11 Universal Routing Protocols

Efficient communication mechanisms are a prerequisite to exploit the performance of large parallel and distributed systems. The problem of moving information from specified sources to specified sinks is called *routing*. The routing problem involves choosing the right paths through the network and scheduling the packet movements along these paths. In this section, we will assume that appropriate routing paths are given so that we only have to focus on finding proper scheduling protocols.

11.1 Notation

In the following, we assume that the communication network can be modeled as an undirected graph $G = (V, E)$. Time is divided into time units and in each time unit, every edge can be crossed by at most one packet in each direction. A routing path of a packet must form a contiguous sequence of edges in G. Usually, these routing paths are *simple*, i.e., every edge is used at most once in a path. Routing a single packet through a network is easy, but when routing multiple packets, local decision rules are needed that determine which packet to prefer if multiple packets want to cross the same link at the same time.

In the *universal routing problem* we are given an arbitrary collection of n simple routing paths with congestion C and dilation D, and the goal is to find a schedule for the movement of packets along these paths, one per path, so that the time until all packets have reached their destinations is minimized. The *dilation* is the length (i.e., the number of edges) of the longest path in the collection, and the *congestion* is defined as the maximum, over all edges e in the network, of the number of paths using e in the same direction.

A classical result in routing theory is that for every collection of simple routing paths with congestion C and dilation D there exists a schedule with runtime $O(C + D)$. Since $\max\{C, D\}$ is a lower bound for any schedule, this is best possible up to a constant factor. An important question has been whether this time bound can also be achieved *online*, i.e., the nodes in the given graph can compute locally, without any coordination and preprocessing, in which order to forward the packets so that a runtime as close to $C + D$ as possible can be achieved. In this section, we will present some of these protocols. All of them are randomized, which comes as no surprise, because it is known that deterministic routing protocols perform poorly in the worst case. In the following, $[x]$ means the set $\{0, \ldots, x - 1\}$ for any $x \in \mathbb{N}$.

11.2 The Random Delay Protocol

The oldest online protocol that deviates only by a factor logarithmic in n, C and D from a best possible runtime of $O(C+D)$ for arbitrary path collections is the protocol presented by Leighton, Maggs and Rao in [2]. We present an extension of it, called here the *random delay protocol*, that can route packets along an arbitrary simple path collection of size n with congestion C and dilation D in $O(C + D \log n)$ steps, w.h.p. (the protocol in [2] requires $O(C + D \log(nD))$ time steps, w.h.p.).

Description of the Protocol

The protocol assumes that all links have bandwidth B (fixed later), that is, up to B packets can traverse a link at one time step. Clearly, each time step for links with bandwidth B can be simulated in B time steps by links with bandwidth 1. The following algorithm is used as a basic building block for the random delay protocol.

> **Algorithm Route(ℓ):**
> Each packet is assigned an initial delay, chosen uniformly and independently at random from the range $[C/\log n]$. A packet that is assigned a delay of δ waits in its initial buffer for δ steps and then moves on without waiting again until it reaches its destination or traversed ℓ links. If more than B packets want to use the same link at the same time then all of them stop.

The random delay protocol works as follows.

> **repeat**
> execute Route($\min\{D, n\}$)
> **until** all packets reached their destinations

Theorem 11.1 *Suppose we are given an arbitrary simple path collection $\mathcal{P}$ of size n with congestion C and dilation D. Then the random delay protocol needs at most $O(C + D\log n)$ time steps to finish routing in $\mathcal{P}$, w.h.p.*

Proof. Let us consider some fixed edge e and time step t during the execution of Route(ℓ). Since at most C packets want to traverse e and each of these packets chooses an initial delay independently at random from a range of size $C/\log n$, the probability that at least $B = \max\{\alpha + 2, 2\mathrm{e}\}\log n + 2$ packets want to traverse e at time step t is at most

$$\binom{C}{B}\left(\frac{1}{C/\log n}\right)^B \leq \left(\frac{\mathrm{e}\log n}{B}\right)^B \leq \left(\frac{1}{2}\right)^{(\alpha+2)\log n+2} = \frac{1}{4n^{\alpha+2}} \ .$$

Let us say that a packet P *fails* at edge e if at least B other packets want to use e at the same time as P. Then the probability that P fails at least $k = \lceil D/n \rceil$ times during the execution of the random delay protocol is bounded by

$$\binom{D+k}{k}\left(\frac{1}{4n^{\alpha+2}}\right)^k \leq \left(\frac{4D}{k}\right)^k\left(\frac{1}{4n^{\alpha+2}}\right)^k \leq \left(\frac{1}{n^{\alpha+1}}\right)^k \leq \frac{1}{n^{\alpha+1}} \ .$$

Since there are n packets to consider, the probability that there exists a packet with at least k failures is at most $n \cdot n^{-\alpha-1} = n^{-\alpha}$. Hence, w.h.p. the random delay protocol successfully routes all packets along the given path collection in time

$$\begin{aligned} & B \cdot (\ell + C/\log n) \cdot (D/\ell + k) \\ & = O((C + \min\{D, n\}\log n) \cdot (D/\min\{D, n\} + \lceil D/n \rceil)) \\ & \overset{C \leq n}{=} O(C + D\log n) \ . \end{aligned}$$

This completes the proof of Theorem 11.1. □

Limitations

The runtime bound of the random delay protocol holds for arbitrary, even non-simple, path collections. However, the definition of C must be changed for non-simple path collections in a way that if a packet traverses an edge e q times then it has to count q times for the congestion at e.

In the following we show how a multiplicative factor of $\log n$ in the time bound can be avoided when routing packets along more restricted classes of path collections. Let us start by demonstrating this for leveled path collections.

11.3 The Random Rank Protocol

A leveled network is a network in which the nodes can be separated into levels $L_0, L_1, L_2, \ldots$ with the property that every edge connects two nodes of consecutive levels. If the leveled network consists of levels from L_0 to L_D, we say its *depth* is D. A *leveled path* in such a network is a path that starts at some node in L_i for some i and then follows edges along the levels $L_i, L_{i+1}, L_{i+2}, \ldots$ until it ends in some level L_j, $j > i$.

The *random rank protocol* has its origin in papers by Aleliunas and Upfal and can be found in a similar form as described below in Leighton's book [1]. It routes packets along an arbitrary leveled path collection of size n with congestion C and depth D in $O(C + D + \log n)$ steps, w.h.p., using edge buffers of size C. (Note that the depth of a leveled path collection is the number of levels formed by its nodes, and not necessarily its dilation.)

Description of the Protocol

At the beginning, every packet p gets a random rank denoted by $\text{rank}(p)$ that is stored in its routing information. We require $\text{rank}(p)$ to be chosen uniformly and independently from the choices of the other packets from some fixed range $[K]$ (K will be determined later). Additionally, each packet stores an identification number $\text{id}(p) \in [n]$ in its routing information that is different from all identification numbers of the other packets. The random rank protocol uses the following contention resolution rule.

> **Priority rule:**
> It two or more packets contend to use the same link at the same time then the one with minimal rank is chosen.

If two packets have the same rank then, in order to break ties, the one with the lowest id wins. The protocol then works as follows in each time step

> For each link with nonempty buffer, select a packet according to the priority rule and send it along that link.

For the random rank protocol the following time bound has been shown (see, e.g., [1]).

Theorem 11.2 *Suppose we are given a leveled path collection $\mathcal{P}$ of size n with congestion C and depth D. Let $K \geq 8C$. Then the random rank protocol needs at most $O(C + D + \log n)$ time steps to finish routing in $\mathcal{P}$, w.h.p., using edge buffers of size C.*

Proof. Consider the runtime of the random rank protocol to be at least $T \geq D + s$. We want to show that it is very improbable that s is large. For this we need to find a structure that witnesses a large s. This structure should become more and more unlikely to exist the larger s becomes.

Let p_1 be a packet that arrived at its destination v_1 in step T. We follow the path of p_1 backwards until we reach a link e_1, where it was delayed the last time. Let us denote the length of the path from the destination of p_1 to e_1 (inclusive) by ℓ_1, and the packet that delayed p_1 by p_2. From e_1 we follow the path of p_2 backwards until we reach a link e_2 where p_2 was delayed the last time by some packet p_3. Let us denote the length of the path from e_1 (exclusive) to e_2 (inclusive) by ℓ_2. We repeat this construction until we arrive at a packet p_{s+1} that prevented the packet p_s at edge e_s from moving forward. Altogether it holds for all $i \in \{1, \ldots, s\}$: packet p_{i+1} leaves the buffer of e_i at time step $T - \sum_{j=1}^{i}(\ell_j + 1) + 1$, and prevents at that time step p_i from moving forward.

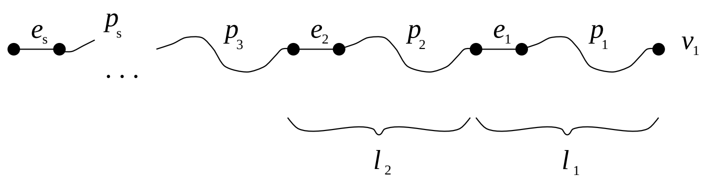

Figure 1: The structure of a delay path.

The path from e_s to v_1 recorded by this process in reverse order is called *delay path* (see Figure 1). It consists of s contiguous parts of routing paths of length $\ell_1, \ldots, \ell_s \geq 0$ with $\sum_{i=1}^{s} \ell_i \leq D$. Because of the contention resolution rule it holds $\text{rank}(p_i) \geq \text{rank}(p_{i+1})$ for all $i \in \{1, \ldots, s\}$. A structure that contains all these features is defined as follows.

Definition 11.3 (s-delay sequence) *An* s-delay sequence *consists of*

- s *not necessarily different* delay links $e_1, \ldots, e_s$;
- $s + 1$ delay packets $p_1, \ldots, p_{s+1}$ *such that the path of* p_i *traverses* e_i *and* e_{i-1} *in that order for all* $i \in \{2, \ldots, s\}$*, the path of* p_{s+1} *contains* e_s*, and the path of* p_1 *contains* e_1*;*
- s *integers* $\ell_1, \ldots, \ell_s \geq 0$ *such that* ℓ_1 *is the number of links on the path of* p_1 *from* e_1 *(inclusive) to its destination, for all* $i \in \{2, \ldots, s\}$ ℓ_i *is the number of links on the path of* p_i *from* e_i *(inclusive) to* e_{i-1} *(exclusive), and* $\sum_{i=1}^{s} \ell_i \leq D$*; and*
- $s + 1$ *integers* $r_1, \ldots, r_{s+1}$ *with* $0 \leq r_{s+1} \leq \ldots \leq r_1 < K$*.*

A delay sequence is called active *if for all* $i \in \{1, \ldots, s+1\}$ *we have* $\text{rank}(p_i) = r_i$*.*

Our observations above yield the following lemma.

Lemma 11.4 *Any choice of the ranks that yields a routing time of* $T \geq D + s$ *steps implies an active* s*-delay sequence.*

Proof. Suppose the random rank protocol needs $T \geq D+s$ steps. Then we get for $\sum_{i=1}^{s} \ell_i \leq D$ that $T \geq \sum_{i=1}^{s} \ell_i + s$ and therefore $T - \sum_{i=1}^{s} \ell_i - s \geq 0$. Hence we can construct an active delay sequence of length s such that packet p_{s+1} leaves the buffer of e_s at time step $T - \sum_{i=1}^{s}(\ell_i+1)+1 \geq 1$. From this the lemma follows. □

Lemma 11.5 *The number of different s-delay sequences is at most*

$$n \cdot C^s \cdot \binom{D+s}{s} \cdot \binom{s+K}{s+1} \ .$$

Proof. There are at most $\binom{D+s}{s}$ possibilities to choose the ℓ_i such that $\sum_{i=1}^{s} \ell_i \leq D$. Furthermore, there are n packets from which p_1 can be chosen. Since p_1 and ℓ_1 determine the link e_1 and the congestion at e_1 is at most C, there are at most C possibilities to choose packet p_2. The same holds for the packets $p_3, \ldots, p_{s+1}$ at the edges $e_2, \ldots, e_s$. Hence we altogether have at most $\binom{D+s}{s} \cdot n \cdot C^s$ possibilities to choose the delay packets. Finally, there are at most $\binom{s+K}{s+1}$ ways to select the r_i such that $0 \leq r_{s+1} \leq \ldots \leq r_1 < K$. □

Note that during the execution of the random rank protocol the packets have a unique ordering w.r.t. their priority levels. (If two or more packets have the same rank, then the id's of the packets are compared.) Hence the packets in an s-delay sequence must be different. Since the packets choose their ranks independently at random, the probability that an s-delay sequence is active is $1/K^{s+1}$. Thus

$$\begin{aligned}
&\Pr[\text{The random rank protocol needs at least } D+s \text{ steps}] \\
&\overset{\text{Lemma 11.4}}{\leq} \quad \Pr[\text{there exists an active } s\text{-delay sequence}] \\
&\overset{\text{Lemma 11.5}}{\leq} \quad n \cdot C^s \cdot \binom{D+s}{s} \cdot \binom{s+K}{s+1} \cdot \frac{1}{K^{s+1}} \\
&\leq \quad n \cdot C^s \cdot 2^{D+s} \cdot 2^{s+K} \cdot \frac{1}{K^{s+1}} \\
&\leq \quad n \cdot 2^{2s+D+K} \cdot \left(\frac{C}{K}\right)^s
\end{aligned}$$

If we set $K \geq 8C$ and $s = K + D + (\alpha+1)\log n$, where $\alpha > 0$ is an arbitrary constant, then

$$\begin{aligned}
&\Pr[\text{The random rank protocol needs at least } D+s \text{ steps}] \\
&\leq \ n \cdot 2^{2s+D+K} \cdot 2^{-3s} = n \cdot 2^{-s+D+K} = \frac{1}{n^\alpha}
\end{aligned}$$

which concludes the proof of Theorem 11.2. □

Limitations

The following observation shows that there are simple path systems for which the random rank protocol performs poorly. Its proof can be found in [3].

Observation 11.6 *There exists a simple path collection of size n with dilation $D = O(\log n/\log\log n)$ and congestion $C = O(\log n/\log\log n)$, where the expected routing time of the random rank protocol is bounded by $\Omega((\log n/\log\log n)^{3/2})$.*

The path collection used for this observation consists of many subcollections of paths. Each subcollection consists of a linear array of length D, with loops of length $\sqrt{D}$ between adjacent nodes (see Figure 2).

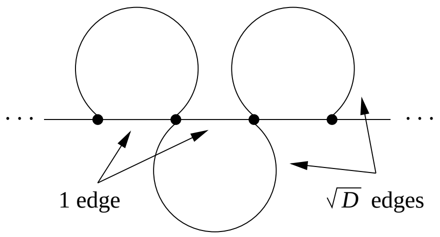

Figure 2: The counterexample.

The packets traversing each subcollection of paths are broken into $\sqrt{D}$ groups numbered 0 through $\sqrt{D}-1$ of $\sqrt{D}$ packets each. The packets in group i use the linear array for $i\sqrt{D}$ steps and then use $\sqrt{D}-i$ loops as their path. Note that if, for all $i \geq 0$, the packets in group i have smaller ranks than the packets in groups with larger numbers, than the packets in group i delay the packets in group $i+1$ by $D-(i+1)\sqrt{D}+i$ steps.

11.4 The Growing Rank Protocol

Now we present a protocol that routes packets along an arbitrary shortcut-free path collection of size n with congestion C and dilation D in $O(C+D+\log n)$ steps, w.h.p., using buffers of size C. It is called *growing rank protocol* [4] and works as follows.

Description of the Protocol

Initially, each packet is assigned an integer rank chosen randomly, independently, and uniformly from $[K]$. For each step, the protocol works as follows.

For each link with nonempty buffer,

- choose a packet p according to the priority rule,
- increase the rank of p by K/D, and
- move p forward along the link.

We generalize the result in [4] by analyzing the performance of the growing rank protocol for the following type of path collections.

Definition 11.7 *A path collection $\mathcal{P}$ is called d-*shortcut-free *if any piece of length at most d of a path in $\mathcal{P}$ can not be shortcut by any combination of other pieces of paths in $\mathcal{P}$.*

We can show the following theorem.

Theorem 11.8 *Suppose we are given a d-shortcut-free path collection $\mathcal{P}$ of size n with congestion C and dilation D, $d \le D$. Let $K \ge 8C$. Then the growing rank protocol needs at most $O(C + \max\{1, \frac{\log(nD)}{d}\}D)$ time steps, w.h.p., to finish routing in $\mathcal{P}$.*

Proof. Similar to the proof of Theorem 11.2 we want to find a structure that witnesses a long runtime of the growing rank protocol. First we introduce the following definitions.

In the following, we denote the rank of a packet p while waiting to traverse a link e by $\text{rank}^e(p)$. Let id : {set of packets} $\to [n]$ be an arbitrary bijective function. We define the *ident-rank* of p at e as $\text{rank}^e(p) + \frac{\text{id}(p)}{n}$ and denote it by $\text{id-rank}^e(p)$. 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 link e it holds $\text{id-rank}^e(p) < \text{id-rank}^e(p')$. The following lemma shows that the rank of any packet can not be greater than $2K - 1$ during the routing.

Lemma 11.9 *Suppose p is a packet which is stored in the buffer of link e in some round. Then* $\text{rank}^e(p) \le 2K - 1$.

Proof. At the beginning, the rank of p is at most $K - 1$. Since the length of the routing path of p is at most D, the rank of p is increased by $\frac{K}{D}$ for at most D times. Thus, $\text{rank}^e(p) \le K - 1 + D \cdot \frac{K}{D} \le 2K - 1$. □

Analogous to the proof for the random rank protocol, the following delay sequence will serve as a witness for a long runtime of the growing rank protocol.

Definition 11.10 ((s, ℓ, K)-delay sequence) *An (s, ℓ, K)-*delay sequence *consists of*

1. *s not necessarily distinct* delay links *$e_1, \dots, e_s$;*
2. *$s+1$* delay packets *$p_1, p_2, \dots, p_{s+1}$ such that the path of p_i moves along the link e_i and the link e_{i-1} in that order for all $i \in \{2, \dots, s\}$, the path of p_{s+1} contains e_s, and the path of p_1 contains e_1;*
3. *s integers $\ell_1, \ell_2, \dots, \ell_s \ge 0$ such that ℓ_1 is the number of links on the routing path of packet p_1 from e_1 (inclusive) to its destination, for all $i \in \{2, \dots, s\}$ ℓ_i is the number of links on the routing path of packet p_i from link e_i (inclusive) to link e_{i-1} (exclusive), and $\sum_{i=1}^{s} \ell_i \le \ell$; and*
4. *s integer keys $r_1, r_2, \dots, r_{s+1}$ such that $0 \le r_{s+1} \le \cdots \le r_2 \le r_1 < 2K$.*

We call s the length *of the delay sequence. Further we say that a delay sequence is* active*, if* $\text{rank}^{e_i}(p_i) = r_i$ *for all $i \in \{1, \dots, s\}$ and* $\text{rank}^{e_s}(p_{s+1}) = r_{s+1}$

Lemma 11.11 *Suppose the routing takes $T \ge 2D$ or more rounds. Then there exists an active $(T - 2D, 2D, K)$-delay sequence.*

Proof. First, we give a construction scheme for a delay sequence. Let p_1 be a packet that arrived at its destination v_1 in step T. We follow p_1's routing path backwards to the last link on this path where it was delayed. We call this link e_1, and the length of the path from v to e_1 (inclusive) ℓ_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 link e_2 at which p_2 was forced to wait, because the packet p_3 was preferred. Let us call the length of the path from e_1 (exclusive) to e_2 (inclusive) ℓ_2. We change the packet again and follow the path of p_3 backwards. We can continue this construction until we arrive at a packet p_{s+1} that prevented the packet p_s at edge e_s from moving forward.

The path from e_s to v_1 recorded by this process in reverse order is called *delay path*. It consists of contiguous parts of routing paths. In particular, the part of the delay path from link e_i (inclusive) to link e_{i-1} (exclusive) is a subpath of the routing path of packet p_i.

Let $r_i = \mathrm{rank}^{e_i}(p_i)$ for all $1 \leq i \leq s$ and $r_{s+1} = \mathrm{rank}^{e_s}(p_{s+1})$. Because of the contention resolution rule we have $0 \leq r_{s+1} \leq \ldots \leq r_1$, and $r_1 \leq 2K - 1$ because of Lemma 11.9. Thus, we have constructed an active (s, ℓ, K)-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 $r_1, \ldots, r_{s+1}$, we denote by r_0 the rank of p_1 at its destination. It follows immediately from the protocol that $r_i + \ell_i \cdot \frac{K}{D} \leq r_{i-1}$ for all $1 \leq i \leq s$. As a consequence,

$$\sum_{i=1}^{s} \ell_i \cdot \frac{K}{D} \leq r_0 \overset{\text{Lemma 11.9}}{\Longrightarrow} \sum_{i=1}^{s} \ell_i \leq (2K-1) \cdot \frac{D}{K} \leq 2D \ . \tag{1}$$

Since the delay sequence consists of $\sum_{i=1}^{s} \ell_i$ moves and s delays, it covers at most $t = \sum_{i=1}^{s} \ell_i + s$ time steps. It follows that

$$t = \sum_{i=1}^{s} \ell_i + s \overset{(1)}{\leq} 2D + s \ .$$

Consequently, if we stop the above construction at packet p_{T-2D+1}, we still have $t \leq T$ and therefore found an active $(T-2D, 2D, K)$-delay sequence. $\square$

Instead of considering the whole delay sequence, we will only consider a piece of it that is chosen in such a way that we can be sure that no packet can appear twice in it. For this we introduce the following definition.

Definition 11.12 ((s', ℓ', K')-delay subsequence) *An* (s', ℓ', K')-delay subsequence *consists of*

1. *s' not necessarily distinct* delay links *$e_1, \ldots, e_{s'}$;*
2. *$s'+1$* delay packets *$p_1, p_2, \ldots, p_{s'+1}$ such that the path of p_i moves along the link e_i and the link e_{i-1} in that order for all $i \in \{2, \ldots, s'\}$, the path of $p_{s'+1}$ contains $e_{s'}$, and the path of p_1 contains e_1;*
3. *s' integers $\ell_1, \ell_2, \ldots, \ell_{s'} \geq 0$ such that ℓ_i is the number of links on the routing path of packet p_i from link e_i (inclusive) to link e_{i-1} (exclusive) for all $i \in \{2, \ldots, s'\}$, and $\sum_{i=2}^{s'} \ell_i \leq \ell'$; and*
4. *s' integer keys $r_1, r_2, \ldots, r_{s'+1}$ such that $0 \leq r_{s'+1} \leq \cdots \leq r_2 \leq r_1 < r_{s'+1} + 2K'$ and $r_1 < 2K$.*

We say that a delay subsequence is active, *if* $\text{rank}^{e_i}(p_i) = r_i$ *for all* $i \in \{1, \ldots, s'\}$ *and* $\text{rank}^{e_{s'}}(p_{s'+1}) = r_{s'+1}$

The following lemma will be helpful to bound the total delay, length, and delay range of a subsequence of a delay sequence. Its proof is similar to a proof in [5] (see Lemma 2.10).

Lemma 11.13 *If there exists an active* (s, ℓ, K)*-delay sequence, then there exists an active* $(\frac{s}{2\alpha}, \frac{2\ell}{\alpha}, \frac{2K}{\alpha})$*-delay subsequence for every* $\alpha \geq 1$.

Proof. Suppose that an (s, ℓ, K)-delay sequence is active. Divide the packet sequence $p_2, \ldots, p_{s+1}$ into α contiguous subsequences such that each subsequence has at least $\lfloor s/\alpha \rfloor \geq s/2\alpha$ packets. This also partitions the delay path into subpaths. Let subsequence 0 consist only of packet p_1. For every subsequence $i \geq 1$, let ℓ_i denote the length of the ith subpath and let $2K_i$ denote the delay range of ranks for the ith subsequence, i.e., $2K_i$ is the difference between the rank of the last packet in subsequence $i-1$ when delayed by the first packet in subsequence i, and the rank of the last packet in subsequence i when delaying the second last. We know that there must be fewer than $\alpha/2$ segments with $K_i > 2K/\alpha$, since $\sum 2K_i \leq 2K$. Furthermore there must be fewer than $\alpha/2$ segments satisfying $\ell_i > 2\ell/\alpha$, since $\sum \ell_i \leq \ell$. Thus there must exist some segment for which $\ell_i \leq 2\ell/\alpha$ and $K_i \leq 2K/\alpha$. □

Next we show that, if we restrict $\frac{2K}{\alpha}$ to be at most $\frac{d}{2} \cdot \frac{D}{K}$, then no packet can appear twice in a $(\frac{s}{2\alpha}, \frac{2\ell}{\alpha}, \frac{2K}{\alpha})$-delay subsequence.

Lemma 11.14 *For any* (s', ℓ', K')*-delay subsequence with* $K' \leq \frac{d}{2} \cdot \frac{D}{K}$ *it holds that no packet can appear twice in it.*

Proof. Suppose, in contrast to our claim, that there is some packet p appearing twice in an (s', ℓ', K')-delay sequence. Then there exist i and j with $1 \leq i < j \leq s' + 1$ and $p = p_i = p_j$. Thus, the routing path of p crosses the delay path at the delay links e_{j-1} and e_i in that order. Since the rank of a packet is increased by $\frac{K}{D}$ each time it traverses an edge and the range of the ranks is bounded by $d \cdot \frac{D}{K}$, the length of the path the packet p traverses from e_{j-1} (inclusive) to e_i (exclusive) can be at most d. Let m denote the distance from link e_{j-1} (inclusive) to link e_i (exclusive) in the delay path. Since the routing paths are d-shortcut-free, the rank of p is increased at most m times while moving from e_{j-1} to e_i, and hence,

$$\text{id-rank}^{e_i}(p) \;\leq\; \text{id-rank}^{e_{j-1}}(p) + m \cdot \frac{K}{D} \;. \tag{2}$$

On the other hand, since for every $k \in \{1, \ldots, s'\}$ packet p_{k+1} delays packet p_k at edge e_k, we have $\text{id-rank}^{e_k}(p_k) > \text{id-rank}^{e_k}(p_{k+1})$ for all $k \in \{1, \ldots, s'\}$. Further, the length of the routing path of packet p_{k+1} from e_{k+1} to e_k is ℓ_{k+1}, and thus the rank of p_{k+1} is increased by $\ell_{k+1} \cdot \frac{K}{D}$ on its path from e_{k+1} to e_k for all $k \in \{1, \ldots, s'-1\}$. It follows that $\text{id-rank}^{e_k}(p_k) > \text{id-rank}^{e_{k+1}}(p_{k+1}) + \ell_{k+1} \cdot \frac{K}{D}$ for all $k \in \{1, \ldots, s'-1\}$. This yields

$$\begin{aligned} \text{id-rank}^{e_i}(p) \;&>\; \text{id-rank}^{e_{j-1}}(p) + \sum_{k=i+1}^{j-1} \ell_k \cdot \frac{K}{D} \\ &=\; \text{id-rank}^{e_{j-1}}(p) + m \cdot \frac{K}{D} \;. \end{aligned} \tag{3}$$

Since (3) contradicts (2), there is no packet that appears twice in the delay subsequence. $\square$

Our goal is therefore to restrict the range of the ranks used in the delay subsequence to be considered to at most $d \cdot \frac{D}{K}$. First we count the number of ways to construct an (s', ℓ', K')-delay subsequence.

Lemma 11.15 *The number of different (s', ℓ', K')-delay subsequences is at most*

$$n \cdot D \cdot 2K \cdot C^{s'} \cdot \binom{\ell' + s'}{s'} \cdot \binom{s' + 2K'}{s' + 1} .$$

Proof. There are n packets from which p_1 can be chosen, and at most D possibilities to choose ℓ_1. Furthermore there are at most $\binom{\ell'+s'}{s'}$ possibilities to choose the ℓ_i such that $\sum_{i=2}^{s} \ell_i \leq \ell'$. Since p_1 and ℓ_1 determine the link e_1 and the congestion at e_1 is at most C, there are at most C possibilities to choose packet p_2. The same holds for the packets $p_3, \ldots, p_{s'+1}$ at the edges $e_2, \ldots, e_{s'}$. Hence we altogether have at most $n \cdot D \cdot \binom{\ell'+s'}{s'} \cdot C^{s'}$ possibilities to choose the delay packets. Finally, there are at most $2K\binom{s'+2K'}{s'+1}$ ways to select the r_i such that $0 \leq r_{s'+1} \leq \ldots \leq r_1 < r_{s'+1} + 2K'$ and $r_1 < 2K$. $\square$

Since the packets choose their ranks independently at random, the probability that an (ℓ', s', K')-delay subsequence is active is $1/K^{s'+1}$. Thus

$$\begin{aligned}
&\Pr[\text{there exists an active } (\ell', s', K')\text{-delay subsequence}] \\
&\leq n \cdot D \cdot 2K \cdot C^{s'} \cdot \binom{\ell' + s'}{s'} \cdot \binom{s' + 2K'}{s' + 1} \cdot \frac{1}{K^{s'+1}} \\
&\leq n \cdot D \cdot C^{s'} \cdot 2^{\ell'+s'} \cdot 2^{s'+2K'} \cdot \frac{1}{K^{s'}} \\
&= n \cdot D \cdot 2^{2s'+\ell'+2K'} \cdot \left(\frac{C}{K}\right)^{s'}
\end{aligned}$$

If we set $K \geq 8C$ and $s' \geq \ell' + 2K' + (\beta+1)\log n + \log D$, where $\beta > 0$ is an arbitrary constant, then

$$\begin{aligned}
&\Pr[\text{there exists an active } (\ell', s', K')\text{-delay subsequence}] \\
&\leq n \cdot D \cdot 2^{2s'+\ell'+2K'} \cdot 2^{-3s'} = n \cdot D \cdot 2^{-s'+\ell'+2K'} \leq \frac{1}{n^{\beta}}
\end{aligned}$$

With $K' = \frac{d}{2} \cdot \frac{K}{D}$ we get from Lemma 11.13 that $\frac{d}{2} \cdot \frac{K}{D} = \frac{2K}{\alpha}$ and therefore $\alpha = \frac{4D}{d}$. Since any (s, ℓ, K)-delay sequence can have at most $2D$ edges, it holds that $\ell' \leq \frac{4D}{\alpha} = d$, which has to be ensured for our analysis to work. Therefore the total delay s of the growing rank protocol is at most

$$\begin{aligned}
2\alpha s' &= 2\frac{4D}{d}\left(d + d \cdot \frac{K}{D} + (k+1)\log n + \log D\right) \\
&= O\left(D + C + \frac{\log(nD)}{d} D\right) .
\end{aligned}$$

This concludes the proof of Theorem 11.8. $\square$

Limitations

In case of bounded buffers, deadlocks can arise. Furthermore, the following observation can be shown (see [6]).

Observation 11.16 *Suppose C satisfies $\log n / \log\log n \le C \le n^{\epsilon}$ for some constant $\epsilon < 1$ and $C \ge D/\log\log n$. Then there is a simple path system of size n with dilation D and congestion C such that the expected routing time of the growing rank protocol is bounded by $\Omega(C + D \cdot \log n/\log\log n)$.*

The observation uses the following path collection.

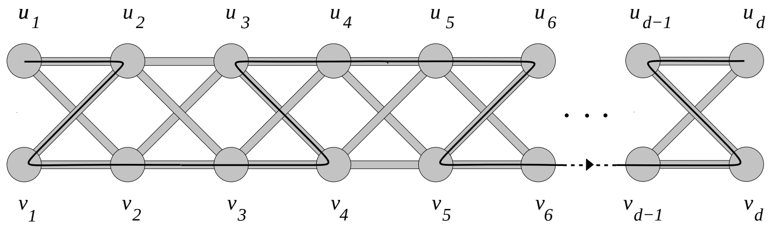

Figure 3: The counterexample.

Let A and B be two sets of packets of size $C/2$ with source node u_1 and v_1 respectively. The routing path of the packets in $\mathcal{A}$ is

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

and the routing path of the packets in $\mathcal{B}$ is

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

Since this path collection is at most 4-shortcut-free, the observation demonstrates that the analysis of the growing rank protocol above is nearly tight. Observation 11.16 further shows that the growing rank protocol can not be efficiently applied to arbitrary simple path collections. Note that it is still an open problem whether efficient shortcut-free path systems exist for any network. In case of shortest path systems, however, networks exist such that any shortest path system has a much higher expected congestion than the best simple path system (consider the union of a mesh and a complete binary tree).

References

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

[2] F.T. Leighton, B.M. Maggs, S.B. Rao. Universal packet routing algorithms. In *Proc. of the 29th Ann. Symp. on Foundations of Computer Science*, pp. 256-271, 1988.

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

[4] F. Meyer auf der Heide and B. Vöcking. A packet routing protocol for arbitrary networks. In *12th Symp. on Theoretical Aspects of Computer Science*, pp. 291-302, 1995.

[5] 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.

[6] C. Scheideler. *Universal Routing Strategies for Interconnection Networks.* Springer Lecture Notes in Computer Science, 1390, 1998.