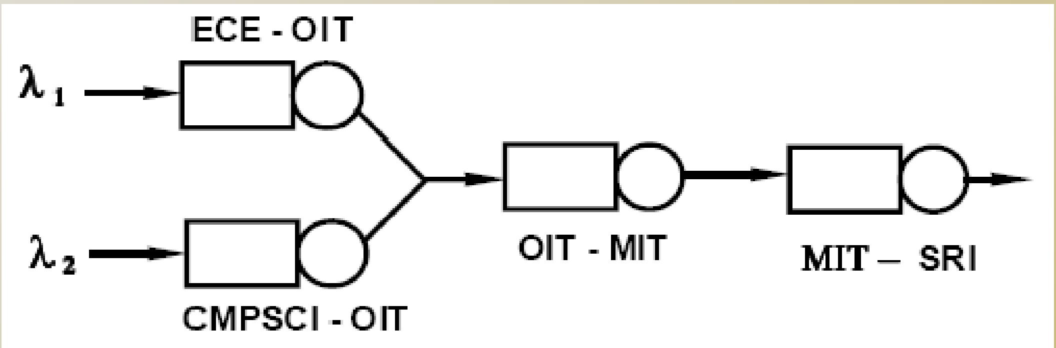
CDA6530: Performance Models of Computers and Networks

Chapter 5: Basic Queuing Networks

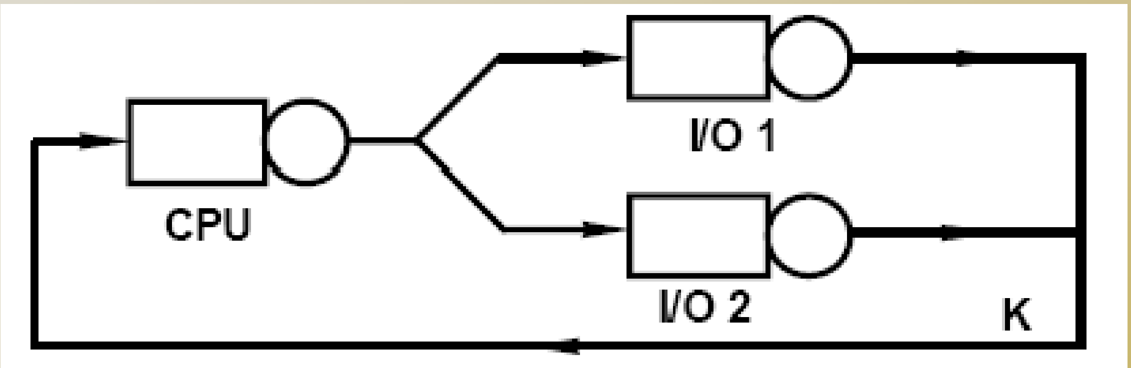
Open Queuing Network

- Jobs arrive from external sources, circulate, and eventually depart



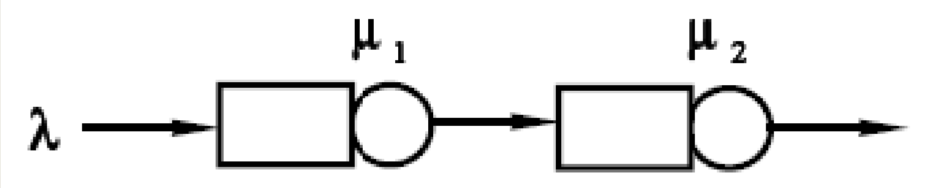
Closed Queuing Network

- Fixed population of *K* jobs circulate continuously and never leave
  - Previous machine-repairman problem



Feed-Forward QNs

- Consider two queue tandem system

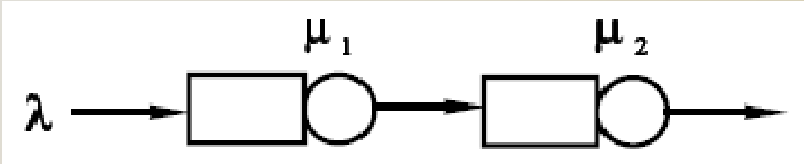


- Q: how to model?
  - System is a continuous-time Markov chain (CTMC)
  - State $(N_1(t), N_2(t))$, assume to be stable
  - $\pi(i,j) = P(N_1=i, N_2=j)$
  - Draw the state transition diagram
    - But what is the arrival process to the second queue?

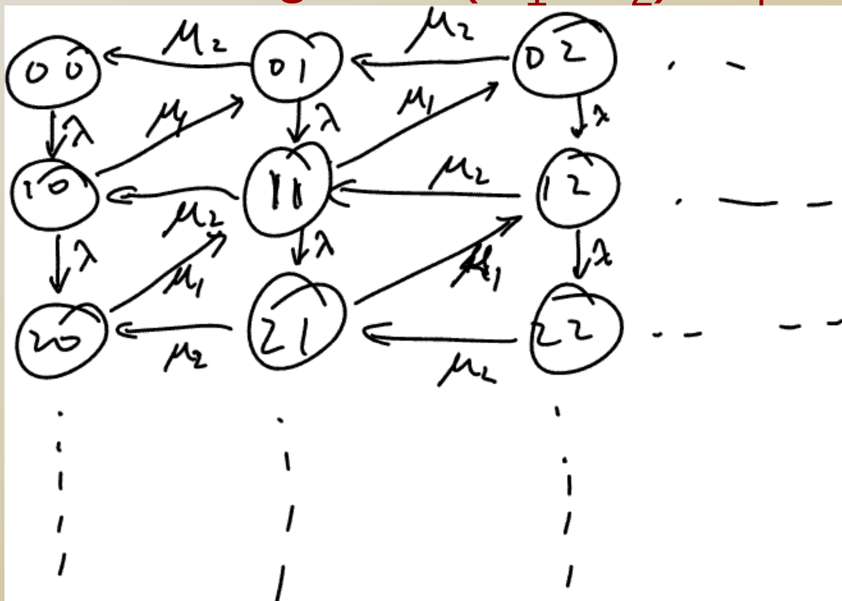
Poisson in $\Rightarrow$ Poisson out

- **Burke's Theorem:** Departure process of *M/M/*1 queue is Poisson with rate *λ* independent of arrival process.

- **Poisson process addition, thinning**
  - Two *independent* Poisson arrival processes adding together is still a Poisson ($\lambda=\lambda_1+\lambda_2$) Why?
  - For a Poisson arrival process, if each customer lefts with prob. p, the remaining arrival process is still a Poisson ($\lambda = \lambda_1 \cdot p$)



- State transition diagram: (N_1, N_2), $N_i = 0,1,2,\cdots$



$$\pi(i,j) = (1-\rho_1)\rho_1^i(1-\rho_2)\rho_2^j \quad i,j \geq 0$$

$$\rho_i = \lambda/\mu_i$$

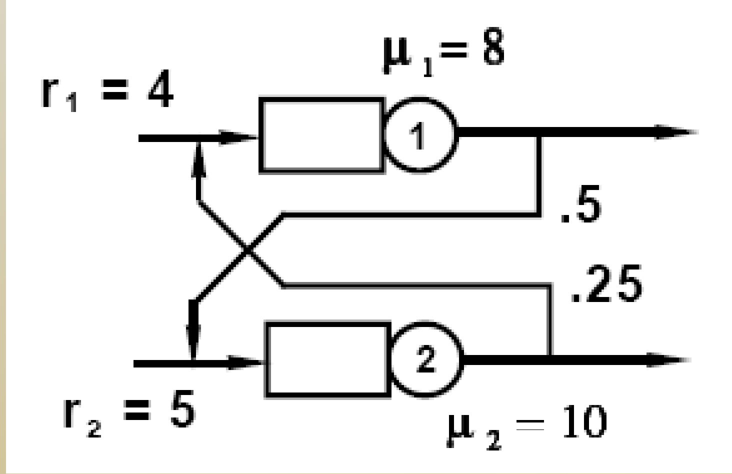
- For a k queue tandem system with Poisson arrival and expo. service time
- Jackson's theorem:

$$P(N_1 = n_1, N_2 = n_2, \ldots, N_k = n_k) = \prod_{i=1}^{k} (1 - \rho_i) \rho_i^{n_i},$$

- Above formula is true when there are feedbacks among different queues
  - Each queue behaves as M/M/1 queue in isolation

Example

- λ_i: arrival rate at queue i



$$\lambda_1 = 4 + \lambda_2/4$$
$$\lambda_2 = 5 + \lambda_1/2$$

Why?

$$\Rightarrow \lambda_1 = 6, \; \lambda_2 = 8$$

$$\pi(n_1, n_2) = \frac{1}{4}\left(\frac{3}{4}\right)^{n_1} \frac{1}{5}\left(\frac{4}{5}\right)^{n_2}$$

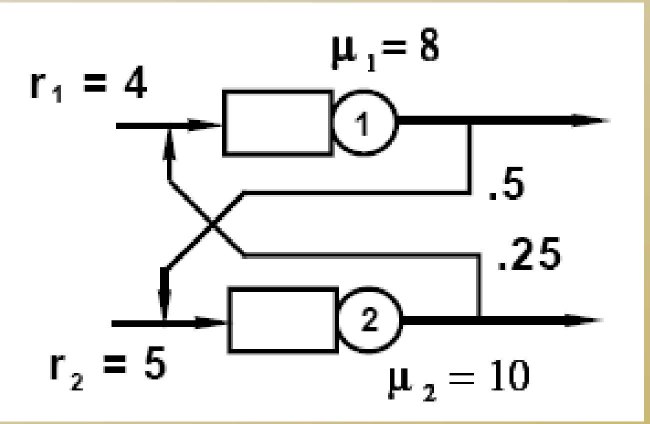
In M/M/1: $E[N] = \dfrac{\rho}{1-\rho} = \dfrac{\lambda}{\mu - \lambda}$

$$\begin{aligned} E[N] = \sum_{i=1}^{2} E[N_i] &= \sum_{i=1}^{2} \lambda_i/(\mu_i - \lambda_i) \\ &= 3 + 4 = 7 \end{aligned}$$

$$E[T] = E[N]/(r_1 + r_2) = 7/9 \text{ time units}$$

Why?

- $T^{(i)}$: response time for a job enters queue i



$$E[T^{(1)}] = 1/(\mu_1 - \lambda_1) + E[T^{(2)}]/2$$
$$E[T^{(2)}] = 1/(\mu_2 - \lambda_2) + E[T^{(1)}]/4$$

Why?

In M/M/1: $E[T] = \frac{1}{\mu - \lambda}$

Extension

- results hold when nodes are multiple server nodes (*M/M/c*), infinite server nodes finite buffer nodes (*M/M/c/K*) (careful about interpretation of results), PS (process sharing) single server with arbitrary service time distr.

Closed QNs

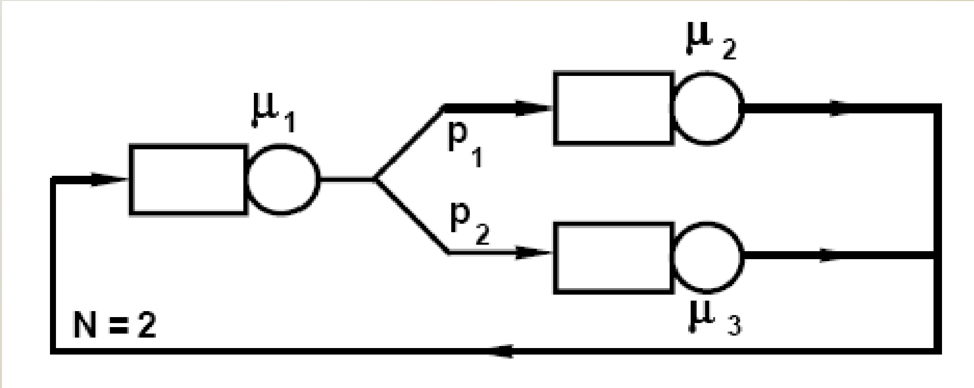
- Fixed population of *N* jobs circulating among *M* queues.
  - single server at each queue, exponential service times, mean $1/\mu_i$ for queue *i*
  - routing probabilities $p_{i,j}$, $1 \le i, j \le M$
  - visit ratios, $\{v_i\}$. If $v_1 = 1$, then v_i is mean number of visits to queue *i* between visits to queue 1

$$v_i = \sum_{j=1}^{M} v_j p_{j,i} \quad i = 2, \ldots M$$

  - γ_i: throughput of queue *i*,

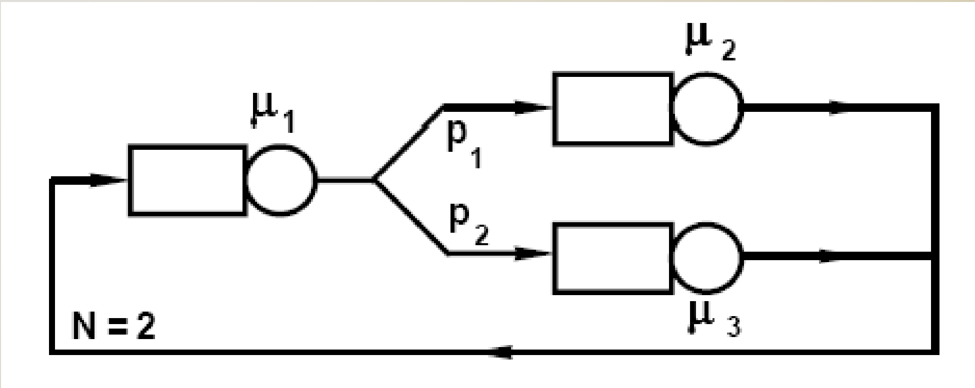
$$\gamma_i/\gamma_j = v_i/v_j, \quad 1 \le i, j \le M$$

Example

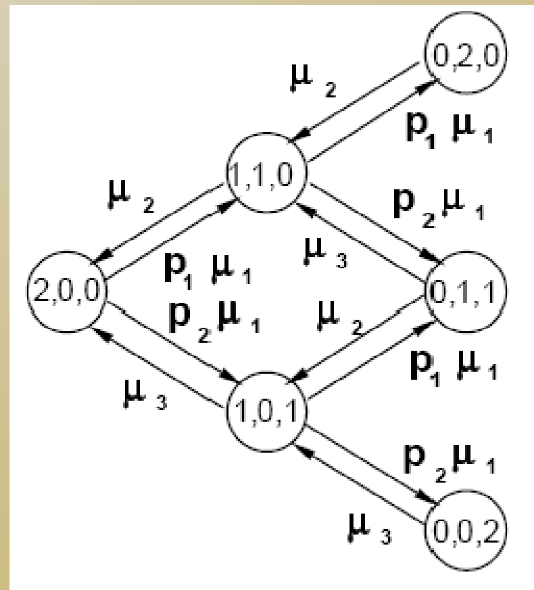


- Open QN has infinite no. of states
- Closed QN is simpler

- How to define states?
  - No. of jobs in each queue



μ_1
p_1
p_2
μ_2
μ_3
N = 2



0,2,0
1,1,0
2,0,0
0,1,1
1,0,1
0,0,2
μ_2
$p_1\mu_1$
$p_2\mu_1$
μ_3

Steady State Solution

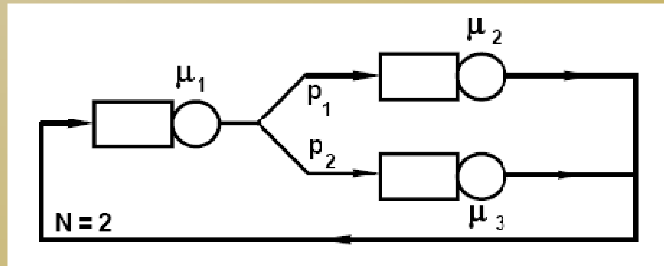
- **Theorem (Gordon and Newell)**

$$\pi(\vec{n}) = \frac{1}{G(N)} \prod_{i=1}^{M} \left(\frac{v_i}{\mu_i}\right)^{n_i} \qquad \vec{n} \geq \vec{0}; \sum_{i=1}^{M} n_i = N$$

where $\vec{n} = (n_1, \ldots, n_M)$, and $G(N)$ is a constant chosen so that $\sum \pi(\vec{n}) = 1$.

- For previous example, v_i?

$$v_1 = 1,\ v_2 = 3/4,\ v_3 = 1/4$$



Mean Value Analysis (MVA) Algorithm

- **Key idea:** a job that moves from one queue to another, at time of arrival to queue sees a system with the same statistics as system with *one less customer*.
  - We only consider single server nodes

MVA Algorithm

- System with population of n jobs

  - $\bar{N}_i(n)$ - average number of jobs at node i
  - $\bar{T}_i(n)$ - average response time at node i
  - $\gamma_i(n)$ - thruput of node i

0. $\bar{N}_i(0) = 0, \quad 1 \le i \le M$ *initialization*

for $n = 1$ **to** N **do**

1. $\bar{T}_i(n) = [1 + \bar{N}_i(n-1)]/\mu_i,$

Why?

2. $\gamma(n) = n/(\sum_{i=1}^{M} v_i \bar{T}_i(n))$

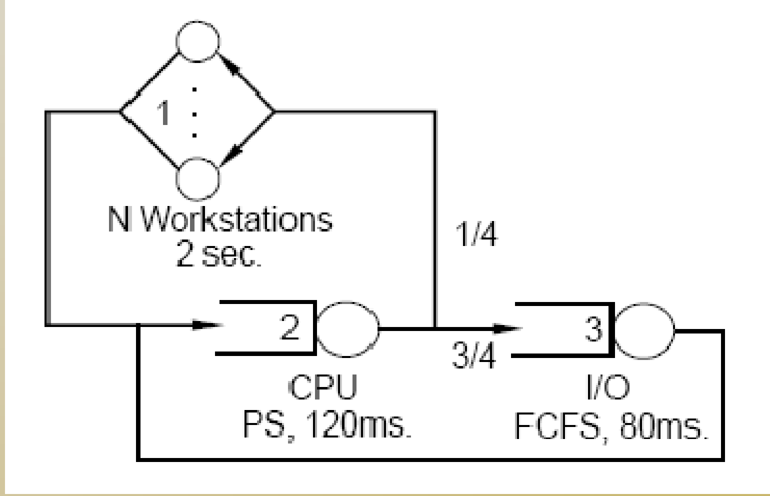
3. $\gamma_i(n) = v_i \gamma(n), \quad 1 \le i \le M$

$\bar{N}_i(n) = \gamma_i(n)\bar{T}_i(n), \quad 1 \le i \le M$

Why?

Example: File Server

- Each workstation requests file server's CPU and I/O service
  - Workstation = job
- What is v_i?



N	$\bar{T}_1$	$\bar{N}_1$	$\bar{T}_2$	$\bar{N}_2$	$\bar{T}_3$	$\bar{N}_3$	γ
1	2sec		120ms.		80ms.		1/2.72
.		.74		.17		.09	.368 job/sec
2	2sec		140ms		87ms		2/2.82
		1.42		.4		.18	.709j/s
3	2sec		168ms		94ms		3/2.952
		2.03		.68		.29	1.02j/s
4	2sec		202ms		103ms		4/3.117
		2.57		1.03		.4	1.28j/s