Lecture 8

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

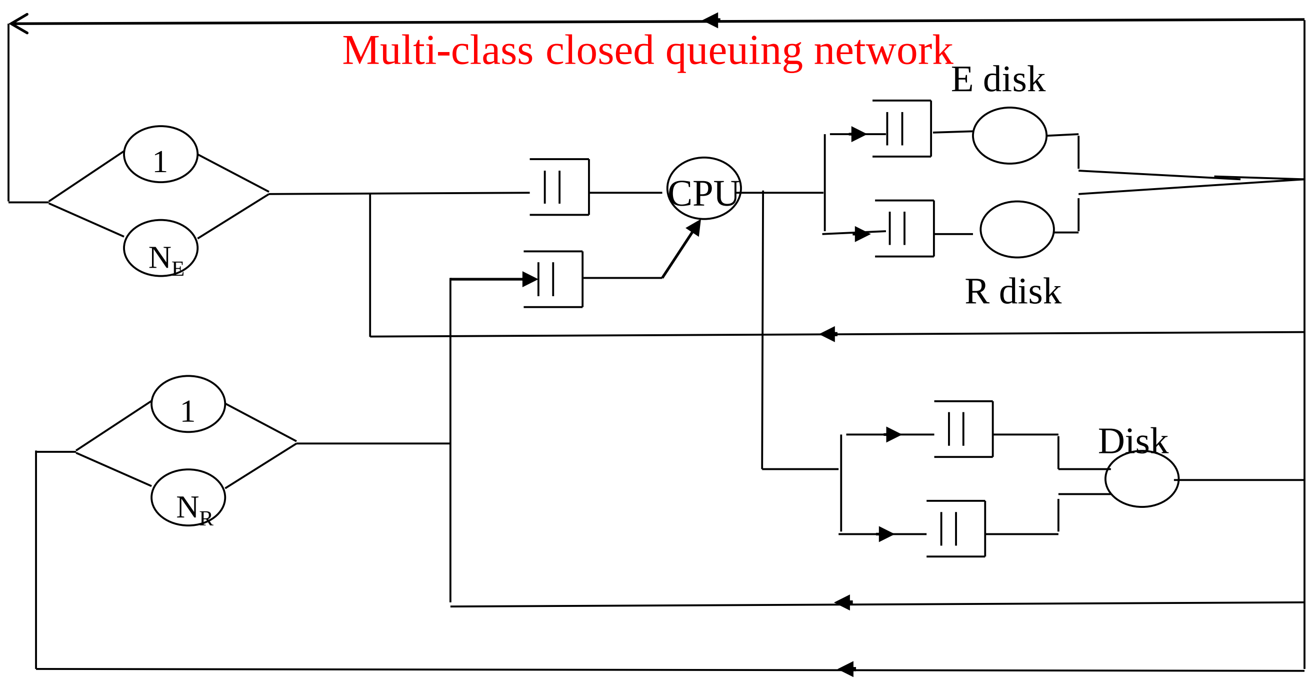
ECE 336
Stochastic Models for the Analysis of Computer Systems and Communication Networks

Outline of Lecture 8

- ❑ Summary of Lecture 7
- ❑ Multi-class Closed Queuing Networks
- ❑ Recursive and Approximation Algorithms
- ❑ Open and Mixed Networks

Why Multi-class Queuing Networks?

- Multiple customer classes arise naturally in practice:
  - end-to-end flow control of packets over several virtual circuits
  - multiple job types in a computer system (interactive, batch, etc.)
  - multiple part types in a manufacturing system



E ~ Edit Jobs
R ~ Run Jobs

Multi-class Closed Queuing Network - 1

- Population vector $(N_1, N_2, \ldots N_J)$, assumed constant in the network

$\underline{N} = (N_1, N_2, \ldots, N_J)$ $\quad J$ = number of customer classes

$|\underline{N}| = N_1 + N_2 + \ldots + N_J$ $\quad$ total population in the network

- M service centers in the network (infinite, single, multi-server, state-dependent)
- A class j customer after finishing processing will next go to node k with probability P_{ikj}, so that

$$v_{ij} = \sum_{k=1}^{M} P_{kij} v_{kj} \quad \forall 1 \le i, k \le M;\ 1 \le j \le J$$

v_{ij} ~ relative # of visits to node i by a class j customer

- Performance measures of interest

$$Q_i(\underline{N}) = \text{Queue length at node } i = \sum_{j=1}^{J} Q_{ij}(\underline{N})$$

$$Q_{ij}(\underline{N}) = \text{Queue length of class } j \text{ at node } i$$

$$R_{ij}(\underline{N}) = \text{Response time of class } j \text{ at node } i$$

$$X_j(\underline{N}) = \text{Network throughput of class } j;\ X_{ij}(N) = \text{Throughput of class } j \text{ at node } i$$

Multi-class Closed Queuing Network - 2

- Service demand, s_{ij}
  - For product-form networks, the queue discipline is restricted to one the following four forms:

Queue Discipline	Service demand distribution
FCFS	Exponential, class independent $s_{ij} = s_i, \forall j = 1,2,\ldots,J$
Processor sharing	any differentiable service time distribution
LCFS preempt resume	,,
Infinite Server	,,

- State $\underline{n}_1, \underline{n}_2, \ldots, \underline{n}_M$ where $\underline{n}_i = (n_{i1}, n_{i2}, \ldots, n_{iJ})$

$\Rightarrow$ has product-form: see Baskett, Chandy, Muntz and Palacios, JACM, Vol.22, April 1975, pp. 248-260 .... BCMP networks

Multi-class Closed Queuing Network - 3

$$p(\underline{n}) = p(\underline{n}_1, \underline{n}_2, \ldots, \underline{n}_M) = \frac{\prod_{i=1}^{M} Y_i(\underline{n}_i)}{G_M(\underline{N})}$$

$$Y_i(\underline{n}_i) = \frac{|\underline{n}_i|!}{n_{i_1}! n_{i_2}! \ldots n_{i_J}!} \frac{\prod_{j=1}^{J} (v_{ij} s_{ij})^{n_{ij}}}{\prod_{k=1}^{|\underline{n}_i|} \mu_i(k)}; 0 \le n_{ij} \le N_j \quad \forall i \text{ and } j$$

Note:

$$Y_i(\underline{n}_i) = \frac{|\underline{n}_i|!}{n_{i_1}! n_{i_2}! \ldots n_{i_J}!} \frac{\prod_{j=1}^{J} (v_{ij} s_{ij})^{n_{ij}}}{\prod_{k=1}^{|\underline{n}_i|} \mu_i(k)} = \frac{n_i}{n_{ij}} \frac{v_{ij} s_{ij}}{\mu_i(n_i)} Y_i(\underline{n}_i - \underline{e}_j) \forall i \text{ and } j$$

$$\sum_{i=1}^{M} n_{ij} = N_j, \ \text{constant } \forall j$$

$$G_M(\underline{N}) = \sum_{\text{all feasible } \underline{n}_i} \prod_{i=1}^{M} Y_i(\underline{n}_i)$$

- All the algorithms extend to multi-class networks.
- We will discuss only MVA (recursive and approximate)

Recursive MVA Equations - 1

- Recursive MVA:

  J=1 ➔ Equilibrium probabilities and performance measures for population N are related to those with population (N-1)

  J > 1 (multi-class) ➔ Probability distribution with $\underline{N}$ related to $\underline{N}-\underline{e}_j$, $1 \le j \le J$

- Basic MVA Equation:

$$p_i(k \mid \underline{n}) = \sum_{j=1}^{J} \frac{X_j(\underline{n}) v_{ij} s_{ij}}{\mu_i(k)} p_i(k-1 \mid \underline{n}-\underline{e}_j); \quad k = 1,2,\ldots,|\underline{n}|$$

$$k = k_{i_1} + k_{i_2} + \ldots + k_{i_J}$$

Proof:

$$p_i(\underline{k} \mid \underline{n}) = \frac{Y_i(\underline{k}) G_{M-\{i\}}(\underline{n}-\underline{k})}{G_M(\underline{n})} \qquad \underline{k} = (k_1, k_2, \ldots, k_J)$$

$$\underline{e}_j = (0 \ldots 0\ 1\ \ldots\ 0)$$

Note that $\quad Y_i(\underline{k}) = \sum_{j=1}^{J} \frac{k_{ij}}{k} Y_i(\underline{k}) \quad$ since $\quad k = k_{i1} + k_{i2} + \ldots + k_{iJ}$

Recursive MVA Equations - 2

$$p_i(\underline{k}\,|\,\underline{n}) = \sum_{j=1}^{J} \frac{k_{ij}}{k} Y_i(\underline{k}) \frac{G_{M-\{i\}}(\underline{n}-\underline{k})}{G_M(\underline{n})}$$

$$= \sum_{j=1}^{J} \frac{k_{ij}}{k} \frac{v_{ij}s_{ij}}{\mu_i(k)} \frac{k}{k_{ij}} Y_i(\underline{k}-\underline{e}_j) \frac{G_{M-\{i\}}(\underline{n}-\underline{k})}{G_M(\underline{n})}$$

$$= \sum_{j=1}^{J} \frac{v_{ij}s_{ij}}{\mu_i(k)} \frac{Y_i(\underline{k}-\underline{e}_j) G_{M-\{i\}}(\underline{n}-\underline{e}_j-\underline{k}+\underline{e}_j)}{G_M(\underline{n}-\underline{e}_j)} \frac{G_M(\underline{n}-\underline{e}_j)}{G_M(\underline{n})}$$

Recall:

$$Y_i(\underline{k}) == \frac{k}{k_{ij}} \frac{v_{ij}s_{ij}}{\mu_i(k)} Y_i(\underline{k}-\underline{e}_j)\ \forall i \text{ and } j$$

$$\Rightarrow p_i(\underline{k}\,|\,\underline{n}) = \sum_{j=1}^{J} \frac{X_j(n)v_{ij}s_{ij}}{\mu_i(k)} p_i(\underline{k}-\underline{e}_j\,|\,\underline{n}-\underline{e}_j)$$

- Summing over each class results in:

$$p_i(k\,|\,\underline{n}) = \sum_{j=1}^{J} \frac{X_j(n)v_{ij}s_{ij}}{\mu_i(k)} p_i(k-1\,|\,\underline{n}-\underline{e}_j)$$

$$Also\ p_i(\geq m_i\,|\,\underline{n}) = \sum_{j=1}^{J} \frac{X_j(\underline{n})v_{ij}s_{ij}}{\mu_i(m_i)} p_i(\geq m_i - 1\,|\,\underline{n}-\underline{e}_j)$$

$$p_i(0\,|\,\underline{n}) = 1 - \sum_{k=1}^{m_i-1} p_i(k\,|\,\underline{n}) - p_i(\geq m_i\,|\,\underline{n})$$

Recursive MVA Equations - 3

- MVA equations

$$know \quad Q_i(\underline{n}) = \sum_{j=1}^{J} Q_{ij}(\underline{n}) = \sum_{j=1}^{J} X_j(\underline{n}) R_{ij}(\underline{n}) = \sum_{k=1}^{|\underline{n}|} k p_i(k \mid \underline{n})$$

$$Q_i(n) = \sum_{j=1}^{|\underline{n}|} k p_i(k \mid \underline{n}) = \sum_{j=1}^{J} X_j(\underline{n}) v_{ij} s_{ij} \sum_{k=1}^{|\underline{n}|} \frac{k}{\mu_i(k)} p_i(k-1 \mid \underline{n} - \underline{e}_j)$$

- Using a procedure similar to single class case, we have

$$Q_i(\underline{n}) = \sum_{j=1}^{J} Q_{ij}(\underline{n}) = \sum_{j=1}^{J} \frac{X_j(\underline{n}) v_{ij} s_{ij}}{\mu_i(m_i)} [1 + Q_i(\underline{n} - \underline{e}_j) + \gamma_i(\underline{n} - \underline{e}_j)]$$

$$R_{ij}(\underline{n}) = \frac{v_{ij} s_{ij}}{\mu_i(m_i)} [1 + Q_i(\underline{n} - \underline{e}_j) + \gamma_i(\underline{n} - \underline{e}_j)]$$

$$where \quad \gamma_i(\underline{n} - \underline{e}_j) = \sum_{k=1}^{m_i - 1} k[\frac{\mu_i(m_i)}{\mu_i(k)} - 1] p_i(k-1 \mid \underline{n} - \underline{e}_j)$$

- Special cases: $R_{ij}(\underline{n}) = \dfrac{v_{ij} s_{ij}}{\mu_i}$ for infinite-server nodes

$R_{ij}(\underline{n}) = \dfrac{v_{ij} s_{ij}}{\mu_i} [1 + Q_i(\underline{n} - \underline{e}_j)]$ for single-server nodes

Recursive MVA Algorithm

- Algorithm:

$$Initialize \;\; Q_i(\underline{0}) = 0 \text{ at single, multi-server and state dependent nodes}$$

$$p_i(0 \mid \underline{0}) = 1 \text{ at multi-server and state dependent nodes}$$

Do $|\underline{n}| = 1, 2, \ldots, |\underline{N}|$

Do $n_1 = \max(0, |\underline{n}| - N_2 - N_3 - \ldots - N_J)$ to $\min(|\underline{n}|, N_1)$

....

Do $n_{J-1} = \max(0, |\underline{n}| - N_1 - N_2 - \ldots - N_{J-2} - N_J)$ to $\min(\underline{n} - n_1 - n_2 - \ldots - n_{J-2}, N_{J-1})$

$$n_J = |\underline{n}| - \sum_{j=1}^{J-1} n_j$$

$$\text{Compute } R_{ij}(\underline{n}) \qquad \forall i$$

$$X_j(\underline{n}) = \frac{n_j}{\sum_{i=1}^{M} R_{ij}(\underline{n})} \qquad \text{Little's Law}$$

$$Q_{ij}(\underline{n}) = R_{ij}(\underline{n}) X_j(\underline{n}) \; ; \quad Q_i(\underline{n}) = \sum_{j=1}^{J} Q_{ij}(\underline{n})$$

$$\text{Evaluate } p_i(k \mid \underline{n})$$

End Do

$$U_i(\underline{N}) = \begin{cases} 0 & \text{Infinite-server nodes} \\ \sum_{j=1}^{J} U_{ij}(\underline{N}), \quad U_{ij}(\underline{N}) = \dfrac{X_j(\underline{N}) v_{ij} s_{ij}}{\mu_i(m_i)} & \text{for all other nodes} \end{cases}$$

Need for Approximate MVA Algorithm

- Computational load $O(MJ\prod_{j=1}^{J}(N_j+1))$ ...exponential
- Storage $O(M\prod_{j=1}^{J}(N_j+1))$ ...exponential $\Rightarrow$ Need approximations

- Key Idea:

At $\underline{n}=\underline{N}$, we have

$$R_{ij}(\underline{N}) = f\{Q_i(\underline{N}-\underline{e}_j), p_i(k\,|\,\underline{N}-\underline{e}_j)\}$$

What if $Q_i(\underline{N}-\underline{e}_j)$ and $p_i(k\,|\,\underline{N}-\underline{e}_j)$ are estimated from statistics for population vector $\underline{N}$, suppose

$$Q_i(\underline{N}-\underline{e}_j) = g_{ij}[Q_i(\underline{N}), Q_{ij}(\underline{N}), X_j(\underline{N}), \ldots]$$

$$p_i(k\,|\,\underline{N}-\underline{e}_j) = h_{ij}[Q_i(\underline{N}), Q_{ij}(\underline{N}), X_j(\underline{N}), \ldots]$$

Approximate MVA Algorithm

- Then the MVA equations become:

$$R_{ij}(\underline{N}) = f\{g_{ij}(Q_i(\underline{N}),\ldots), h_{ij}(Q_i(\underline{N}),\ldots)\}$$

$$X_j(\underline{N}) = \frac{N_j}{\sum_{i=1}^{M} R_{ij}(\underline{N})}$$

$$Q_{ij}(N) = X_j(\underline{N}) R_{ij}(\underline{N})$$

$p_i(k \mid \underline{N})$ from MVA equation

$\underline{Y} = f(\underline{Y})$ suggests the following iteration

continue until convergence

- Start with an initial guess of performance statistics for population vector $\underline{N}$
- Estimate $g_{ij}(\underline{N})$, $h_{ij}(\underline{N})$
- Compute R_{ij}, X_j, Q_{ij}, $p_i(k|\underline{N})$

S-B and C-N MVA Approximations

- $\exists$ Two approximation schemes: Schweitzer-Bard (S-B) and Chandy-Neuse (C-N)
- S-B is a special case of C-N
- C-N Linearizer approximation:

$$know\ \ Q_i(\underline{N}-\underline{e}_j)=\sum_{m=1}^{J} Q_{im}(\underline{N}-\underline{e}_j)$$

1. $Q_{im}(\underline{N}-\underline{e}_j)$ is a linear function of the population of class m, N_m

$$Q_{im}(\underline{N}-\underline{e}_j)=\begin{cases} Q_{im}(\underline{N})+N_m D_{imj}; & m\neq j \\ \dfrac{\mathrm{N_j\text{-}1}}{\mathrm{N_j}}Q_{im}(\underline{N})+(N_j-1)D_{ijj}; & m=j \end{cases}$$

Alternatively, $Q_{im}(\underline{N}-\underline{e}_j)=(\underline{N}-\underline{e}_j)_m[F_{im}(\underline{N})+D_{imj}]$

$$where\ \ F_{im}(\underline{N})=\frac{Q_{im}(\underline{N})}{N_m}$$

C-N (Linearizer) MVA Approximation -1

- We find D_{imj} from

$$D_{imj} = F_{im}(\underline{N} - \underline{e}_j) - F_{im}(\underline{N})$$

Note that
$$\sum_{i=1}^{M} D_{imj} = \sum_{i=1}^{M} \frac{Q_{im}(\underline{N} - \underline{e}_j)}{(\underline{N} - \underline{e}_j)_m} - \sum_{i=1}^{M} \frac{Q_{im}(\underline{N})}{N_m} = 0 \quad \forall m \text{ and } j$$

- To get D_{imj}, we need performance statistics for population vector $\underline{N}$ and for population vectors $\underline{N}$-$\underline{e}_j$, 1≤j≤J
- If we assume $D_{imj} = 0 \quad \forall i,m \text{ and } j,$ The result is the S-B heuristic:

$$Q_{im}(\underline{N} - \underline{e}_j) = \begin{cases} Q_{im}(\underline{N}); & j \neq m \\ \dfrac{N_j\text{-}1}{N_j} Q_{ij}(\underline{N}); & j = m \end{cases}$$

- Thus, the S-B heuristic assumes that: 1) *classes don't interfere with each other*, and 2) *queue lengths are proportional to population*.

C-N (Linearizer) MVA Approximation -2

- We also need $p_i(k \mid \underline{N}-\underline{e}_j)$ to estimate $\gamma_i(\underline{N}-\underline{e}_j)$ at state-dependent nodes.
- The approximation for probability estimation is based on the following two assumptions:

1) Throughput approximation:

$$x_m(\underline{N}-\underline{e}_j) \approx x_m(\underline{N}) \qquad \forall m \,\&\, j$$

2) A closed network node *i* behaves as an open network node with arrival rate of class *m*, $x_m(\underline{N}-\underline{e}_j) \approx x_m(\underline{N})$ with a finite waiting place of $|\underline{N}|$-1.

Let $\hat{p}_i(k) \approx p_i(k \mid \underline{N}-\underline{e}_j)$

$$\mu_i(k)\,\hat{p}_i(k) = (\overbrace{\sum_{m=1}^{J} x_m(\underline{N}) v_{im} s_{im}}^{\lambda_i})\,\hat{p}_i(k-1)$$

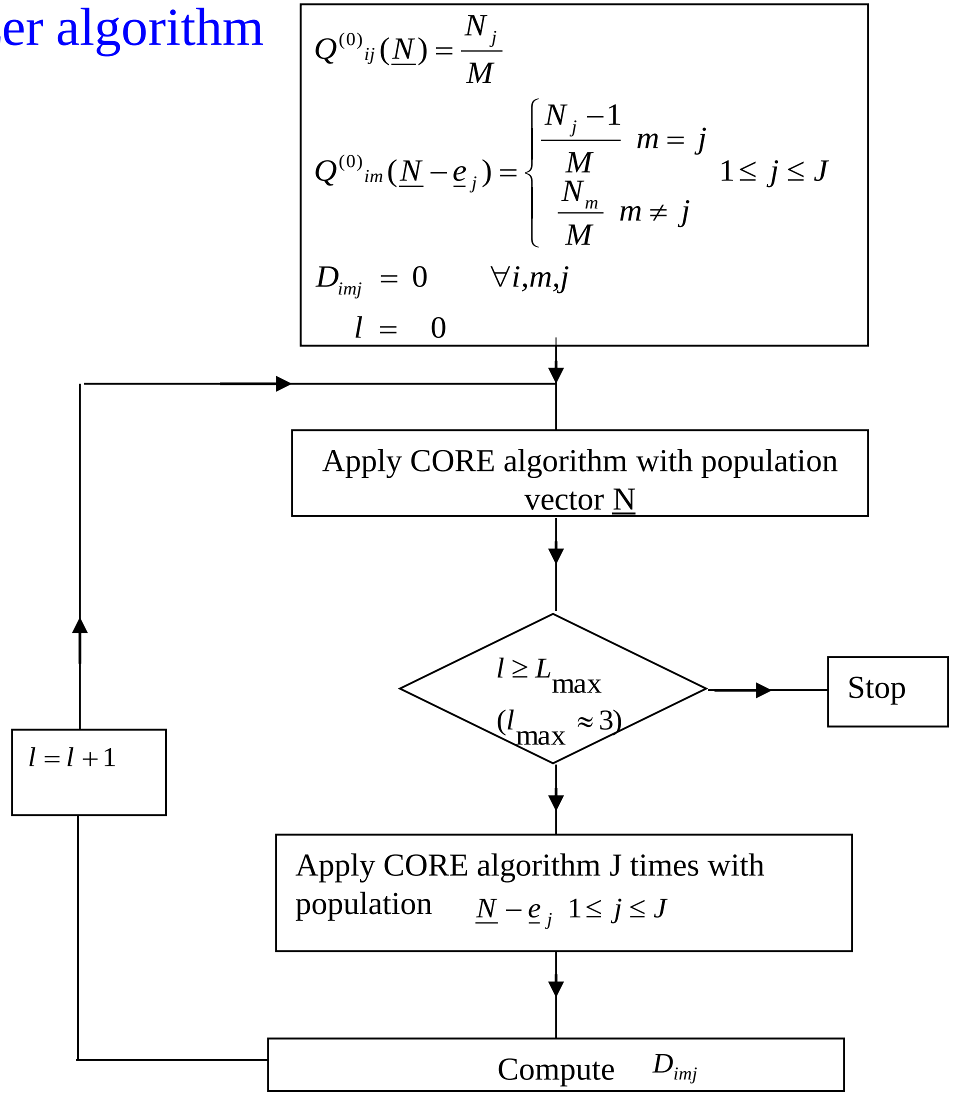
$$\hat{p}_i(k) = \frac{\lambda_i}{\mu_i(k)}\,\hat{p}_i(k-1), \quad k = 1,2,..,|\underline{N}|-1$$

Obtain $\hat{p}_i(0)$ from the normalization constraint: $\sum_{k=0}^{|\underline{N}|-1} \hat{p}_i(k) = 1$

This scheme is due to Krzesinski & Greyling and Pattipati *et al*

C-N Linearizer Algorithm

- C-N Linearizer algorithm



CORE Algorithm

- CORE algorithm:
  - Population vector: $\underline{n}$ → $\underline{N}$, → $\underline{N} - \underline{e}_j$

$$r = 0 \qquad TOL = \frac{1}{4000 + 16|n|}$$

Compute $$Q_{im}(\underline{n} - \underline{e}_j) = (\underline{n} - \underline{e}_j)_m \left[\frac{Q^{(r)}{}_{im}(\underline{n})}{n_m} + D_{imj} \right]$$

$$Q_i(\underline{n} - \underline{e}_j) = \sum_{m=1}^{J} Q_{im}(\underline{n} - \underline{e}_j)$$

Compute $p_i(k \mid \underline{n} - \underline{e}_j)$

Update $R_{ij}^{(r+1)}, \quad X_j^{(r+1)}, \quad Q_{ij}^{(r+1)}, \quad p_i^{(r+1)}(k \mid \underline{n})$ etc.

Check if $$\frac{|Q_{ij}^{(r+1)} - Q_{ij}^{(r)}|}{n_j} \leq TOL$$ Stop

else $r = r + 1$

Multi-class Network Example

- Example: J=2 customer classes; M=3, N_1=4, N_2=1

  node 1 ~ inf. Server; node 2 ~ single server; node 3 ~ multi-server (4 servers)

v_{ij}

↓i \ j→	1	2
1	1	1
2	1	1
3	1	0.1

s_{ij}

↓i \ j→	1	2
1	1	1
2	1	1
3	2	2

$$\mu_1 = \mu_2 = 1; \quad \mu_3(k) = \begin{cases} k; & k \le 4 \\ 4; & k > 4 \end{cases}$$

- Exact algorithm:

$$[Q_{ij}] = \begin{bmatrix} .673 & .262 \\ 1.979 & .685 \\ 1.348 & .053 \end{bmatrix}; \quad [R_{ij}] = \begin{bmatrix} 1 & 1 \\ 2.941 & 2.618 \\ 2.003 & 2.020 \end{bmatrix}$$

- S-B:

$$[Q_{ij}] = \begin{bmatrix} .636 & .233 \\ 2.092 & .72 \\ 1.272 & .047 \end{bmatrix}; \quad [R_{ij}] = \begin{bmatrix} 1 & 1 \\ 3.289 & 3.081 \\ 2.00 & 2.00 \end{bmatrix} \quad \text{error} = 12.4\%$$

C-N Linearizer provides near-exact answers

Open Research Problems

- Open research problems:

  1. Uniqueness of C-N linearizer solutions and convergence of those algorithms for both single and multi-class networks

  2. Uniqueness of S-B solution and convergence of S-B algorithms to multi-class networks *for finite populations*.

  3. Investigation of the accuracy of probability estimation schemes

  4. $\exists$ a relationship between the approximate MVA based on S-B approximation and convex minimization: exploration of computational techniques for solving the MVA equations via optimizations techniques.

  5. MVA algorithm is parallelizable. Investigation of parallel MVA algorithms.

See Pattipati *et al*., JACM, July 1990

Multi-class Open Queuing Networks -1

- <u>Open networks:</u> Analysis of open product-form networks is much simpler than their closed network counterparts.

$$p(\underline{n}) = p(\underline{n}_1, \underline{n}_2, \ldots, \underline{n}_M) = \prod_{i=1}^{M} Y_i(\underline{n}_i)\, p_i(\underline{0});$$

$$Y_i(\underline{n}_i) = \frac{|\underline{n}_i|!}{n_{i1}!\,n_{i2}!\ldots n_{iJ}!} \frac{\prod_{j=1}^{J} (\lambda_j v_{ij} s_{ij})^{n_{ij}}}{\prod_{k=1}^{|\underline{n}_i|} \mu_i(k)}$$

$\lambda_j =$ arrival rate of class j customers to the network

$$v_{ij} = p_{sij} + \sum_{k=1}^{M} P_{kij} v_{kj}$$

$$\begin{aligned}
p_i(\underline{n}_i) &= Y_i(\underline{n}_i)\, p_i(\underline{o}) \\
&= \sum_{j=1}^{J} \frac{n_{ij}}{|\underline{n}_i|} Y_i(\underline{n}_i)\, p_i(\underline{n}) \\
&= \sum_{j=1}^{J} \frac{\lambda_j v_{ij} s_{ij}}{\mu_i(|\underline{n}_i|)} Y_i(\underline{n}_i - \underline{e}_j)\, p_i(\underline{o}) \\
&= \sum_{j=1}^{J} \frac{\lambda_j v_{ij} s_{ij}}{\mu_i(|\underline{n}_i|)} p_i(\underline{n}_i - \underline{e}_j)
\end{aligned}$$

Multi-class Open Queuing Networks -2

- Let $k = |\underline{n}_i|$ and sum over all customers $\ni |\underline{n}_i| = k$, then

$$p_i(k) = \rho_i(k)\, p_i(k-1)$$

where $$\rho_i(k) = \frac{\sum_{j=1}^{J} \lambda_j v_{ij} s_{ij}}{\mu_i(k)} = \frac{r_i}{\mu_i(k)}$$

- So, each node behaves like a birth-death process: *M|M|1, M|M|m,* $M|M|\infty$, *SD* node. We can find performance measures via:

1) Infinite – server $M|M|\infty$ queue with multiple customer classes:

$$Q_i = \rho_i = \frac{r_i}{\mu_i} = \sum_{j=1}^{J} Q_{ij}\,;\; Q_{ij} = \rho_{ij} = \frac{\lambda_j v_{ij} s_{ij}}{\mu_i}$$

$$R_{ij} = \frac{v_{ij} s_{ij}}{\mu_i}$$

$$U_{ij} = U_i = 0$$

Multi-class Open Queuing Networks -3

2) Single server M|M|1 queue with multiple customer classes

$$Q_i = \frac{\rho_i}{1-\rho_i} = \sum_{j=1}^{J} Q_{ij} \ ; \quad Q_{ij} = \frac{\rho_{ij}}{1-\rho_i}$$

$$R_{ij} = \frac{v_{ij} s_{ij}}{\mu_i (1-\rho_i)}$$

$$U_i = \rho_i = \sum_{j=1}^{J} U_{ij}; \ U_{ij} = \rho_{ij}$$

3) Multi-server and state-dependent nodes ($\mu_i(m_i) = m_i \mu_i$ for multi-server nodes)

$$Q_i = \frac{\rho_i(m_i)}{1-\rho_i(m_i)}[1+\gamma_i] \ ; \quad Q_{ij} = \frac{\rho_{ij}(m_i)}{1-\rho_i(m_i)}[1+\gamma_i]; \ \rho_i(m_i) = \frac{r_i}{\mu_i(m_i)}$$

$$\gamma_i = \sum_{k=1}^{m_i-1} k\left[\frac{\mu_i(m_i)}{\mu_i(k)} - 1\right] p_i(k-1)$$

$$R_{ij} = \frac{v_{ij} s_{ij}}{\mu_i(m_i)(1-\rho_i(m_i))}[1+\gamma_i]$$

$$U_i = \frac{r_i}{\mu_i(m_i)} = \rho_i(m_i) = \sum_{j=1}^{J} U_{ij}; \ U_{ij} = \frac{\lambda_j v_{ij} s_{ij}}{\mu_i(m_i)}$$

Mixed Queuing Networks -1

- Mixed Networks: Idea is to convert into an equivalent closed network

$$v_{ij} = p_{sij} + \sum_{k=1}^{M} P_{kij} v_{kj}$$

$$P_{sij} \begin{cases} \rightarrow 0 \Rightarrow \text{closed class} \\ \rightarrow \neq 0 \Rightarrow \text{open class} \end{cases}$$

- We will consider only two class network for simplicity. Extension to multi-class networks is straightforward.

  - State at each node: $(n_0 \ \ n_1)_i = \underline{n}_i$ $\quad n_0 \sim open$, $\ n_1 \sim closed$

$$p(\underline{n}_1, \underline{n}_2, \underline{n}_M \mid N) = \frac{\prod_{i=1}^{M} Y_i(\underline{n}_i)}{G_M(N)}$$

where

$$Y_i(\underline{n}_i) = \frac{(n_{i0} + n_{i1})!}{n_{i0}! n_{i1}!} \frac{(\lambda v_{i0} s_{i0})^{n_{i0}} (v_{i1} s_{i1})^{n_{i1}}}{\prod_{k=1}^{n_{i0}+n_{i1}} \mu_i(k)}$$

Mixed Queuing Networks - 2

- Marginal probabilities are given by

$$p_i(n_0,n_1 \mid N) = \frac{(n_0+n_1)!}{n_0!n_1!}\frac{(\lambda v_{i0}s_{i0})^{n_0}(v_{i1}s_{i1})^{n_1}}{\prod_{k=1}^{n_0+n_1}\mu_i(k)} \cdot \frac{G_{M-\{i\}}(N-n_1)}{G_M(N)}$$

since $$\sum_{n_0=0}^{\infty}\sum_{n_1=0}^{N} p_i(n_0,n_1)=1$$

$$G_M(N) = \sum_{n_0=0}^{\infty}\sum_{n_1=0}^{N}\frac{(n_0+n_1)!}{n_0!n_1!}\frac{(\lambda v_{i0}s_{i0})^{n_0}(v_{i1}s_{i1})^{n_1}}{\prod_{k=1}^{n_0+n_1}\mu_i(k)} G_{M-\{i\}}(N-n_1)$$

Letting

$$\tilde{Y}_i(n_1) = \sum_{n_0=0}^{\infty}\frac{(n_0+n_1)!}{n_0!n_1!}\frac{(\lambda v_{i0}s_{i0})^{n_0}(v_{i1}s_{i1})^{n_1}}{\prod_{k=1}^{n_0+n_1}\mu_i(k)}$$

$$G_M(N) = \sum_{n_1=0}^{N}\tilde{Y}_i(n_1)G_{M-\{i\}}(N-n_i) \Rightarrow$$

$\tilde{Y}_i(n_1)$ is the equivalent closed network capacity function

Mixed Queuing Networks - 3

- Let us look at $\tilde{Y}_i(n_1)$ closely:

$$\tilde{Y}_i(n_1) = \frac{(v_{i1}s_{i1})^{n_1}}{\prod_{k=1}^{n_1}\mu_i(k)} \cdot \underbrace{\sum_{n_0=0}^{\infty}\frac{(n_0+n_1)!}{n_0!n_1!}\frac{(\lambda v_{i0}s_{i0})^{n_0}}{[\prod_{k=n_1+1}^{n_1+n_0}\mu_i(k)]}}_{\alpha_i(n_1)}$$

$$= \frac{(v_{i1}s_{i1})^{n_1}}{\prod_{k=1}^{n_1}\mu_i(k)} \cdot \alpha_i(n_1) = \frac{(v_{i1}s_{i1})^{n_1}}{\prod_{k=1}^{n_1}\mu_{eqi}(k)}$$

$$\Rightarrow \quad \mu_{eqi}(n_1) = \frac{\tilde{Y}_i(n_1 - 1)}{\tilde{Y}_i(n_1)} v_{i1}s_{i1} = \frac{\mu_i(n_1)\cdot\alpha_i(n_1-1)}{\alpha_i(n_1)}$$

where:

$$\alpha_i(n_1) = \sum_{n_0=0}^{\infty}\frac{(n_0+n_1)!}{n_0!n_1!}\frac{(\lambda v_{i0}s_{i0})^{n_0}}{\prod_{k=n_1+1}^{n_1+n_0}\mu_i(k)}$$

Mixed Queuing Networks - 4

- For single server nodes:

$$\alpha_i(n_1) = \sum_{n_0=0}^{\infty} \frac{(n_0+n_1)!}{n_0!\,n_1!} \rho_i^{n_0};$$

$$\rho_i = \frac{\lambda v_{i0} s_{i0}}{\mu_i} \quad \Leftarrow \text{ due to open classes only}$$

$$n_1 = 0 \quad \Rightarrow \quad \alpha_i(0) = \frac{1}{1-\rho_i}$$

$$n_1 = 1 \quad \Rightarrow \quad \alpha_i(1) = \frac{1}{(1-\rho_i)^2} \quad \text{etc.}$$

$$\text{so,} \quad \alpha_i(n_1) = \frac{1}{(1-\rho_i)^{n_1+1}}$$

- So, $\mu_{eqi}(k) = \dfrac{\mu_i \alpha_i(k-1)}{\alpha_i(k)} = (1-\rho_i)\mu_i$

➔ Open class has a localized effect of changing the service rate of a node.

Mixed Queuing Networks - 5

- Infinite server node:

$$\alpha_i(n_i) = \sum_{n_0=0}^{\infty} \frac{\rho_i^{n_0}}{n_0!} = e^{\rho_i} \quad \Rightarrow \quad \mu_{eq}(n_i) = \mu(n_i) \; as\ we\ should\ \mathrm{exp}ect!!$$

- State-dependent node:
  - For $n_1 \geq m_i$, $\alpha_i(n_1) = \dfrac{1}{[1-\rho_i(m_i)]^{n_1+1}} \quad \Rightarrow \mu_{eqi}(n_1) = [1 - \rho_i(m_i)]\mu_i(m_i)$
  - For $0 \leq n_1 \leq m_i - 1$, slightly more complex

$$\alpha_i(n_1) = \sum_{n_0=0}^{\infty} \frac{(n_0+n_1)!}{n_0!n_1!} \frac{(\lambda v_{i0} s_{i0})^{n_0}}{\prod_{k=n_1+1}^{n_1+n_0} \mu_i(k)}$$

$$= \sum_{n_0=0}^{m_i-n_1-1} \binom{n_0+n_1}{n_0} [\rho_i(m_i)]^{n_0} [\prod_{k=n_1+1}^{n_1+n_0} \frac{\mu_i(m_i)}{\mu_i(k)} - 1] + \frac{1}{[1-\rho_i(m_i)]^{n_1+1}}$$

Mixed Queuing Networks - 6

- So,

$$\mu_{eqi}(n_1) = \mu_i(n_1)[1-\rho_i(m_i)]\left[\frac{1+\Phi(n_1-1)\cdot[1-\rho_i(m_i)]^{n_1}}{1+\Phi(n_1)\cdot[1-\rho_i(m_i)]^{n_1+1}}\right]$$

where

$$\Phi(n_1) = \sum_{n_0=0}^{m_i-n_1-1}\binom{n_0+n_1}{n_0}[\rho_i(m_i)]^{n_0}[\prod_{k=n_1+1}^{n_1+n_0}\frac{\rho_i(m_i)}{\rho_i(k)}-1]$$

- So, to solve for performance measures of a mixed network
  - Compute ρ_i due to open classes
  - Compute effective service rates for closed classes
  - Solve closed queuing network via MVA for closed network statistics

Mixed Queuing Networks - 7

- Compute statistics of open classes
  - Statistics of open classes:

$$p_i(n_0 \mid N) = \sum_{n_1=0}^{N} p_i(n_0, n_1 \mid N)$$

$$= \sum_{n_1=0}^{N} \frac{(n_1+n_0)!}{n_1! n_0!} \frac{(\lambda v_{i0} s_{i0})^{n_0} (v_{i1} s_{i1})^{n_1}}{\prod_{k=1}^{n_1+n_0} \mu_i(k)} \frac{G_{M-\{i\}}(N-n_1)}{G_M(N)}$$

$$Q_i^{op}(N) = \sum_{n_0=1}^{\infty} p_i(n_0|N) n_0$$

$$= \sum_{n_0=1}^{\infty} \sum_{n_1=0}^{N} (n_1+n_0) \frac{\lambda v_{i0} s_{i0}}{\mu_i(n_1+n_0)} p_i(n_0-1, n_1 \mid N)$$

- Infinite-server node:

$$Q_i^{OP}(N) = \frac{\lambda v_{i0} s_{i0}}{\mu_i} = \rho_i$$

- Single-server node:

$$Q_i^{op}(N) = \rho_i [Q_i^{CL}(N) + 1 + Q_i^{OP}(N)]$$

$$\Rightarrow Q_i^{op}(N) = \frac{\rho_i}{1-\rho_i}[1 + Q_i^{CL}(N)]$$

Mixed Queuing Networks - 8

- State-dependent node

$$Q_i^{OP}(N)=\sum_{n_0=0}^{\infty} n_0 p_i(n_0\,|\,N)=\sum_{n_0=1}^{\infty}\sum_{n_1=0}^{N} n_0 p_i(n_0,n_1\,|\,N)$$

$$=\sum_{n_1=0}^{N}\sum_{n_0=1}^{\infty}\frac{(n_1+n_0)!}{(n_0-1)!n_1!}\frac{(\lambda v_{i0}s_{i0})^{n_0}}{\prod_{k=1}^{n_0+n_1}\mu_i(k)}(v_{i1}s_{i1})^{n_1}\frac{G_{M-\{i\}}(N-n_1)}{G_M(N)}$$

$$=\lambda v_{i0}s_{i0}\sum_{n_1=0}^{N}(n_1+1)\sum_{n_0=0}^{\infty}\frac{(n_1+n_0+1)!}{n_0!(n_1+1)!}\frac{(\lambda v_{i0}s_{i0})^{n_0}}{\prod_{k=1}^{n_0+n_1+1}\mu_i(k)}(v_{i1}s_{i1})^{n_1}\frac{G_{M-\{i\}}(N-n_1)}{G_M(N)}$$

using $\tilde{Y}_i(n_1)v_{i1}s_{i1}=\tilde{Y}_i(n_1+1)\mu_{eqi}(n_1+1)\Rightarrow\dfrac{\tilde{Y}_i(n_1+1)}{v_{i1}s_{i1}}=\dfrac{\tilde{Y}_i(n_1)}{\mu_{eqi}(n_1+1)}$

Mixed Queuing Networks - 9

$$Q_i^{OP}(N) = \lambda v_{i0} s_{i0} \sum_{n_1=0}^{N} \frac{n_1+1}{\mu_{eqi}(n_1+1)} p_i^{CL}(n_1 \mid N)$$

$$= \lambda v_{i0} s_{i0} \sum_{k=1}^{N+1} \frac{k}{\mu_{eqi}(k)} p_i^{CL}(k-1 \mid N)$$

$$= \frac{\lambda v_{i0} s_{i0}}{\mu_{eqi}(m_i)} \sum_{k=1}^{N+1} \frac{(k-1+1)\mu_{eqi}(m_i)}{\mu_{eqi}(k)} p_i^{CL}(k-1 \mid N)$$

$$= \frac{\rho_i(m_i)}{1-\rho_i(m_i)} \left\{ \sum_{k=1}^{m_i-1} \left[\frac{\mu_{eqi}(m_i)}{\mu_{eqi}(k)} - 1\right] k \; p_i^{CL}(k-1 \mid N) + Q_i^{CL}(N) + 1 \right\}$$

$$= \frac{\rho_i(m_i)}{1-\rho_i(m_i)} [1 + \gamma_i^{mixed} + Q_i^{CL}(N)]$$

Recall

$$p_i^{CL}(n_1 \mid N) = \frac{\tilde{Y}_i(n_1) G_{M-\{i\}}(N-n_1)}{G_M(N)}$$

where

$$\gamma_i^{mixed} = \sum_{k=1}^{m_i-1} k \left[\frac{\mu_{eqi}(m_i)}{\mu_{eqi}(k)} - 1\right] p_i^{CL}(k-1 \mid N)$$

Note the similarity to open & closed network measures.

Summary

- ❑ Multi-class Closed Queuing Networks
- ❑ Recursive and Approximation Algorithms
- ❑ Open and Mixed Networks

Reference: Lavenberg's Handbook