A Packet Routing Protocol for Arbitrary Networks*

Friedhelm Meyer auf der Heide and Berthold Vöcking

Department for Mathematics and Computer Science
and Heinz Nixdorf Institute, University of Paderborn,
33098 Paderborn, Germany

Abstract. In this paper, we introduce an on-line protocol which routes any set of packets along shortest paths through an arbitrary N-node network in $O(\text{congestion} + \text{diameter} + \log N)$ rounds, with high probability. This time bound is optimal up to the additive $\log N$, and it was previously only reached for bounded-degree levelled networks.
Further, we prove bounds on the congestion of random routing problems for Cayley networks and general node symmetric networks based on the construction of shortest paths systems. In particular, we give construction schemes for shortest paths systems and show that if every processor sends p packets to random destinations along the paths described in the paths system, then the congestion is bounded by $O(p \cdot \text{diameter} + \log N)$, with high probability.
Finally, we prove an (apparently suboptimal) congestion bound for random routing problems on randomly chosen regular networks.

1 Introduction

Communication among the processors of a parallel computer usually requires a large portion of runtime of a parallel algorithm. These computers are often realized as relatively sparse networks of a large number of processors such that each processor can communicate directly only with a few neighbours. Thus, most of the communication must proceed through intermediate processors. One of the basic problems in this context is to route simultanously many *packets* through the network. Most previous theoretical research on packet routing concentrates on special classes of networks. We are interested in packet routing problems on networks with arbitrary topology.

Assume that we are given an arbitrary network with N nodes (processors) and with directed edges (channels). A *packet routing problem* on this network is defined by a set of packets. Each of the packets is assigned a destination and a source node, and the goal is to route each packet from its source to its destination. A routing problem in which every node is the source of p packets

* Supported in part by the DFG Forschergruppe 'Effiziente Nutzung massiv paralleler Systeme, Teilprojekt 4', by the Esprit Basic Research Action No. 7141 (ALCOM II), and by the Volkswagen Foundation.

and the destination of p packets is called a *p-to-p-routing problem*, and a routing problem in which every node sends p packets to random destinations chosen independently and uniformly from the set of nodes is called a *random p-routing problem*.

Our investigations are based on the *store-and-forward* model. In this model, the packets are viewed as atomic objects, and it is assumed that the routing proceeds in synchronized rounds such that each edge can transmit only one packet in each round. At the beginning of the first round, a packet is stored in an *initial queue* at its source node. During the routing it moves forward step by step, and at each node, it is stored in a *packet buffer* until it is allowed to move forward to the next node. Arriving at its destination the packet is inserted into a *final queue*. The path traversed by the packet from its source to its destination is called the *routing path* of the packet.

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

First, we turn to the second problem. A *scheduling protocol* describes the rules for moving forward packets along preset routing paths. In particular, it specifies for each node, which packets are allowed to move forward in a round and which have to wait. If the schedule is produced while the packets are routed through the network, this is called *on-line* routing. If we allow a global controller to precompute the schedule, we talk about *off-line* routing. We are interested in the construction of on-line scheduling protocols.

Of course, the following parameters greatly influence the routing time for a set of packets with preset routing paths: the *congestion C*, i.e. the maximum number of routing paths that pass through the same edge, and the *dilation D*, i.e. the maximum length of the packet's routing paths. Clearly, if packets only have to wait at an edge since another packet moves forward along this edge, then each packet waits at most $C-1$ rounds at each edge on its path. Thus, it arrives at its destination in at most $C \cdot D$ rounds. On the other hand $\max(\{C, D\}) = \Omega(C+D)$ is a lower bound, since at least one edge is traversed by C packets, and at least one packet has to traverse D edges.

The path selection problem is defined as follows. We are given the sources and the destinations of the packets, and we have to determine the routing paths. This we do by a *shortest paths system W* which is a set of N^2 shortest paths through the network. It includes a path $w(u, v) = (u \rightarrow \cdots \rightarrow v)$ for every pair u and v of nodes. For every packet with source u and destination v we choose the path $w(u, v)$ as its routing path.

In the following, we represent the underlying processor network by a digraph $\mathcal{G} = (V, E)$, where V is the set of nodes or processors, and $E \subseteq V \times V$ is the set of directed edges or channels. Of course, any network description which is based on undirected graphs can be represented in the digraph model just by replacing each undirected edge by two directed edges in opposite direction. We call a network an *undirected network*, if there is an opposite edge (v, u) for every edge (u, v).

1.1 Known Results

Leighton, Maggs, and Rao [LMR88, LMR94] show that for any set of packets whose paths have congestion C and dilation D there exists a schedule that delivers all packets in $O(C + D)$ rounds using constant-size packet buffers at each edge, thereby achieving the naive lower bound for scheduling on arbitrary networks. The proposed protocol is off-line, and the best known sequential algorithm [LM94] for computing the schedule takes time $O((PM)^{1+\epsilon})$ for any fixed constant ϵ, where P is the number of packets and M the number of edges in the network.

Further, Leighton, Maggs, and Rao [LMR94] present a probabilistic on-line scheduling protocol for routing on arbitrary networks. The protocol completes the routing of P packets along arbitrary preset paths with congestion C and dilation D in $O(C+D\log(DP))$ rounds[2] using packet buffers of size $O(\log(DP))$ at every node. This is the best known result for on-line routing on arbitrary networks.

Much better results are known for the class of bounded-degree levelled networks. In a *levelled network of depth L*, the nodes can be partitioned into levels $0, \ldots, L$ such that each edge in the network leads from some node on level i to some node on level $i + 1$ for $0 \leq i \leq L - 1$.

Ranade [Ran91] proposes a probabilistic on-line routing protocol for butterfly networks which is based on techniques developed by Valiant [Val82] and Upfal [Up84]. It can be easily extended to the class of bounded-degree levelled networks [Lei92] [LMRR94]. Ranade's protocol uses packet buffers of constant size at each edge. The protocol completes the routing of any set of packets along preset routing paths in $O(C + L + \log N)$ rounds, w.h.p.[3], where C is the congestion of the routing paths, L the depth of the network, and N the size of the network. Moreover, several standard networks can emulate a levelled network, e.g. the hypercube and the shuffle-exchange networks [LMRR94]. As a result, the above protocol routes with optimal delay on these networks using constant-size packet buffers.

Leighton [Lei92] introduces a simple probabilistic protocol for butterfly networks. It is called the *random-rank protocol*. This protocol is a simple version of Ranade's protocol. Initially, each packet is assigned a random rank. The ranks determine which packets move forward and which have to wait in a round. As Ranade's protocol, the random-rank protocol can easily be extended to the class of bounded-degree levelled networks. It routes any set of packets along preset paths in $O(C + L + \log N)$ rounds, w.h.p., using buffers of size C at each edge,

[2] Throughout this paper $\log N$ denotes $\log_2 N$ and $\log^2 N$ denotes $(\log N)^2$.

[3] Throughout this paper *w.h.p.* (with high probability) means with probability at least $1 - N^{-\alpha}$ for any fixed constant α, where N is the size of the network.

where C is the congestion of the paths, L the depth of the network, and N the size of the network.

1.2 Overview — New Results

In Section 2, we introduce a new probabilistic on-line scheduling protocol which we call *growing-rank protocol*. We show that the growing-rank protocol routes any set of packets along shortest paths with congestion C and dilation $D \leq D^*$ on an arbitrary N-node network in $O(C + D^* + \log N)$ rounds, w.h.p., where D^* is an upper bound on the dilation which is known by every node. D^* e.g. can be chosen to be the diameter of the network. Thus, we obtain the same bound for arbitrary networks as previously known only for bounded-degree levelled networks.

The protocol requires buffers of size C at each edge. If the size is bounded, then the possibility of *deadlock* arises. Suppose there are k nodes $v_0, \ldots, v_{k-1}$ with full packet buffers, and every node v_i is trying to move forward a packet to node $v_{(i+1) \bmod k}$ for $0 \leq i \leq k-1$. Then each node is blocked, i.e. a deadlock occurs. The deadlock problem does not occur on levelled networks even with constant-size buffers, since the packets move forward only in the direction of increasing levels. Avoiding deadlocks seems to be the major problem to be solved in order to generalize our results to networks with bounded buffers.

Whereas the diameter of the network is a good bound for the dilation, the congestion heavily depends on the routing problem. In Section 3, we study the congestion of random p-routing problems on node symmetric and on randomly chosen regular networks.

For node symmetric networks, we construct shortest paths systems such that the congestion of random p-routing problems is $O(p \cdot \mathit{diam} + \log N)$, w.h.p.. Combining this bound and the runtime bound of our growing-rank protocol, we achieve routing time $O(p \cdot \mathit{diam} + \log N)$ for random p-routing problems on node symmetric networks. Applying Valiant's paradigm *first routing to a random destination*, the above bounds for random p-routing problems hold for any p-to-p-routing problem as well.

Finally, we show that the congestion of random p-routing problems (w.r.t. shortest paths) on random regular networks is bounded by $O(p \cdot \log^2 N)$ with probability $1 - o(1)$. We do not believe that this result is optimal, but that $O(p \cdot \log n)$ is the truth. Getting a better result for shortest paths systems seems to be a hard problem.

2 The New Protocol

In this section, we introduce the growing-rank protocol. Suppose $\mathcal{G}$ is an arbitrary network of size N, and fix a set of packets with preset routing paths which are shortest paths in $\mathcal{G}$. We denote the congestion of the routing paths by C, the dilation of these paths by D, and we assume that each node knows upper bounds of C and D denoted by C^* and D^*. For example, we can use the total number of packets for C^* and the diameter of the network for D^*. (Better congestion bounds which depend on the underlying paths systems are given in Section 3.) Further, we assume that each packet p has a unique *ident-number*, denoted by $\mathrm{id}(p)$. For example, the ith packet in the initial queue of the jth node has the ident-number $i \cdot N + j$ for $i \geq 0$ and $0 \leq j \leq N-1$. The maximum ident-number of all packets we denote by $\mathrm{id}_{\max}$.

2.1 Description of the Growing-Rank Protocol

During the routing, a node stores all packets that wait for moving forward along an outgoing edge a in an edge buffer $\mathcal{Q}_a$. At the beginning of each round, each node examines all its edge buffers and selects one packet from each non-empty buffer. The selected packets are forwarded along their routing paths. The other packets have to wait for moving forward in a later round. Thus, a node moves forward at the most one packet over each outgoing edge in a round.

The selection of the packets is determined by random ranks. Initially, each packet is assigned an integer rank chosen randomly, independently, and uniformly from the range $[0, R-1]$. R is a multiple of D^* with $R \geq 12eC^* + 2D^* + (\alpha + 2)\log N$ for α specified later. The rank of a packet is increased by $\frac{R}{D^*}$ whenever the packet traverses an edge. If two or more packets in a buffer $\mathcal{Q}_a$ are contending to move forward along the edge a in a round, then one of those with minimum rank is chosen. Thus, for each outgoing edge a, a round looks like this:

1. choose a packet $p \in \mathcal{Q}_a$ with minimum rank,
2. increase the rank of p by $\frac{R}{D^*}$,
3. move p forward along the edge a, and finally,
4. insert all arriving packets that must move along a into $\mathcal{Q}_a$.

In order to break ties, if there are multiple packets with the same minimum rank, the one with smallest ident-number is chosen.

Remark 1. The major difference between the growing-rank protocol and the random-rank protocol is that the rank of a packet is increased whenever the packet traverses an edge. As a result, packets that are often delayed have a tendency to be preferred. In levelled networks, the increase of the ranks has only slight effect. In particular, if all packets start at level 0, then there is no effect at all, since every packet that passes level i has moved forward along i edges on its path, and thus the ranks of all packets that arrive at a node are increased by the same value.

In the following, we denote the rank of a packet p while waiting at a node v by $\mathrm{rank}^v(p)$. Further, we define the *ident-rank* of p at v as $\mathrm{rank}^v(p) + \frac{\mathrm{id}(p)}{\mathrm{id}_{\max}+1}$ and denote it by $\text{id-rank}^v(p)$. Note that, in each round, the ident-ranks of all packets are distinct. The protocol ensures that whenever a packet p delays a packet p' at a node v it is $\text{id-rank}^v(p) < \text{id-rank}^v(p')$. The following lemma shows that none of the ranks becomes greater than $2R-1$.

Lemma 2. *Suppose p is a packet which is stored at a node v in some round. Then* $\operatorname{rank}^v(p) \leq 2R-1$.

Proof. Initially, the rank of p is at most $R-1$. Since the length of the routing path of p is at most D, the rank of p is increased by $\frac{R}{D^*}$ for at most D times. Thus, $\operatorname{rank}^v(p) \leq R-1+D\cdot\frac{R}{D^*} \leq 2R-1$. □

2.2 Analysis of the Protocol

We will show that the growing-rank protocol completes the routing along arbitrary shortest paths in $O(C+D^*+\log N)$ rounds, w.h.p.. Our analysis is based on a delay sequence argument similar to that in [Lei92], [LMRR94] and [Ran91].

Definition 3 ((s,ℓ)-delay sequence). An (s,ℓ)-delay sequence consists of

1. $s+1$ not necessarily distinct *collision nodes* $v_0, v_1, \ldots, v_s$;
2. s *delay packets* $p_1, p_2, \ldots, p_s$ such that the routing path of p_i crosses the node v_i and the node v_{i-1} in that order for $1 \leq i \leq s$, and the path of p_i leaves the node v_{i-1} along the same edge as the path of p_{i-1} for $2 \leq i \leq s$;
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 node v_i to node 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 $0 \leq r_s \leq \cdots \leq r_2 \leq r_1 \leq 2R-1$.

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

Lemma 4. *Suppose the routing takes $T \geq 2D^*$ or more rounds. Then a $(T-2D^*, 2D^*)$-delay sequence is active.*

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 node v_0. We follow p_1's routing path backwards to the last node on this path where it was delayed. This node 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 node 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 node v_i to node 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 $1 \leq i \leq s$. Further, p_i leaves the node v_{i-1} along the same edge as p_{i-1}, since p_i delays p_{i-1} at this node for $2 \leq i \leq s$. Figure 1 illustrates the situation.

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

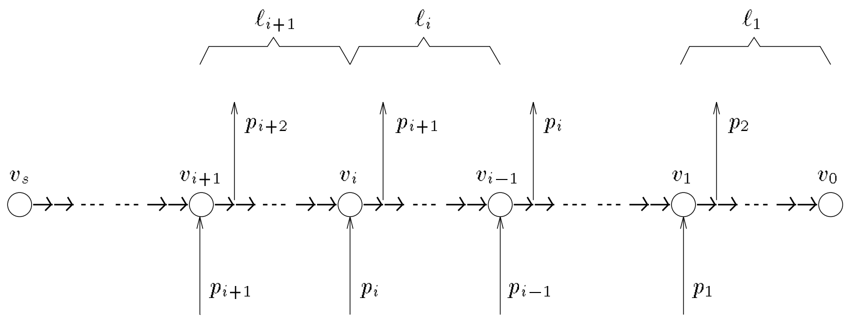

Fig. 1. The components of the delay-path.

Our next goal is to 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 $r_i + \ell_i \cdot \frac{R}{D^*} \leq r_{i-1}$ for $1 \leq i \leq s$. As a consequence,

$$\sum_{i=1}^{s} \ell_i \cdot \frac{R}{D^*} \leq r_0 \overset{\text{Lemma 2}}{\Longrightarrow} \sum_{i=1}^{s} \ell_i \leq (2R-1) \cdot \frac{D^*}{R} \leq 2D^* \ . \tag{1}$$

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

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

Consequently, if we stop the above construction at packet p_{T-2D^*}, we have found an active $(T-2D^*, 2D^*)$-delay sequence. □

Lemma 5. *If the routing paths of the packets are shortest paths, then the delay packets in the above construction are pairwise distinct.*

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

Let m denote the distance from the node v_j to the node 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,

$$\text{id-rank}^{v_i}(p) = \text{id-rank}^{v_j}(p) + m \cdot \frac{R}{D^*} \ . \tag{2}$$

On the other hand, each packet p_{k+1} delays the packet p_k at node v_k, and consequently, $\text{id-rank}^{v_k}(p_k) > \text{id-rank}^{v_k}(p_{k+1})$ for $1 \leq k \leq s-1$. 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 $\ell_{k+1} \cdot \frac{R}{D^*}$ on its path from v_{k+1} to v_k for $1 \leq k \leq s-1$. It follows that $\text{id-rank}^{v_k}(p_k) > \text{id-rank}^{v_{k+1}}(p_{k+1}) + \ell_{k+1} \cdot \frac{R}{D^*}$ for $1 \leq k \leq s-1$. This yields

$$\text{id-rank}^{v_i}(p) > \text{id-rank}^{v_j}(p) + \sum_{k=i}^{j-1} \ell_{k+1} \cdot \frac{R}{D^*} \geq \text{id-rank}^{v_j}(p) + m \cdot \frac{R}{D^*} \quad . \qquad (3)$$

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

Lemma 6. *The number of different active* (s, ℓ)*-delay sequences is at most*

$$N^2 2^\ell \left(\frac{2eC(s+2R)}{s} \right)^s \quad .$$

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

- Since $\sum_{i=1}^s \ell_i \leq \ell$, there are $\binom{s+\ell}{s}$ ways to choose the ℓ_i's.
- Once the ℓ_i's are chosen, there are at most $N^2 C^s$ choices for the delay packets and the collision nodes. This is because there are at most N possibilities to determine the node v_0, and if v_0 is fixed, there are at most $N \cdot C$ choices for the packet p_1, since v_0 has at most N incoming edges. Furthermore, if v_0, p_1 and ℓ_1 are fixed, then v_1 is fixed, since v_1 is the node on the routing path of p_1 with distance ℓ_1 to v_0.

  Now, suppose v_{i-1}, p_{i-1} and ℓ_i are fixed for $2 \leq i \leq s$. From the definition of the delay sequence, we know that p_i leaves the node v_{i-1} along the same edge as p_{i-1}. Thus, we have at most C choices for p_i. Moreover, if v_{i-1}, p_i and ℓ_i are determined, then v_i is determined as well, since it is the node on the routing path of p_i with distance ℓ_i to v_{i-1}.
- Finally, there are $\binom{s+2R-1}{s} \leq \binom{s+2R}{s}$ possibilities to choose the r_i's such that $2R - 1 \geq r_1 \geq r_2 \geq \cdots \geq r_s \geq 0$.

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

$$N^2 C^s \binom{s+\ell}{s} \binom{s+2R}{s} \quad .$$

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

$$N^2 C^s 2^{s+\ell} \left(\frac{e(s+2R)}{s} \right)^s \leq N^2 2^\ell \left(\frac{2eC(s+2R)}{s} \right)^s \quad .$$

□

Theorem 7. *Let $\mathcal{G}$ be an arbitrary network of size N. Then the growing-rank protocol completes the routing of any set of packets whose routing paths are shortest paths in $\mathcal{G}$ with congestion $C \leq C^*$ and dilation $D \leq D^*$ in $O(C + D^*) + (\alpha + 2) \log N$ rounds using buffers of size C at each edge with probability $1 - N^{-\alpha}$ for every α.*

Remark 8. The above time bound depends on the dilation bound D^*, but not on the congestion bound C^*. The bound C^* only influences the range of the ranks.

Remark 9. The shortest paths condition is necessary to show that packets can not appear twice in the delay sequence. This condition can be slightly weakened: A set of paths on a network $\mathcal{G} = (V, E)$ is said to be *shortcut-free*, if there is a subnetwork $\mathcal{G}' = (V, E')$ with $E' \subseteq E$ such that the paths in the set are shortest path in $\mathcal{G}'$. Of course, every set of shortest paths is shortcut-free. It is easy to check that Lemma 5, holds also for shortcut-free paths. Hence, the growing-rank protocol routes any set of packets whose routing paths are shortcut-free and have congestion $C \leq C^*$ and dilation $D \leq D^*$ in $O(C + D^* + \log N)$ rounds, w.h.p..

Proof of Theorem 7. The probability that a particular delay sequence with s distinct packets is active is R^{-s}. This is because a sequence with s distinct packets determines s ranks. As a consequence,

$$\begin{aligned} &\text{Prob(the routing takes } T = s - 2D^* \text{ or more rounds)} \\ \overset{\substack{\text{Lemma} \\ 4\,+\,5}}{\leq} \quad & \text{Prob}\begin{pmatrix} \text{an } (s, 2D^*)\text{-delay sequence with} \\ \text{distinct delay packets is active} \end{pmatrix} \\ \leq \quad & N^2 2^{2D^*} \left(\frac{2eC(s+2R)}{s}\right)^s \cdot R^{-s} \ . \end{aligned}$$

We choose $T = 12eC + 4D^* + (\alpha + 2)\log N$. This yields

$$s \geq 12eC \ , \tag{4}$$

$$s \geq (\alpha + 2)\log N + 2D^* \ , \text{ and} \tag{5}$$

$$R \geq s \ , \tag{6}$$

because $R \geq 12eC^* + 2D^* + (\alpha + 2)\log N$. As a consequence,

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

□

3 Congestion Bounds

The congestion is an upper bound for the running time of oblivious routing protocols. It depends on the network, on the path system, and on the routing problem. The following lemma is a simple application of a Chernoff bound. It relates the expected individual congestions of the edges to the congestion of the network, and gives a tail estimate.

Lemma 10. *Let $\mathcal{G} = (V, E)$ be an arbitrary strongly connected network of size N. Suppose the packets of a random routing problem move forward along the paths of a paths system W on $\mathcal{G}$. Let $E(C_a)$ denote the expected congestion of edge $a \in E$, i.e. the expected number of routing paths that pass through the edge a. Then the congestion of the routing paths is $O(\max(\{E(C_a) \mid a \in E\}) + \log N)$, w.h.p..*

Proof. Fix an edge $a \in E$ arbitrarily. We define $X_{v,i}$ to be 0-1-random variables such that $X_{v,i} = 1$, iff the path of the ith packet of node v traverses a for $v \in V$ and $0 \leq i \leq p-1$. Then $C_a = \sum_{v \in V} \sum_{i=0}^{p-1} X_{v,i}$. Since the random variables $X_{v,i}$ are pairwise independent, we can apply the following Chernoff bound.

$$\mathrm{Prob}(C_a \geq 2eE(C_a) + (\alpha+2)\log N) \leq 2^{-(2eE(C_a)+(\alpha+2)\log N)} \leq N^{-\alpha+2}$$

for any α (cf. [HR90]). Up to now we have bounded the congestion of the edge a. Taking into account all edges, we can bound the congestion C as follows.

$$\mathrm{Prob}(C \geq 2eE(C_a) + (\alpha+2)\log N) \leq |E| \cdot N^{-\alpha+2} \leq N^{-\alpha}$$

for any α. □

3.1 Node Symmetric Networks

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

Cayley Networks. The class of Cayley networks is an important subclass of node symmetric networks. Many standard networks belong to this class, for instance the multidimensional arrays (generalized hypercubes), the cube-connected-cycles, the wrapped butterflies, the bubble-sort networks, and the star networks.

Definition 11 (Cayley network). Let Γ be a finite algebraic group with identity 1, and suppose Σ is a set of generators of Γ with $1 \notin \Sigma$. Then the Cayley network $\mathcal{G}_{\Gamma,\Sigma} = (V, E)$ is defined by

$$V = \Gamma \quad \text{and} \quad E = \{(a, b) \mid a^{-1}b \in \Sigma\} \ .$$

Definition 12 (symmetric paths system). Let W be a paths system on a network $\mathcal{G} = (V, E)$. Then we call W symmetric, if given any two nodes u und v of $\mathcal{G}$ there is a permutation $\psi : V \longrightarrow V$ such that for every path $(w_0 \to w_1 \to \cdots \to w_\ell) \in W$ with $w_i = u$ there is a path $(\psi(w_0) \to \psi(w_1) \to \cdots \to \psi(w_\ell)) \in W$ with $\psi(w_i) = v$ for $0 \le i \le \ell$.

Roughly speaking, a *symmetric paths system* has the property that it looks the same viewed from any node of the network.

Lemma 13. *For every Cayley network, there is a symmetric shortest paths system.*

Proof. Let $\mathcal{G}_{\Gamma,\Sigma} = (V, E)$ be a Cayley network. Then there is a transitive automorphism group U of size $|V|$ [Biggs93]. We denote by ϕ_u^v the automorphism of U which maps the node u onto the node v for $u, v \in V$. Thus, $U = \{\phi_u^v \mid v \in V\}$ for any $u \in V$.

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

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

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

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

In the first step we have chosen N *prototype paths* (inclusive the trivial one from 0 to 0). In the second step we have made $N - 1$ copies of each prototype path. Thus, every automorphism of U, except for the identity, has been used once for copying each prototype path. In a full version of this paper, we show that this paths system is symmetric. □

The following theorem can be concluded using Lemmata 10 and 13.

Theorem 14. *Let $\mathcal{G}_{\Gamma,\Sigma} = (V, E)$ be a Cayley network of size N. Let W be a symmetric shortest paths system. Then the congestion of a random p-routing problem is bounded by $O(p \cdot \operatorname{diam}(\mathcal{G}_{\Gamma,\Sigma}) + \log N)$, w.h.p..*

Node Symmetric Non-Cayley Networks. For bounding the congestion in Theorem 14 we used a symmetric paths system. As seen, this can be easily constructed for Cayley networks. For non-Cayley node symmetric networks, like for example the Petersen graph [Yap86], the construction in the proof of Lemma 13 fails. In a full version of this paper, we show how to reach the same congestion bound as in Theorem 14 for general node symmetric networks.

3.2 Random Regular Networks

An undirected network is called $\varDelta$-regular if the degree of each node is $\varDelta$. Define

$$G(N,\varDelta) := \left\{ \mathcal{G} = (V,E) \;\middle|\; \begin{matrix} \mathcal{G} \text{ is a } \varDelta\text{–regular undirected network} \\ \text{with } V = \{0,1,\ldots,N-1\} \end{matrix} \right\} .$$

Bollobás and de la Vega [BV81] show that the diameter of a network which is chosen randomly from $G(N,\varDelta)$ is at most $\log_{\varDelta-1} N + \log_{\varDelta-1} \ln N + 5$ with probability $1 - o(1)$. In a full paper, we use this bound to show the following theorem.

Theorem 15. *Choose $\mathcal{G} = (V,E)$ randomly from $G(n,\varDelta)$, where $\varDelta$ is a fixed constant. Suppose W is an arbitrary shortest path system on $\mathcal{G}$. Then the congestion of a random p-routing problem is $O(p \cdot \log^2 N)$ with probability* $1 - o(1)$.

References

[Biggs93] N.L. Biggs, *Algebraic graph theory*, Second Edition, Cambridge University Press (Cambridge 1993).

[BV81] B. Bollobás and W. Fernandez de la Vega, The diameter of random regular graphs, *Combinatorica* 2 (2) (1982) pp. 125–134.

[HR90] T. Hagerup and C. Rüb, A guided tour of Chernoff bounds, *Information Processing Letters* 33/6 (1989/90) pp. 305–308.

[Lei92] F.T. Leighton, *Introduction to parallel algorithms and architectures: arrays · trees · hypercubes*, Morgan Kaufmann Publishers (San Mateo, CA 1992).

[LM94] F.T. Leighton and B.M. Maggs, Fast algorithms for finding O(congestion + dilation) packet routing schedules, Unpublished Manuscript (1994).

[LMR88] F.T. Leighton, B.M. Maggs, and S.B. Rao, Universal packet routing algorithms (Extended Abstract), *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, IEEE (White Plains, NY 1988) pp. 256–271.

[LMR94] F.T. Leighton, B.M. Maggs, and S.B. Rao, Packet routing and job-shop scheduling in O(congestion + dilation) steps, *Combinatorica* 14 (2) (1994) pp. 167-186.

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

[Ran91] A.G. Ranade, How to emulate shared memory, *Journal of Computer and System Sciences* 42 (1991) pp. 307–326.

[Up84] E. Upfal, Efficient schemes for parallel communication, *Journal of the Association for Computing Machinery* Vol. 31, No. 3 (July 1984) pp. 507–517.

[Val82] L.G. Valiant, A scheme for fast parallel communication, *SIAM Journal on Computing* 11/2 (1982) pp. 350–361.

[Yap86] H.P. Yap, *Some topics in graph theory*, Cambridge University Press (Cambridge 1986).

This article was processed using the LaTeX macro package with LLNCS style