4 Offline Switching

Suppose that a collection of simple (i.e. loop free) paths is given to us. Along each path one packet has to be sent. All we know about the path collection is that is has a congestion of C and a dilation of D. In this case it is clear that at least $\max[C, D]$ steps are needed to send the packets to their destinations. On the other hand, for any greedy method (every time when at least one packet is waiting to traverse a link, this link forwards a packet) at most $C \cdot D$ steps are needed. However, would it be also possible to send the packets to their destinations in $O(C+D)$ steps? We will investigate this problem in this section.

4.1 Probabilistic Methods

First, we introduce two methods that will be important for proving the existence of good switching strategies.

Lemma 4.1 (Chernoff Bounds) *Let $X_1, \ldots, X_n$ are independent binary random variables, and let $X = \sum_{i=1}^{n} X_i$. Then it holds for all $\delta \geq 0$ and $\mu \geq \mathrm{E}[X]$ that*

$$\begin{aligned} \Pr[X \geq (1+\delta)\mu] &\leq \left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \\ &\leq \mathrm{e}^{-\frac{\delta^2 \mu}{2(1+\delta/3)}} \\ &\leq \mathrm{e}^{-\min[\delta^2,\, \delta]\cdot\mu/3} , \end{aligned} \tag{1}$$

where e *denotes the Euler number* 2.718.... *Furthermore, it holds for all $0 \leq \delta \leq 1$ and $\mu \leq \mathrm{E}[X]$ that*

$$\begin{aligned} \Pr[X \leq (1-\delta)\mu] &\leq \left(\frac{\mathrm{e}^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu} \\ &\leq \mathrm{e}^{-\delta^2 \mu/2} . \end{aligned} \tag{2}$$

The Chernoff bounds are very useful for proving that certain outcomes occur with high probability. Sometimes, however, it is not known whether some outcome can occur at all. In this case the Lovász Local Lemma may be used.

Lemma 4.2 (Lovász Local Lemma, Symmetric Case) *Let $A_1, \ldots, A_n$ be "bad" events in an arbitrary probability space. Suppose that every event A_i is independent of all the other events A_j but at most d, and that $\Pr[A_i] \leq p$ for all $1 \leq i \leq n$. If*

$$\mathrm{e}p(d+1) \leq 1 , \tag{3}$$

then $\Pr[\bigcap_{i=1}^{n} \bar{A}_i] > 0$, i.e. it is possible for no bad event to hold.

4.2 Offline Switching

The central result of this section is the following theorem.

Theorem 4.3 (Leighton, Maggs, Rao) *For any collection of paths with congestion C and dilation D, there is an offline schedule that needs at most $(C + D) \cdot 2^{O(\log^*(C+D))}$ time steps to send a packet along each of these paths.*

Proof. W.l.o.g. we assume in the following that $\max[C, D]$ is beyond some constant value. Otherwise, we simply use a brute force schedule to arrive at a schedule of time $O(C + D)$.

We will present a proof given by Leighton, Maggs and Rao. Our strategy for constructing an efficient schedule is to make a succession of refinements, starting with an initial schedule S_0 in which each packet is forwarded at every time step until it reaches its destination (see Figure 1).

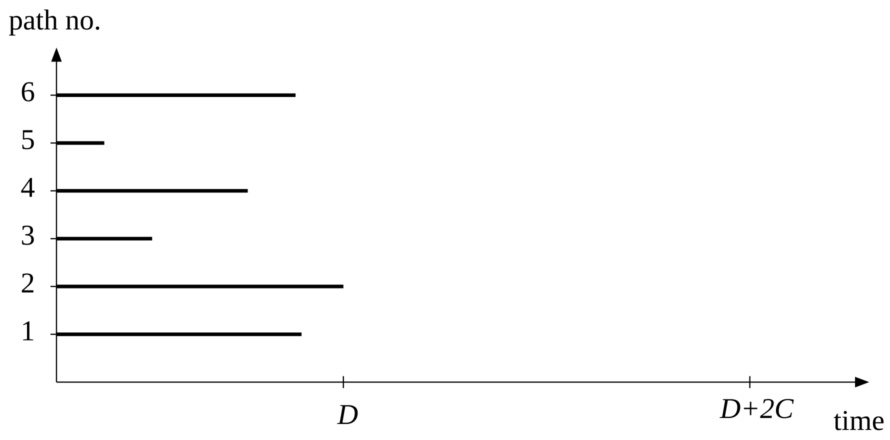

Figure 1: Schedule S_0.

This initial schedule is as short as possible. Its length is exactly D. Unfortunately, up to C packets may have to use an edge at a single time step in S_0, whereas in the final schedule at most one packet is allowed to use an edge at each step. Each refinement will bring us closer to meeting this requirement by bounding the congestion within smaller and smaller time intervals. The proof uses the Chernoff bounds and the Lovász Local Lemma at each refinement step. In the following, a *t-interval* denotes a time interval consisting of t consecutive time steps. A *t-frame* is defined as a t-interval that starts at an integer multiple of t.

Let $I_0 = \max[C, D]$ and $I_j = \lceil 18(\ln I_{j-1} + 1)\rceil$ for all $j \geq 1$. (Since $I_{j-1} \geq 1$, this means that $I_j \geq 18$ for all j.) The first step is to assign an initial delay to each packet, chosen independently and uniformly at random from the range $\Delta_1 = [2C]$. In the resulting schedule, S_1, a packet that is assigned a delay of δ waits in its source node for δ steps, then moves on without waiting again until it reaches its destination (see Figure 2). The length of S_1 is at most $D + 2C$. We use the Lovász Local Lemma to show that if the delays are chosen independently and uniformly at random and I_1 is sufficiently large, then with nonzero probability the congestion at any edge in any I_1-interval is at most I_1. Thus, such a set of delays must exist.

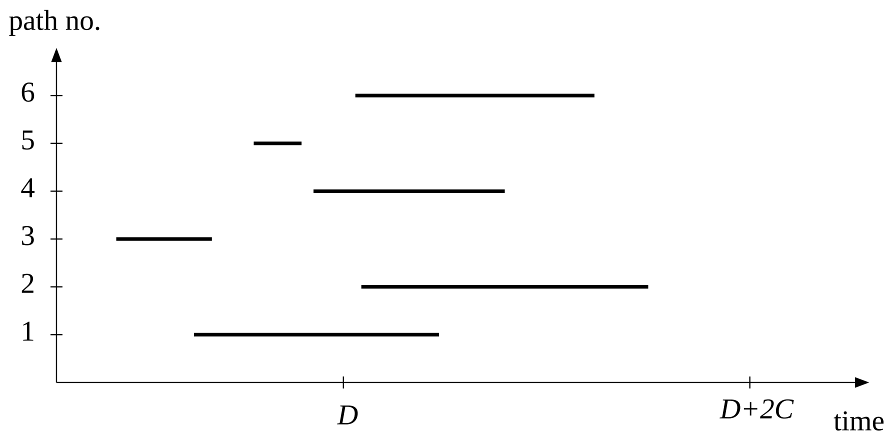

Figure 2: Schedule S_1.

To apply the Lovász Local Lemma, we associate a bad event with each edge. The bad event for edge e is that more than I_1 packets use e in some I_1-interval. For this we have to bound the dependence b among the bad events and the probability p that a bad event occurs.

We first bound the dependence. Whether or not a bad event occurs depends solely on the delays assigned to the packets that pass through the corresponding edge. Thus, two bad events are independent unless some packet passes through both of the corresponding edges. Since at most C packets pass through an edge and each of these packets passes through at most D other edges, the dependence b of the bad events is at most $C \cdot D$.

Next we bound the probability that a bad event occurs. Let us consider some fixed edge e and I_1-interval J. Let the packets traversing e be numbered from 1 to $m \leq C$. For every $i \in \{1, \ldots, m\}$, let the binary random variable X_i be one if and only if packet i traverses e during J. Let $X = \sum_{i=1}^{m} X_i$. Since the packets choose their delays uniformly at random from a range of size $2C$, we get $\Pr[X_i = 1] \leq I_1/(2C)$ for all $i \in \{1, \ldots, m\}$. Hence,

$$\mathrm{E}[X] = \sum_{i=1}^{m} \mathrm{E}[X_i] \leq I_1/2 \ .$$

Together with the Chernoff bounds (see Lemma 4.1, (1)) we therefore get with $\epsilon = 1$ that

$$\Pr[X \geq I_1] = \Pr[X \geq (1+\epsilon)I_1/2] \leq \mathrm{e}^{-I_1/6} \leq \mathrm{e}^{-3(\ln(\max[C,D])+1)} = (\mathrm{e} \cdot \max[C,D])^{-3} \ .$$

Since for each edge there are at most $D+2C$ many I_1-intervals to consider, the probability that a bad event occurs for edge e is bounded by

$$p \leq (D+2C) \cdot (\mathrm{e} \cdot \max[C,D])^{-3} < \frac{1}{\mathrm{e}(C \cdot D + 1)}$$

for $\max[C,D] \geq 18$. Thus, $\mathrm{e}p(b+1) \leq 1$ and therefore, by the Lovász Local Lemma (see Lemma 4.2, (3)), there is some assignment of delays such that the congestion in each I_1-interval is bounded by I_1.

We now break schedule S_1 into I_1-frames and continue to schedule each frame independently. So each frame can be viewed as a separate scheduling problem where the origin of a packet is its location at the beginning of the frame, and its destination is its location at the end of the frame. Our next refinement step will be to choose, for each frame, a random initial delay for each packet that visits at least one edge within this frame. In the resulting schedule S_2, the frames (enlarged by their delay ranges) are executed one after the other in a way that a packet that is assigned a delay δ in some frame F waits at its first edge in F for (additional) δ steps, and then moves on without waiting until it traverses its last edge in F (see Figure 3).

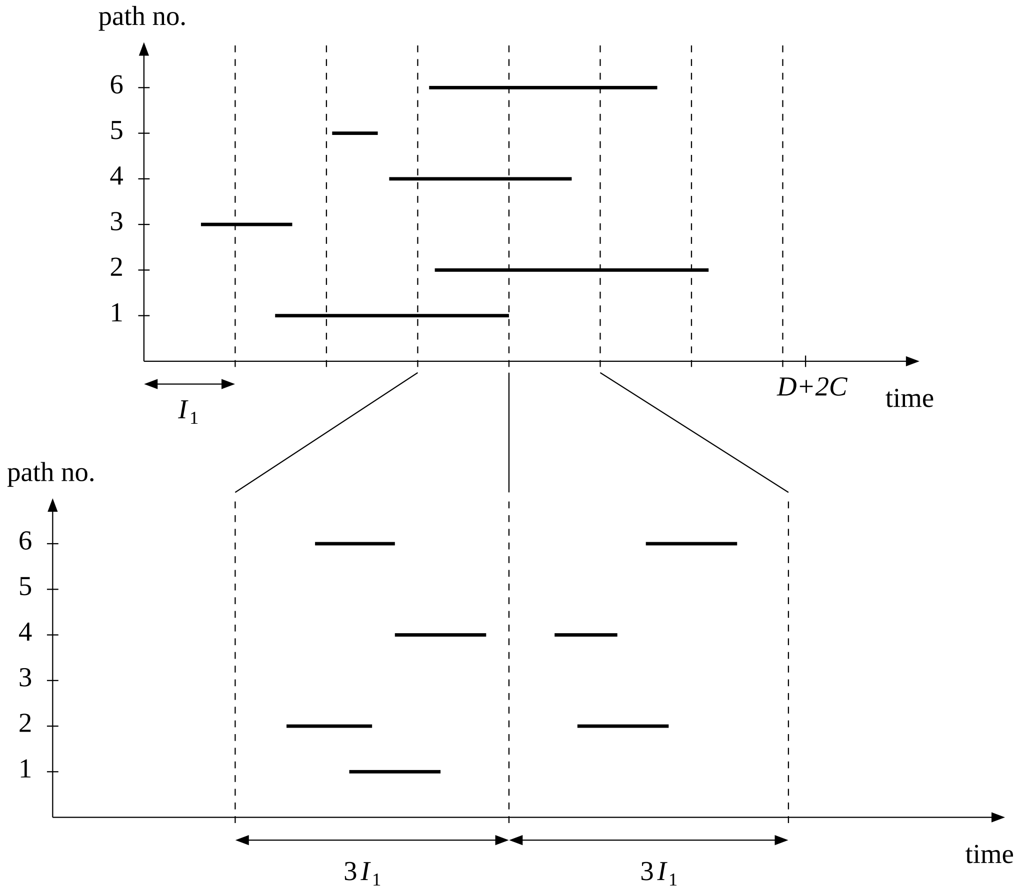

Figure 3: Refinement from schedule S_1 to schedule S_2.

We concentrate in the following on some fixed frame F. Let each packet choose an additional delay out of the range $\Delta_2 = [2I_1]$. Hence, the length of the resulting schedule for this frame is at most $3I_1$. We use the Lovász Local Lemma to show that if the delays are chosen independently and uniformly at random, then with nonzero probability the congestion in any I_2-interval is at most I_2. We again associate a bad event with each edge. The bad event for edge e is that more than I_2 packets use e in some I_2-interval.

We first bound the dependence of these events. Whether or not a bad event occurs solely depends on the delays assigned to the packets that pass through the corresponding edge. Since at most I_1 packets pass through an edge in any frame and each of these packets passes through at most I_1 other edges, the dependence of the bad events is at most I_1^2.

Next we bound the probability that a bad event occurs. Consider some fixed edge e and I_2-interval J. Let the packets traversing e in F be numbered from 1 to $m \le I_1$. For every $i \in \{1, \dots, m\}$, let the binary random variable X_i be one if and only if packet i traverses e during J. Let $X = \sum_{i=1}^m X_i$. Since the packets choose their delays uniformly at random from a range of size $2I_1$, we get $\Pr[X_i = 1] \le I_2/(2I_1)$ for all $i \in \{1, \dots, m\}$. Hence,

$$\mathrm{E}[X] = \sum_{i=1}^{m} \mathrm{E}[X_i] \le I_1 \cdot \frac{I_2}{2I_1} = I_2/2 \ .$$

Together with the Chernoff bounds we therefore get with $\epsilon = 1$ that

$$\Pr[X \ge I_2] = \Pr[X \ge (1+\epsilon) I_2/2] \le \mathrm{e}^{-I_2/6} \le \mathrm{e}^{-3(\ln I_1 + 1)} = (\mathrm{e} \cdot I_1)^{-3} \ .$$

Since for each edge there are at most $3I_1$ many I_2-intervals to consider, the probability that a bad event occurs for edge e is bounded by

$$p \le 3I_1 \cdot (\mathrm{e} \cdot I_1)^{-3} < \frac{1}{\mathrm{e}(I_1^2 + 1)}$$

for $I_1 \ge 18$. Thus, $\mathrm{e}p(b+1) \le 1$ and therefore, by the Lovász Local Lemma, there is some assignment of delays such that the congestion in each I_2-interval is bounded by I_2.

We continue to refine each I_2-frame in F recursively until we reach a round k in which $I_{k+1} \ge I_k/18$, i.e. I_k is a constant. We end up with a schedule S_k with total length at most

$$(D + 2C) \prod_{j=1}^{k-1} 3 = (D + C) \cdot 2^{O(\log^*(C+D))}$$

and with a congestion C_k in each I_k-interval of at most I_k. In order to obtain a valid schedule, we simply schedule each I_k-frame in schedule S_k in a greedy way. Since each I_k-frame has a constant congestion and dilation, we altogether end up with a routing schedule of runtime $(C + D) \cdot 2^{O(\log^*(C+D))}$. $\square$

4.3 References

- T. Leighton, B. Maggs, and S. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica* 2:167–186, 1994. (Conference article: Universal packet routing algorithms. In *Proc. of the 29th Symposium on Foundations of Computer Science (FOCS)*, pp. 256–271, 1988.)