers of size B. The routing occurs in *stages*, each of which corresponds to a step in the wormhole routing algorithm and requires B time units. We redefine the notion of a *site* to be an ordered pair consisting of an edge and a stage.

Next, the random delays are selected from different ranges. Specifically, $C_0 = C$ and for $1 \leq r \leq k$, $C_r = 240e^2(\alpha + 2)\log n$. Finally, the lengths of the stages (and thus the sizes of the buffers being used) is changed after round 0. Specifically, the algorithm uses stages of length $B = 10\ln(C \cdot D)/\ln\ln(C \cdot D)$ in round 0, and of length $B' = 1$ in all other rounds. The analysis of round 0 is identical to that of Algorithm 1, so it follows that after round 0 the probability that any packet has more than $B \cdot m$ conflicting packets that were not successfully delivered is at most $n^{-(\alpha+1)}$.

Now consider round 1. Note that during round 1 each active packet has at most $2B \cdot m$ conflicting packet copies (assuming that round 0 succeeded) because two copies are made of each of the at most $B \cdot m$ packets that were discarded at sites in that active packet's region. Let $q = C_1 = 240e^2(\alpha + 2)\log n$, $y' = 10(\alpha + 2)\log n$, and $y = 8\lceil y'/8 \rceil$. Consider any active packet copy in round 1 and let Y be the set of its at most $2B \cdot m \leq y$ conflicting packet copies. Let U' be the set of packet copies in Y that are discarded in round 1 at a site at which at least one other packet copy in Y is discarded. Let V be the set of packet copies in Y that are discarded in round 1 at a site at which at least one packet copy not in Y is discarded. Let $u' = |U'|$ and $v = |V|$ and note that at most $u' + v$ packet copies in Y are discarded in round 1.

We will first bound the probability that $u' \geq y/4$. We will require the following definitions.

Definitions: Two packet copies *collide* if they both attempt to pass through the same link during the same stage. A (u, s)-*configuration* consists of:

- a set $U \subseteq U'$ where $|U| = u$,
- a partition of U into s nonempty, disjoint subsets called *groups*,
- a designation of one packet copy per group as the *representative* of the group, and
- an assignment of a random delay to each packet copy in U.

A (u, s)-configuration is *active* if, given the assigned random delays, every nonrepresentative packet copy collides with its representative.

Note that if $u' \geq y/4$, there must exist a set of packets copies $U \subseteq U'$, where $|U| = u = y/4$, and a set of sites S, where $1 \leq |S| \leq u/2$, such that every packet copy in U was discarded at one of the sites in S. As a result, it is possible to construct a (u, s)-configuration, where $1 \leq s \leq u/2$, by grouping the packet copies in U according to the sites at which they were discarded, arbitrarily selecting a representative from each group, and recording the random delays. Furthermore, note that this (u, s)-configuration must be active, as all of the packet copies within each group were discarded at the same site. Therefore, if $u' \geq y/4$, there must exist an active (u, s)-configuration where $u = y/4$ and $1 \leq s \leq u/2$. We will now bound the probability that such an active configuration exists.

Note that there are at most $\binom{y}{s}$ different ways of selecting the representatives, and once the representatives have been chosen, there are at most $\binom{y}{u-s}s^{u-s}$ different ways of selecting the nonrepresentatives and grouping them with representatives.

Also, note that in any shortest path collection, if packet copies p_1 and p_2 collide when they have random delays d_1 and d_2, respectively, then they will not collide when they have delays d_1 and d_3, respectively, for any $d_3 \neq d_2$. Therefore, once we have selected the random delays of the representatives, there is at most one way to set the random delays of the nonrepresentatives in order to make the configuration active. Thus there are a total of at most

$$\binom{y}{s}\binom{y}{u-s}s^{u-s}q^s \le \left(\frac{ey}{s}\right)^s \left(\frac{eys}{u-s}\right)^{u-s} q^s$$

active (u,s)-configurations. Finally, the probability that a given (u,s)-configuration occurs is $(1/q)^u$. Therefore, the probability that there exists an active (u,s)-configuration, given any fixed values of the parameters u and s where $1 \le s \le u/2$, is at most

$$\begin{aligned}\left(\frac{ey}{s}\right)^s \left(\frac{eys}{u-s}\right)^{u-s} \left(\frac{1}{q}\right)^{u-s} &\le \left(\frac{ey}{s}\right)^s \left(\frac{eys}{q(u-s)}\right)^{u-s} \\ &\le \left(\frac{ey}{s}\right)^s \left(\frac{2eys}{qu}\right)^{u-s} \\ &= \left(\frac{4eu}{s}\right)^s \left(\frac{8es}{q}\right)^{u-s}.\end{aligned}$$

Let $f(s) = (4eu/s)^s(8es/q)^{u-s}$ and $g(s) = \ln f(s)$. Note that $g(s) = s\ln(4eu/s) + (u-s)\ln(8es/q)$ and $g'(s) = \ln(qu/2s^2) + u/s - 2$. Because $q > e^2 s$, $g'(s) > \ln(q/4s) \ge 0$ and $g(s)$ and $f(s)$ are maximized when $s = u/2 = y/8$. As a result, the probability that there exists an active (u,s)-configuration, where $u = y/2$ and $1 \le s \le u/2$, is at most

$$\sum_{s=1}^{u/2} f(\tfrac{u}{2}) = \left(\frac{u}{2}\right)(8e)^{u/2}\left(\frac{4eu}{q}\right)^{u/2} = \left(\frac{u}{2}\right)\left(\frac{32e^2u}{q}\right)^{u/2}.$$

Because $q \ge 32e^3u$, it follows that $f(u/2) \le (1/e)^{u/2} \le (1/n)^{\alpha+2}$ and that the probability that $u' \ge y/4$ is at most $(y/8)(1/n)^{\alpha+2}$.

Now we will bound the probability that $v \ge y/4$. Select an arbitrary packet copy $p \in Y$. Packet copy p has at most y conflicting packet copies that are not in Y, and for each of these conflicting packet copies the probability that p will collide with it is at most $1/q$ (because regardless of the random delay selected by the conflicting packet copy, there is at most one random delay that p can select to cause such a collision). Therefore, the probability that p is in V is at most y/q. We can view the selection of a random delay for each packet copy in Y as being a Bernoulli trial with probability of success at most y/q (where success is defined as being in the set V), and these trials are independent. Therefore, the expected number of successes is at most y^2/q, and the probability of at least $y/4$ successes is at most $(4ey/q)^{y/4}$. Because $q \ge 24e^2y$, the probability that $v \ge y/4$ is at most $(1/6e)^{y/4} \le (1/n)^{10(\alpha+2)}$.

Combining these results, it follows that the probability that more than $y/2$ packet copies in Y are discarded in round 1 is at most $(1/n)^{\alpha+1}$. The remaining analysis is analogous to that given in Section 2.

4.3 Proof of Theorem 1.13

Let us choose the following routing strategy on a d-dimensional mesh with side length m. Given a random destination chosen independently and uniformly from the set of nodes, each worm chooses a random startup delay in $[d \cdot m]$ and afterwards is first routed according dimension 1, then according to dimension 2 and so on, until it reaches its destination.

In the following, A denotes an arbitrary linear array of length m in one dimension of the mesh. Let $E(A)$ be the expected number of worms that want to use links of A. Because of symmetry reasons, for uniformly and independently chosen destinations, $E(A)$ is the same for each linear array A. (Note that this argument is not true if we considered links instead of linear arrays.) Since the total number of linear arrays used by worms is at most $q \cdot N \cdot d$ and the number of linear arrays is $d \cdot m^{d-1}$, $E(A)$ is at most $q \cdot N \cdot d/(d \cdot m^{d-1}) = q \cdot m$. Replacing the edge congestion by this array congestion in the proof of Main Theorem 2 yields the upper bound in Theorem 1.13.

4.4 Proof of Theorem 1.14

4.4.1 Proof of Part (a)

Suppose the qN worms, q per processor, have to be sent to random destinations. Let $\ell := \min\{L, \log N\}$. Then the worms can be given delays such that the routing can be viewed as divided into $\lfloor \log N/\ell \rfloor$ independent passes in which every input has to send out at most $q' = \lceil q/\lfloor \log N/\ell \rfloor \rceil$ worms. Therfore, we have to show that there exists a rank allocation such that each pass takes $O(\frac{1}{B}(q' \cdot \log^{1/B} N + \log N))$ rounds of time $2\log N + L - 1$, w.h.p. In the following we do not only show the existence but also describe a specific allocation that ensures the above routing time.

The ith worm $(0 \le i \le q')$ starting at node $(x_{\log N-1}, \ldots, x_0)$ on level 0 is assigned rank $i \cdot N + (x_0, \ldots, x_{\log N-1})_2$. Thus, the rank of a worm is essentially the bit-reversal number of its input node plus an offset. In the following we identify a worm with its rank. (Note that the ranks of all worms participating in the same phase are distinct.)

For an integer $m \ge 1$ with binary representation $(\ldots, m_2, m_1, m_0)$, we define

$$\delta(m) := \max\{i \le \log N \mid (m_{i-1}, \ldots, m_0) = 0^i\}.$$

Suppose $\omega_1 < \ldots < \omega_{B+1}$ are $B+1$ worms. Then there are only $N \cdot \left(\lfloor 2^{\min\{\delta(\omega_2-\omega_1),\ldots,\delta(\omega_{B+1}-\omega_1)\}-1} \rfloor\right)^B$ ways to choose the destinations of the worms such that all $B+1$ routing paths share an edge, i.e. the worms can not reach their destination altogether in the same round. This is because any two worms ω and ω' can not use the same edge before level $\log N - \delta(|\omega' - \omega|)$.

Now, consider an arbitrary worm ω_1 with some fixed destination, and let m be an arbitrary integer. Then the number of possibilities to choose B worms $\omega_2, \ldots, \omega_{B+1}$ and their destinations such that $\omega_1 < \omega_2 < \ldots < \omega_B < \omega_{B+1} = \omega_1 + m$ and such that all $B+1$ routing paths share an edge is at most

$$\begin{aligned}&\sum_{\substack{\omega_2,\ldots,\omega_B \\ \omega_1<\omega_2<\ldots<\omega_1+m}} 2^{(\min\{\delta(\omega_2-\omega_1),\ldots,\delta(\omega_B-\omega_1),\delta(m)\}-1)\cdot B} \\ &\le \binom{m}{B-1}\sum_{i=0}^{\delta(m)-1} 2^{-i(B-1)} \cdot 2^{(i-1)B} + \\ &\quad \sum_{i=\delta(m)}^{\infty} 2^{-i(B-1)} \cdot 2^{(\delta(m)-1)\cdot B} \\ &\le \binom{m}{B-1} 2^{\delta(m)}. \qquad (1)\end{aligned}$$

Now we are able to estimate the number of (s,B)-delay sequences. For this purpose we have to count the number of ways to choose $s\cdot B+1$ worms $\omega_1,\ldots,\omega_{s\cdot B+1}$ and their destinations such that $0\le\omega_1<\omega_2<\ldots<\omega_{s\cdot B+1}<q'N$ and such that for every $1\le i\le s$, the routing paths of the worms $\omega_{i\cdot B+1},\omega_{i\cdot B+2},\ldots,\omega_{(i+1)\cdot B+1}$ share an edge. Of course, there are at most $q'N^2$ possibilities to choose ω_1 and its destination. Define $R:=q'N$, and let $m_i:=\omega_{i\cdot B+1}-\omega_{(i-1)\cdot B+1}$. Then it follows by repeatedly applying inequation 1 that the number of (s,B)-delay sequences is at most

$$\begin{aligned}
& q'N^2\cdot \sum_{\substack{m_1,m_2,\ldots,m_s\in\mathbb{N}_+\\ \sum m_i\le R}} \prod_{i=1}^{s}\binom{m_i}{B-1}\cdot 2^{\delta(m_i)}\\
\le\ & q'N^2\cdot\binom{R}{s(B-1)}\underbrace{\sum_{\substack{m_1,\ldots,m_s\in\mathbb{N}_+\\ \sum m_i\le R}}\prod_{i=1}^{s}2^{\delta(m_i)}}_{=:A_{s,R}}
\end{aligned}$$

We now show by induction on s that $A_{s,R}\le\log^s N\cdot\binom{R}{s}$. For $s=0$, this inequality trivially holds. For $s\ge 1$, we have

$$\begin{aligned}
A_{s,R} &= \sum_{m_s=1}^{R-1}2^{\delta(m_s)}\sum_{\substack{m_1,m_2,\ldots,m_{s-1}\in\mathbb{N}_+\\ \sum m_i\le R-m_s-1}}\prod_{i=1}^{s-1}2^{\delta(m_i)}\\
&= \sum_{m=1}^{R-1}2^{\delta(m)}\cdot A_{s-1,R-m}\\
&\le \sum_{m=1}^{R-1}2^{\delta(m)}\cdot\log^{s-1}N\cdot\binom{R-m}{s-1}\le\ldots\\
&\le \log^s N\cdot\sum_{m=1}^{R-1}\binom{R-m}{s-1}\\
&= \log^s N\cdot\binom{R}{s}\ .
\end{aligned}$$

Therefore, the probability that a worm is still active after

$$\begin{aligned}
T &\ge \frac{2e\cdot q'\cdot\log^{1/B}N+(\alpha+2)\log(q'N)}{B}\\
&= O(\frac{q'\cdot\log^{1/B}N+\log N}{B})
\end{aligned}$$

rounds can be bounded by

$$\begin{aligned}
& q'N^2\cdot\binom{R}{T(B-1)}\cdot\log^T N\cdot\binom{R}{T}\cdot N^{-TB}\\
\le\ & q'N^2\cdot\left(\frac{2e\cdot q'N\cdot\log^{1/B}N}{TB\cdot N}\right)^{TB}\\
\le\ & (q'\cdot N)^{-\alpha}\ .
\end{aligned}$$

This yields part (a) of Theorem 1.14.

4.4.2 Proof of Part (b)

Consider $m\ge 7BN/\log N$ arbitrary worms. We first bound the probability P that all these worms can be routed simultaneously, i.e. they can arrive at their random destinations in L consecutive time steps. In the following we say the worms *collide* if more than B of them want to travel through the same edge.

In order to cope with dependencies between the collision probabilities on different levels we use an idea from Ranade *et al.* [RSW94] which is to divide the butterfly into many small subbutterflies. Let $k:=\lceil\log\log N\rceil$ and $K=2^k$. Then the first k levels consists of N/K small distinct butterflies. We denote the number of worms starting in the ith of these butterflies by m_i.

We first estimate the probability P_i that the m worms do not collide in one of the $2K$ edges on level $k+1$ incident to the K output nodes of the ith small butterfly, under the assumption $B+1\le m_i\le 2KB$. Let $e_1,\ldots,e_{2K}$ denote these edges on level $k+1$. Further, let E_j denote the event that more than B worms want to travel along the edge e_j, and let E'_j denote the event that exactly $B+1$ worms want to travel along this edge. Then

$$\begin{aligned}
\mathrm{Prob}(E_j) &\ge \mathrm{Prob}(E'_j)\\
&= \binom{m_i}{B+1}\left(\frac{1}{2K}\right)^{B+1}\left(1-\frac{1}{2K}\right)^{m_i-B-1}\\
&\overset{B+1\le m_i}{\ge} \left(\frac{m_i}{2K(B+1)}\right)^{B+1}3^{-(m_i-B-1)/2K}\\
&\overset{m_i\le 2KB}{\ge} \left(\frac{m_i}{2K(B+1)}\right)^{B+1}\cdot 3^{-B}\ ,
\end{aligned}$$

and therefore,

$$\begin{aligned}
P_i &\le \mathrm{Prob}\left(\bigwedge_{j=1}^{2K}\neg E_j\right)\\
&\le \prod_{j=1}^{2K}\mathrm{Prob}\left(\neg E_j\,\middle|\,\bigwedge_{\ell=1}^{j-1}\neg E_\ell\right)\le\prod_{j=1}^{2K}\mathrm{Prob}(\neg E_j)\\
&\le \left(1-\left(\frac{m_i}{2K(B+1)}\right)^{B+1}\cdot 3^{-B}\right)^{2K}\\
&\le \exp\left(-m_i^{B+1}\cdot(B+1)^{-(B+1)}\cdot(6K)^{-B}\right)\ .
\end{aligned}$$

Now we can bound the probability P that the m worms do not collide in level $k+1$. If there is at least one small butterfly in which more than $2KB$ worms start, then it is $P=0$. Otherwise,

$$\begin{aligned}
P &\le \prod_{\substack{i=1\\ m_i\ge B+1}}^{N/K}P_i\\
&\le \exp\left(-\left(\sum_{\substack{i=1\\ m_i\ge B+1}}^{N/K}m_i^{B+1}\right)\cdot(B+1)^{-(B+1)}\cdot(6K)^{-B}\right).
\end{aligned}$$

Let ℓ denote the number of small butterflies in which more

than $B+1$ worms start, and define

$$m' := \sum_{\substack{i=1 \\ m_i \geq B+1}}^{N/K} m_i \geq m - \frac{BN}{K} \geq \frac{6}{7}m \ .$$

Then it is

$$\sum_{\substack{i=1 \\ m_i \geq B+1}}^{N/K} m_i^{B+1} \geq \sum_{\substack{i=1 \\ m_i \geq B+1}}^{N/K} \left(\frac{m'}{\ell}\right)^{B+1} \geq \ell \left(\frac{6m}{7\ell}\right)^{B+1} .$$

Thus, we can conclude

$$\begin{aligned} P &\leq \exp\left(-\ell\left(\frac{6m}{7\ell}\right)^{B+1} \cdot (B+1)^{-(B+1)} \cdot (6K)^{-B}\right) \\ &\overset{\ell \leq N/K}{\leq} \exp\left(-(6/7)\cdot(7N)^{-B} \cdot m^{B+1} \cdot (B+1)^{-(B+1)}\right) \ . \end{aligned}$$

We can repeat this argument to obtain upper bounds for the probability of having no collisions in the levels $2k+2$, $3k+3$, $4k+4$ and so on. Since we obtain all these bounds by using only information regarding the k levels in front of the respective collision level, we may multiply these probabilities to obtain an upper bound on the probability that the worms do not collide on their way through the whole butterfly. Thus,

$$\begin{aligned} &\left(\exp\left(-(6/7)\cdot(7N)^{-B} \cdot m^{B+1} \cdot (B+1)^{-(B+1)}\right)\right)^{\left\lfloor \frac{\log N}{k+1} \right\rfloor} \\ &\quad \leq \exp\left(-\frac{\log N \cdot m^{B+1}}{3\log\log N \cdot (7N)^B \cdot (B+1)^{B+1}}\right) . \end{aligned}$$

Therefore, the probability that there exists a set of

$$m \geq 7N \cdot \left(\frac{3(\alpha+2)\log^2 \log N}{\log N} \cdot (B+1)^{B+1}\right)^{1/B}$$

worms that can be routed simultaneously is

$$\begin{aligned} &\binom{N\log^\alpha N}{m} \cdot \exp\left(-\frac{\log N \cdot m^{B+1}}{3\log\log N \cdot (7N)^B \cdot (B+1)^{B+1}}\right) \\ &\quad \leq \exp\left(m\left((\alpha+1)\log\log N - \right.\right. \\ &\qquad\qquad \left.\left.\frac{\log N \cdot m^{B+1}}{3\log\log N \cdot (7N)^B \cdot (B+1)^{B+1}}\right)\right) \\ &\quad \leq \exp(-m) \ . \end{aligned}$$

As a consequence, the routing takes at least time

$$\frac{LN\log^\alpha N}{m} = \Omega\left(\left(\frac{\log N}{\log^2 \log N}\right)^{1/B} \frac{L(\log N)^\alpha}{B}\right)$$

with probability $\exp(-m)$. This proves part (b) of Theorem 1.14.

References

[AS92] A. Acampora, S. Shah. Multihop Lightwave Networks: a Comparison of Store-and-Forward and Hot-Potato Routing. *IEEE Transaction on Communications*, pp. 1082-1090, 1992.

[BFU92] A. Broder, A. Frieze, E. Upfal. Existence and Construction of Edge Disjoint Paths on Expander Graphs. In *Proc. of the 24th Symp. on Theory of Computing*, pp. 140-149, 1992.

[BRST93] A. Bar-Noy, P. Raghavan, B. Schieber, H. Tamaki. Fast Deflection Routing for Packets and Worms. In *Proc. of the 12th Symp. on Principles of Distributed Computing*, pp. 75-86, 1993.

[D90] W. Dally. Virtual-Channel Flow Control. In *Proc. of the 17th Intl. Symp. on Computer Architecture*, pp. 60-68, 1990.

[DS87] W. Dally, C. Seitz. Deadlock-Free Message Routing in Multicomputer Networks Using Virtual Channels. *IEEE Trans. on Computers* 36, pp. 547-553, 1987.

[FR92] U. Feige, P. Raghavan. Exact Analysis of Hot-Potato Routing. In *Proc. of the 33rd Symp. on Foundations of Computer Science*, pp. 553-562, 1992.

[FRU92] S. Felperin, P. Raghavan, E. Upfal. A Theory of Wormhole Routing in Parallel Computers. In *Proc. of the 33rd Symp. on Foundations of Computer Science*, 1992.

[GH92] A. Greenberg, B. Hajek. Deflection Routing in Hypercube Networks. *IEEE Transactions on Communications*, pp. 1070-1081, 1992.

[GO93] R. Greenberg, H.C. Oh. Universal Wormhole Routing. Technical Report, University of Maryland, 1993.

[L92] T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, San Mateo, CA, 1992.

[LMR94] T. Leighton, B. Maggs, S. Rao. Packet Routing and Job-Shop Scheduling in O(Congestion + Dilation) Steps. *Combinatorica* 14, pp. 167-186, 1994.

[LMRR94] T. Leighton, B. Maggs, A. Ranade, S. Rao. Randomized Routing and Sorting on Fixed-Connection Networks. *Journal of Algorithms* 17, pp. 157-205, 1994.

[M89] N. Maxemchuk. Comparison of Deflection and Store-and-Forward Techniques in the Manhattan Street and Shuffle-Exchange Networks. In *Proc. of the IEEE INFOCOM*, pp. 800-809, 1989.

[MS95] F. Meyer auf der Heide, C. Scheideler. Routing with Bounded Buffers and Hot-Potato Routing in Vertex-Symmetric Networks. In *3rd European Symposium on Algorithms*, pp. 341-354, 1995.

[MV95] F. Meyer auf der Heide, B. Vöcking. A Packet Routing Protocol for Arbitrary Networks. In *Proc. of the 12th Symp. on Theoretical Aspects of Computer Science*, pp. 291-302, 1995.

[MW95] F. Meyer auf der Heide, M. Westermann. Hot-Potato Routing on Multi-Dimensional Tori. To appear in *Int. Workshop on Distributed Algorithms*, 1995.

[P84] N. Pippenger. Parallel Communication with Limited Buffers. In *Proc. of the 25th Symp. on Foundations of Computer Science*, pp. 127-136, 1984.

[RSW94] A. Ranade, S. Schleimer, D.S. Wilkerson. Nearly Tight Bounds for Wormhole Routing. In *Proc. of the 35th Symp. on Foundations of Computer Science*, pp. 347-355, 1994.

[SM93] M. Settembre, F. Matera. All Optical Implementations of High Capacity TDMA Networks. *Fiber and Integrated Optics* 12, pp. 173-186, 1993.

[V82] L.G. Valiant. A Scheme for Fast Parallel Communication. *SIAM Journal of Computing* 11/2, pp. 350-361, 1982.