"You were proving this on a friday evening? You need to go get a life!".

— Da Chuang, Colleague Extraordinaire†

D

Proofs for Chapter 5

In this appendix, we will prove that a buffered crossbar with a speedup of two using arbitrary input and output scheduling algorithms achieves 100% throughput. We will use the traffic models and definitions that were defined in Appendix B.

D.1 Achieving 100% Throughput in a Buffered Crossbar - An Outline

Figure D.1 shows the scheduling phases in a buffered crossbar with a speedup of two. The two scheduling phases each consist of two parts: input scheduling and output scheduling. In the input scheduling phase, each input (independently and in parallel) picks a cell to place into an empty crosspoint buffer. In the output scheduling phase, each output (independently and in parallel) picks a cell from a non-empty crosspoint buffer to take from.

We know that the scheduling algorithm in a buffered crossbar is determined by the input and output scheduling policy that decides how inputs and outputs pick cells in the scheduling phases. The randomized algorithm that we considered in Chapter 5 to achieve 100% throughput was as follows:

†Da Chuang, Humorous Moments in Proof, Stanford, 2002.

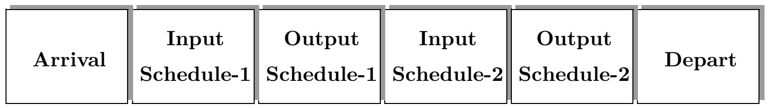

Figure D.1: *The scheduling phases for the buffered crossbar. The exact order of the phases does not matter, but we will use this order to simplify proofs.*

Randomized Algorithm: *In each scheduling phase, the input picks any non-empty VOQ, and the output picks any non-empty crosspoint.*

We will adopt the following notation and definitions. The router has N ports, and VOQ_{ij} holds cells at input i destined for output j. X_{ij} is the occupancy of VOQ_{ij},[1] and $Z_{ij} = X_{ij} + B_{ij}$ is the sum of the number of cells in the VOQ and the corresponding crosspoint. We will assume that all arrivals to input $i \in 1, 2, 3, ..., N$ are Bernoulli i.i.d. with rate λ_i, and are destined to each output $j \in 1, 2, 3, ...N$ with probability λ_{ij}. We will denote the arrival matrix as $A \equiv [\lambda_{ij}]$, where for all i, j,

$$\lambda_i = \sum_{j=1}^{N} \lambda_{ij}, \lambda_j = \sum_{i=1}^{N} \lambda_{ij}, 0 \leqslant \lambda_{ij} < 1. \tag{D.1}$$

We will also assume that the traffic is admissible (see Definition B.1), *i.e.*, $\sum_i \lambda_{i,j} < 1$, $\sum_j \lambda_{i,j} < 1$. In what follows, we will show that the buffered crossbar can give 100% throughput. The result is quite strong in the sense that it holds for any arbitrary work-conserving input and output scheduling policy with a speedup of two. In other words, each input i can choose to serve any non-empty VOQ for which $B_{ij} = 0$, and each output j can choose to serve any crosspoint for which $B_{ij} = 1$.

First we describe an intuition and outline of the proof. Then, in the next section, we will give a rigorous proof.

[1] We will see later that other queueing structures are useful and that it is not necessary to place cells in VOQs.

Theorem D.1. *(Sufficiency) A buffered crossbar can achieve 100% throughput with speedup two for any Bernoulli i.i.d. admissible traffic.*

Proof. **Intuition and Outline:** For each VOQ_{ij}, let C_{ij} denote the sum of the cells waiting at input i and the cells waiting at all inputs destined to output j (including cells in the crosspoint for output j),

$$C_{ij} = \sum_{k} X_{ik} + \sum_{k} (X_{kj} + B_{kj}). \tag{D.2}$$

It is easy to see that when VOQ_{ij} is non-empty (*i.e.*, $X_{ij} > 0$), then C_{ij} decreases in every scheduling phase. There are two cases:

- **Case 1:** $B_{ij} = 1$**.** Output j will receive one cell from the buffers destined to it, and $\sum_k (X_{kj} + B_{kj})$ will decrease by one.
- **Case 2:** $B_{ij} = 0$**.** Input i will send one cell from its VOQs to a crosspoint, and $\sum_k X_{ik}$ will decrease by one.[2]

With $S = 2$, C_{ij} will decrease by two per time slot. When the inputs and outputs are not oversubscribed, the expected increase in C_{ij} is strictly less than two per time slot. So the expected change in C_{ij} is negative over the time slot, and this means that the expected value of C_{ij} is bounded. This in turn implies that the expected value of X_{ij} is bounded and the buffered crossbar has 100% throughput. ❒

[2] If a cell from VOQ_{ij} is sent to crosspoint B_{ij}, then $\sum_k (X_{kj} + B_{kj})$ stays the same at the end of the input scheduling phase, since X_{ij} decreases by one and B_{ij} increases by one. In the output schedule, *Case 1* applies and C_{ij} will further decrease by one. As a result, if a cell from VOQ_{ij} is sent to crosspoint B_{ij}, then C_{ij} decreases by two in that scheduling phase.

D.2 Achieving 100% Throughput in a Buffered Crossbar - A Rigorous Proof

Lemma D.1. *Consider a system of queues whose evolution is described by a discrete time Markov chain (DTMC) that is aperiodic and irreducible with state vector* $Y_n \in \mathbb{N}^M$*. Suppose that a lower bounded, non-negative function* $F(Y_n)$*, called Lyapunov function,* $F : \mathbb{N}^M \rightarrow \mathbb{R}$ *exists such that* $\forall Y_n, E[F(Y_{n+1})|Y_n] < \infty$*. Suppose also that there exist* $\gamma \in \mathbb{R}^+$ *and* $C \in \mathbb{R}^+$*, such that* $\forall ||Y_n|| > C$,

$$E[F(Y_{n+1}) - F(Y_n)|Y_n] < -\gamma, \tag{D.3}$$

then all states of the DTMC are positive recurrent and for every $\epsilon > 0$*, there exists* $B > 0$ *such that* $\lim_{n\rightarrow\infty} Pr\{\sum_{i,j} X_{i,j}(n) > B\} < \epsilon$.

Proof. This is a straightforward extension of Foster's criteria and follows from [39, 216, 217, 218]. ❐

We will use the above lemma in proving Theorem D.1. We are now ready to prove the main theorem, which we repeat here for convenience.

Theorem D.2. *(Sufficiency) Under an arbitrary scheduling algorithm, the buffered crossbar gives 100% throughput with speedup of two.*

Proof. In the rest of the proof we will assume that all indices i, j, k vary from $1, 2, ..N$. Denote the occupancy of VOQ_{ij} at time n by $X_{ij}(n)$. Also, let Z_{ij} denote the combined occupancy of the VOQ_{ij} and the crosspoint B_{ij} at time n. By definition, $Z_{ij}(n) = X_{ij}(n) + B_{ij}(n)$.

Define,

$$f_1(n) = \sum_{i,j} X_{ij}(n)\Big(\sum_k X_{ik}(n)\Big), \tag{D.4}$$

$$f_2(n) = \sum_{i,j} Z_{ij}(n)\Big(\sum_k Z_{kj}(n)\Big), \tag{D.5}$$

$$F(n) = f_1(n) + f_2(n). \tag{D.6}$$

Observe that from Equation D.4

$$\begin{aligned} f_1(n) &= \sum_{i,j} X_{ij}(n)\Big(\sum_k X_{ik}(n)\Big) \\ &= \sum_{i,j,k} X_{ij}(n) X_{ik}(n). \end{aligned}$$

Denote $D_{ij}(n) = 1$ if a cell departs from VOQ_{ij} at time n and zero otherwise. Also, let $A_{ij}(n) = 1$ if a cell arrives to VOQ_{ij} and zero otherwise. Then, $X_{ij}(n+1) = X_{ij}(n) + A_{ij}(n) - D_{ij}(n)$. Henceforth, we will drop the time n from the symbol for $D_{ij}(n)$ and $A_{ij}(n)$, and refer to them as D_{ij} and A_{ij} respectively, since in the rest of the proof, we will only be concerned with the arrivals and departures of cells at time n.

Therefore, $[f_1(n+1) - f_1(n)]$

$$= \sum_{i,j,k} [X_{ij}(n+1) X_{ik}(n+1) - X_{ij}(n) X_{ik}(n)]$$

Then we get $[f_1(n+1) - f_1(n)]$

$$\begin{aligned}
&= \sum_{i,j,k} (X_{ij}(n) + A_{ij} - D_{ij})(X_{ik}(n) + A_{ik} - D_{ik}) - \\
&\quad X_{ij}(n) X_{ik}(n) \\
&= \sum_{i,j,k} (A_{ij} - D_{ij}) X_{ik}(n) + (A_{ik} - D_{ik}) X_{ij}(n) + \\
&\quad (A_{ij} - D_{ij})(A_{ik} - D_{ik}) \\
&= \sum_{i,j,k} 2(A_{ik} - D_{ik}) X_{ij}(n) + (A_{ij} - D_{ij})(A_{ik} - D_{ik})
\end{aligned}$$

Since $|A_{ij} - D_{ij}| \leq 1$ and similarly $|A_{ik} - D_{ik}| \leq 1$, we get[3]

$$E[f_1(n+1) - f_1(n)] \leq N^3 + \sum_{i,j,k} 2E[A_{ik} - D_{ik}] X_{ij}(n). \tag{D.7}$$

Denote $E_{ij}(n) = 1$ if a cell departs from the combined queue of VOQ_{ij} and the crosspoint B_{ij}, and zero otherwise. Note that $E_{ij}(n) = 1$ only when a cell departs from the crosspoint B_{ij} to the output at time n, since all departures to the output must occur from the crosspoint. Also recall that the arrival rate to the combined queue, VOQ_{ij} and B_{ij}, is the same as the arrival rate to VOQ_{ij}. So we can write $Z_{ij}(n+1) = Z_{ij}(n) + A_{ij}(n) - E_{ij}(n)$. Again we will drop the time n from the symbol for $E_{ij}(n)$ and $A_{ij}(n)$, and refer to them as E_{ij} and A_{ij} respectively.

Then, similar to the derivation in Equation D.7, we can derive using Equation D.5,

$$E[f_2(n+1) - f_2(n)] \leq N^3 + \sum_{i,j,k} 2E[A_{kj} - D_{kj}] Z_{ij}(n). \tag{D.8}$$

[3]This is in fact the conditional expectation given knowledge of the state of all queues and crosspoints at time n. For simplicity in the rest of the proof (since we only use the conditional expectation), we will drop the conditional expectation sign and simply use the symbol for expectation as its meaning is clear.

So from Equation D.7 and Equation D.8, $E[F(n+1) - F(n)]$

$$
\begin{aligned}
&\leq 2N^3 + 2\sum_{i,j,k}\Big(E[A_{ik} - D_{ik}]X_{ij}(n) \\
&\qquad + E[A_{kj} - E_{kj}]Z_{ij}(n)\Big) \\
&= 2N^3 + 2\sum_{i,j}\Big(X_{ij}(n)\sum_k E[A_{ik} - D_{ik}] \\
&\qquad + Z_{ij}(n)\sum_k E[A_{kj} - E_{kj}]\Big)
\end{aligned}
$$

Re-substituting $Z_{ij} = X_{ij} + B_{ij}$, we get $E[f(n+1) - f(n)]$,

$$
\begin{aligned}
&\leq 2N^3 + 2\sum_{i,j}\Big(X_{ij}(n)\sum_k E[A_{ik} - D_{ik}] \\
&\qquad + \big(X_{ij}(n) + B_{ij}(n)\big)\sum_k E[A_{kj} - E_{kj}]\Big) \\
&= 2N^3 + 2\sum_{i,j}\Big(X_{ij}(n)\sum_k E[A_{ik} - D_{ik} + A_{kj} - E_{kj}] \\
&\qquad + B_{ij}(n)\sum_k E[A_{kj} - E_{kj}]\Big)
\end{aligned}
$$

We can substitute $R_{ij} = \sum_k E[A_{ik} - D_{ik} + A_{kj} - E_{kj}]$ and $S_j = \sum_k E[A_{kj} - E_{kj}]$ and re-write this as,

$$
E[F(n+1) - F(n)] \leq 2N^3 + 2\sum_{i,j}\Big(X_{ij}(n)R_{ij} + B_{ij}(n)S_j\Big) \tag{D.9}
$$

But, we also have from Equation D.2,

$$
E[C_{ij}(n+1) - C_{ij}(n)] \equiv R_{ij} \tag{D.10}
$$

$$
E[\sum_k\Big(Z_{kj}(n+1) - Z_{kj}(n)\Big)] \equiv S_j. \tag{D.11}
$$

In Section D.1, it was shown that for a buffered crossbar with speedup of two, R_{ij}

is strictly negative when $X_{ij}(n) > 0$ and the traffic is admissible. So the first product term inside the summation sign in Equation D.9

$$X_{ij}(n)R_{ij} \leq 0. \tag{D.12}$$

Similarly, if the traffic is admissible, then $\sum_k E[A_{kj}] < 1$. Also, when $B_{ij}(n) = 1$, then from Equation D.2 and *case 1* of Theorem D.1 in Section D.1, we know that the output j will receive at least one cell, and so at least one cell must have departed one of the crosspoints destined to output j at time n. And so when the traffic is admissible and $B_{ij}(n) = 1$, then $S_j < 0$. This implies that the second product term inside the summation sign in Equation D.9,

$$B_{ij}(n)S_j \leq 0. \tag{D.13}$$

In both cases, $X_{ij}(n)R_{ij}$ and $B_{ij}(n)S_j$ are equal to zero only if $X_{ij} = 0$ and $B_{ij} = 0$ respectively. Now we want to use Lemma D.1 and show that the whole right hand side of Equation D.9 is strictly negative. All that needs to be done is to ensure that one of the $VOQs$ X_{ij} in the summation in Equation D.9 is large enough so that $2X_{ij}(n)R_{ij}$ can negate the positive constant $2N^3$.

In order to show this, let $\lambda_{max} = \max(\sum_k \lambda_{ik}, \sum_k \lambda_{kj}), i, j \in (1, 2, ..N)$. Choose any $\gamma' > 0$, and let

$$F \equiv \sum_{ijk} X_{ij}X_{ik} + Z_{ij}Z_{kj} >$$
$$N^3 \left[\left(\frac{(1+\gamma')N^3}{1-\lambda_{max}} \right)^2 + \left(1 + \frac{(1+\gamma')N^3}{1-\lambda_{max}} \right)^2 \right] \equiv C$$

where, C corresponds to the constant in Lemma D.1. Recall that $Z_{ij} \leq X_{ij} + 1$. Then the above inequality can only be satisfied if there exists X_{ij} such that:

$$X_{ij} > \frac{(1+\gamma')N^3}{1-\lambda_{max}}$$

As shown in Section D.1, when $X_{ij} > 0$,

$$R_{ij} \leq -(2 - 2\lambda_{max}) < -(1 - \lambda_{max})$$

Therefore, we have

$$X_{ij}R_{ij} < -(1 + \gamma')N^3$$

If we substitute this in Equation D.9, then for all n such that $F(n) > B$,

$$E[F(n+1) - F(n)] < -2\gamma' N^3$$

Let γ correspond to the variable in Lemma D.1 and set $\gamma = 2\gamma' N^3$. Also it is easy to see that,

$$E[F(n+1)|F(n)] < \infty$$

From Lemma D.1, for every $\epsilon > 0$, there exists $B > 0$ such that $\lim_{n\to\infty} Pr\{\sum_{i,j} X_{i,j}(n) > B\} < \epsilon$. From Definition B.3, the scheduling algorithm gives 100% throughput. ❒