5 Online Switching

We saw in the previous section that for every collection of simple paths there is a schedule to send packets along these paths, one packet per path, so that in at most $O(C+D)$ time steps all packets reach their destinations. The question we want to investigate in this section is whether a similar runtime can also be achieved for the online case. By "online" we mean that the system has no global knowledge about the routing problem. At the beginning, every processor only knows the packets that start in it (and, maybe, a few global parameters such as the congestion of the path collection). A processor can only use the information it receives during the routing. Thus, the processors may have do base their switching decisions on imcomplete information. We will show that nevertheless efficient switching can be done.

5.1 A Switching Protocol for Directed Acyclic Graphs

A *directed acyclic graph (or short DAG)* is a directed graph that has no directed cycle. We demand that any path chosen in a DAG for some routing problem must traverse its edges in the direction of their orientation, that is, it must be directed. We say that a DAG has a *depth* of D if the maximum number of edges a directed path can have in it is equal to D.

Suppose that we are given a collection of n paths in a DAG. Along each of these paths a packet has to be sent. Suppose w.l.o.g. that these packets are numbered from 1 to n. (Any numbering that ensures that no two packets have the same number would suffice.) The *random rank protocol* sends the packets along their paths in the following way:

At the beginning, every packet i chooses uniformly and independently at random an integer r_i out of some range $[K]$ (K will be determined later). Let the *rank* of packet i be defined as $\text{rank}(i) = r_i + \frac{i}{n+1}$. These ranks are used in the following way to resolve conflicts among the packets.

> For every time step and every edge with nonempty buffer, select the packet with minimum rank waiting at it and send it along that edge.

5.2 Analysis of the Switching Protocol

The following time bound has been shown by Leighton, Maggs, Ranade, and Rao.

Theorem 5.1 *Suppose we are given an arbitrary path collection $\mathcal{P}$ of size n and congestion C in a DAG of depth D. Let $K \geq 8C$. Then the random rank protocol needs at most $O(C+D+\log n)$ time steps, with high probability, to send all the packets to their destinations.*

Proof. Suppose that the runtime of the random rank protocol is equal to $T \geq D+S$. 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. The structure we are looking for will be the result of the following backwards argument that is similar to the argument used in Section 1.

Let p_1 be some packet that arrived at its destination v_0 in step T. We follow p_1 backwards in time until we reach an edge e_1 where it was delayed the last time before it reached its destination. Let us denote the length (i.e., the number of edges) 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. Starting from e_1, we now follow p_2 backwards in time until we reach an edge e_2 where it was delayed by some other packet. We call this packet p_3 and denote the length of the path from e_1 (exclusive) to e_2 (inclusive) by ℓ_2. Afterwards, we follow p_3 backwards in time, and so on, until we arrive at a packet p_{s+1} that prevented the packet p_s at edge e_s from moving forward. (We will specify later how to choose s in relation to S.) The path from e_s to v_0 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 $\sum_{i=1}^{s} \ell_i \leq D$, since no directed path can be longer than D. Because of the contention resolution rule it holds that $\mathrm{rank}(p_i) > \mathrm{rank}(p_{i+1})$ for all $i \in \{1, \ldots, s\}$. A structure that contains all of these features is defined as follows.

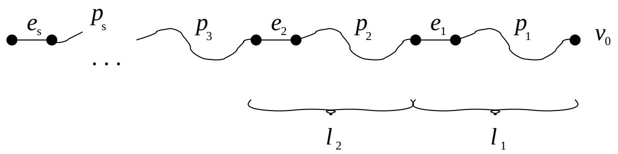

Figure 1: The structure of a delay path.

Definition 5.2 (delay sequence) *A* delay sequence of length s *consists of*

- s *not necessarily different* delay edges $e_1, \ldots, e_s$*;*
- $s+1$ 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 edges in 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 edges in 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* $1 \leq r_{s+1} \leq \ldots \leq r_1 \leq K$*.*

A delay sequence is called active *if*

1. *for all* $i \in \{1, \ldots, s\}$ *we have* $\mathrm{rank}(p_{i+1}) < \mathrm{rank}(p_i)$ *and*
2. *for all* $i \in \{1, \ldots, s+1\}$ *we have* $\lfloor \mathrm{rank}(p_i) \rfloor = r_i$*.*

Now, how does s relate to S? The following lemma gives an answer to this.

Lemma 5.3 *If the runtime of the random rank protocol is at least* $D+s$*, then there must exist an active delay sequence of length* s*.*

Proof. Suppose the random rank protocol needs $T \geq D+s$ steps. Since for any delay sequence of length s we have $\sum_{i=1}^{s} \ell_i \leq D$, it holds that $T \geq \sum_{i=1}^{s} \ell_i + s$ and therefore $T - \sum_{i=1}^{s} \ell_i - s \geq 0$. In this case, the backwards argument we used above is guaranteed to end with a packet p_{s+1}

that delayed p_s at time step $T - \sum_{i=1}^{s}(\ell_i + 1) + 1 \geq 1$, which implies that there must be such a packet p_{s+1}. Due to the contention resolution rule, the packets must fulfill requirement (1) of a delay sequence to be active. Requirement (2) can be fulfilled by simply setting $r_i = \lfloor \text{rank}(p_i) \rfloor$. $\square$

Thus, if no active delay sequence of length s can be constructed from the given path collection and the ranks chosen by the packets, then the runtime of the random rank protocol can be at most $D+s$. Hence, all we need to do is to bound the probability that it is possible to construct a delay sequence of length s. For this we need the following two lemmata.

Lemma 5.4 *The number of different delay sequences of length s 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 edge 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 $1 \leq r_{s+1} \leq \ldots \leq r_1 \leq K$. $\square$

Lemma 5.5 *No packet can appear twice in an active delay sequence.*

Proof. Since the ranks of the packets do not change during the routing, the lemma follows directly from requirement (1) for a delay sequence to be active. $\square$

Because the packets choose their ranks independently at random, Lemma 5.5 implies that the probability that some fixed delay sequence of length s is active (or more precisely, that requirement (2) for a delay sequence to be active is fulfilled) is at most $1/K^{s+1}$. Thus,

$$\begin{aligned}
&\Pr[\text{The random rank protocol needs at least } D+s \text{ steps}] \\
&\overset{\text{Lemma 5.3}}{\leq} \quad \Pr[\text{there exists an active delay sequence of length } s] \\
&\overset{\text{Lemmata 5.4, 5.5}}{\leq} \quad n \cdot C^s \cdot \binom{D+s}{s} \cdot \binom{s+K}{s+1} \cdot \frac{1}{K^{s+1}} \\
&\leq \quad n \cdot C^s \cdot 2^{D+s} \cdot 2^{s+K} \cdot \frac{1}{K^{s+1}} \\
&\leq \quad n \cdot 2^{2s+D+K} \cdot \left(\frac{C}{K}\right)^s
\end{aligned}$$

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

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

which concludes the proof of Theorem 5.1. $\square$

5.3 References

- T. Leighton, B. Maggs, A. Ranade, and S. Rao. Randomized routing and sorting in fixed-connection networks. *Journal of Algorithms* 17:157–205, 1994.