Space-Efficient Routing in Vertex-Symmetric Networks*

(Technical Report)

Friedhelm Meyer auf der Heide and Christian Scheideler
Department of Mathematics and Computer Science
and Heinz Nixdorf Institute, University of Paderborn
33095 Paderborn, Germany
email: {fmadh,chrsch}@uni-paderborn.de

Abstract

In this paper we prove an upper bound for the trade-off between routing time and space needed in vertex-symmetric networks to store routing information in the processors and the packets. In particular, we prove that for any vertex-symmetric network with n vertices, degree d, and diameter $D = \Omega(\log n)$ it holds for all $s \in [2, n]$:

A randomly chosen function and any permutation can be routed in time $O(\log_s n \cdot D)$, with high probability, if $O(s \cdot D \cdot \log d)$ space is available at each processor and

(1) $O(D \cdot \log d)$ space is available for storing routing information in each packet (this suffices to route to arbitrary destinations), or

(2) $O(\log(s \cdot D))$ space is available for storing routing information in each packet (this suffices to route to randomly chosen destinations, w.h.p.).

E.g., for arbitrary bounded degree vertex-symmetric networks with diameter $O(\log n)$ (among them expanders) this result shows: routing time $O(\log n)$ can be achieved already if $O(n^\epsilon)$ space is available in each vertex, $\epsilon > 0$ arbitrary. If we allow $O(\log^2 n)$ routing time, space can be reduced to $O(\log n)$.

This is the first result that relates space to routing time; previous approaches only consider space and dilation, ignoring congestion and the design of routing protocols.

1 Introduction

The communication cost among the processors of a parallel system is usually measured by two parameters: the time and the routing space necessary to route all packets from any point to any point of the system. Whereas much is known about the runtime for all kinds of networks under the condition that enough space is available (see, e.g., [MV95]), little is known about how space-efficiency can influence the runtime. But space-efficiency will be important for large parallel systems to keep the price of the routing hardware low. Moreover, the design of the routing hardware for these systems should be independent of the topology of the network to be realized. On the other hand, the communication among the processors usually requires a large portion of the runtime of a parallel algorithm. Therefore, designing routing hardware and routing protocols that find an optimal trade-off between routing time and space is an important task in parallel systems.

* Supported in part by DFG-Forschergruppe "Effiziente Nutzung massiv paralleler Systeme, Teilprojekt 4", by Volkswagen Foundation and by the Esprit Basic Research Action Nr 7141 (ALCOM II)

In this paper we prove an upper bound for the trade-off between space and routing time that holds for all vertex-symmetric networks with diameter $\Omega(\log n)$. This result is a consequence of a new strategy for the simulation between arbitrary networks. The simulation strategy and its analysis is inspired by the routing protocol for arbitrary networks presented in [MV95]. We will apply our techniques for the simulation of networks to demonstrate space-efficient routing strategies for vertex-symmetric networks.

1.1 Space-Efficient Routing

The *routing network* is represented by a connected graph $H = (V, E)$, where $V = [n](= \{0, \ldots, n-1\})$ is the set of all vertices (or processors) and $E \subset V \times V$ is the set of all edges (or links) in H. Each $\{v, w\} \in E$ consists of two links, one in each direction. Each link entering a vertex v contains an input buffer that is able to store packets.

We only consider *oblivious* routing strategies, i.e., a packet with origin u and destination v has to travel along a prescribed *routing path* $p(u, w)$ in H. The set of these paths for all $\binom{n}{2}$ pairs (u, v) of vertices in H is called a *path system* and denoted by $\mathcal{P}$. A *shortest path system* contains only paths $p(u, v)$ that are shortest paths from u to v in H.

A *packet* consists of a *source* $v \in V$, a *destination* $w \in V$, additional *routing information* and a *message*. The source and destination need $\log n$ bits, each. Throughout this paper we restrict the routing information to be very small, namely of length at most $O(\log n)$. We assume the messages to have uniform length.

Given a path system $\mathcal{P}$ in H, a *routing protocol* consists of a *contention resolution protocol* and a *routing structure* for each vertex v in H.

The contention resolution protocol chooses a packet from those currently stored in v's input buffers. The choice depends on the source, destination and routing information of these packets. Our contention resolution protocol works with $O(\text{degree of } H)$ operations, each on $\log n$-bit words.

The edge along which a packet has to be sent is determined with the help of a *routing structure* stored in v. This is a (static) data structure that, given the destination and the routing information of a packet, enables v to compute the next edge the packet has to use w.r.t. its path prescribed in $\mathcal{P}$, and (maybe) update the packet's routing information. We demand that this access needs constant time, i.e. a constant number of operations on $\log n$-bit words.

The routing protocol used in this paper proceeds in rounds. Initially, every vertex $v \in V$ has one packet. A function $f : V \to V$ assigns a destination vertex to each packet. The set of all functions is denoted by $\mathcal{F}$. In a round, each vertex v chooses a packet from one of its input buffers with the help of its contention resolution protocol, computes the next edge it has to go by accessing its routing structure, and sends it along the respective edge.

Clearly, the following parameters greatly influence the time needed to route an h-function f in H:

- the *dilation* D of $\mathcal{P}$, that is, the length of the longest path in $\mathcal{P}$, and
- the *congestion* C_f, i.e. the maximum number of routing paths $p(u, f(u, i))$ in $\mathcal{P}$ that pass through the same vertex in H.

Note that $C_f \cdot D$ rounds suffice to route f. This upper bound follows from the facts that a packet is delayed at most C_f times at any vertex and that the length of its path is at most D. On the other hand, if there is at least one vertex that transmits C_f packets and one packet that traverses $\Omega(D)$ vertices in H, the routing takes $\Omega(D + C_f)$ time.

1.2 Routing Networks

In this paper we mainly deal with space-efficient routing in vertex-symmetric networks. This class is defined as follows.

Definition 1.1 *A network* $H = (V, E)$ *is called* vertex-symmetric *if for any pair* u, v *of vertices in* H *there exists an automorphism* $\varphi : V \to V$ *mapping* u *to* v *such that for the graph* $H_\varphi = (V, E_\varphi)$ *with* $E_\varphi = \{\{\varphi(x), \varphi(y)\} \mid \{x, y\} \in E\}$ *it holds* $H_\varphi = H$.

Vertex-symmetric networks form a very general class and include most of the standard networks such as the d-dimensional torus, the butterfly, the hypercube, etc.. Furthermore, the best expanders that have an explicit construction are all Cayley graphs and therefore vertex-symmetric (see, e.g., [LPS88], [M88] or [M94]).

Besides the notion of vertex-symmetric networks we need in our proofs the notion of s-ary Butterfly networks. This class is defined as follows.

Definition 1.2 *The* s*-ary* d*-dimensional Butterfly network* (s, d)*-BF is an undirected graph* $G = (V, E)$ *with vertex set*

$$V = \{(l, x) \mid l \in [d],\ x = (x_{d-1}, \ldots, x_0) \in [s]^d\}$$

and edge set

$$E = \bigcup_{i=0}^{s-1} \{\{(l, x), ((l+1) \bmod d, f(x, l, i))\} \mid (l, x) \in V\}$$

where $f(x, l, i)$ *is defined as*

$$f(x, l, i) = (x_{d-1}, \ldots, x_{l+1}, i, x_{l-1}, \ldots, x_0)$$

For $k \in \{1, \ldots, s\}$ *let us call the subgraph* $G_k = (V_k, E_k)$ *of an* (s, d)*-BF with vertex set*

$$V_k = \{(l, x) \mid l \in [d],\ x \in [k] \times [s]^{d-1}\}$$

and edge set $E_k = E|_{V_k \times V_k}$ *the* (s, d, k)*-BF.*

The following example will clarify how a $(3, 2, 2)$-BF is located in a $(3, 2)$-BF (the vertices in the highest and lowest level are the same).

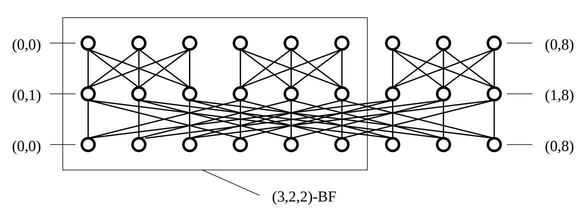

Figure 1: A $(3, 2, 2)$-BF in a $(3, 2)$-BF

Note that the $(n, 1)$-BF is the complete graph consisting of n vertices.

The goal of this paper is to show that for any vertex-symmetric network H with diameter $\Omega(\log n)$ and $s \in \{2, n\}$ there is a suitable routing protocol that can efficiently simulate a suitably chosen (s, d, k)-BF in H using little space for the routing structures in the vertices and routing information in the packets.

1.3 Previous Results

If no restrictions are imposed on the routing space then, according to [MV95], it holds for arbitrary networks with diameter D that any h-function f with congestion C_f can be routed in time $O(D + C_f + \log n)$, w.h.p.. Their results can be used to prove that, for all vertex-symmetric networks with diameter D, a randomly chosen h-function can be routed in time $O(h \cdot D + \log n)$, w.h.p., if space $O(n \cdot d)$ in each vertex and routing information of length $O(\log n)$ in each packet is available. (By 'w.h.p.' we mean a probability of at least $1 - \frac{1}{n^\alpha}$ for every constant $\alpha > 0$.)

The most commonly used strategies for space-efficient routing are interval routing and hierarchical routing.

The interval routing protocol works as follows: every outgoing link e of a vertex v with $id(v) \in [n]$ is attached by intervals of id's of vertices, e.g. $[i_1, i_2]$, telling v that whenever it has a packet that has to be sent to a destination vertex with id i such that $i_1 \leq i \leq i_2$, v has to send the packet along link e. Cleary, if each vertex is allowed to have k intervals the routing protocol requires only a space of $O(k \log n)$ for every vertex of the network. Analyses of the interval routing protocol and generalizations of it can be found, e.g., in [FJ88] and [FGS93]. In [B93] a lower bound can be found for the number of intervals necessary to obtain optimal interval routing for arbitrary networks of degree 3, namely $\Omega(\frac{n}{\log^2 n})$. Thus a dilation O(diameter) can only be achieved using space $\Omega(\frac{n}{\log n})$. No analysis is known so far for the routing time of interval routing in arbitrary vertex-symmetric networks.

Space-efficient hierarchical routing schemes can be found, e.g., in [FJ90], [PU89] and [ABLP90]. These papers analyze the relationship between the routing space and the stretch factor for a class of so-called c-decomposable graphs (see [FJ90]) or arbitrary graphs (see [PU89], [ABLP90]). A routing scheme has stretch factor k if the length of the path a packet from vertex v to vertex w has to take according to the scheme is at most k times longer then the length of the shortest path between v and w. In [ABLP90] it is shown that, in order to guarantee stretch factor k, routing structures of size $O(k \cdot n^{1/k} \cdot \log n)$ in each vertex and routing information of size $O(\log n)$ in each packet are sufficient. So their routing scheme needs routing structures of size at least $O(\log^2 n)$. According to [PU89] any routing scheme that achieves a stretch factor of k must use an average of $\Omega(n^{1/(2k+4)})$ bits for the routing structure of a vertex.

All hierarchical schemes have the great disadvantage that the routing is done with the help of a clustering of the graph, where some vertices are declared as routing centers for a set of other vertices. It is not difficult to prove that this strategy causes a congestion of $\Theta(n)$, w.h.p., if randomly chosen 1-functions have to be routed. Therefore hierarchical routing schemes are not useful to obtain a fast routing time.

1.4 New Results

Our main result is a trade-off between routing time and space requirement in arbitrary vertex-symmetric networks. In particular, we prove:

Main Theorem: *Let $H = (V, E)$ be an arbitrary vertex-symmetric network with n vertices, degree d, and diameter $D = \Omega(\log n)$. Then for every $s \in [2, n]$ it holds:*
A randomly chosen function can be routed in $O(\log_s n \cdot D)$ rounds, w.h.p., if $O(s \cdot D \cdot \log d)$ space is available at each vertex and

(1) $O(D \cdot \log d)$ space is available for storing routing information in each packet. This

suffices to route arbitrary functions.

(2) $O(\log(s \cdot D))$ *space is available for storing routing information in each packet. This suffices to route random functions, w.h.p..*

Consequences of this result are described in Section 1.4 below. It is easy to extend the results of the Main Theorem to routing arbitrary permutations in H by simulating each routing phase in a way that the packets are first sent to random destinations before they are sent to their original destinations (see, e.g., [V82]).

Our approach to achieve this trade-off between routing time and space is 'Routing via Simulation'. The line of proof chosen here contains new results about the simulation of arbitrary and vertex-symmetric networks.

Consider networks $G = (V, R)$ and $H = (V, E)$. Fix a shortest path system $\mathcal{P}_R^H$ in H which contains shortest paths $p_H(u, v)$ in H only for pairs $\{u, v\} \in R$. Further fix a shortest path system $\mathcal{P}_G$ in G, consisting of paths $p_G(u, v)$ for all $u, v \in V$. Our strategy to simulate routing in G by H then works as follows:

Suppose, a packet with origin u and destination v travels along the path $p_G(u, v)$. In order to simulate the traversal of an edge $\{x, y\} \in R$, it chooses the path $p_H(x, y)$.

The resulting path system in H is called $\mathcal{P}^*$. Let D_G denote the dilation of $\mathcal{P}_G$, D_H the dilation of $\mathcal{P}_R^H$, and D^* the dilation of $\mathcal{P}^*$.

Let us call a packet at *stage* q if it is currently routed along the path in H simulating the q-th edge of the packet's path in G. Let C_f^q be an upper bound for the number of packets at stage q that pass a vertex v in H and $C_f^* = \max_{q \in [D_G]} C_f^q$. Obviously, this definition of congestion is stronger then the one in Section 1.1, because there C_f was defined to be the sum over all C_f^q. So in the following we consider C_f^* instead of C_f.

Clearly, $D^* \leq D_H \cdot D_G$, and routing f using the path system $\mathcal{P}^*$ needs time $\Omega(D^* + C_f^*)$. We will present a routing protocol that uses the path system $\mathcal{P}^*$ and prove the following performance bound.

Let f be some routing function with congestion C_f^* w.r.t. $\mathcal{P}^*$. Then routing f in H needs at most $O(D_G(D_H + C_f^*) + \log n)$ rounds, w.h.p..

Before we give upper bounds on the routing time and routing space in vertex-symmetric networks we describe what our random experiments are.

- Let S_n be the set of all permutations on V. Let $\pi \in S_n$ define an *embedding* of the vertices of G into vertices of H in a way that u in G is embedded in $\pi(u)$ in H. In the following we will mean by *'G is randomly embedded in H'* that π is chosen uniformly at random from S_n.

- Let $\mathcal{SP}_G$ denote the collection of all shortest path systems $\mathcal{P}_G$ in G. We say that $\mathcal{P}_G$ is a *random shortest path system* if it is chosen uniformly at random from $\mathcal{SP}_G$.

- Further let $\mathcal{SP}_H$ denote the collection of all shortest path systems $\mathcal{P}_H$ in H. We say that $\mathcal{P}_R^H$ is a *random shortest path system* if its paths are taken from a path system $\mathcal{P}_H$ chosen uniformly at random from $\mathcal{SP}$.

- We call f a *random function* if it is chosen uniformly at random from $\mathcal{F}$.

For a permutation $\pi \in S_n$ let $\pi \circ R = \{\{\pi(u), \pi(v)\} \mid \{u, v\} \in R\}$. Let D_G be the diameter of G. Then, for the experiment of randomly choosing a shortest path system $\mathcal{P}_G$ and a function f, the *expected stage congestion* σ of G is defined as

$$\sigma = \max_{e \in R,\, q \in [D_G]} E(\#\text{ packets that want to use } e \text{ as } q\text{-th edge})$$

Using the random experiments described above we are able to prove the following results:

Let $G = (V, R)$ be a d_G-regular network with expected stage congestion σ, $H = (V, E)$ be vertex-symmetric with diameter $D_H = \Omega(\log n)$, π be a randomly chosen embedding of G into H, and $\mathcal{P}_G$ and $\mathcal{P}^H_{\pi \circ R}$ be randomly chosen shortest path systems. Then the congestion C^*_f of a random function f is at most $O(\sigma \cdot d_G D_H)$, w.h.p.. Therefore routing f in H needs at most $O(D_G \cdot \sigma \cdot d_G D_H)$ rounds, w.h.p.. Furthermore, there are two strategies for space-efficient routing that imply routing structures of size $O(d_G \cdot D_H \cdot \log d_H)$ plus the size for storing $\mathcal{P}_G$, and

(1) routing information of size $O(D_H \log d_H)$. This suffices to route arbitrary functions.

(2) routing information of size $O(\log(d_G \cdot D_G \cdot D_H))$. This suffices for random functions f, w.h.p..

Finally, in order to get fast and space-efficient routing protocols for H, we will use as guest graph G a well-known vertex-symmetric network, the s-ary Butterfly. Its regular structure allows very space-efficient routing structures for $\mathcal{P}_G$. Furthermore, we show that $\sigma = \frac{1}{d_G}$ which implies that the congestion C^*_f of a random function is bounded by $O(D_H)$, w.h.p.. The Main Theorem then follows immediately from the fact that the s-ary Butterfly has degree $2s$ and diameter at most $2 \log_s n$.

1.5 Discussion of the Main Theorem

According to the Main Theorem it holds for all bounded degree vertex-symmetric networks with diameter $D = \Omega(\log n)$: If only space $O(D)$ is allowed for each vertex and space $O(\log D)$ is allowed for storing routing information in a packet the routing of a randomly chosen function finishes after $O(\log n \cdot D)$ rounds, w.h.p.. If a space of $O(n^\epsilon \cdot D)$, $\epsilon > 0$, is allowed for each vertex and space $O(\log n)$ is allowed for each packet, the routing finishes after $O(D)$ rounds, w.h.p..

As noted above, the best expanders that have an explicit construction are all Cayley graphs and therefore vertex-symmetric. Although it seems to be very difficult to design space-efficient routing schemes for these graphs with the help of an analysis of the underlying algebraic structure, the Main Theorem shows that space $O(\log n)$ suffices to route almost all functions $f \in \mathcal{F}$ in time $O(\log^2 n)$ and a space of $O(n^\epsilon)$ suffices, for arbitrary $\epsilon > 0$, to achieve a routing time of $O(\log n)$, w.h.p..

1.6 Organization of the Paper

In the next section we describe the routing protocol used for our simulations. Section 3 proves an upper bound for the congestion C^*_f if G is edge-symmetric and H is vertex-symmetric. Section 4 presents a suitable design for space-efficient routing structures. Finally, in Section 5 the Main Theorem is proved.

2 The Extended Growing-Rank Protocol

In this section we describe an extension of the *growing-rank protocol* presented in [MV95]. As we will see, it is especially suitable for simulations among vertex-symmetric networks.

Let $G = (V, R)$ and $H = (V, E)$ be arbitrary networks and $f \in \mathcal{F}$. Let $\mathcal{P}_G$, $\mathcal{P}_R^H$, $\mathcal{P}^*$, D_G, D_H, and C_f^* be defined as in Section 1.3.

Initially, each packet P_v is assigned an integer rank(P_v), chosen uniformly at random and independently from the set $\{0, \ldots, K-1\}$, where

$$K := \left\lceil \frac{12e \cdot C_f^* + 2D_H + (\alpha+1)\log n / D_G}{D_H} \right\rceil \cdot D_H$$

for some constant $\alpha > 0$. Thus K is a multiple of D_H.

Whenever a packet is forwarded in H, its rank is increased by $\frac{K}{D_H}$. When a packet P reaches the vertex in H that simulates the q'th vertex on P's path in $\mathcal{P}_G$, $q = 0, 1, \ldots$, a new rank is chosen independently and uniformly at random from the set $q \cdot 2K + [K] = \{q \cdot 2K, \ldots, q \cdot 2K + (K-1)\}$.

If two or more packets are contending to leave the same vertex, then the one with the smallest rank is chosen. A round for a vertex within a stage looks the same as in the growing rank protocol described in [MV95]:

- choose a packet P with minimum rank;
- rank(P) :=rank$(P) + \frac{K}{D_H}$;
- move P forward on its routing path.

If there is more than one packet with smallest rank, then in order to break ties the packet P_v with lowest value v is chosen (note that for this purpose v has to be stored in the routing information).

The following theorem will give a bound for the routing time of an arbitrary function on G simulated by H for arbitrary networks G and H. The proof will be an extension of the proof in [MV95] which itself is modification of analyses presented in [R91], [L92], and [LMRR94]. The result in [MV95] only holds for *shortest* path systems. The problem we have to handle in our proof is that different phases of our routing protocol overlap and that we do not have shortest path systems any more.

Theorem 2.1 *Let G and H be two arbitrary graphs, let $\mathcal{P}^*$, D_H, and D_G be defined as above. Furthermore, let $f \in \mathcal{F}$ be some routing function with congestion C_f^* w.r.t. $\mathcal{P}^*$. Then the extended growing-rank protocol routes f in H within $O(D_G(D_H + C_f^*) + \log n)$ rounds, w.h.p..*

Proof. In the following, we denote the rank of a packet P while waiting at a vertex v by rank$^v(P)$. Let $\mathrm{id}_{\max} = n$. We define the *ident-rank* of P at v as rank$^v(P) + \frac{\mathrm{id}(P)}{\mathrm{id}_{\max}+1}$ and denote it by id-rank$^v(P)$. Note that, in each round, the ident-ranks of all packets are distinct. This type of rank ensures that whenever a packet P delays a packet P' at a vertex v it holds id-rank$^v(P) <$ id-rank$^v(P')$. The following lemma shows that the rank of any packet at stage q can not be greater than $2(q+1)K - 1$.

Lemma 2.2 *Suppose P is a packet at stage q which is stored at a vertex v in some round. Then* rank$^v(P) \leq 2(q+1)K - 1$.

Proof. At the beginning of stage q, the rank of P is at most $q \cdot 2K + K - 1$. Since the length of the routing path of P within two stages is at most D_H, the rank of P is increased by $\frac{K}{D_H}$ for at most D_H times. Thus, $\text{rank}^v(P) \leq q \cdot 2K + K - 1 + D_H \cdot \frac{K}{D_H} \leq 2(q+1)K - 1$. ∎

Note that the rank of any packet during any stage of the routing will be bounded above by $2D_GK - 1$. The following analysis will be based on a delay sequence argument.

Definition 2.3 (((s,ℓ)-delay sequence)) *An (s,ℓ)-delay sequence consists of*

(1) $s+1$ not necessarily distinct collision vertices $v_0, v_1, \ldots, v_s$;

(2) s delay packets $P_1, P_2, \ldots, P_s$ such that the routing path of P_i crosses the vertex v_i and the vertex v_{i-1} in that order for $1 \leq i \leq s$;

(3) s integers $\ell_1, \ell_2, \ldots, \ell_s$ such that ℓ_i is the number of edges on the routing path of packet P_i from vertex v_i to vertex v_{i-1} for $1 \leq i \leq s$, and $\sum_{i=1}^{s} \ell_i \leq \ell$; and

(4) s integer keys $r_1, r_2, \ldots, r_s$ such that $0 \leq r_s \leq \cdots \leq r_2 \leq r_1 \leq 2D_GK - 1$.

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

Lemma 2.4 *Suppose the routing takes $T \geq 2D_GD_H$ or more rounds. Then there exists an active $(T - 2D_GD_H, 2D_GD_H)$-delay sequence.*

Proof. First, we give a construction scheme for a delay sequence. Let P_1 be a packet that moves forward in round T to a vertex v_0. We follow P_1's routing path backwards to the last vertex on this path where it was delayed. This vertex we call v_1. Let P_2 be the packet that caused the delay, since it was preferred against P_1. We now follow the path of P_2 backwards until we reach a vertex v_2 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 round 1. Here it ends with a packet P_s starting at its source v_s.

The path from v_s to v_0 recorded by this process in reversed order is called *delay path*. It consists of contiguous parts of routing paths. In particular, the part of the delay path from vertex v_i to vertex v_{i-1} is a subpath of the routing path of packet P_i; we define ℓ_i to be the length of this subpath for $1 \leq i \leq s$.

We set $r_i := \text{rank}^{v_i}(P_i)$ for $1 \leq i \leq s$. Because of the rules of the protocol we have $r_1 \geq r_2 \geq \cdots \geq r_s \geq 0$. Moreover, Lemma 2.2 yields that $2D_GK - 1 \geq r_1$. Thus, we have constructed an active (s,ℓ)-delay sequence for every $\ell \geq \sum_{i=1}^{s} \ell_i$.

Our next goal is to bound the sum of the ℓ_i's. In addition to the ranks $r_1, \ldots, r_s$, we denote by r_0 the rank of P_1 in v_0. It follows immediately from the protocol that $r_i + \ell_i \cdot \frac{K}{D_H} \leq r_{i-1}$ for $1 \leq i \leq s$. As a consequence,

$$\sum_{i=1}^{s} \ell_i \cdot \frac{K}{D_H} \leq r_0 \overset{\text{Lemma 2.2}}{\Longrightarrow} \sum_{i=1}^{s} \ell_i \leq (2D_GK - 1) \cdot \frac{D_H}{K} \leq 2D_GD_H \ . \tag{1}$$

Since the delay sequence covers up T rounds and consists of $\sum_{i=1}^{s} \ell_i$ moves and $s-1$ delays, we have $T = \sum_{i=1}^{s} \ell_i + s - 1$. It follows that

$$s = T - \sum_{i=1}^{s} \ell_i + 1 \overset{(1)}{\geq} T - 2D_GD_H + 1 \ .$$

Consequently, if we stop the above construction at packet $P_{T-2D_GD_H}$, we have found an active $(T - 2D_GD_H, 2D_GD_H)$-delay sequence. ∎

Lemma 2.5 *If the routing paths of the packets are shortest paths, then the tuples (P, q) of delay packets P at stage q in the above construction are pairwise distinct.*

Proof. Suppose, in contrast to our claim, that there is some packet P appearing twice at the same stage q in the delay sequence. Then there exist i and j with $1 \le i < j \le s$ and $P = P_i = P_j$. Thus, the routing path of P crosses the delay path at the collision vertices v_j and v_i in that order.

Let m denote the distance from the vertex v_j to the vertex v_i. If the routing paths are shortest paths, then the rank of P is increased m times while moving from v_j to v_i, and hence,

$$\text{id-rank}^{v_i}(P) \quad = \quad \text{id-rank}^{v_j}(P) + m \cdot \frac{K}{D_H} \ . \tag{2}$$

On the other hand, each packet P_{k+1} delays the packet P_k at vertex v_k, and consequently, $\text{id-rank}^{v_k}(P_k) > \text{id-rank}^{v_k}(P_{k+1})$ for $1 \le k \le s-1$. Further, the length of the routing path of packet P_{k+1} from v_{k+1} to v_k is ℓ_{k+1}, and thus the rank of P_{k+1} is increased by $\ell_{k+1} \cdot \frac{K}{D_H}$ on its path from v_{k+1} to v_k for $1 \le k \le s-1$. It follows that $\text{id-rank}^{v_k}(P_k) > \text{id-rank}^{v_{k+1}}(P_{k+1}) + \ell_{k+1} \cdot \frac{K}{D_H}$ for $1 \le k \le s-1$. This yields

$$\text{id-rank}^{v_i}(P) > \text{id-rank}^{v_j}(P) + \sum_{k=i}^{j-1} \ell_{k+1} \cdot \frac{K}{D_H} \ge \text{id-rank}^{v_j}(P) + m \cdot \frac{K}{D_H} \ . \tag{3}$$

Since (3) contradicts (2), there is no packet that appears twice at the stage in the delay sequence. ∎

Lemma 2.6 *The number of different active (s, ℓ)-delay sequences in H is at most*

$$n \cdot 2^{\ell} \left(\frac{2eC_f^*(s + 2D_GK)}{s} \right)^s \ .$$

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

- There are n possibilities to determine the starting point v_0 of the delay path.
- Since $\sum_{i=1}^{s} \ell_i \le \ell$, there are $\binom{s+\ell}{s}$ ways to choose the ℓ_i's.
- Finally, there are $\binom{s+2D_GK-1}{s} \le \binom{s+2D_GK}{s}$ possibilities to choose the r_i's such that $2D_GK - 1 \ge r_1 \ge r_2 \ge \cdots \ge r_s \ge 0$.
- Once the ℓ_i's and r_i's are chosen, there are at most $(C_f^*)^s$ choices for the delay packets. This is because there are at most C_f^* choices for the packet P_1. We follow the routing path of P_1 backwards for ℓ_1 rounds, until we reach vertex v_1. Now we have at most C_f^* choices for P_2. We follow again the routing path of this packet to vertex v_2 an so on, until we reach packet P_s.

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

$$n \cdot (C_f^*)^s \binom{s+\ell}{s} \binom{s+2D_GK}{s} \ .$$

Applying the inequalities $\binom{a}{b} \leq 2^a$ and $\binom{a}{b} \leq \left(\frac{ea}{b}\right)^b$, the desired upper bound is

$$n(C_f^*)^s 2^{s+\ell} \left(\frac{e(s+2D_GK)}{s}\right)^s \quad \leq \quad n \cdot 2^\ell \left(\frac{2eC_f^*(s+2D_GK)}{s}\right)^s \ .$$

∎

The probability that a particular delay sequence with s distinct packets is active is at most K^{-s}. This is because a sequence with s distinct packets determines s ranks. As a consequence,

$$\begin{aligned}
&\text{Prob(the routing takes } T = s - 2D_GD_H \text{ or more rounds)} \\
&\leq \quad \text{Prob}\begin{pmatrix}\text{an } (s, 2D_GD_H)\text{-delay sequence with} \\ \text{distinct delay packets is active}\end{pmatrix} \\
&\leq \quad n2^{2D_GD_H}\left(\frac{2eC_f^*(s+2D_GK)}{s}\right)^s \cdot K^{-s} \ .
\end{aligned}$$

We choose $T = 12eC_f^*D_G + 4D_GD_H + (\alpha+1)\log n$. This yields

$$s \ \geq \ 12eC_f^*D_G \ , \tag{4}$$

$$s \ \geq \ (\alpha+1)\log n + 2D_GD_H \ , \text{ and} \tag{5}$$

$$s \ \leq \ D_GK \tag{6}$$

for $K \geq 12eC_f^* + 2D_H + (\alpha+1)\log n/D_G$. As a consequence,

$$\begin{aligned}
&\text{Prob(the routing takes } T = s - 2D_GD_H \text{ or more rounds)} \\
&\overset{(6)}{\leq} \quad n2^{2D_GD_H}\left(\frac{6eC_f^*D_G}{s}\right)^s \overset{(4)+(5)}{\leq} n2^{2D_GD_H}\left(\frac{1}{2}\right)^{(\alpha+1)\log n + 2D_GD_H} = n^{-\alpha} \ .
\end{aligned}$$

This proves Theorem 2.1. ∎

Note that, if we use priority queues as buffers for the packets and so-called ghost-packets according to a strategy used in [R91], then it takes only $O(d_H)$ time for a vertex to find the packet with the lowest rank. So if we consider only networks H with constant degree, then the time to find the packet with lowest rank in each round is constant.

In the next section this theorem will be used to obtain efficient simulations of arbitrary networks on vertex-symmetric networks.

3 Bounding the Congestion

In this section we bound the congestion C_f^* for the case that G is a d_G-regular network and H is vertex-symmetric. Recall our strong notion of congestion as defined in Section 1.3.

Theorem 3.1 *Let G be a d_G-regular network with expected stage congestion σ, and H be vertex-symmetric with diameter $D_H = \Omega(\log n)$, π be a random embedding of G into H, $\mathcal{P}_G$ be a random shortest path system in G, and $\mathcal{P}^H_{\pi \circ R}$ be a random shortest path system in H. Then, for a random function f,*

$$C^*_f = O(\sigma \cdot d_G D_H) \ ,$$

w.h.p.. Thus, by Theorem 2.1, $O(D_G \cdot \sigma \cdot d_G D_H)$ rounds suffice to route a random function f, w.h.p..

Proof. We have to prove the bound on C^*_f. For a fixed $v \in V$ and $e = \{u, w\} \in R$, let the binary random variable $X_{e,v}$ be 1 if and only if for a randomly chosen embedding π and shortest path system $\mathcal{P}^H_{\pi \circ R}$ the path $p_H(\pi(u), \pi(w))$ contains v. Further, for a fixed edge $e \in R$ and packet $u \in V$, let the binary random variable $X^q_{u,e}$ be 1 if and only if for a randomly chosen shortest path system $\mathcal{P}_G$ and function f, e is the q-th edge in the path from u to $f(u)$ in G prescribed by $\mathcal{P}_G$.

For $v \in V$, let the random variable C^q_v denote the congestion at v in H caused by packets at stage q if π, the shortest path systems $\mathcal{P}_G$ and $\mathcal{P}^H_{\pi \circ R}$, and $f \in \mathcal{F}$ are chosen independently at random. Clearly, it holds:

$$C^q_v = \sum_{e \in R} X_{e,v} \left(\sum_{u \in V} X^q_{u,e} \right)$$

We first want to calculate the expected congestion $E(C^q_v)$ for each vertex v in H and $q \in [D_G]$.

Let $p_{e,v}$ be the probability that $X_{e,v} = 1$ and $p^q_{u,e}$ be the probability that $X^q_{u,e} = 1$. Because H is vertex-symmetric there is an automorphism φ for every pair of vertices v, v' in H that maps v to v'. Consequently, by the choice of the random experiments, $p_{e,v} = p_{e,\varphi(v)} = p_{e,v'}$. Since the $X_{e,v}$ are independent from the $X^q_{u,e}$ it holds:

$$\begin{aligned} E(C^q_v) &= \sum_{e \in R} \sum_{u \in V} E(X_{e,v}) \cdot E(X^q_{u,e}) \\ &= \sum_{e \in R} \sum_{u \in V} p_{e,v} \cdot p^q_{u,e} \ = \ \sum_{e \in R} \sum_{u \in V} p_{e,\varphi(v)} \cdot p^q_{u,e} \\ &= \sum_{e \in R} \sum_{u \in V} p_{e,v'} \cdot p^q_{u,e} \ = \ E(C^q_{v'}) \end{aligned}$$

Thus $E(C^q_v)$ is the same for every vertex $v \in V$, namely at most $D_H + 1$, because there are n packets that have to be routed along paths of length at most $D_H + 1$.

Unfortunately, we can not use the well-known Chernoff bound to prove an upper bound for C^q_v that holds w.h.p., because the products $X_{e,v} \cdot X^q_{(x,i),e}$ are not independent from each other. Nevertheless, the following lemma enables us to use the high moments version of the well-known Markov Inequality (see, e.g., [SSS93]) in such a way that we can bound the congestion at each vertex in H by $O(\sigma \cdot d_G D_H)$, w.h.p..

Lemma 3.2 *Let X be an arbitrary random variable. Then, for every $\epsilon > 0$ and $k \geq 0$, it holds:*

$$\mathit{Prob}\left(|X - E(X)| \geq \epsilon \cdot \sqrt[k]{E(|X - E(X)|^k)} \right) \leq \left(\tfrac{1}{\epsilon} \right)^k$$

∎

Let $m \in \{\alpha \log n, \alpha \log n + 1\}$ be even. Then we get:

$$E(|C_v^q - E(C_v^q)|^m) = E((C_v^q - E(C_v^q))^m) = \sum_{k=0}^{m} \binom{m}{k} E((C_v^q)^k)(-E(C_v^q))^{m-k}$$

It remains to bound $E((C_v^q)^k)$ for every $0 \le k \le m$. Let $s(k,j) = \sum_{\ell=0}^{j} (-1)^\ell \binom{j}{\ell} (j-\ell)^k \le j^k$ be the number of surjective mappings from $[k]$ to $[j]$. Then it holds

$$E((C_v^q)^k) = \sum_{j=1}^{k} s(k,j) \sum_{\substack{\{(u_1,e_1),\ldots,(u_j,e_j)\} \\ \subseteq V \times R}} E(X_{e_1,v} X_{u_1,e_1}^q \cdot \ldots \cdot X_{e_j,v} X_{u_j,e_j}^q)$$

In the following let the operator $\bar{E}(.)$ denote the average value of $E(.)$ over all subsets $\{(u_1,e_1),\ldots,(u_k,e_k)\} \subseteq V \times R$, where '.' denotes some formula over random variables. In other words,

$$\bar{E}(.) = \frac{1}{\binom{n \cdot |R|}{k}} \sum_{\substack{\{(u_1,e_1),\ldots,(u_j,e_j)\} \\ \subseteq V \times R}} E(.)$$

Then it remains to prove the following lemma to get a bound for $E((C_v^q)^k)$.

Lemma 3.3 *For $k \le \alpha \cdot \log n$, α constant, and the four random experiments described above it holds:*

$$\bar{E}(X_{e_1,v} X_{u_1,e_1}^q \cdot \ldots \cdot X_{e_k,v} X_{u_k,e_k}^q) \le \left(\frac{e\sigma(D_H + 4k)}{n^2} \right)^k$$

Proof. Since the $X_{e,v}$ are independent from the $X_{u,e}^q$ it holds:

$$E(X_{e_1,v} X_{u_1,e_1}^q \cdot \ldots \cdot X_{e_k,v} X_{u_k,e_k}^q) = E(X_{e_1,v} \cdot \ldots \cdot X_{e_k,v}) \cdot E(X_{u_1,e_1}^q \cdot \ldots \cdot X_{u_k,e_k}^q)$$

From this we conclude that

$$\begin{aligned}
&\bar{E}(X_{e_1,v} X_{u_1,e_1}^q \cdot \ldots \cdot X_{e_k,v} X_{u_k,e_k}^q) \\
&= \frac{1}{\binom{n \cdot |R|}{k}} \sum_{\substack{\{(u_1,e_1),\ldots,(u_j,e_j)\} \\ \subseteq V \times R}} E(X_{e_1,v} X_{u_1,e_1}^q \cdot \ldots \cdot X_{e_j,v} X_{u_j,e_j}^q) \\
&= \frac{1}{\binom{n \cdot |R|}{k}} \sum_{e_1,\ldots,e_k \in R} E(X_{e_1,v} \cdot \ldots \cdot X_{e_k,v}) \sum_{\substack{u_1,\ldots,u_k \in V \\ \text{s.t. all } (u_j,e_j) \text{ distinct}}} E(X_{u_1,e_1}^q \cdot \ldots \cdot X_{u_k,e_k}^q) \cdot \frac{1}{k!}
\end{aligned}$$

The factor $\frac{1}{k!}$ is necessary to eliminate superflous permutations of the (u_j, e_j). It is easy to see that the X_{u_j,e_j}^q can be regarded as independent, because the destinations of the packets are chosen independently at random and there is at most one edge e a packet can take at stage q. Thus, according to the definition of the expected stage congestion σ, it holds for all $e_1, \ldots, e_k \in R$ with M denoting the set of all $(u_1, \ldots, u_k) \in V^k$ such that all (u_j, e_j) are distinct:

$$\frac{1}{|M|} \sum_{(u_1,\ldots,u_k) \in M} E(X_{u_1,e_1}^q \cdot \ldots \cdot X_{u_k,e_k}^q) \le \left(\frac{\sigma}{n} \right)^k$$

It remains to prove an upper bound for $\bar{E}(X_{e_1,v} \cdot \ldots \cdot X_{e_k,v})$. This is done in the following claim.

Claim 3.4 *For the random experiments described above it holds:*

$$\bar{E}(X_{e_1,v}\cdot\ldots\cdot X_{e_k,v}) \le \left(\tfrac{e(D_H+4k)}{n}\right)^k$$

Proof. According to the definition of $\bar{E}(.)$ we get

$$\bar{E}(X_{e_1,v}\cdot\ldots\cdot X_{e_k,v}) \le \frac{1}{\binom{n\cdot|R|}{k}}\sum_{j=1}^{k} s(k,j)\sum_{\{e'_1,\ldots,e'_j\}\subseteq R} E(X_{e'_1,v}\cdot\ldots\cdot X_{e'_j,v})\sum_{(u_1,\ldots,u_k)\in V^k}\frac{1}{k!}\ ,$$

since there are $s(k,j)$ possibilities to map $\{e_1,\ldots,e_k\}$ to $\{e'_1,\ldots,e'_j\}$. Therefore it holds (note that $[m]_k = m!/(m-k)!$):

$$\begin{aligned}\bar{E}(X_{e_1,v}\cdot\ldots\cdot X_{e_k,v}) &\le \frac{1}{\binom{n\cdot|R|}{k}}\sum_{j=1}^{k} s(k,j)\sum_{\{e'_1,\ldots,e'_j\}\subseteq R} E(X_{e'_1,v}\cdot\ldots\cdot X_{e'_j,v})\cdot\frac{n^k}{k!}\\ &= \sum_{j=1}^{k}\frac{s(k,j)\cdot n^k}{[n|R|]_k}\sum_{\{e'_1,\ldots,e'_j\}\subseteq R}\underbrace{\mathrm{Prob}(X_{e'_1,v}\cdot\ldots\cdot X_{e'_j,v}=1)}_{(*)}\end{aligned}$$

Before we can proceed with our calculation we have to find an upper bound for $(*)$ if $\{e'_1,\ldots,e'_j\}$ is randomly chosen out of R.

Let the random variable I be i if and only if $\{e'_1,\ldots,e'_j\}$ has a maximal independent set of size i, that is, $\{e'_1,\ldots,e'_j\}$ has a set $\{\bar{e}_1,\ldots,\bar{e}_i\}$ of maximal size i for which all vertices adjacent to $\bar{e}_1,\ldots,\bar{e}_i$ are distinct. We first want to show that for any edge $e\in\{\bar{e}_1,\ldots,\bar{e}_i\}$ we can independently assume a probability of $\frac{D_H+1}{n-4i}$ that, for a randomly chosen embedding of G into H, the path simulating e in H traverses a fixed vertex v in H.

Since H is vertex-symmetric, it holds for every fixed vertex v in H that, for a randomly chosen shortest path system $\mathcal{P}_H$ in H,

$$\frac{1}{\binom{n}{2}}\sum_{\{u,w\}\subseteq V} p_{\{u,w\},v} = \frac{D_H+1}{n}$$

Consider the edges $\bar{e}_1,\ldots,\bar{e}_{i-1}$ to be embedded into some set of vertices $W=\{w_1,\ldots,w_{2(i-1)}\}$ in H. Then we get

$$\frac{1}{\binom{n}{2}}\sum_{\{u,w\}\subseteq V\setminus W} p_{\{u,w\},v} \le \frac{D_H+1}{n}$$

From this we conclude that, for a randomly chosen 1-1 embedding π and shortest path system $\mathcal{P}_H$,

$$\begin{aligned}&\mathrm{Prob}(X_{\bar{e}_i,v}=1\mid X_{\bar{e}_1,v}\cdot\ldots\cdot X_{\bar{e}_{i-1},v}=1)\\ &= \frac{\mathrm{Prob}(X_{\bar{e}_1,v}\cdot\ldots\cdot X_{\bar{e}_i,v}=1)}{\mathrm{Prob}(X_{\bar{e}_1,v}\cdot\ldots\cdot X_{\bar{e}_{i-1},v}=1)}\\ &= \frac{\sum_{W=\{w_1,\ldots,w_{2(i-1)-1}\}\subseteq V}\prod_{j=1}^{i-1} p_{\{w_{2j-1},w_{2j}\},v}\cdot\frac{1}{\binom{n-2(i-1)}{2}}\sum_{\{u_1,u_2\}\subseteq V\setminus W} p_{\{u_1,u_2\},v}}{\sum_{W=\{w_1,\ldots,w_{2(i-1)-1}\}\subseteq V}\prod_{j=1}^{i-1} p_{\{w_{2j-1},w_{2j}\},v}}\\ &\le \frac{\sum_{W=\{w_1,\ldots,w_{2(i-1)}\}\subseteq V}\prod_{j=1}^{i-1} p_{\{w_{2j-1},w_{2j}\},v}\cdot\overbrace{\frac{\binom{n}{2}}{\binom{n-2(i-1)}{2}}\cdot\frac{D_H+1}{n}}^{(**)}}{\sum_{W=\{w_1,\ldots,w_{2(i-1)-1}\}\subseteq V}\prod_{j=1}^{i-1} p_{\{w_{2j-1},w_{2j}\},v}}\end{aligned}$$

$(**)$ can be bounded by

$$\left(\frac{n}{n-2(i-1)-1}\right)^2 \frac{D_H+1}{n} \le \frac{D_H+1}{n-4i}$$

Therefore it holds:

$$\mathrm{Prob}(X_{e'_1,v}\cdot\ldots\cdot X_{e'_j,v}=1) \le \sum_{i=1}^{j} \mathrm{Prob}(I=i)\left(\frac{D_H+1}{n-4i}\right)^i$$

It remains to prove an upper bound for $\mathrm{Prob}(I=i)$.

Consider any fixed $i \in \{1,\ldots,j\}$. Let $\{\bar{e}_1,\ldots,\bar{e}_i\}$ be a maximal independent set in $\{e'_1,\ldots,e'_j\}$. Then the set $\{e'_1,\ldots,e'_j\}$ can be decomposed into ℓ trees T_ℓ containing $\bar{e}_\ell$ in such a way that we obtain the following structure (each Δ in a tree T_ℓ denotes a set of edges incident to one of the vertices of e_ℓ):

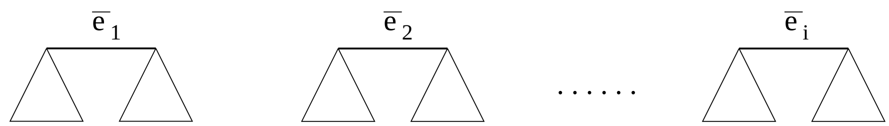

Figure 2: A decomposition of $\{e'_1,\ldots,e'_j\}$ into i trees.

Assume in the contrary this is not true. Then there exists a tree T_ℓ that has an edge e with distance 2 from $\bar{e}_\ell$ that has no vertex that is adjacent to an edge in $\{\bar{e}_1,\ldots,\bar{e}_i\}$. But then we can extend the independent set by e, that is, $\{\bar{e}_1,\ldots,\bar{e}_i\}$ can not be a maximal independent set. Thus the decomposition above is correct.

Clearly, there are at most $|R|^i$ possibilities to choose the edges of the independent set. For the remaining edges there are $(2i)^{j-i}$ possibilities to determine to which subtree of which tree T_ℓ they belong, and d_G^{j-i} possibilities to choose the second vertex adjacent to them. Since we do not want to count permutations among these j edges we get that altogether there are at most

$$\binom{|R|}{i}\binom{i(2d_G-1)}{j-i}$$

possibilities to choose an edge set $\{e'_1,\ldots,e'_j\}$ that correponds to the decomposition described above. Since there are $\binom{|R|}{j}$ ways to choose a subset of j different edges it holds with $|R|=\frac{d_G}{2}n$ that

$$\begin{aligned}\mathrm{Prob}(I=i) &= \frac{\binom{|R|}{i}\binom{i(2d_G-1)}{j-i}}{\binom{|R|}{j}} \\ &\le \frac{[|R|]_i(i(2d_G-1))^{j-i}}{[|R|]_j}\cdot\frac{j!}{i!(j-i)!} \le \binom{j}{i}\left(\frac{4i}{n}\right)^{j-i}\end{aligned}$$

So altogether we get

$$\begin{aligned}\mathrm{Prob}(X_{e_1,v}\cdot\ldots\cdot X_{e_j,v}=1) &\le \sum_{i=1}^{j}\binom{j}{i}\left(\frac{D_H+1}{n-4i}\right)^i\left(\frac{4i}{n}\right)^{j-i} \\ &\le \left(\frac{D_H+4j}{n-4j}\right)^j\end{aligned}$$

Using this bound in $(*)$ and the fact that $\binom{n}{k} \le \frac{1}{\sqrt{\pi k}}(\frac{en}{k})^k$ for all $n \in \mathbb{N}$ and $k \in \{1, \ldots, n\}$ yields

$$\begin{aligned}
\bar{E}(X_{e_1,v} \cdot \ldots \cdot X_{e_k,v}) &\le \sum_{j=1}^{k} \frac{s(k,j)\cdot n^k}{[n|R|]_k} \sum_{\{e_1,\ldots,e_j\}\subseteq R} \left(\frac{D_H+4j}{n-4j}\right)^j \\
&\le \sum_{j=1}^{k} \frac{j^k \cdot n^k}{[n|R|]_k} \binom{|R|}{j} \left(\frac{D_H+4j}{n-4j}\right)^j \\
&\le \sum_{j=1}^{k} \frac{j^k \cdot n^k}{[n|R|]_k} \frac{1}{\sqrt{\pi j}} \left(\frac{e|R|}{j}\right)^j \left(\frac{D_H+4j}{n-4j}\right)^j \\
&\le \sum_{j=1}^{k} \frac{e^j}{\sqrt{\pi j}} \left(\frac{j}{|R|}\right)^{k-j} \left(\frac{D_H+4j}{n-4j}\right)^j \\
&\le \left(\frac{e(D_H+4k)}{n}\right)^k
\end{aligned}$$

This proves the claim. ∎

So altogether we get

$$\bar{E}(X_{e_1,v}X^q_{(x_1,i_1),e_1} \cdot \ldots \cdot X_{e_k,v}X^q_{(x_k,i_k),e_k}) \le \left(\frac{e(D_H+4k)}{n}\right)^k \left(\frac{\sigma}{n}\right)^k$$

This proves the lemma. ∎

With the help of Lemma 3.3 we get

$$\begin{aligned}
E((C_v^q)^k) &= \sum_{j=1}^{k} s(k,j) \sum_{\substack{\{(u_1,e_1),\ldots,(u_j,e_j)\} \\ \subseteq V\times R}} E(X_{e_1,v}X^q_{u_1,e_1} \cdot \ldots \cdot X_{e_j,v}X^q_{u_j,e_j}) \\
&\le \sum_{j=1}^{k} s(k,j) \binom{n\cdot|R|}{j} \left(\frac{e\sigma(D_H+4j)}{n^2}\right)^j \\
&\le \sum_{j=1}^{k} j^k \cdot \frac{1}{\sqrt{\pi j}} \left(\frac{en|R|}{j}\right)^j \left(\frac{e\sigma(D_H+4j)}{n^2}\right)^j \\
&\le \sum_{j=1}^{k} \frac{e^j}{\sqrt{\pi j}} j^{k-j} \left(\frac{n|R|\cdot e\sigma(D_H+4j)}{n^2}\right)^j \\
&\le (e[\tfrac{e}{2}\sigma \cdot d_G(D_H+4k)+k])^k
\end{aligned}$$

for all $k \le m$, m chosen as above. Thus it holds

$$\begin{aligned}
E(|C_v^q - E(C_v^q)|^m) &= E((C_v^q - E(C_v^q))^m) \\
&= \sum_{k=0}^{m} \binom{m}{k} E((C_v^q)^k)(-E(C_v^q))^{m-k} \\
&\le (\underbrace{e[\tfrac{e}{2}\sigma \cdot d_G(D_H+4m)+m]}_{(*)})^m
\end{aligned}$$

With the help of Lemma 3.2 we get that for any $\epsilon > 0$, $|C_v^q - E(C_v^q)| \geq \epsilon \cdot (*)$ with probability at most $(\frac{1}{\epsilon})^k$ and therefore

$$\text{Prob}(|C_v^q - E(C_v^q)| \geq \epsilon \cdot (*)\ ) \leq \left(\frac{1}{n}\right)^{\alpha \log \epsilon}$$

Hence, the congestion $C_v = \max_{q \in [D_G]} C_v^q$ in each vertex v is at most $O(\sigma \cdot d_G D_H)$, w.h.p.. With the help of the simple Markov Inequality ($k = 1$) we can finally show that for a randomly chosen π, $\mathcal{P}_G$, and $\mathcal{P}^H_{\pi \circ R}$ it holds, w.h.p., that afterwards a function f chosen independently at random has a congestion C_f^* of at most $O(\sigma \cdot d_G D_H)$, w.h.p.. Applying Theorem 2.1 with this result completes the proof of Theorem 3.1. ∎

Theorem 3.1 implies that it is easy to construct a fast probabilistic sequential algorithm that builds up a path system in H that guarantees a congestion of $O(\sigma \cdot d_G D_H)$ for almost every function f, w.h.p.. It is not clear whether there exists a fast parallel algorithm for this purpose. This will be an interesting problem for the situation that the configuration of the parallel system often changes.

4 Design of Space-Efficient Routing Structures

In this section we present two methods to design space-efficient routing structures and routing information.

Theorem 4.1 *Let G be d_G-regular and H vertex-symmetric, G be randomly embedded in H by π, and $\mathcal{P}_G$ and $\mathcal{P}^H_{\pi \circ R}$ be random shortest path systems. Then there are two strategies for space-efficient routing that imply routing structures of size $O(d_G \cdot D_H \cdot \log d_H)$ plus the size for storing $\mathcal{P}_G$, and*

(1) routing information of size $O(D_H \log d_H)$ suffice to route arbitrary functions f in H, for arbitrary h.

(2) routing information of size $O(\log(d_G \cdot D_G \cdot D_H))$ suffice to route random functions f in H, w.h.p..

Proof. We first prove (1). Consider a vertex v in H. Let $x = \pi^{-1}(v)$ and R_x be the set of all edges in G incident to x. Clearly, the number of x (which needs space $\log n$) has to be stored in v. Furthermore, the routing structure for v consists of the following two tables.

- $T_{v,1} : V^2 \to R_x \cup \{\emptyset\}$ is arranged in such a way that for every path $p_G(y, z)$ in $\mathcal{P}_G$ that crosses x, $T_{v,1}(y, z)$ contains the edge following x in $p_G(y, z)$.
- $T_{v,2} : R_x \to \mathcal{P}^H_{\pi \circ R}$ is arranged such that $T_{v,2}(e)$ contains that path in H representing the edge e in G.

Clearly, it takes $d_G \cdot D_H \cdot \log d_H$ bits to store $T_{v,2}$ in v. Since $D_H \log d_H = \Omega(\log n)$, the routing structure for v needs space $O(d_G \cdot D_H \cdot \log d_H)$ plus the space needed for storing $T_{v,1}$, which depends on $\mathcal{P}_G$.

In this case, the simulation will work as follows. Suppose, a packet P_x stored in v has to be sent to a vertex y in G. Initially, v chooses the edge $e \in T_{v,1}(x, y)$ and stores the corresponding path $T_{v,2}(e)$ in the routing information $r(x)$ of packet P_x.

If this packet is received by vertex w in H during the routing, w first checks whether the packet has already completed the path stored in its routing information. If this is true and the packet P_x has not reached its destination vertex yet, then w chooses the edge $e \in T_{w,1}(x,y)$ and stores $T_{w,2}(e)$ in $r(x)$. Otherwise, w will take the information about the next edge to be chosen out of $r(x)$ and routes the packet along this edge.

Thus the routing information consists of a path in H of length at most D_H, which needs space $O(D_H \log d_H)$, and h and a random rank for the extended growing rank protocol, which needs space $O(\log n)$, because $C_f^* \leq n$ for all functions f. This proves part (1) of Theorem 4.1.

In order to prove part (2) we have to find an upper bound for the number of paths in $\mathcal{P}^H_{\pi\circ R}$ that traverse a vertex v in H, w.h.p., if $D_H = \Omega(\log n)$.

Lemma 4.2 *For a randomly chosen π and $\mathcal{P}^H_{\pi\circ R}$ it holds, w.h.p., that the number of paths in $\mathcal{P}^H_{\pi\circ R}$ traversing v is at most $O(d_G \cdot D_H)$ for every vertex v in H.*

Proof. For $v \in V$, let the random variable P_v denote the number of paths in $\mathcal{P}^H_{\pi\circ R}$ traversing vertex v, let $X_{e,v}$ and $p_{e,v}$ be defined as in the proof of Theorem 3.1. Clearly, it holds:

$$E(P_v) = \sum_{e\in R} X_{e,v}$$

Choose two arbitrary vertices v and v'. Since H is vertex-symmetric, there exists an automorphism φ that maps v to v'. By the conditions of the random experiments we conclude $p_{e,v} = p_{e,\varphi(v)} = p_{e,v'}$. Consequently, it holds $E(P_v) = E(P_{v'})$. Hence the expected congestion for all vertices is the same, namely at most $\frac{d_G}{2}(D_H+1)$, because $|R| = \frac{d_G}{2}\cdot n$ and every path in H has at most length $D_H + 1$.

As mentioned in the proof of Theorem 3.1, the random variables used for P_v are not independent from each other. Nevertheless Claim 3.4 can be used to show that for all $k \leq \alpha \log n$, α constant, it holds:

$$E(P_v^k) \leq \left(|R| \left(\frac{e(D_H+4k)}{n}\right)\right)^k$$

Using this in the high moment version of the Markov Inequality yields that the number of paths P_v traversing a vertex v is at most $O(d_G \cdot D_H)$, w.h.p.. ∎

Let $B = O(d_G \cdot D_H)$ be an upper bound for the number of paths in $\mathcal{P}^H_{\pi\circ R}$ traversing a vertex v in H such that a path collides with at most $B \cdot D_H$ other paths in H. Suppose $G' = (V', E')$ is a graph in which each vertex represents a path in $\mathcal{P}_{\pi\circ R}$ and vertices $x, y \in V'$ are connected with each other if their respective paths collide with each other in H. Then G' has a degree of at most $d' = B \cdot D_H$. Because $d'+1$ colors suffice to color every d'-regular graph in such a way that no two adjacent vertices have the same color it is possible to attach numbers to the paths in $\mathcal{P}^H_{\pi\circ R}$ out of $[B \cdot D_H + 1]$ in such a way that no two colliding paths have the same number.

Let $\psi : R \to [B \cdot D_H + 1]$ be the function that assignes a number to all edges in R represented as paths in $\mathcal{P}^H_{\pi\circ R}$ such that the condition above is fulfilled. Then we choose the following strategy to store space-efficient routing information.

Consider a vertex v in H. Let $x = \pi^{-1}(v)$ and R_x be the set of all edges in G incident to x. Let the routing structure for v consist of the following three tables.

- $T_{v,1} : V^2 \to R_x \cup \{\emptyset\}$ is arranged in such a way that for every path $p_G(y,z)$ in $\mathcal{P}_G$ that crosses x, $T_{v,1}(y,z)$ contains the edge following x in $p_G(y,z)$.
- $T_{v,2} : R_x \to [B \cdot D_H + 1], e \to \psi(e)$ maps each edge in R_x to a suitable color.
- $T_{v,3} : [B \cdot D_H + 1] \to (E \cup \{\emptyset\})^2$ is arranged such that $T_{v,3}(k)$ contains both edges the path with number k uses to traverse v.

Clearly, it takes at most $O(d_G \log n)$ space to store $T_{v,2}$. If we apply perfect hashing techniques described in [SS90] we can reduce the size of $T_{v,3}$ from $O(B \cdot D_H \cdot \log d_H)$ to $O(B \cdot \log d_H)$ in such a way that we can still evaluate $T_{v,3}$ in constant time. Altogether this results in a routing structure of size

$$\begin{aligned} & O((d_G \cdot D_H + \log n)\log d_H + d_G \log n) \\ = \; & O((d_G \cdot D_H)\log d_H) \end{aligned}$$

because of $D_H = \Omega(\log n)$, plus the size for storing $T_{v,1}$, which depends on $\mathcal{P}_G$.

The simulation will then work as follows: Suppose, a packet P_x stored in v has to be sent to a vertex y in G. Initially, v chooses the edge $e \in T_{v,1}(x,y)$, transforms it into a number $k = T_{v,2}(e)$, and stores k in the routing information $r(x)$ of packet P_x.

If this packet is received by vertex w in H during the routing, w first checks with the help of $T_{v,3}$ whether the packet has already completed the path whose number is stored in its routing information. If this is true and the packet P_x has not reached its destination vertex yet, then w chooses the edge $e \in T_{w,1}(x,y)$, transforms it into a number $k' = T_{w,1}(e)$, and stores it in $r(x)$. Otherwise, w will ask table $T_{v,3}$ for the two edges in H which lie on the path with number k and routes the packet along the edge not used before.

Hence for a random f the routing information of each packet consists of its actual rank which can be stored in $O(\log(D_G \cdot D_H))$ bits, and a number $k \in [B \cdot D_H + 1]$ which can be stored in $O(\log(d_G \cdot D_H))$ bits, w.h.p.. This proves part (2) of Theorem 4.1. ∎

5 Space-Efficient Routing in Vertex-Symmetric Networks

We now finally prove the Main Theorem. For this we introduce the *modified s-ary Butterfly network* denoted by (s,d,k)-mBF.

Definition 5.1 *For any (s,d,k)-BF $G' = (V', E')$, let the (s,d,k)-mBF $G = (V,E)$ be an undirected graph with vertex set $V = \{(\ell, x) \mid \ell \in [d+1], x = (x_d, \ldots, x_0) \in [k] \times [s]^{k-1}\}$ and edge set*

$$\begin{aligned} E \;=\; & E' \setminus \{\{(d-1,x),(0,y)\} \mid x,y \in [k] \times [s]^{k-1}\} \\ & \cup \{\{(d-1,x),(d,(y+r) \bmod k \cdot s^{k-1})\}, \{(d,(y+r) \bmod k \cdot s^{k-1}),(0,y)\} \mid \\ & \{(d-1,x),(0,y)\} \in E', r \in [\lceil \tfrac{s}{k} \rceil]\} \end{aligned}$$

To clarify how an (s,d,k)-mBF looks like we give the following picture.

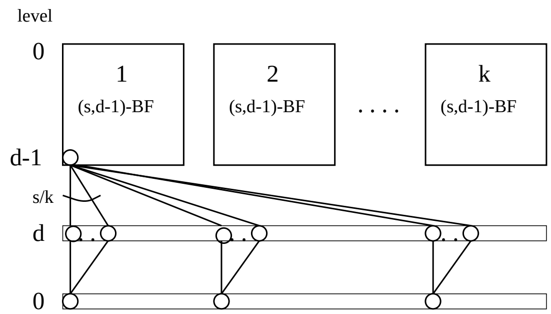

Figure 3: The structure of an (s,d,k)-mBF.

Let us perform the following simple routing strategy on an (s,d,k)-mBF:

Suppose vertex $v=(\ell,x)$ wants to send a packet P to vertex $w=(\ell',y)$. This will be done in two phases. In Phase 1, P is first sent along the vertices

$$\begin{aligned} v=(\ell,(x_{d-1},\ldots,x_0)) &\rightarrow (\ell+1,(x_{d-1},\ldots,x_{\ell+1},y_\ell,x_{\ell-1},\ldots,x_0)) \\ &\rightarrow (\ell+2,(x_{d-1},\ldots,x_{\ell+2},y_{\ell+1},y_\ell,x_{\ell-1},\ldots,x_0)) \rightarrow \ldots \\ &\rightarrow (d-2,(x_{d-1},y_{d-2},\ldots,y_\ell,x_{\ell-1},\ldots,x_0)) \\ &\rightarrow (d-1,[(y_{d-1},\ldots,y_\ell,x_{\ell-1},\ldots,x_0)+r] \bmod k\cdot s^{k-1}) \\ &\rightarrow (d,(y_{d-1},\ldots,y_\ell,x_{\ell-1},\ldots,x_0)) \rightarrow \ldots \\ &\rightarrow (\ell,(y_{d-1},\ldots,y_0)) \end{aligned}$$

where r is randomly chosen out of $[\lceil \frac{s}{k} \rceil]$.

In Phase 2, P is moved from $(\ell,(y_{d-1},\ldots,y_0))$ to $(\ell+1,(y_{d-1},\ldots,y_0))$, etc. until it reaches its destination w.

In the following section we bound the expected stage congestion for these two phases.

5.1 Bounding the Expected Stage Congestion

In this section we want to give bounds on the expected stage congestion at Phase 1, σ_1, and the expected stage congestion at Phase 2, σ_2.

Lemma 5.2 *For any (s,d,k)-mBF it holds that $\sigma_1=\frac{1}{s}$ and $\sigma_2=1$.*

Proof. Let us first consider a simple greedy routing strategy in an (s,d,k)-BF. Clearly, this strategy determines exactly one path for any pair of starting and destination vertex. Because of the symmetry properties of the resulting path system it is clear that the expected congestion for any stage is the same for every vertex within the same level. Therefore the expected congestion for any stage is the same for every vertex in any (s,d,k)-BF, namely 1. Furthermore, the Butterfly-like structure of an (s,d,k)-BF ensures during Phase 1 that, for any vertex v in level ℓ with c edges to the next higher level, each of these edges has the same probability to be chosen by a packet with random destination leaving v. In Phase 2, each vertex only has one edge to choose. Therefore, $\sigma_2=1$.

In order to bound σ_1, let us now change to the routing strategy on an (s,d,k)-mBF described above. Because this strategy is equivalent to the greedy routing strategy on an (s,d,k)-BF except for routing from level $d-1$ to d and d to 0 in Phase 1, the expected stage congestion for all edges outside these levels is $\frac{1}{s}$.

It remains to analyze the expected stage congestion of Phase 1 for all edges inbetween level $d-1$ and 0. Since the edges from level $d-1$ to 0 in an (s,d,k)-BF all have the same probability to be chosen it follows that each of the edges from level $d-1$ to d and d to 0 in an (s,d,k)-mBF have the same probability to be chosen, namely $\frac{1}{k} \cdot \frac{1}{\lceil s/k \rceil} \leq \frac{1}{s}$. Therefore, $\sigma_1 = \frac{1}{s}$. ∎

Whereas in Phase 1 the degree of the subgraph in the (s,d,k)-mBF used for routing is at most $3s$, in Phase 2 the degree of the subgraph used for routing is at most 2, since in Phase 2 the packets only use edges of type $\{(l,x),(l+1,x)\}$. Thus the congestion for all stages within Phase 1 and 2 is bounded by $C_f^* = O(\max\{\sigma_1 \cdot 3s, \sigma_2 \cdot 2\} \cdot D_H) = O(D_H)$, w.h.p..

5.2 Simulations using Butterfly Networks

We now finally prove the Main Theorem. First of all, it is easy to check that for any $s \in \{2,\ldots,n\}$ there is an (s,d,k)-BF of size m such that $|V_H| \leq m \leq 2|V_H|$. Let G be the corresponding (s,d,k)-mBF. If we attach to each vertex (ℓ,x) in G with $\ell < d$ the number $\mathrm{id}(\ell,x) = d \cdot x + \ell$ and for each vertex (d,x) the number $\mathrm{id}(d,x) = d \cdot k \cdot s^{d-1} + x$, the vertices in G are numbered consecutively in such a way that all vertices in level d have numbers greater than $n-1$. So if we force G to be embedded into H such that each vertex in H gets at most 3 vertices of G and the vertices with numbers 0 to $n-1$ in G are embedded 1 to 1 in H we have a systematic numbering for all vertices in H using only vertices of lower levels than d. (Note that a systematic numbering is necessary to avoid that vertices in H have to use additional space for storing the numbers of the other vertices.) We prevented vertices in level d to have a number less than n to ensure that $\sigma_1 = \frac{1}{s}$, even for stage 0, since only the vertices with numbers 0 to $n-1$ in G are considered to have packets at the beginning of a routing problem. It remains to show that a restriction to this kind of embedding does not hurt our analysis.

The only place where we have to consider the way G is embedded in H is in the proof of Claim 3.4. There we assume that for $i \leq \alpha \log n$ independent edges $\bar{e}_1,\ldots,\bar{e}_i$ it holds that, for a randomly chosen embedding and any fixed vertex v in H,

$$Prob(X_{\bar{e}_1,v} \cdot \ldots \cdot X_{\bar{e}_i,v} = 1) \leq \left(\frac{D_H+1}{n-4i}\right)^i$$

A similar analysis to that in Claim 3.4 shows that this bound also holds for the kind of embedding of G described above.

Since the greedy routing strategy on an (s,d,k)-BF has dilation at most $2\log_s n$ it follows that if π is a random embedding of G into H obeying the above restrictions and $\mathcal{P}^H_{\pi\circ R}$ is a randomly chosen shortest path system with dilation D, then according to Theorem 3.1 a randomly chosen routing function f can be routed in H in time $O(\log_s n \cdot D_H)$ w.h.p.. Furthermore, the routing strategy on G described above implies a path system $\mathcal{P}_G$ that needs no space in the routing structures of the vertices in H. The Main Theorem then immediately follows by choosing the strategies described in Theorem 4.1.

6 Conclusions

Changing the view point from a simulation of G by H to a space-efficient path system that supports communication between any two vertices in H we have established in this paper a way to run arbitrary parallel algorithms on H in a space-efficient way.

References

[ABLP90] B. Awerbuch, A. Bar-Noy, N. Linial, D. Peleg. Improved Routing Strategies with Succinct Tables. *Journal of Algorithms* 11, pp. 307-341, 1990.

[B93] V. Braune. *Theoretische und experimentelle Analyse von Intervall-Routing-Algorithmen.* Master Thesis, Department of Mathematics and Computer Science, University of Paderborn, 1993.

[FGS93] M. Flammini, G. Gambosi, S. Salomone. Boolean Routing. In *Proc. 7th Int. Workshop on Distributed Algorithms* (WDAG 93), LNCS 725, Springer Verlag, pp. 219-233, 1993.

[FJ88] G.N. Frederickson and R. Janardan. Designing Networks with Compact Routing Tables. *Algorithmica* 3, pp. 171-190, 1988.

[FJ90] G.N. Frederickson and R. Janardan. Space-Efficient Message Routing in c-Decomposable Networks. *SIAM Journal of Computing* 19/1, pp. 164-181, 1990

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

[LPS88] A. Lubotzky, R. Phillips, R. Sarnak. Ramanujan Graphs. *Combinatorica* 8/3, pp. 261-277, 1988.

[LMRR94] F.T. Leighton, B.M. Maggs, A.G. Ranade, S.B. Rao. Randomized routing and sorting on fixed-connection networks. *Journal of Algorithms* 17, pp. 157-205, 1994.

[M88] G.A. Margulis. Explicit Group Theoretical Constructions of Combinatorial Schemes and their Application to the Design of Expanders and Superconcentrators. *Problems Inform. Transmission* 11, pp. 39-46, 1988.

[M94] M. Morgenstern. Existence and Explicit Constructions of $q+1$ Regular Ramanujan Graphs for Every Prime Power q. *Journal of Comb. Theory, Series B* 62, pp. 44-62, 1994.

[MV95] F. Meyer auf der Heide and B. Vöcking. A Packet Routing Protocol for Arbitrary Networks. In *12th Symp. on Theoretical Aspects of Computer Science* (STACS 95), pp. 291-302, 1995.

[PU89] D. Peleg and E. Upfal. A Tradeoff between Size amd Efficiency for Routing Tables. *Journal of the ACM* 36, pp.510-530, 1989.

[R91] A.G. Ranade. How to Emulate Shared Memory. *Journal of Computer and System Sciences* 42, pp. 307-326, 1991.

[SS90] J.P. Schmidt, A. Siegel. The Spatial Complexity of Oblivious k-Probe Hash Functions. *SIAM Journal of Computing* 19/5, pp. 775-786, 1990.

[SSS93] J.P. Schmidt, A. Siegel, A. Srinivasan. Chernoff-Hoeffding Bounds for Applications with Limited Independence. In *Proc. 4th Symp. on Discrete Algorithms* (SODA 93), pp. 331-340, 1993.

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