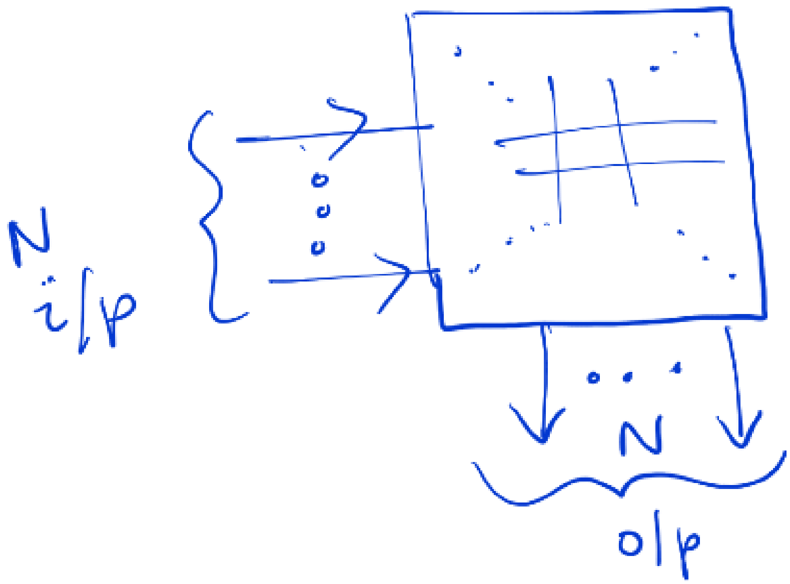
EE5902 MIN - Part 1: Crossbar Switch Analy...

- Design a Crossbar Switch $(N \times N)$ that allows strictly non-blocking connections.



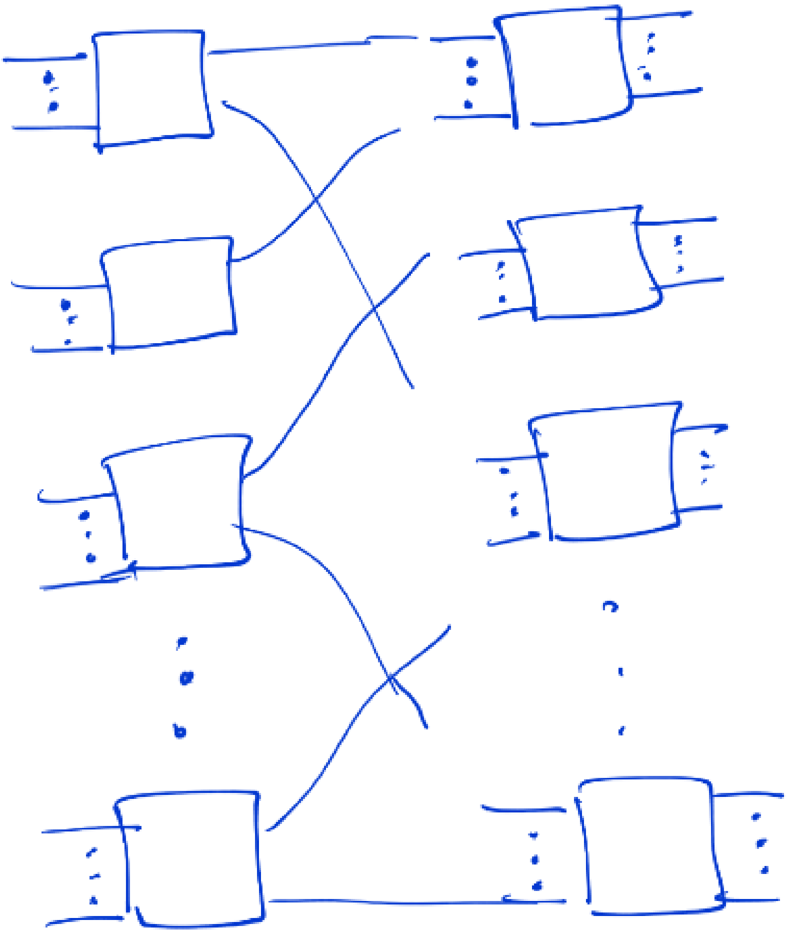
$O(N^2)$ switches

Non-blocking ⟹ whenever there is a free i/p & a free o/p port, regardless of other connections, we need to establish a connection between those free ports if a request arrives, without disturbing the existing connections.

- $\boxed{\text{Switch Controllers:}}$ Complexity of computation @ the controller increases as $O(N2)$!!

Scalability

- Smaller crossbars put together to form a bigger n/w as follows.

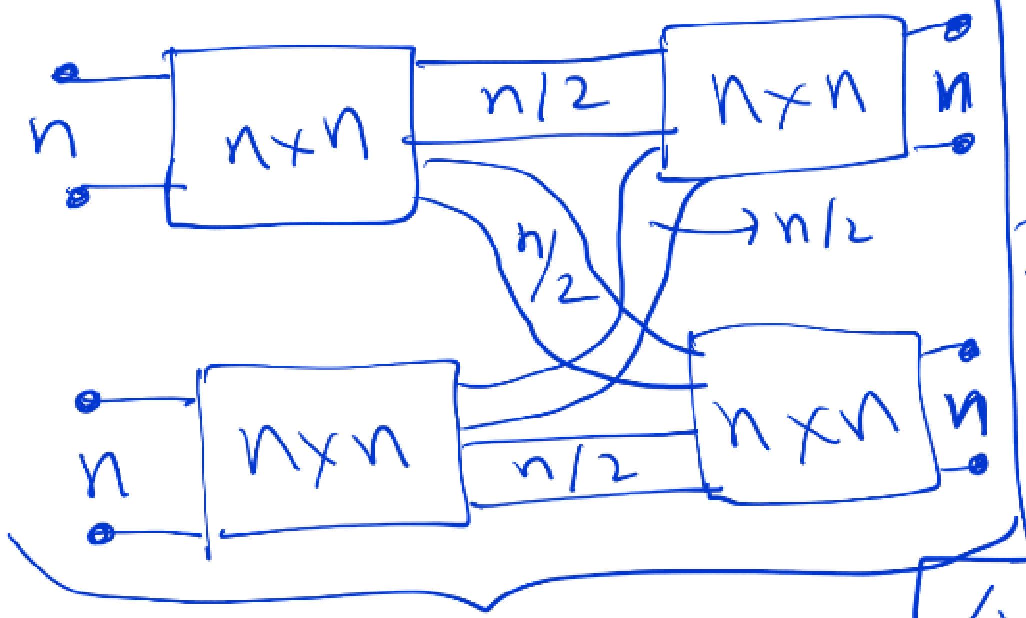


This is a MIN!

. . .

more stages possible

- **Design:** $2n \times 2n$ switch using $n \times n$ crossbar switches



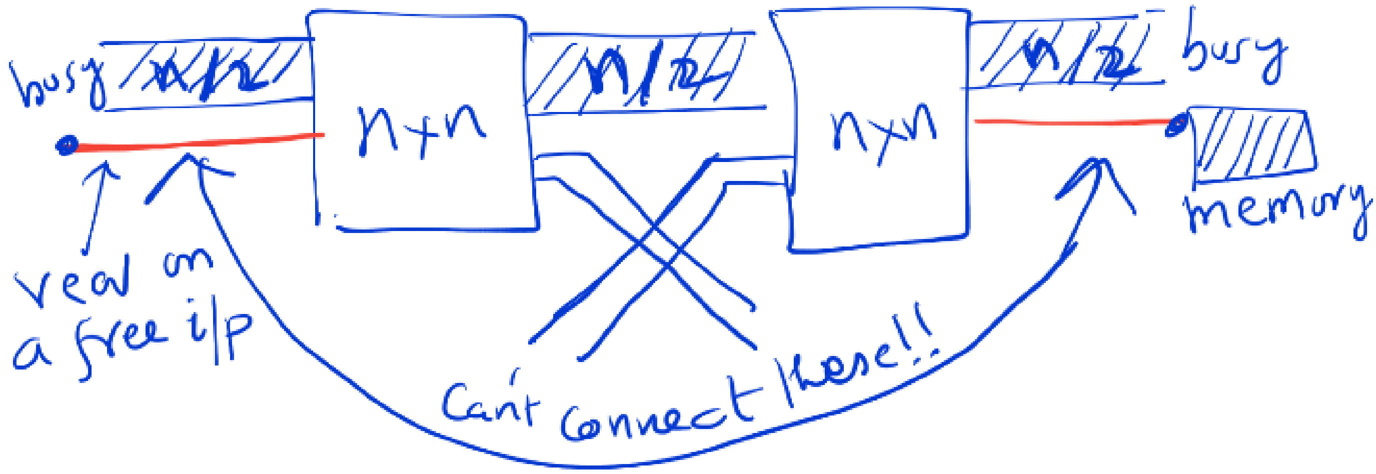
proposed design

$n/2$ connections from i/p stage to next stage;

$4n^2$ switches for a $2n \times 2n$ sys

Q: Is the above interconnection <u>strictly</u> <u>non-blocking</u>?

Consider a $n/2$ i/p lines as shown that are busy.

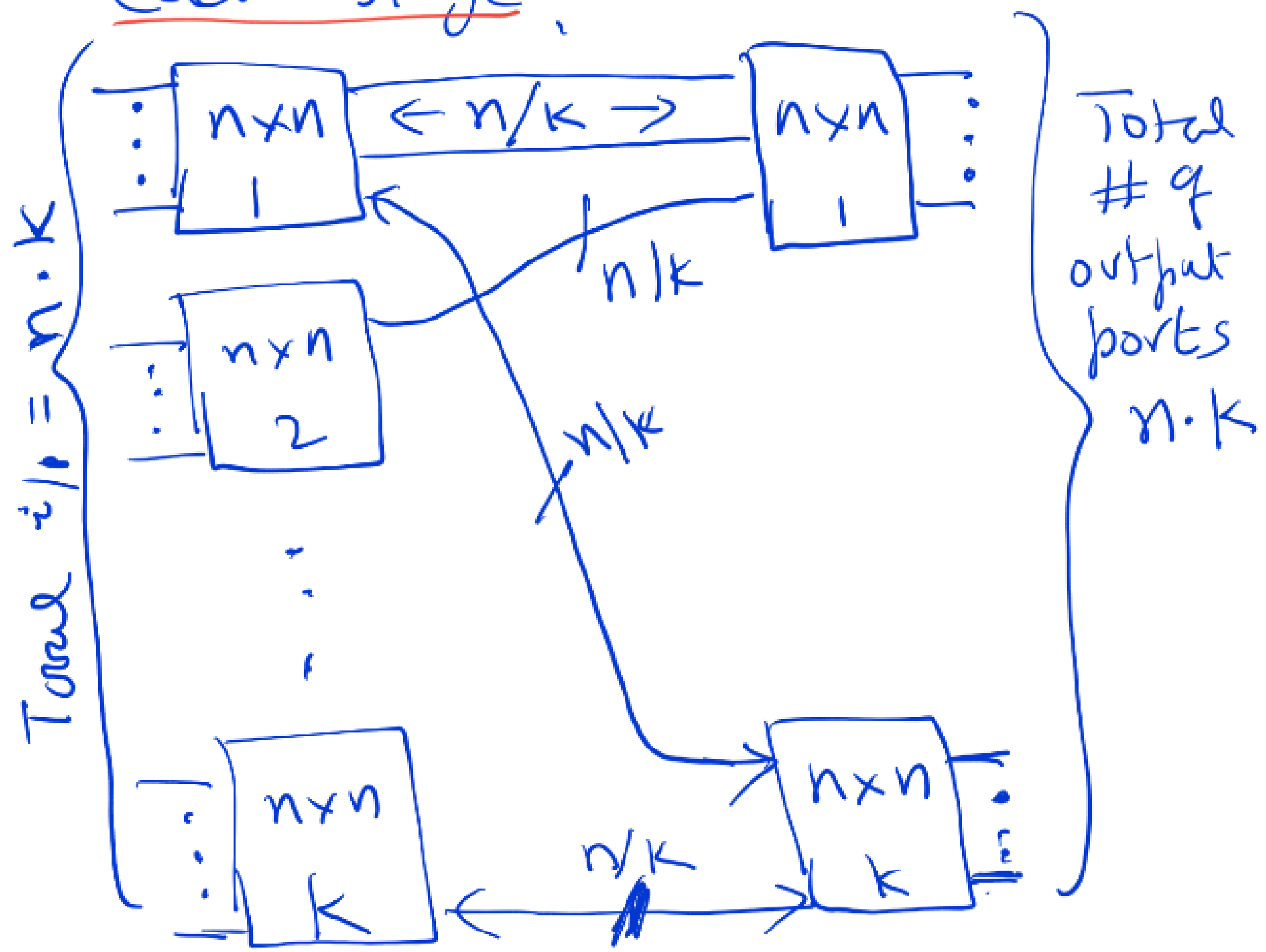


$\Rightarrow$ with a 2-stage it is not possible to build a non-blocking system; we need more stages

Complexity of $2n \times 2n$ System

- $4n^2$

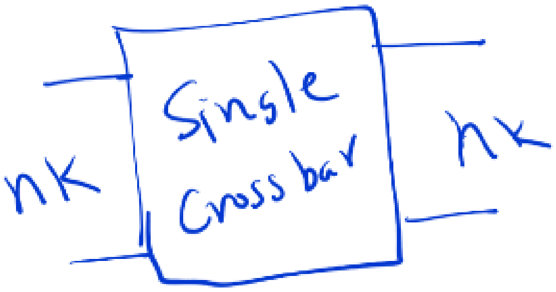
Q: What if there are K switches in each stage?



$\therefore$ # of switching elements $2kn^2$

$\rightarrow 2k\,(n \times n)$ # of cross points

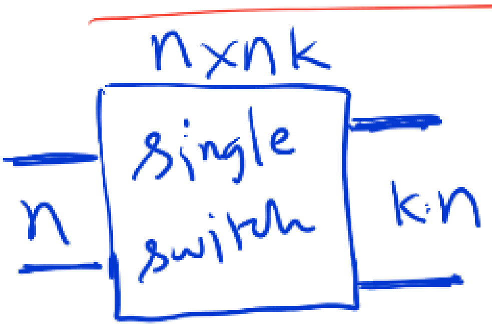
i.e, 2k switches, each with $(n \times n)$ cross points. Thus, with



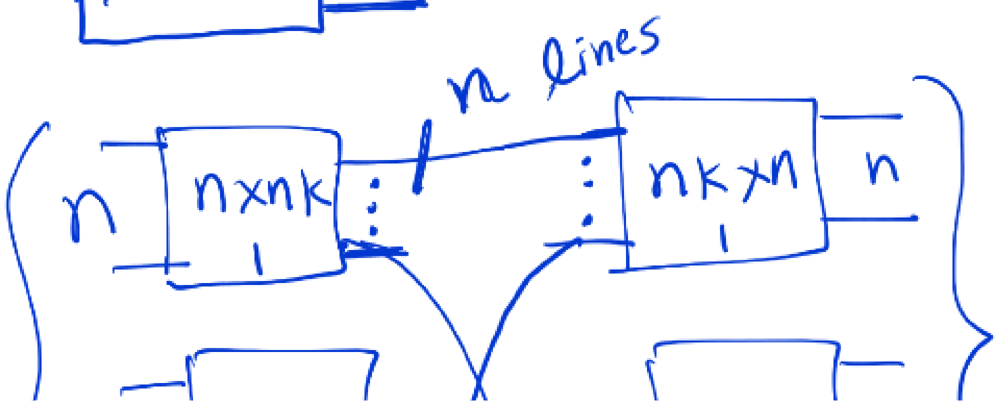
$\Rightarrow (n^2 k^2)$ switch points

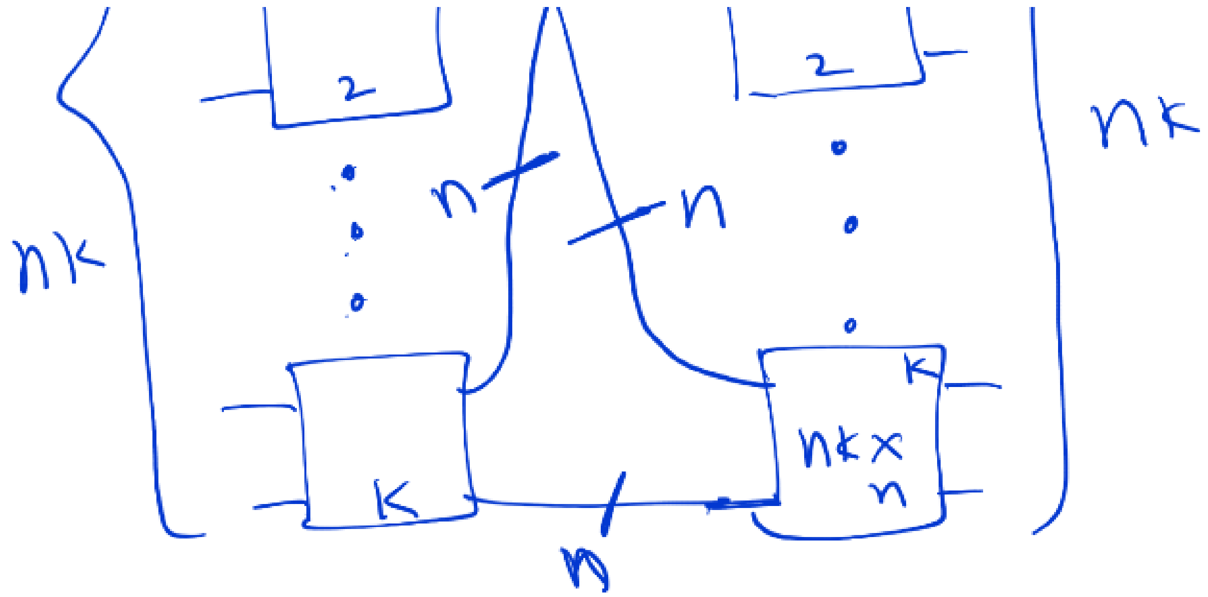
• With 2 stages we have only $(2kn^2)$ as complexity

Q: How to make it as a strictly non-blocking system?



$|o/p| > |i/p|$





So all n connections from a single switch can have n connections to 2nd stage regardless of existing connections.

Complexity: $2k(n \times nk)$
$= 2n^2k^2$!!!

- No advantage in terms of total switching elements, except that each $(n \times nk)$ switch has a separate controller (distributed control.)

Remarks: (Towards improving the design)

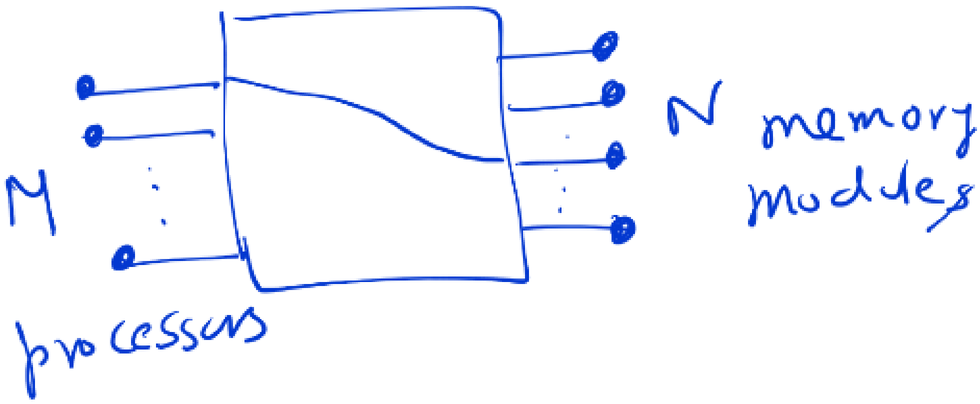
- Most real-life situations aim to minimize the cost of hardware when compared to achieving a non-blocking mode of working.
- System can block a few requests as cost of realizing this system with small blocking is preferred.

→ one in 10k or 30k requests get blocked, which is fine!

⇒ small probability of blocking.

So, let us increase the # of stages & then see if we can estimate this blocking probability.

Estimating the Blocking Probability of a simple $M \times N$ Crossbar Switch



rate of arrival of requests } poisson distribution

$$\text{Prob}(n \text{ reqs. in } T \text{ duration}) = \frac{(\lambda T)^n \cdot (e^{-\lambda T})}{n!}$$

$$\underbrace{\vert \qquad \vert}_{T}$$

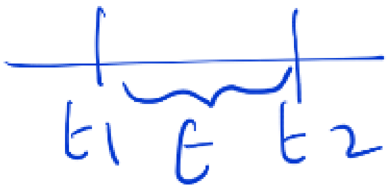
We know, $\displaystyle\sum_{\forall n} \frac{(\lambda T)^n e^{-\lambda T}}{n!} = 1$

$$\left\{ \sum_{0}^{\infty} \frac{(\lambda T)^n}{n!} = e^{\lambda T} \right.$$

- What about req processing duration?

• We consider req proc/duration as an exponential distribution, with a mean value of every req: $(1/\mu)$

If we want to measure the time duration between two arrivals, how to determine its PDF?



$P(t < T) \equiv$ one or more reqs in time t

$\equiv (1 -$ no arrival in $T)$

$\Rightarrow (1 - e^{-\lambda T})$

To get PDF take its derivative:

$\Rightarrow \lambda e^{-\lambda T}$

- <u>When blocking will happen?</u> $\boxed{N<M}$

— Concept of Markov Chains

$\boxed{\text{State}}$: One o/p link is occupied means the system in State 1

no o/p link is occupied
⇒ state 0

⋮

When in state N, as M > N, reqs can arrive but can't be connected to any memory block.

- <u>At what rate we will be transiting between the states?</u>

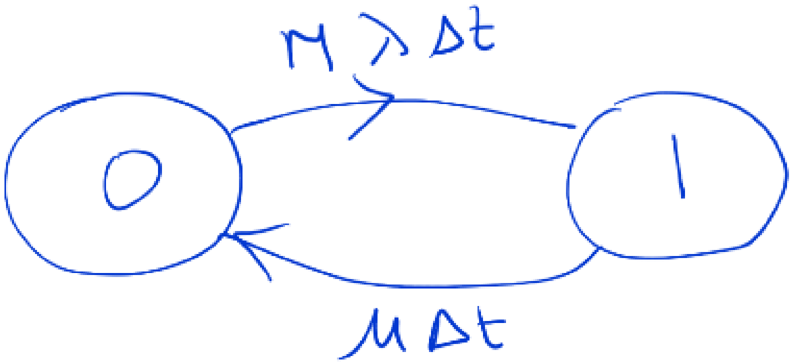
Prob (1 req will arrvie in a time Δt)?

$= (\lambda \Delta t)\, e^{-\lambda \Delta t}/1!$ $\quad$ Δt is so small that the prob of 2 arrivals is very small.

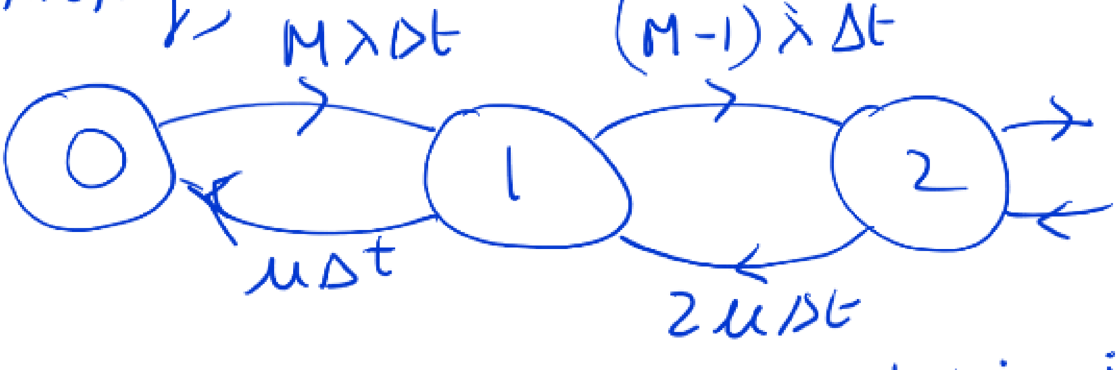
State Space design

- With (M) input lines, the prob that 1 req arrives will be $(M \cdot \lambda \cdot \Delta t)$

- When in state (1), we can go back to (0) after the req processing by memory block at a rate $(\mu \Delta t)$



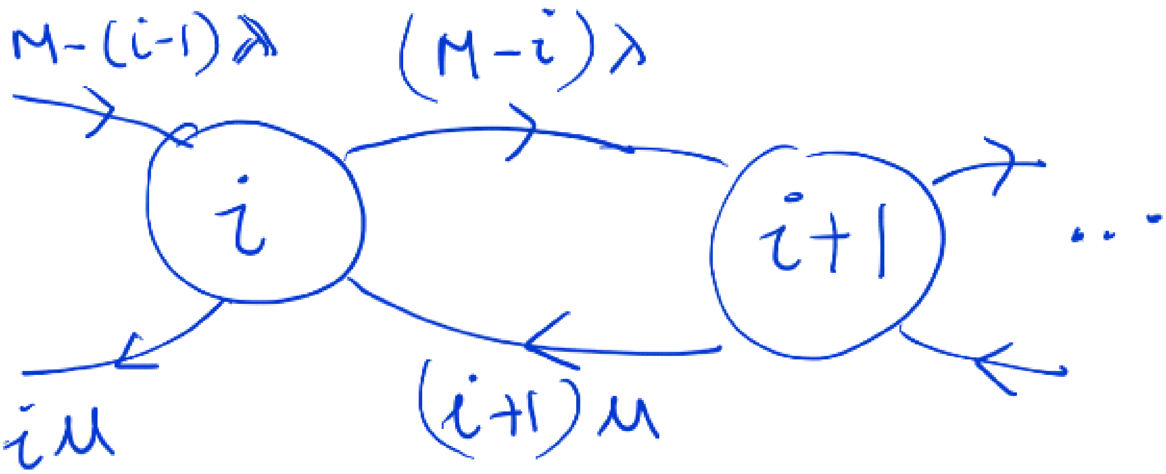
Continuing,



Under steady-state, prob of staying in a specific state is a constant.

- Writing steady-state equations

→ (Δt gets cancelled when we write balance equations)



① $P_0 \cdot M\lambda = P_0 \mu \Rightarrow P_1 = M\left(\frac{\lambda}{\mu}\right)P_0$

② $\underbrace{P_0 M\lambda + 2\mu P_2 = \mu P_1 + (M-1)\lambda P_1}_{\text{solve } P_2 \text{ in terms of } P_0 \text{ (use ①)}}$

Similarly,

$$P_i = {}^{M}C_i \left(\frac{\lambda}{\mu}\right)^i P_0$$

Using $\sum_{\forall i} P_i = 1$, we obtain

$$\boxed{P_0 = \frac{1}{\sum_{j=0}^{N} {}^{M}C_j \left(\frac{\lambda}{\mu}\right)^j}}$$

Thus, the blocking probability is given by

$$\left. P_B = P_N = \frac{{}^{M}C_N (\lambda/\mu)^N}{\sum_{\forall j} {}^{M}C_j (\lambda/\mu)^j} \right\}$$

Last modified: 11:15 pm