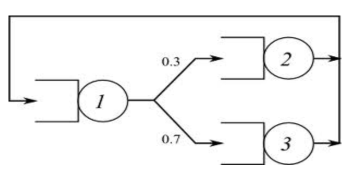
Queueing Networks

David Tipper
Associate Professor
Graduate Telecommunications and Networking Program
University of Pittsburgh

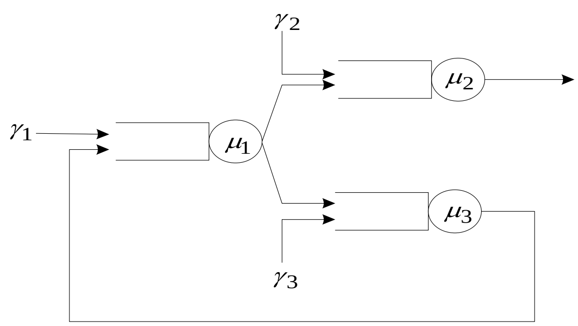
Networks of Queues

- Many communication systems must be modeled as a set of interconnected queues – which is termed a queueing network.
- Systems modeled by queueing networks can roughly be grouped into four categories
  - Open networks
  - Closed networks
  - Networks with population constraints (Loss Networks)
  - Mixed network



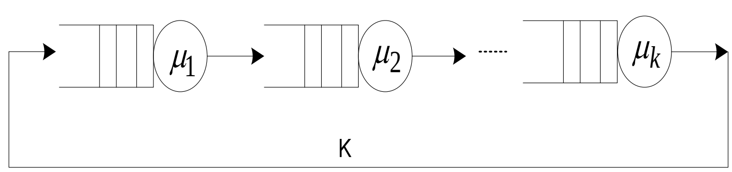
Open Networks

- Customers arrive from outside the system are served and then depart.
- Example: Packet switched data network.



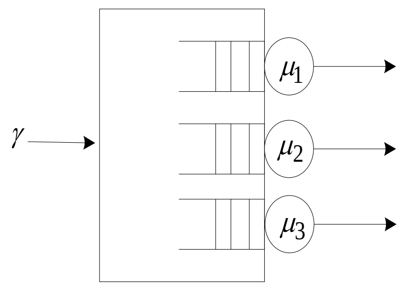
Closed Networks

- Fixed number of customers (K) are trapped in the system and circulate among the queues.
- Example: CPU job scheduling problem



Loss Networks with Population Constraints

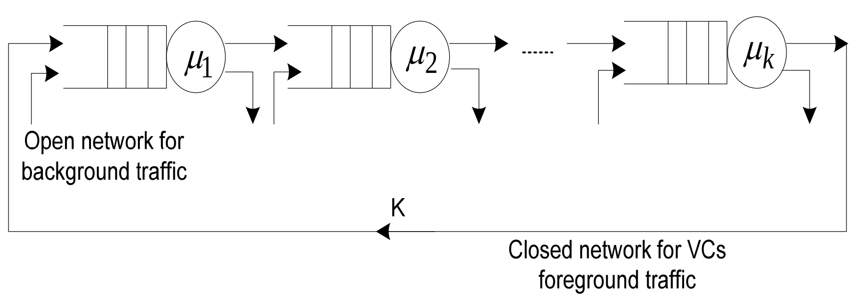
- Customers arrive from outside the system if there is room in the system. They enter, served and then depart.
- Example: queues sharing a common buffer pool – customers are lost when arriving to full system



5

Mixed Network

- Any combination of previous types.
- Example: simple model of virtual circuit that is window flow controlled.



6

Properties in Queueing Networks

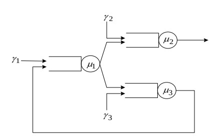
Queueing Networks exhibit behavior not seen in single queue scenarios

- **Jockeying**: Customers moving among parallel queues.
- **Blocking** – Customer waiting depart a server and join next queue is unable to due to limited waiting space, and therefore stays in server (blocking it.)
- **Routing** - Customer leaving a queue may have options as to where to go next
- **Forking** – Customer leaving a queue clones into multiple customers possibly going along different routes.
- **Joining** – Multiple customers are combined into a single customer
  - Forking and joining are used in models of parallel processing systems, packet fragmentation and reassembly.

7

Open Networks

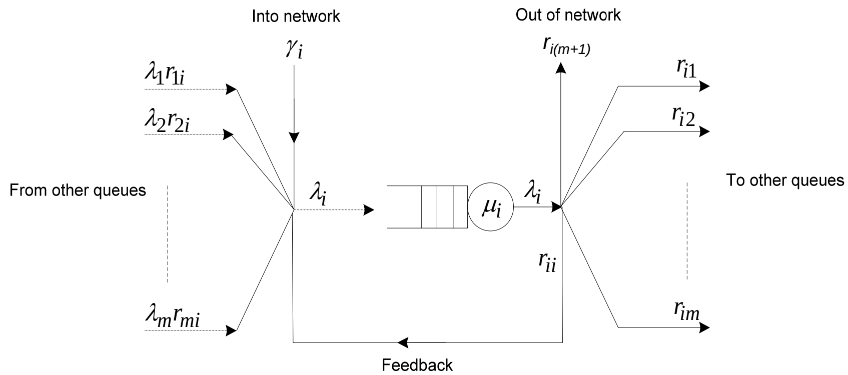
- Consider an open network
  - Assume arbitrary network of *M* queues with infinite waiting space
  - Customers arrive from outside the system are served and then depart. Note customer my visit several queues before departing including possibly visiting some queues more than once.
  - Service time of queue *i* is non-negative generally distributed with rate μ_i
  - Arrivals from outside the network to queue *i* occur according to general i.i.d. process with mean rate γ_i
  - The total mean customer arrival rate to queue *i* is denoted λ_i
  - Queues are G/G/1



8

Open Networks

An arbitrary queue *i* can be represented as



r_{ij} – routing probability that a customer completing service at queue *i* goes to queue *j*.

$r_{i(m+1)}$ – routing probability that a customer completing service at queue *i* leaves the network. (customer sink is dummy queue m+1)

Open Networks

$$\sum_{j=1}^{m+1} r_{i,j} = 1$$ routing fractions sum to one

- Let λ_i be the total mean customer arrival rate to queue *i*.

$$\rho_i = \frac{\lambda_i}{\mu_i} \qquad \lambda_i = \gamma_i + \sum_{j=1}^{m+1} r_{ji} \lambda_j$$

$$\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_m] \qquad \gamma = [\gamma_1, \gamma_2, \ldots, \gamma_m]$$

$$R = [r_{ij}] \quad 1 \le i \le m \quad 1 \le j \le m$$ ← Routing matrix – doesn't include sink

- The flow conservation equation can be written in matrix vector form as

$$\lambda = \gamma + \lambda R$$

$$\lambda(I - R) = \gamma \quad \Rightarrow \quad \lambda = \gamma (I - R)^{-1}$$

Relates external arrival rates and routing to determine the total flow at each queue

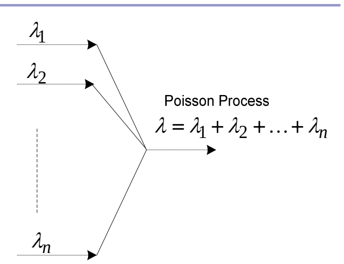
Jackson Networks

- James Jackson (UCLA Math professor) did the basic work on queueuing networks
- Jackson Networks – special class of open queueing networks
  - Network of *M* queues
  - There is only one class of customers in the network
  - A job can leave the network from any node
  - All service times are exponentially distributed with rate μ_i at queue i
  - The service discipline at all nodes is FCFS.
  - All external customer arrival processes are Poisson processes with rate γ_i at queue *i*

Open Jackson Networks

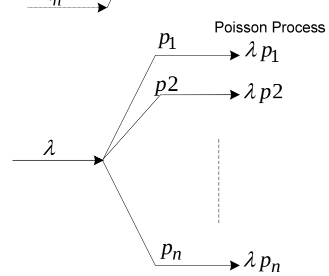
Now consider queue *i* in the Jackson network, from previous analysis we know

1. Merging of independent Poisson processes is Poisson with rate equal to the sum of the individual rates.



1. The departure process of an M/M/1 queue is Poisson with rate equal to input rate of the queue λ

2. Probabilistic splitting of a Poisson process results in a Poisson process.



Open Networks

Combining these results, we can see that the input and output processes of each queue *i* in the network is a Poisson process.

Let $\tilde{n}_i(t)$ be the number of customers in the system at queue *i* at the time *t*.

The state of the network is defined by the vector $\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$

$\left\{\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right),\ t \geq 0\right\}$ is a *m dimensional Markov process*

$P(n)$ denote steady state probability.

$$P(n) = \lim_{t\to\infty} P\left\{\tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_m(t) = n_m\right\}$$

$$P(n-1_i) = \lim_{t\to\infty} P\left\{\tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_i(t) = n_i - 1, \tilde{n}_m(t) = n_m\right\}$$

$\Rightarrow$ ***decrease by 1 in the ith queue***

13

Open Networks

$$P(n+1_i) = \lim_{t\to\infty} P\left\{\tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_i(t) = n_i + 1, \tilde{n}_m(t) = n_m\right\}$$

$\Rightarrow$ ***increase by 1 in the ith queue***

Writing the steady state flow balance equation

rate in to state *n* = rate out of state *n*

$$\sum_{i=1}^{m} \gamma_i \cdot P(n-1_i) + \sum_{i=1}^{m} \mu_i \cdot r_{i\,m+1} \cdot P(n+1_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} r_{ji} \cdot \mu_j \cdot P(n+1_j-1_i) = \left[\sum_{i=1}^{m} \lambda_i + \sum_{i=1}^{m} \mu_i\right] \cdot P(n)$$

The solution to the steady state flow balance equation is the **Product Form Solution**

$$P(n) = C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_m^{n_m} = C\prod_{i=1}^{m}\rho_i^{n_i}$$

where $\rho_i = \dfrac{\lambda_i}{\mu_i}$ and *C* is a constant.

14

Open Networks

$$\sum_{n_m=0}^{\infty}\sum_{n_{m-1}=0}^{\infty}\cdots\sum_{n_2=0}^{\infty}\sum_{n_1=0}^{\infty} C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_m^{n_m} = 1$$

results in $C = \prod_{i=1}^{m}(1-\rho_i)$

Hence, $P(n) = \prod_{i=1}^{m}(1-\rho_i)\rho_i^{n_i} \Rightarrow \rho_i < 1 \quad ;\forall i$ for stability

Essentially the product of M independent M/M/1 queues steady state probabilities,

$\pi_n = (1-\rho)\rho^n \Rightarrow$ (M/M/1 steady state)

$$P(n) = \prod_{i=1}^{m}\pi_{n_i}$$

Jackson's Theorem

- If in an open network ($\lambda_i < \mu_i$) holds for all queues i=1,…,M
  - the arrival rates λ_i can be computed by

$$\lambda = \gamma(I-R)^{-1}$$

  - The steady-state probability of the network can be expressed as the product of the state probabilities of the individual queues.

$$\pi(k_1,k_2,\ldots,k_N) = \pi_1(k_1)\cdot\pi_2(k_2)\cdot\ldots\cdot\pi_N(k_N)$$

  - The nodes of the network can be considered at independent M/M/1 queues with arrival rate λ_i and service rate μ_i .

Open Networks – Performance Measure

Since each queue i is a M/M/1 queue with ρ_i

$$L_i = \frac{\rho_i}{1-\rho_i} \qquad \pi_{n_i} = (1-\rho_i)\rho_i^{n_i}$$

$$W_i = \frac{1}{\mu_i - \lambda_i} \qquad W_{q_i} = \frac{\rho_i}{\mu_i - \lambda_i}$$

$\Rightarrow$ all M/M/1 measures apply (e.g., percentile of delay distribution, etc.)

For the network as a whole

LN – Average number of customers in network.

$$LN = \sum_{i=1}^{m} L_i = \sum_{i=1}^{m} \frac{\rho_i}{1-\rho_i}$$

γN – total average load on network.

$$\gamma N = \sum_{i=1}^{m} \gamma_i$$

17

Open Networks – Performance Measure

WN – Average delay through network.

$$WN = \frac{LN}{\gamma N} = \frac{1}{\gamma N}\sum_{i=1}^{m} \frac{\rho_i}{1-\rho_i} = \sum_{i=1}^{m} \frac{\lambda_i}{\gamma N} W_i$$

Note that in applying this solution to packet switched networks

$$\rho_i = \frac{\lambda_i}{\mu C_i}$$

where μ–average packet length, C_i–capacity of link i

Can extend model to includde deterministic delay d_{ij} corresponding to the time it takes a customer to move from the i th queue to the j th queue (propagation delay) still get Jackson network results as above, ***only WN changes***.

$$WN = \sum_{i=1}^{m} \frac{\lambda_i}{\gamma N}\left[W_i + \sum_{j=1}^{m} r_{ij} \cdot d_{ij}\right]$$

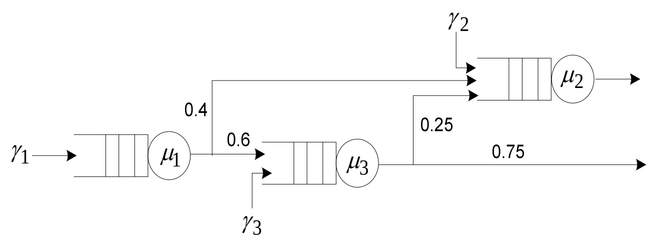
18

Open Networks – Example

Three node network shown below

Poisson external arrivals with $\gamma_1 = 0.5$, $\gamma_2 = 0.25$, $\gamma_3 = 0.25$

Exponential service at each queue with $\mu_1 = 1$, $\mu_2 = 1$, $\mu_3 = 1$



From the diagram $r_{12} = 0.4$, $r_{13} = 0.6$, $r_{32} = 0.25$, $r_{24} = 1.0$, $r_{34} = 0.75$

Example

- Given $\gamma_1 = 0.5$, $\gamma_2 = 0.25$, $\gamma_3 = 0.25 \Rightarrow \gamma = [0.5, \quad 0.25, \quad 0.25]$
- Solving the flow conservation equation for λ_i

$$R = \begin{bmatrix} 0 & 0.4 & 0.6 \\ 0 & 0 & 0 \\ 0 & 0.25 & 0 \end{bmatrix} \qquad \lambda = \gamma(I - R)^{-1}$$

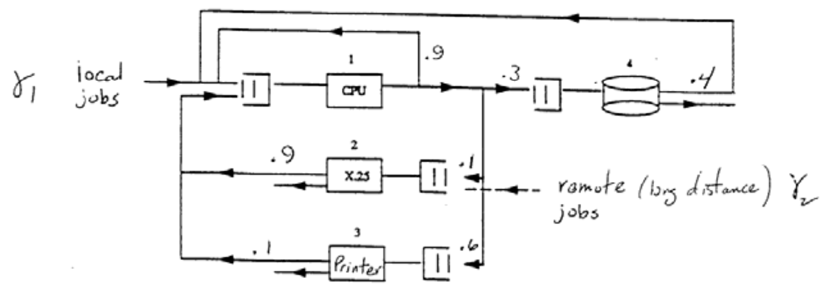
- using Matlab λ = [0.5, 0.5875, 0.55] $\Rightarrow$ $\rho_1 = 0.5$, $\rho_2 = 0.5875$, $\rho_3 = 0.55$
- The resulting average delay is

$$WN = \frac{1}{\gamma N} \sum_{i=1}^{3} \frac{\rho_i}{1 - \rho_i} = 3.646$$

Example 2

Consider a node in the SITA network (circa 1992) shown below

The interarrival of local and long distance jobs are exponentially distributed with rates $1/\gamma_1 = 10$ and $1/\gamma_2 = 60$. The processing time of jobs at the CPU, X.25, Printer and Disk queues are exponentially distributed with rates $\mu_1 = 10$, $\mu_2 = 1/6$, $\mu_3 = 1/5$, and $\mu_4 = 0.5$ respectively. (a) Determine the average delay W (b) Determine the requirements on γ_1 for maintaining system stability



Example 2

$\gamma = [.1, 1/60, 0,0]$ ➔ $\gamma N = .1167$

$$R = \begin{bmatrix} .9 & .01 & .06 & .03 \\ .9 & 0 & 0 & 0 \\ .1 & 0 & 0 & 0 \\ .4 & 0 & 0 & 0 \end{bmatrix} \qquad \lambda = \gamma(I-R)^{-1} = [1.5753, .0324, .0945, .0473]$$

$\rho = [\lambda_i/\mu_i] = [.1575, .1945, .4726, .0945]$

$$WN = \frac{1}{\gamma N}\sum_{i=1}^{m}\frac{\rho_i}{1-\rho_i} = 12.2484$$

Example 2

From the network diagram we get the following set of equations

(1) $\lambda_1 = \gamma_1 + .9\lambda_1 + .9\lambda_2 + .1\lambda_3 + .4\lambda_4$

(2) $\lambda_2 = \gamma_2 + .01\lambda_1 = 1/60 + .01\lambda_1$

(3) $\lambda_3 = .06\lambda_1$

(4) $\lambda_4 = .03\lambda_1$

Solving for λ_1 results in $\lambda_1 = 13.698\gamma_1 + .2055$, since $\rho_i = \lambda_i/\mu_i < 1$ for stability

Get for ρ_1 ➔ $\gamma_1 < 0.715$

Similarly from (2) get $\lambda_2 = .0187 + .13698\gamma_1$ ➔ $\rho_2 = .1123 + .822\gamma_1$ ➔ $\gamma_1 < 1.08$

Similarly from (3) get $\lambda_3 = .06\lambda_1$ ➔ $\rho_3 = 4.1\gamma_1 + .0616$ ➔ $\gamma_1 < 0.2283$

Similarly from (4) get $\lambda_4 = .03\lambda_1$ ➔ $\rho_4 = .8219\gamma_1 + .0123$ ➔ $\gamma_1 < 1.2$

The most restrictive constraint is at queue 3 the printer and is $\gamma_1 < 0.2283$

23

Additional Open Networks

- Many extensions to Jackson Networks exist – focus on cases were one gets a product form solution

$$P(n) = C\prod_{i=1}^{m}\rho_i^{n_i}$$

- Form of *C* depends on the system modeled
- Some of the additional features that can be modeled include: multiple classes of jobs, state dependent exponential servers, multiple servers, coxian service distribution with ∞ number of servers, fixed path routing, etc. See Chapter 6 in text
- Baskett, Chandy, Muntz and Palacios (BCMP) Networks are a widely used extension – different service disciplines (e.g., processor sharing, LIFO)

24

Closed Queueing Networks

Simplest case K customers circulating among M queues.
Each queue i has exponentially distributed service time μ_i

The routing probability for a customer completing service at queue i to go to queue j is r_{ij}

$$\sum_{j=1}^{m} r_{ij} = 1$$

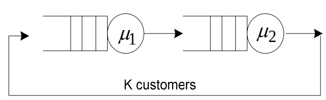
State of network defined by $\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$

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

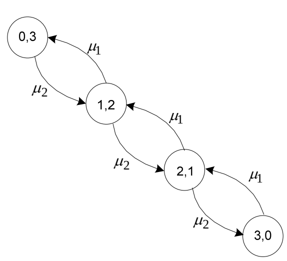
$$S = \left\{ (n_1, n_2, \ldots, n_m): 0 \le n_i \le K \;\; \forall i; \sum_{i=1}^{m} n_i = K \right\}$$

Closed Queueing Networks

For example, M = 2, K = 3



(n_1, n_2) state diagram



Steady state probabilities

$$P(n) = \lim_{t \to \infty} P\left\{ \tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_m(t) = n_m \right\}$$

Flow balance equation in steady state

rate in = rate out

$$\sum_{i=1}^{m} \sum_{j=1}^{m} r_{ji} \cdot \mu_j \cdot P(n + 1_j - 1_i) = \left(\sum_{i=1}^{m} \mu_i \right) \cdot P(n)$$

Closed Queueing Networks

The solution of the flow balance equation is once again a product form with

$$P(n) = \frac{1}{G(K,M)} \prod_{i=1}^{M} \rho_i^{n_i}$$

where $\rho_i = \dfrac{\lambda_i}{\mu_i}$ and

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

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

In order to determine $G(K,M)$ and $P(n)$ need $\lambda_i \,; \forall i$

Flow conservation equation is

$$\lambda_i = \sum_{j=1}^{m+1} r_{ji} \lambda_j$$ $\Leftarrow$ same as open network case without external arrivals or departures.

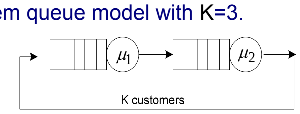
arrival rates are found relative to each other, set $\lambda_1 = 1$ or set $\lambda_1 = \mu_1 \rightarrow \rho_1 = 1$

27

Closed Queueing Networks

For example, consider the tandem queue model with K=3.

Customer with $\mu_1 = 1$, $\mu_2 = 2$



From the diagram $r_{12} = r_{21} = 1 \Rightarrow \lambda_1 = \lambda_2$

State space S = { (0,3), (1,2), (2,1), (3,0) }

$$G(K,M) = G(3,2) = \sum_{n \in S} \prod_{i=1}^{M} \rho_i^{n_i} = \rho_2^3 + \rho_1 \rho_2^2 + \rho_1^2 \rho_2 + \rho_1^3$$

choosing $\lambda_1 = 1 \Rightarrow \lambda_2 = 1 \Rightarrow \rho_1 = 1$, $\rho_2 = 0.5$

$G(3,2) = 1.875$ and $P(n) = \dfrac{1}{G(K,M)} \prod_{i=1}^{M} \rho_i^{n_i}$ results in

$$P(0,3) = \rho_2^3 / G(3,2) = 0.0667 \qquad P(1,2) = \rho_1 \rho_2^2 / G(3,2) = 0.1333$$

$$P(2,1) = \rho_1^2 \rho_2 / G(3,2) = 0.2667 \qquad P(3,0) = \rho_1^3 / G(3,2) = 0.5333$$

28

Closed Queueing Networks

To illustrate the arbitrary value for λ_1

Let λ_1 = 0.5 $\Rightarrow \lambda_2$ = 0.5 $\Rightarrow$ ρ_1 = 0.5 , ρ_2 = 0.25 $\Rightarrow$ $G(3,2)$ = 0.5333

From $P(n)$, one can compute the standard mean performance measures

$$L_i = \sum_{j=0}^{K} j\left(\sum_{n_i=j;\, n\in S} P(n) \right) \quad \Rightarrow \quad \sum_{i=1}^{M} L_i = K$$

From the example above,

$$L_1 = 1P(1,2) + 2P(2,1) + 3P(3,0) = 2.2667$$

$$L_2 = 1P(2,1) + 2P(1,2) + 3P(0,3) = 0.7333$$

Closed Queueing Networks

Note that to find W_i , one needs to find the effective arrival rate $e_i = \mu_i \left(1 - \sum_{n_i=0;\, n\in S} P(n) \right)$

The effective server utilization $\rho_{e_i} = \frac{e_i}{\mu_i} = \left(1 - \sum_{n_i=0;\, n\in S} P(n) \right)$ $\quad$ *note* $\rho_{e_i} < 1$

For the two queues example above

$$e_1 = \mu_1 (1 - P(0,3)) = 0.9333 \qquad \rho_{e_1} = 0.9333$$

$$e_2 = \mu_2 (1 - P(3,0)) = 0.9333 \qquad \rho_{e_2} = 0.4667$$

$$W_1 = L_1 / e_1 = 2.4286$$

$$W_2 = L_2 / e_2 = 0.7857$$

Closed Queueing Networks

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

$$\text{Number of states} = \binom{K+M-1}{M-1}$$

For even small networks, this is large. For example $K = 9$, $M = 2 \Rightarrow$ 3,628,800 states

One popular technique to determine $G(K,M)$ is **Buzen's algorithm**
(also called the **convolution algorithm**.)

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

Closed Queueing Networks

Note that
$$G(0,m) = 1 \qquad m = 1,2,\ldots,M$$
$$G(k,1) = \rho_1^k \qquad k = 1,2,\ldots,K$$

This can be computed in a simple tabular form

$$\begin{array}{cc}
 & \begin{array}{ccccc} \rho_1 & \rho_2 & \rho_3 & & \rho_M \\ 1 & 2 & 3 & \cdots & M \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ K \end{array} & \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \rho_1 & \rho_1+\rho_2 & \rho_1+\rho_2+\rho_3 & \cdots & \\ \rho_1^2 & \rho_1^2+\rho_2(\rho_1+\rho_2) & \cdots & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \\ \rho_1^K & \cdots & \cdots & \cdots & \end{bmatrix}
\end{array}$$

The *i j* element in the table is computed by taking the *i*,(*j*-1) element adding
$\rho_j \cdot (i-1,\ j)$ *element*

Closed Queueing Networks

For the two queue example previously discussed.

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

	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 Queueing Networks

One of the advantages of this technique is that the performance measures can be written in terms of $G(K,M)$

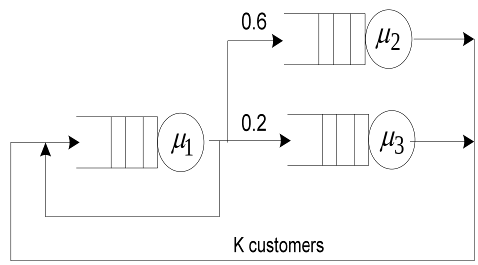
$$L_i = \frac{1}{G(K,M)} \sum_{k=1}^{K} \rho_i^k G(K-k,M)$$

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

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

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

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

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

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

37

Summary

- Overview of basic queueing networks
  - Categories
    - Open Networks
    - Closed Networks
  - Focused on queueing networks cases that yield a *Product Form* for state probabilities

$$P(n) = C \prod_{i=1}^{m} \rho_i^{n_i}$$

  - Efficient algorithms for closed networks

38