Queueing Networks

Two types of networks

	known	estimate
Open	λ	L, W
Closed	L	λ, W

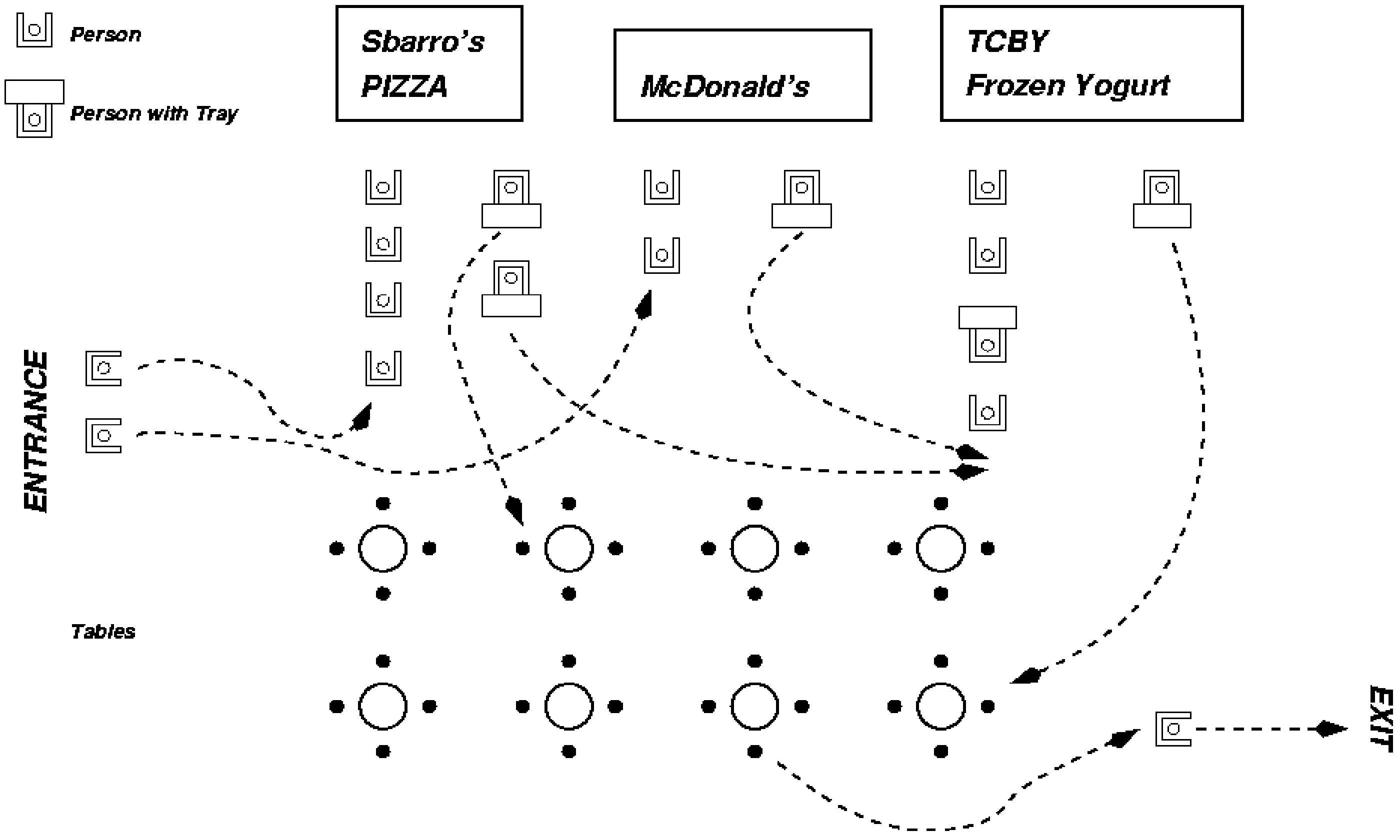
Examples

- internet traffic
- emergency room (Denardo)
- food court
- airport (*arrive*, ticket counter, security, passport control, gate, *board plane*)
- factory with serial production system and no material control after it enters

Examples

Open networks

Food Court



Examples

- factory with material controlled by keeping the number of items constant (CONWIP)
- factory with limited fixtures or pallets

Benefits

Queueing networks are often modeled as *Jackson networks.*

- Easy to compute performance measures (capacity, average time in system, average queue lengths).
- Easily gives intuition.
- Easy to optimize and to use for design.
- Valid for a large class of systems ...

Limitations

- ... but not everything. Storage areas must be infinite (ie, large enough so that blocking never occurs).

Open Jackson Networks

K single-server stations

$\mu_i =$ service rate at station i -- exponential

$\alpha_i =$ arrival rate to station i -- Poisson

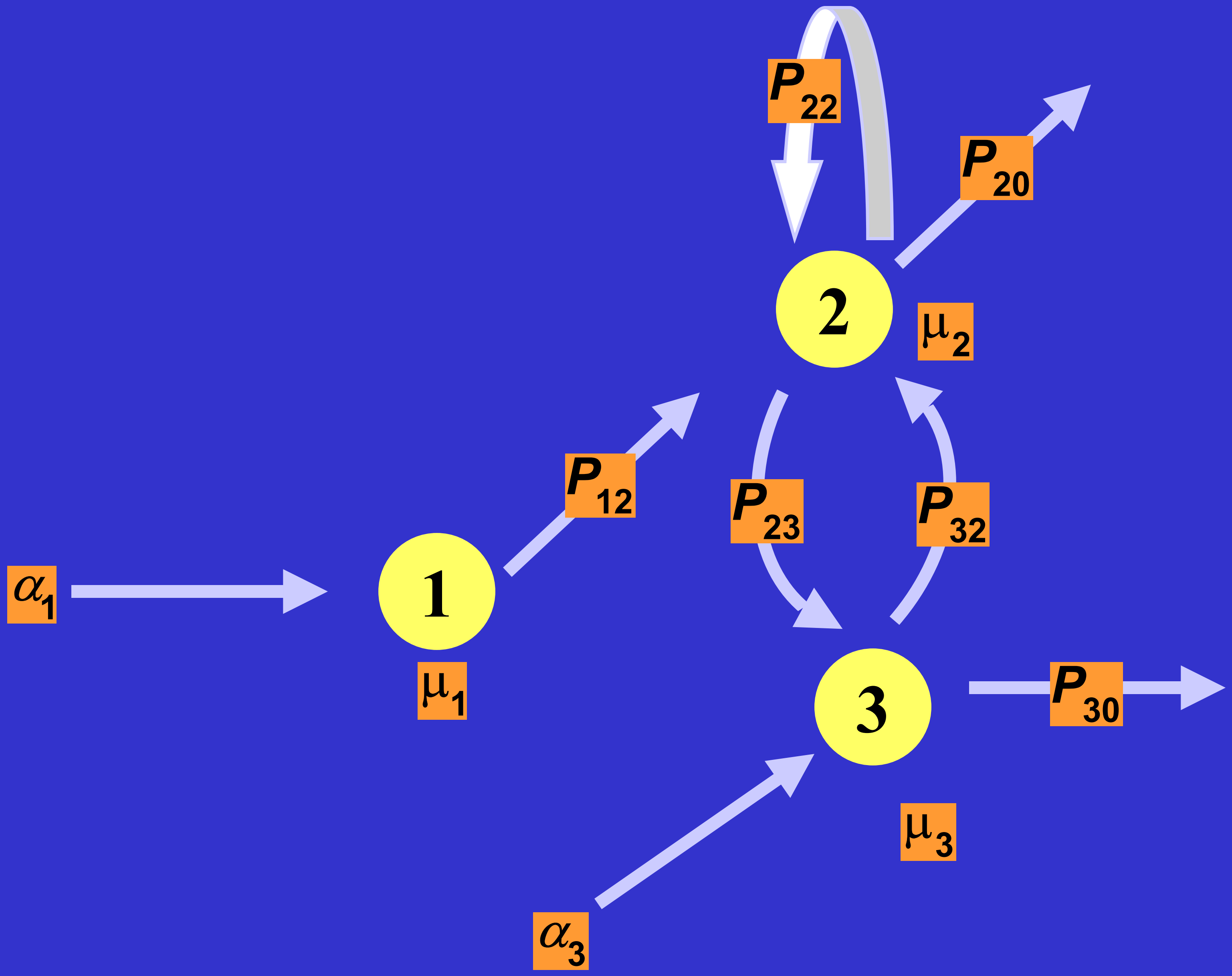
$P_{ij} =$ Pr(customer departing station i goes next to station j)

$$P_{i0} = 1 - \sum_{j=1}^{K} P_{ij} = \text{Pr(customer exits system after station i)}$$

Note:

1. **customers are probabilistically indistinguishable**
2. **routing is memoryless**
3. **FCFS at each station**
4. $P_{i0} > 0$ **for at least one i**

Example:



Solution to Open Jackson Networks

λ_i = effective arrival rate to station *i*

$$\lambda_i = \alpha_i + \sum_{j=1}^{K} \lambda_j P_{ji}$$ traffic equations *(Flow in = Flow out)*

$\lambda = \alpha + \lambda P$ (row vectors)

$\lambda = \alpha (I - P)^{-1}$

Let $\rho_i = \frac{\lambda_i}{\mu_i} < 1$

n_i = steady-state number of customers at station *i*

$\pi(n_1, \ldots, n_k) =$ steady-state probability

Product-form Solution:

$$\pi(n_1, \ldots, n_k) = \pi_1(n_1) \ldots \pi_k(n_k)$$

$$= \rho_1^{n_1}(1-\rho_1) \ldots \rho_k^{n_k}(1-\rho_k)$$

$\rho^n(1-\rho)$ *is the probability of n items in a single M/M/1 queue.*

$$L_i = E[n_i] = \frac{\rho_i}{1-\rho_i}$$

W_i = mean waiting time at station i in isolation $= \frac{1}{\mu_i(1-\rho_i)}$

Comments

1) Jackson guessed solution and verified that

$$\text{flux out of state } (n_1, \ldots, n_k) = \text{ flux into state } (n_1, \ldots, n_k)$$

2) Each station looks like its getting Poisson input, but it is not if there is any feedback in routing

3) If no customer overtaking is possible, then mean waiting time in system = sum of mean waiting times at the various stations in isolation

4) Product-form solution holds for multi-server stations

Closed Jackson Networks

Consider Open Jackson Network, but assume:

$\alpha_i = 0$ no external arrivals

$\sum_{j=1}^{K} P_{ij} = 1$ no departures

$\sum_{i=1}^{K} n_i(0) = N$ initial customers

Then $\sum_{i=1}^{K} n_i(t) = N$ for all t

Traffic equations $\lambda_i = \sum_{j=1}^{K} \lambda_i \, \rho_{j_i}$ can only be solved up to a scale constant

This gives relative throughout rates λ_i

Let $\rho_i = \dfrac{\lambda_i}{\mu_i}$

Closed Jackson Networks Cont.

Traffic equations $\lambda_i = \sum_{j=1}^{K} \lambda_i \, \rho_{j_i}$ **can only be solved up to a scale constant**

This gives relative throughout rates λ_i

Let $\rho_i = \dfrac{\lambda_i}{\mu_i}$

Product Form Solution:

$$\pi(n_1, \ldots, n_K) = \frac{\rho_1^{n_1} \ldots \rho_k^{n_k}}{C(K, N)}$$

where normalizing constant is

$$C(K,N) = \sum_{\substack{\{n : n_1 + \ldots + n_K = N\} \\ n_i \geq 0}} \rho_1^{n_k} \ldots \rho_k^{n_k}$$

Buzen's Algorithm

Goal:

Derive $C(K,N) = \sum_{\{x \geq 0: x_1 + \ldots + x_k = N\}} \pi_{i=1}^{K} f_i(x_i)$

* I changed notation $f_i(x_i) = \rho_i^{n_i}$

Step 1: Define an auxiliary function

$$c(k,n) = \sum_{\{x \geq 0: x_1 + \ldots + x_k = n\}} \pi_{i=1}^{K} f_i(x_i)$$

Step 2: Develop a recursive relationship

condition on $X_K = j$ and sum over j

$$c(k,n) = \sum_{j=0}^{n} \left[\sum_{\{x \geq 0: x_1 + \ldots + x_{k-1} + j = n\}} \pi_{i=1}^{K} f_i(x_i) \right]$$

$$= \sum_{j=0}^{n} f_k(j) \left[\sum_{\{x \geq 0: x_1 + \ldots + x_{k-1} = n-j\}} \pi_{i=1}^{K-1} f_i(x_i) \right]$$

$$= \sum_{j=0}^{n} f_k(j) c(k-1, n-j)$$

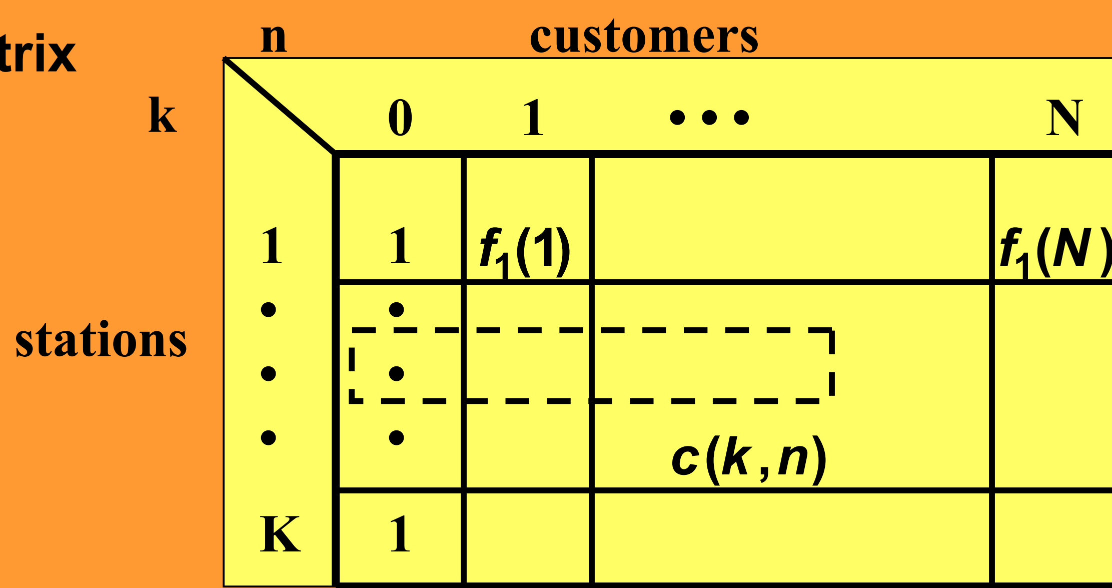
Buzen's Algorithm Cont.

Step 3: Initialization

no customers $c(k,0) = \underset{i=1}{\overset{K}{\pi}} f_i(0) = 1$

1 node $c(1,n) = \sum_{\{x_1=n\}} \underset{i=1}{\overset{1}{\pi}} f_i(x_i) = f_1(n)$

Step 4: Fill in KxN matrix



Work your way down the columns, starting from top left

Let u_i = utilization at station i

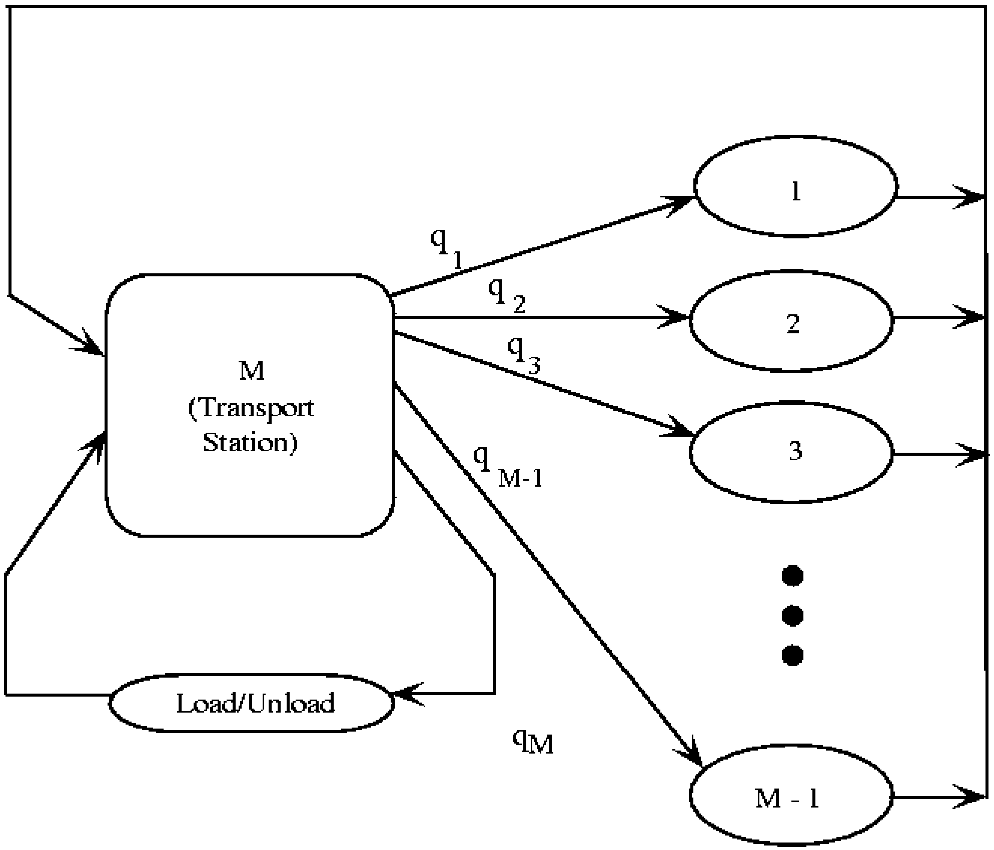
$$u_i = \frac{\rho_i C(K, N-1)}{C(K,N)}$$

Throughout rate at station i = $\mu_i u_i$

Closed Networks

Application

Simple FMS model



$$P_{iM} = 1 \text{ if } i \neq M$$

$$P_{Mj} = q_j \text{ if } j \neq M$$

$$P_{ij} = 0 \text{ otherwise}$$

The production rate is

$$\frac{C(M, N-1)}{C(M, N)}\mu_m$$

and $C(M, N)$ is easy to calculate in this case.