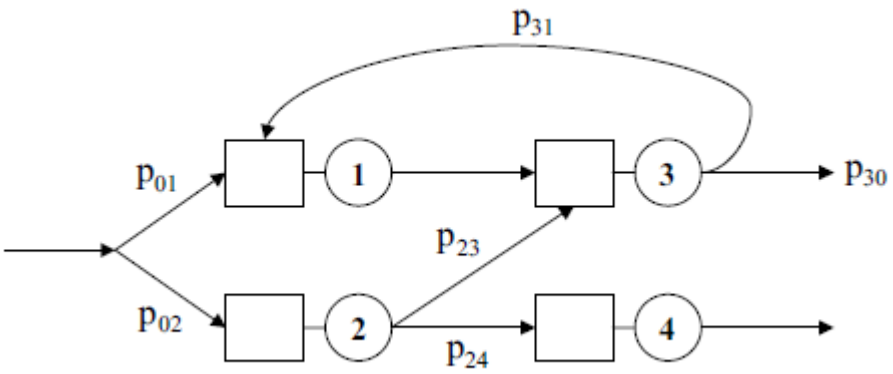
Queuing Network

Dr. Sk. Md. Masudul Ahsan
Professor, Dept. of CSE, KUET

Queueing Networks

- A queueing network is a system composed of several interconnected stations, each with a queue.

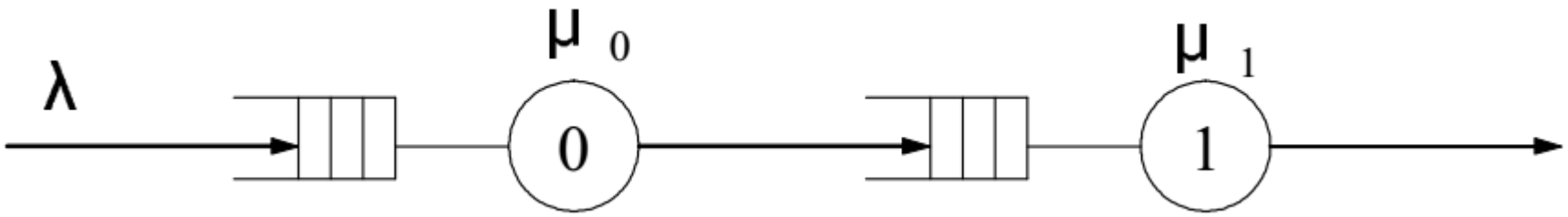
- Customers, upon the completion of their service at a station, moves to another station for additional service or leave the system according some **routing rules** (deterministic or probabilistic).



Exponential Queues in Series

- In the queuing models that we have studied so far, a customer's entire service time is spent with a single server.
- In many situations the customer's service is not complete until the customer has been served by more than one server.

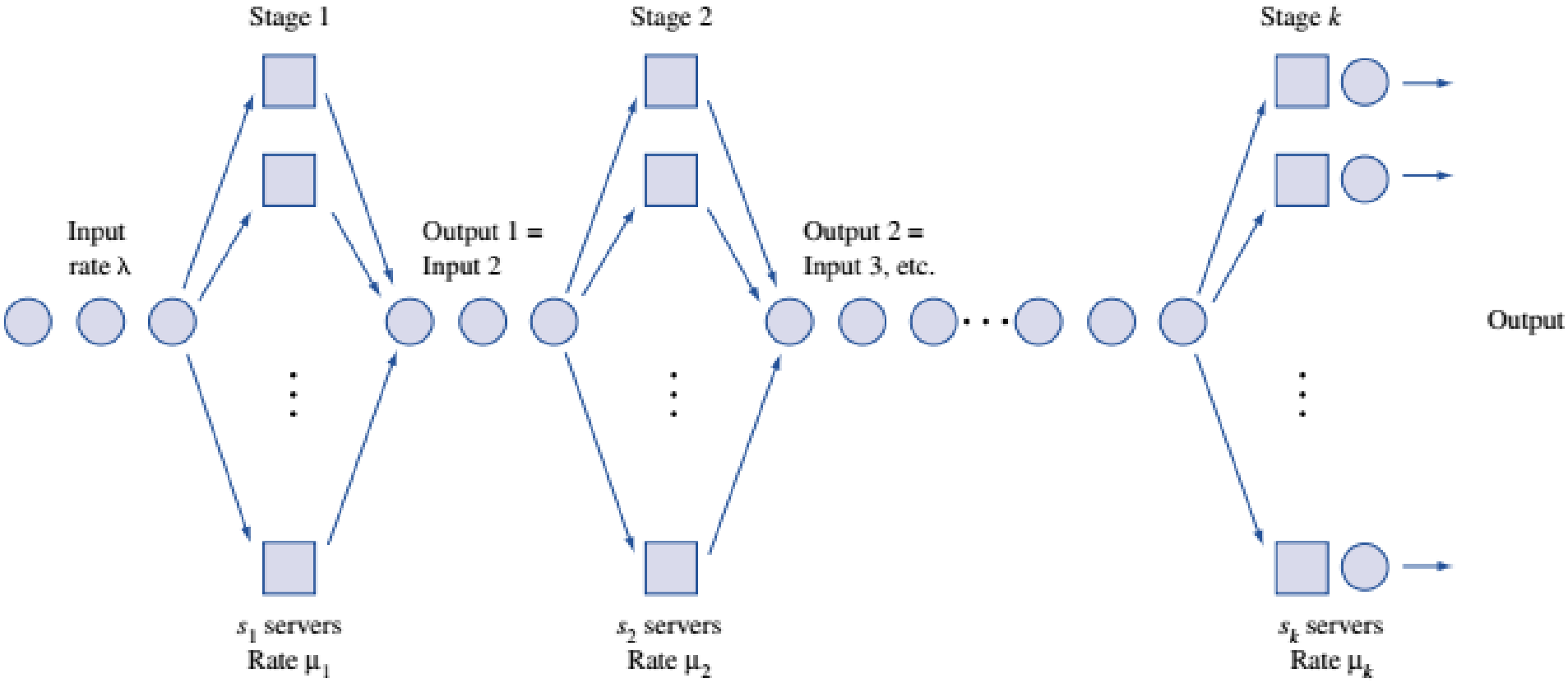
- **2-stage tandem network with Independent Service Times**



- What the arrival process and rate at node 1 ?

Exponential Queues in Series

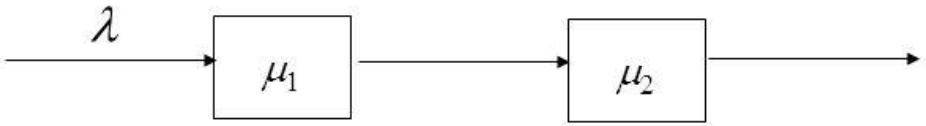
- k-stage series queuing system



Exponential Queues in Series Networks

- **Theorem**
- **If**
  1) interarrival times for a series queuing system are exponential with rate λ,
  2) service times for each stage ***i*** server are exponential, and
  3) each stage has an infinite-capacity waiting room,
- **then** interarrival times for arrivals to each stage of the queuing system are exponential with rate λ.

Tandem Queue



If arrivals to the first server follow a Poisson process and service times are exponential, then arrivals to the second server also follow a Poisson process and the two queues behave as independent M/M/1 systems:

Tandem network of M/M/1 queues

- M/M/1 queue, Poisson(λ) arrivals, exponential(μ) service
- Equilibrium distribution

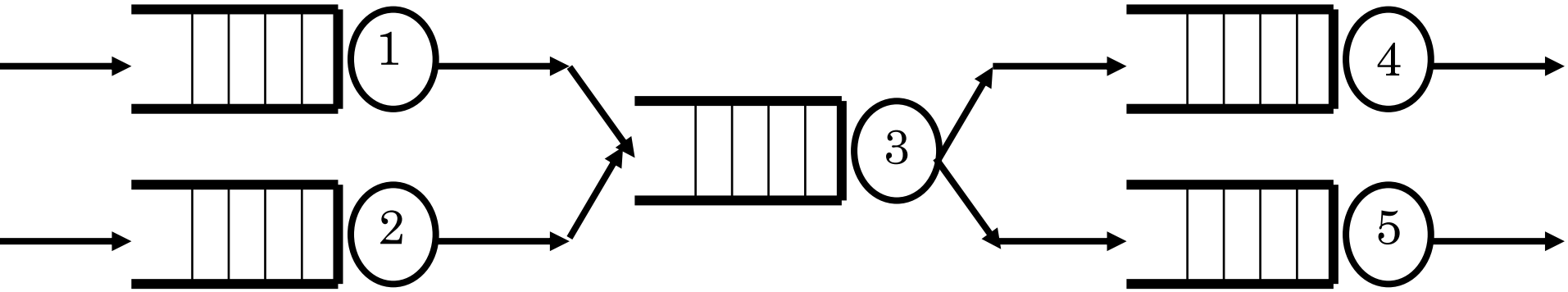
$$\pi_j = (1-\rho)\rho^j, \quad j = \{0,1,2,\ldots\}, \rho = \lambda/\mu < 1$$

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

$$\pi(j_1,\ldots,j_k) = \prod_{i=1}^{k} (1-\rho_i)\rho_i^{j_i}, \quad \rho_i = \lambda/\mu_i < 1$$

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

Example: feed forward network of M/M/1 queues



arrival rate $\lambda_i, i = 1,2$

service rate $\mu_i, i = 1,...,5$

routing probabilities p_{34}, p_{35}

$$\lambda_3 = \lambda_1 + \lambda_2$$

$$\lambda_4 = \lambda_3 p_{34}$$

$$\lambda_5 = \lambda_3 p_{35}$$

$$\pi(j_1,...,j_5) = \prod_{i=1}^{5} (1-\rho_i)\rho_i^{j_i}, \quad \rho_i = \lambda_i / \mu_i < 1$$

Exponential Queues in Series Networks

- The last two works in a car manufacturing process are installing the engine and putting on the tires. An average of 54 cars per hour arrive requiring these two tasks.
- One worker is available to install the engine and can service an average of 60 cars per hour.
- After the engine is installed, the car goes to the tire station and waits for its tires to be attached. Three workers serve at the tire station. Each works on one car at a time and can put tires on a car in an average of 3 minutes.
- Both interarrival times and service times are exponential.
  - Determine the mean queue length at each work station.
  - Determine the total expected time that a car spends waiting for service

Exponential Queues in Series Networks

- This is a series queuing system with
  - λ=54 cars per hour, s_1 = 1, μ_1 = 60 cars per hour,
  - s_2 = 3, and μ_2 = 20 cars per hour
  - Since $\lambda < \mu_1$ and $\lambda < 3\mu_2$, neither queue will "blow up,"
- For stage 1 (engine), ρ = 54/60 = 0.9

$$L_q \text{ (for engine)} = \left(\frac{\rho^2}{1-\rho}\right) = \left[\frac{(.90)^2}{1-.90}\right] = 8.1 \text{ cars}$$

$$W_q \text{ (for engine)} = \frac{L_q}{\lambda} = \frac{8.1}{54} = 0.15 \text{ hour}$$

Exponential Queues in Series Networks

- For stage 2 (Tires), ρ = 54/(3*20) = 0.9

$$L_q = \frac{(s\rho)^s}{s!}\pi_0 \frac{\rho}{(1-\rho)^2} \qquad L_q = \frac{P(j \geq s)\rho}{1-\rho} \qquad P(j \geq s) = \frac{(s\rho)^s \pi_0}{s!(1-\rho)}$$

$$\pi_0 = \left(\sum_{j=0}^{S-1} \frac{(s\rho)^j}{j!} + \frac{(s\rho)^s}{s!(1-\rho)} \right)^{-1} \qquad P(j \geq s) = 0.82$$

$$= (1 + 2.7 + 3.645 + 32.805)^{-1} = 0.025$$

$$L_q = \frac{0.82 * 0.9}{1-0.9} = 7.4 \quad cars$$

total expected waiting time is

0.15 +0.137 = 0.287 hour

$$W_q = \frac{7.4}{54} = 0.137 hrs$$

Open Queuing Network

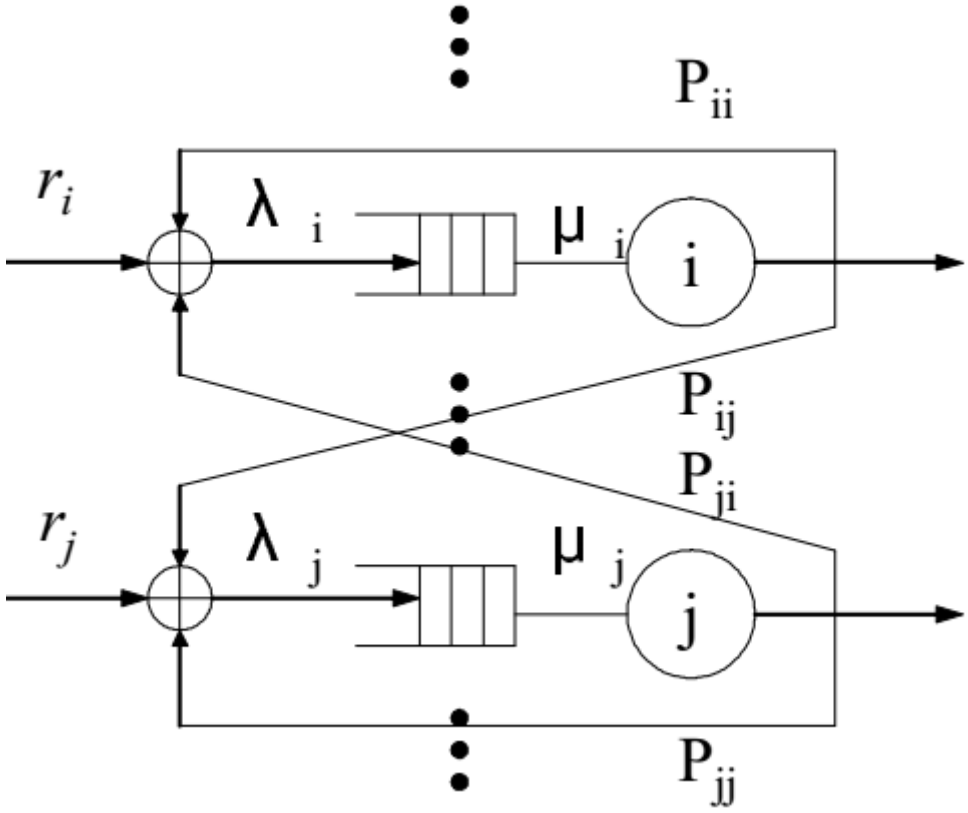
- Jobs arrive from external sources, circulate, and eventually depart

Jackson Network Definition

1. All outside arrivals to each queuing station in the network must follow a Poisson process.
2. All service times must be exponentially distributed.
3. All queues must have unlimited capacity.
4. When a job leaves one station, the probability that it will go to another station is independent of its past history and is independent of the location of any other job.

In essence, a Jackson network is a collection of connected $M/M/s$ queues with known parameters.

Jackson Network Definition



In essence, a Jackson network is a collection of connected $M/M/s$ queues with known parameters.

Jackson's Theorem

1. Each node is an independent queuing system with Poisson input determined by partitioning, merging and tandem queuing example.

2. Each node can be analyzed separately using $M/M/1$ or $M/M/s$ model.

3. Mean delays at each node can be added to determine mean system (network) delays.

Computation of Input Rate

Let r_i = external arrival rate to station $i = 1, \ldots, k$

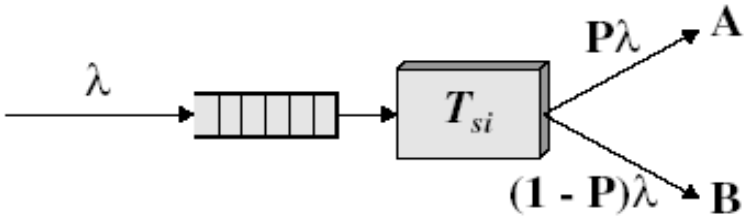
P_{ij} = probability of going from station i to j in network

λ_i = total input to station i

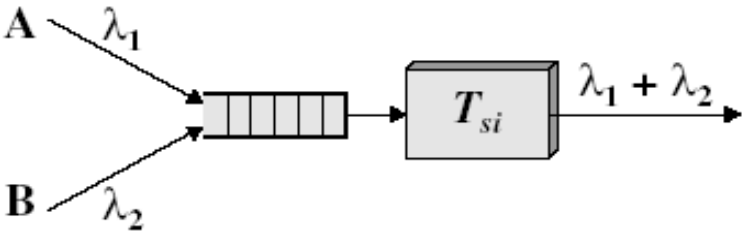
In steady state there must be flow balance at each station.

$$\lambda_i = r_i + \sum_{m=1}^{k} P_{mi}\lambda_m, \quad i = 1, \ldots, k$$

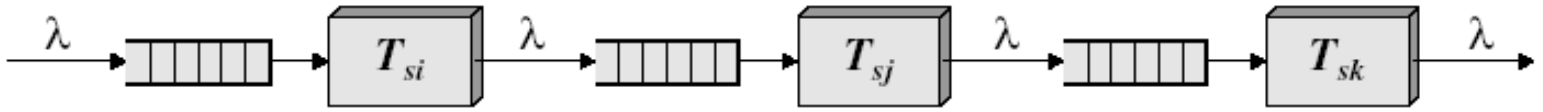
Element of a Queuing Network



(a) Traffic partitioning



(b) Traffic merging



(c) Simple tandem queue

Matrix Form of Computations

Property 1: Let **P** be the $k \times k$ probability matrix that describes the routing of units within a Jackson network, and let r_i denote the mean arrival rate of units going directly to station i from outside the system. Then

$$\boldsymbol{\lambda} = \mathbf{r}(\mathbf{I} - \mathbf{P})^{-1}$$

where $\mathbf{r} = (r_1,\ldots,r_k)$ give the external arrival rates into the various station; and **I** is the identity matrix,

λ_i is the net arrival rate into station i.

Note: Unlike the state-transition matrix used for Markov chains, the rows of the **P** matrix here need not sum to one; that is $\Sigma_j P_{ij} \leq 1$

Simplification of Network

After the net rate into each node is known, the network can be decomposed and each node treated as if it were an independent queuing system with Poisson input.

Property 2: Consider a Jackson network comprising k nodes. Let N_i denote a random variable indicating the number of jobs at node i (the number in the queue plus the number in service). Then,

$$\Pr\{ N_1 = n_1, \ldots, N_m = n_m\} = \Pr\{ N_1 = n_1\} \times \cdots \times \Pr\{ N_m = n_m\}$$

and

$\Pr\{ N_i = n_i\}$ for all $n_i = 0, 1, \ldots$ can be calculated using the equations for independent *M/M/s* seen previously.

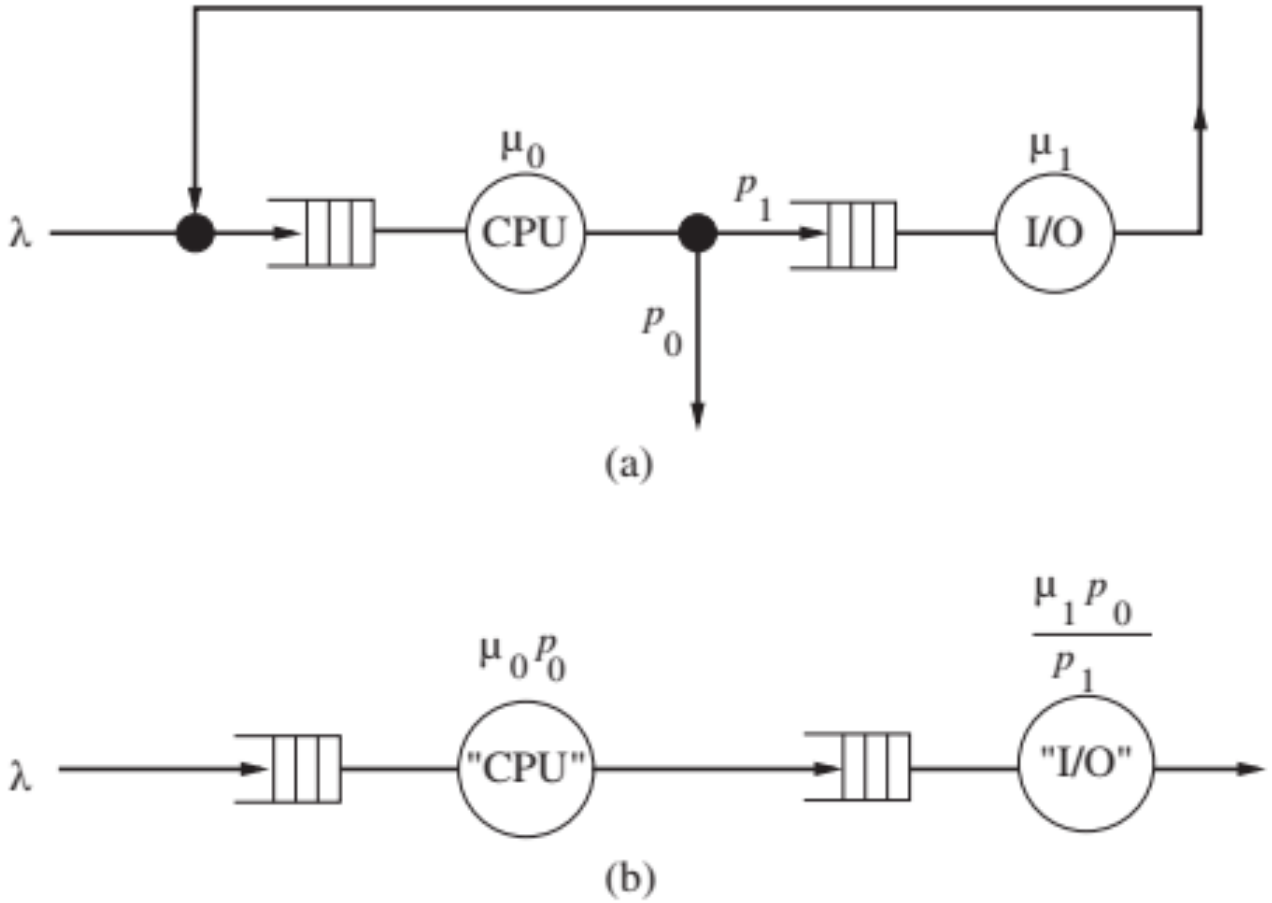


Figure 9.4. (a) An open network with feedback; (b) an "equivalent" network without feedback

$$p(k_0, k_1) = (1 - \rho_0)\rho_0^{k_0}(1 - \rho_1)\rho_1^{k_1}$$

where $\lambda_0/\mu_0 = \rho_0$ and $\lambda_1/\mu_1 = \rho_1$.

$$\rho_0, \rho_1 < 1$$

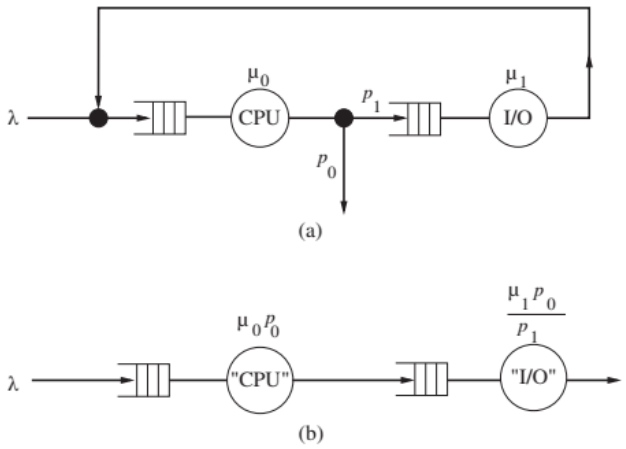


Figure 9.4. (a) An open network with feedback; (b) an "equivalent" network without feedback

$$\lambda_0 = \lambda + \lambda_1.$$

$$\lambda_1 = \lambda_0 p_1$$

$$\lambda_0 = \frac{\lambda}{1 - p_1} = \frac{\lambda}{p_0}$$

$$\lambda_1 = \frac{p_1 \lambda}{p_0}.$$

$$\rho_0 = \frac{\lambda}{p_0 \mu_0} \quad \text{and} \quad \rho_1 = \frac{p_1 \lambda}{p_0 \mu_1}.$$

W= $$E[R] = \left(\frac{\rho_0}{1 - \rho_0} + \frac{\rho_1}{1 - \rho_1}\right)\frac{1}{\lambda}$$

$$= \frac{1}{p_0\mu_0 - \lambda} + \frac{1}{\frac{p_0\mu_1}{p_1} - \lambda}$$

$$W_{s0} = 1/(p_0\mu_0) \qquad W_{s1} = p_1/(p_0\mu_1)$$

For, M/M/1

$$L_s = 0\pi_0 + 1(\pi_1 + \pi_2 + \cdots) = 1 - \pi_0$$

$$= 1 - (1 - \rho) = \rho$$

$$W_s = \frac{L_s}{\lambda} = \frac{\rho}{\lambda} = \frac{1}{\mu}$$

Computation Center Example

- A high performance computation center is composed of 3 work stations comprising: (1) input processors, (2) central computers, and (3) a print center.
- All jobs submitted must first pass through an input processor for error checking before moving on to a central processor → 80% go through and 20% are rejected.
- Of the jobs that pass through the central processor, 40% are routed to a printer.
- Jobs arrive randomly at the computation center at an average rate of 10/min. To handle the load, each station may have several parallel processors.

Data for the Computation Center

We know from previous statistics that the time for the three steps have exponential distributions with means as follows:

10 seconds for an input processor

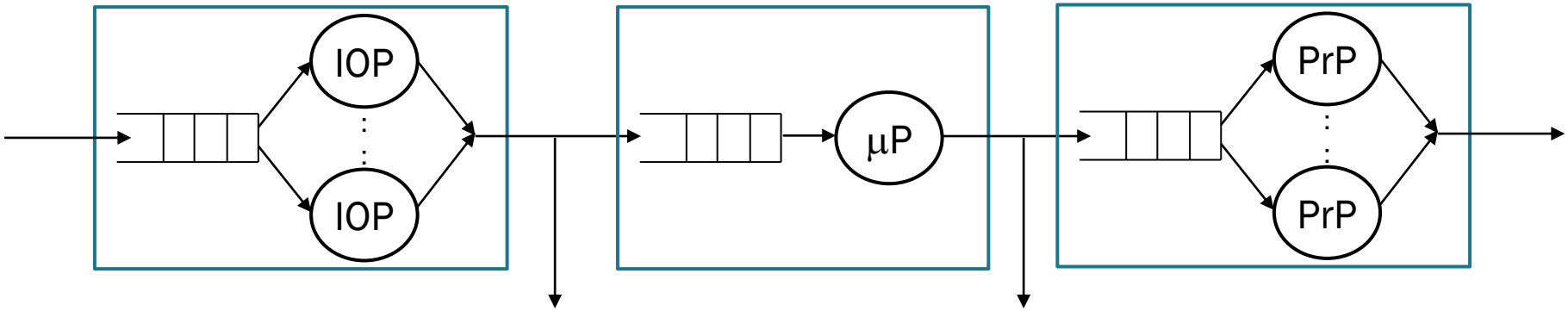
3 seconds for a central processor

70 seconds for a print processor

All queues are assumed to have unlimited capacity.

Goal

Model system as a Jackson network. Find the minimum number of processors of each type and compute the average time require for a job to pass through the system.


IOP
IOP
μP
PrP
PrP

Arrival Rate Computations

Using general equation:

With $k = 3$, $r_1 = 10$, $p_{12} = 0.8$, $p_{23} = 0.4$

we get:

$$\lambda_1 = 10$$

$$\lambda_2 = 0.8\lambda_1 = 8$$

$$\lambda_3 = 0.4\lambda_2 = 3.2$$

I/O Data for the Computation Center

System measure	Input processor	Central processor	Printer
External arrival rate, γ_i	10/min	0	0
Total arrival rate, λ_i	10/min	8/min	3.2/min
Service rate, μ_i	6/min	20/min	0.857/min
Minimum channels, s_i	2	1	4
Traffic intensity, ρ_i	0.833	0.400	0.933

Results for Computation Center

for $M/M/1$ $\quad \pi_0 = 1 - \rho \qquad \rho = \lambda/\mu$

for $M/M/s$ $\quad \pi_0 = \left(\sum_{j=0}^{S-1} \frac{(s\rho)^j}{j!} + \frac{(s\rho)^s}{s!(1-\rho)} \right)^{-1} \qquad \rho = \lambda/s\mu$

	ρ	π_0
Input processor (M/M/2)	0.833	$10.976^{-1} = 0.09$
Central Processor (M/M/1)	0.40	0.6
Printer (M/M/4)	0.933	$\approx 141^{-1}$

for $M/M/1$

$$L_q = L - L_s = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$$

for $M/M/s$

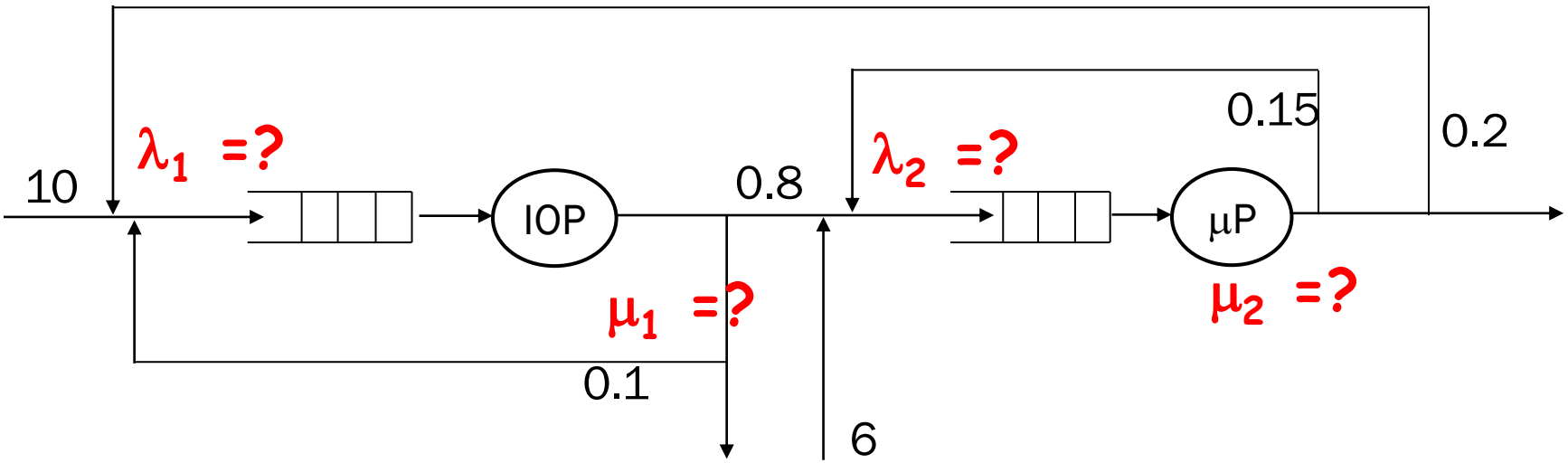
$$L_q = \frac{(s\rho)^s}{s!}\pi_0 \frac{\rho}{(1-\rho)^2} \qquad L_q = \frac{P(j \geq s)\rho}{1-\rho} \qquad P(j \geq s) = \frac{(s\rho)^s \pi_0}{s!(1-\rho)}$$

$$L = L_q + \frac{\lambda}{\mu} \qquad L_s = \frac{\lambda}{\mu} \qquad \text{Since } W_s = \frac{1}{\mu}$$

Results for Computation Center

Measure	Input processor	Central processor	Printer station	Total
Model	*M/M/2*	*M/M/1*	*M/M/4*	
L_q	3.788	0.267	12.023	16.077
W_q	0.379	0.033	3.757	4.169
L_s	1.667	0.400	3.734	5.801
W_s	0.167	0.050	1.167	1.384

Problem



$\lambda_1 = 10 + 0.1\lambda_1 + 0.2\lambda_2$

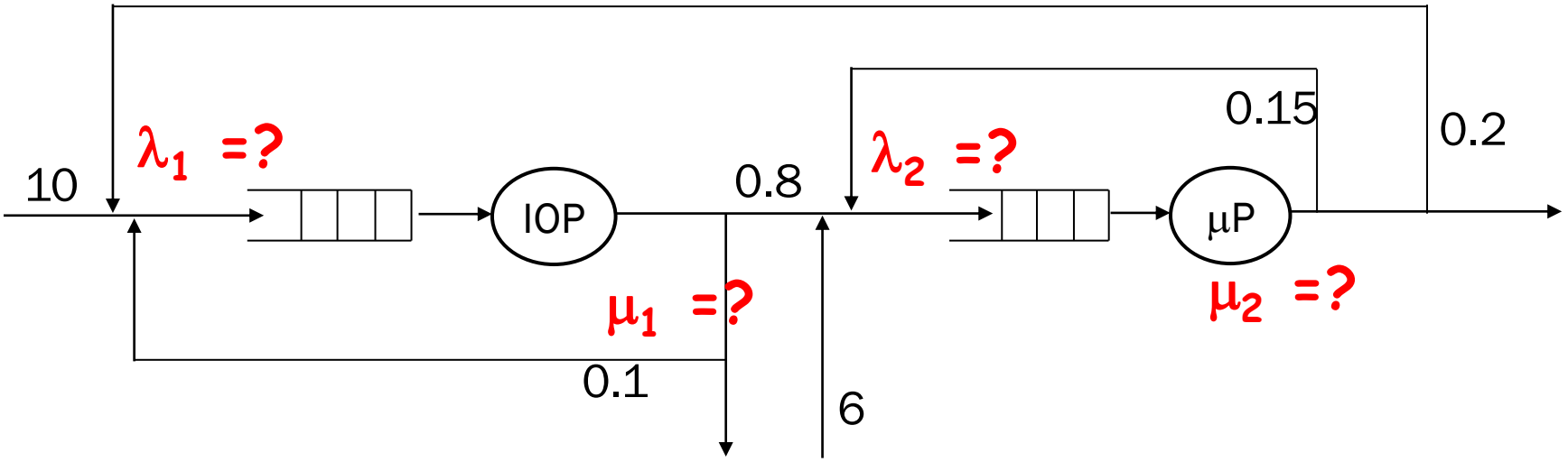
$\lambda_2 = 6 + 0.8\lambda_1 + 0.15\lambda_2$

Solves to

$\lambda_1 \approx 15.21$

$\lambda_2 \approx 22.15$

Problem



$$\mathbf{r} = [r_1 \quad r_2] = [10 \quad 6]$$

$$\mathbf{P} = \begin{bmatrix} 0.1 & 0.8 \\ 0.2 & 0.15 \end{bmatrix}$$

$$\mathrm{I} - \mathbf{P} = \begin{bmatrix} 0.9 & -0.8 \\ -0.2 & 0.85 \end{bmatrix}$$

$$(\mathrm{I} - \mathbf{P})^{-1} = \frac{1}{0.765 - 0.16} \begin{bmatrix} 0.85 & 0.8 \\ 0.2 & 0.9 \end{bmatrix}$$

$$[\lambda_1 \quad \lambda_2] = \mathbf{r(I - P)^{-1}}$$

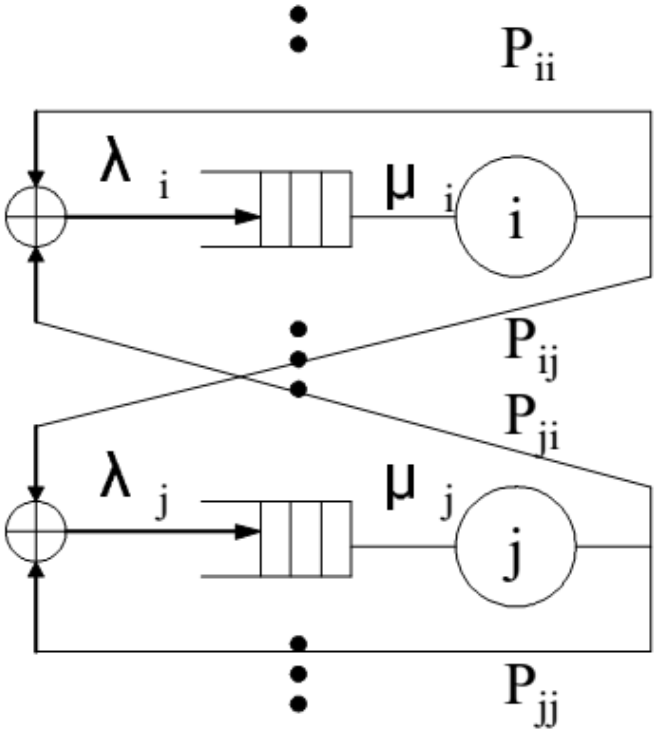
$$= [15.21 \quad 22.15]$$

Closed Queuing Network

- Fixed population of jobs circulate continuously and never leave
  - No arrivals from outside and no departures from the network
- Example: CPU job scheduling problem
- Since the number of jobs in the system is always constant, the distribution of jobs at different servers cannot be independent.

- Simplest case, *K* customers circulating among *m* queues
  - Each queue *i* has a server with exponentially distributed service time μ_i
  - P_{ij} be the routing probability from Q_i to Q_m

$$\sum_{j=1}^{m} P_{ij} = 1 \; ; \quad \forall\, i = 1, \ldots, m$$



Closed Queuing Network

- State of network at time t defined by

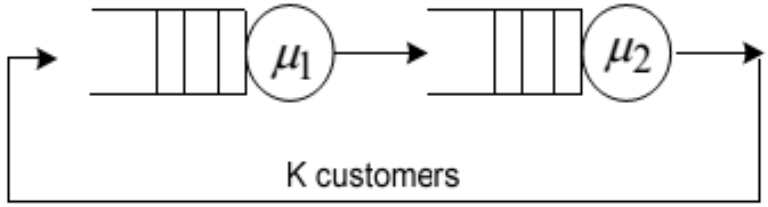
$$\mathbf{n} = \left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$$

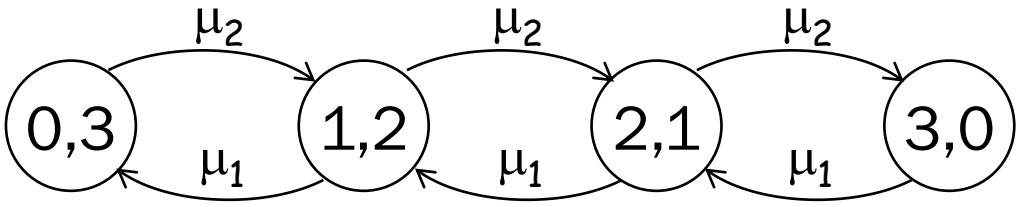
or simply, $= (n_1, n_2, \ldots, n_m)$

- which is m dimensional Markov process.
- The state space S is determined by

$$S = \left\{(n_1, n_2, \ldots, n_m): 0 \le n_i \le K \ \forall i; \sum_{i=1}^{m} n_i = K\right\}$$

- For example, $M = 2$, $K = 3$
- ($n1$, $n2$) state diagram





Closed Queuing Network

- Gordon and Newell (1967) showed that any arbitrary closed networks of m-server queues with exponentially distributed service times also have a product form solution
- The solution of the flow balance equation

$$\pi(\mathbf{n}) = \frac{1}{G(K,m)} \prod_{i=1}^{m} \rho_i^{n_i} \text{ where } \quad \rho_i = \frac{\lambda_i}{\mu_i}$$

- G(K, m) is a normalization constant so that $\sum_{\mathbf{n} \in S} \rho_i = 1$ given by

$$G(K,m) = \sum_{\mathbf{n} \in S} \prod_{i=1}^{m} \rho_i^{n_i}$$

Closed Queuing Network

- In order to determine $G(K, m)$ we need λ_i ; $\forall i$
- Flow conservation equation is

$$\lambda_i = \sum_{j=1}^{m} P_{ji}\lambda_j \quad ; \quad \forall\, i = 1,\ldots,m$$

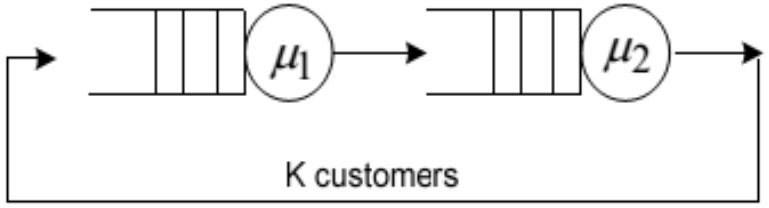
  - same as open network case without external arrivals or departures.
  - This equation have no unique solution. Fortunately, it turns out that we can use any solution to help us get steady-state probabilities

  - arrival rates are found relative to each other,

    set $\lambda_1 = 1$ or set $\lambda_1 = \mu_1$ => $\rho 1 = 1$

Closed Queuing Network

- For example, consider the tandem queue model with K=3. with $\mu_1 = 1$ and $\mu_2 = 2$



- From the diagram $P_{12} = P_{21} = 1$
- *i.e.* $\lambda_1 = \lambda_2$

- State space S = { (0,3), (1,2), (2,1), (3,0) }
- $G(K, m) = G(3,2) = \sum_{\mathbf{n}\in S} \prod_{i=1}^{m} \rho_i^{n_i} = \rho_2^3 + \rho_1\rho_2^2 + \rho_1^2\rho_2 + \rho_1^3$
- choosing $\lambda_1 = 1$ $\lambda_2 = 1$ $\rho_1 = 1$, $\rho_2 = 0.5$, $G(3,2) = 1.875$
- $\pi(0,3) = \frac{\rho_1^0\rho_2^3}{G(K,m)} = 0.0667$ $\pi(1,2) = \frac{\rho_1\rho_2^2}{G(K,m)} = 0.1333$

$\pi(2,1) = \frac{\rho_1^2\rho_2^1}{G(K,m)} = 0.2667$ $\pi(3,0) = \frac{\rho_1^3\rho_2^0}{G(K,m)} = 0.5333$

Closed Queuing Network

- The computation of $G(K, m)$ is difficult when the state space become large.
- For a closed network of m queues with K customers the number of states is given by

$$\text{Number of states} = {}_{K+m-1}C_{m-1}$$

- For even small networks, this is large.
  - For example $K = 9$, $m = 2$ = 3,628,800 states
  - And, direct computation of G(K,m) is very tedious

- One popular technique to determine is **Buzen's algorithm** (also called the **convolution algorithm**)

$$G(K, m) = G(K, m-1) + \rho_m G(K-1, m)$$

- With initial condition

$$G(0,m) = 1 \qquad m = 1,2,\ldots,M$$

$$G(k,1) = \rho_1^k \qquad k = 1,2,\ldots,K$$

Closed Queuing Network

- This can be computed in a simple tabular form

	ρ_1	ρ_2	ρ_3	...	ρ_M
	1	2	3	...	M
0	1	1	1	...	1
1	ρ_1	$\rho_1 + \rho_2$	$\rho_1 + \rho_2 + \rho_3$	...	
2	ρ_1^2	$\rho_1^2 + \rho_2(\rho_1 + \rho_2)$	...	...	
⋮	⋮	⋮	⋮	⋮	
K	ρ_1^K	...	...	...	

- The *i j* element in the table is computed by taking the *i*,(*j*-1) element and adding ρ_j . (*i* -1, *j*) *element*
- For the two queue example.

$$\lambda_1 = 0.5 \Rightarrow \lambda_2 = 0.5 \Rightarrow \rho_1 = 0.5, \quad \rho_2 = 0.25$$

	ρ_1	ρ_2
	1	2
0	1	1
1	0.5	0.75
2	0.25	0.4315
3	0.125	0.2344

Closed Queuing Network

- The performance measures can be written in terms of $G(K, M)$

$$L_i = \frac{1}{G(K,M)} \sum_{k=1}^{K} \rho_i^k G(K-k, M)$$

$$P(n_i \geq k) = \rho_i^k \frac{G(K-k,M)}{G(K,M)}$$

$$e_i = \lambda_i \frac{G(K-1,M)}{G(K,M)}$$

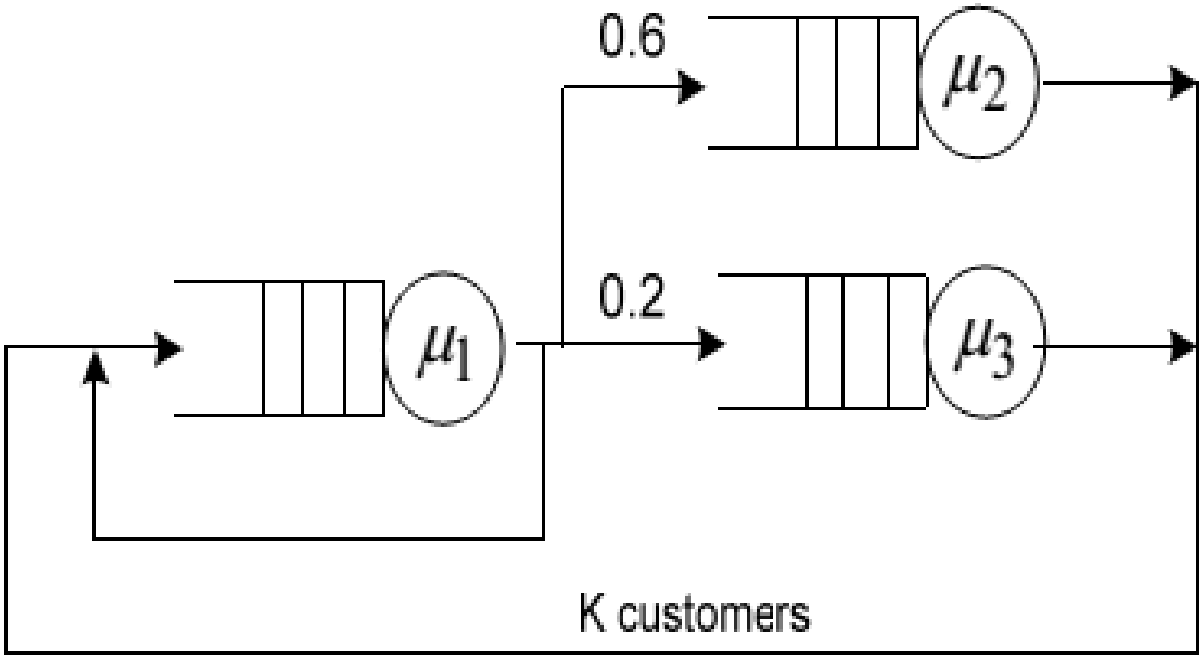
$$W_i = \frac{L_i}{e_i}$$

The effective server utilization $\rho_{e_i} = \frac{e_i}{\mu_i}$

- L_i = average customer at queue i, e_i = effective arrival rate

Example

Consider the simple model of a computer system shown below, queue 1– the CPU, queue 2–disk drive, and queue 3–I/O.
Given $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 1$, K = 4 jobs



From the diagram

$$r_{11} = 0.2\ ,\ r_{12} = 0.6\ ,\ r_{13} = 0.2\ ,\ r_{21} = r_{31} = 1,$$

Example

- Choosing $\lambda_1 = 10 \Rightarrow \lambda_2 = 6, \lambda_3 = 2$, and $\rho_1 = 1, \rho_2 = 1.2, \rho_3 = 2$
- Computing $G(4,3)$

	$\rho_1 = 1$	$\rho_2 = 1.2$	$\rho_3 = 2$
	1	2	3
0	1	1	1
1	1	2.2	4.2
2	1	3.64	12.04
3	1	5.368	29.448
4	1	7.4416	66.3376

Example

- Computing the effective arrival rates

$$e_1 = \lambda_1 \frac{G(3,3)}{G(4,3)} = 10 \times \frac{29.448}{66.3376} = 4.4391 \quad , e_2 = 2.6635 \quad , \quad e_3 = 0.8878$$

- The mean number in system at each queue

$$L_1 = \frac{1}{G(4,3)} \sum_{k=1}^{4} \rho_1^k G(4-k,3) = \frac{1}{G(4,3)} \left[\rho_1 G(3,3) + \rho_1^2 G(2,3) + \rho_1^3 G(1,3) + \rho_1^3 G(0,3) \right]$$

$$L_1 = 0.7038 \quad , \quad L_2 = 0.9347 \, , \quad L_3 = 2.3615$$

$$W_1 = L_1 / e_1 = 0.1585 \qquad W_2 = 0.3509 \qquad W_3 = 2.6599$$