Queueing Networks

- used to model contention for multiple resources
- open queueing network: jobs arrive from external sources, circulate, and eventually depart

![Diagram of queueing networks with nodes labeled ECE-OIT, CMPSCI-OIT, OIT-MIT, and MIT-SRI, showing transitions with rates λ₁ and λ₂.]
- **closed queueing network**: fixed population of K jobs circulate continuously and never leave

- need to specify routing, service times, scheduling policies, arrival process
Feed Forward QNs

Consider two queue tandem system

\[ \begin{align*}
\lambda & \quad \mu_1 \\
\text{Box} & \quad \text{Circle} \quad \text{Box} \\
\mu_2
\end{align*} \]

Q: how to model?
by CTMC with state \((N_1(t), N_2(t))\). Assume stability; let \(\pi(i, j) = P(N_1 = i, N_2 = j)\) where \(N_i = \lim_{t \to \infty} N_i(t)\).
Feed Forward QNs

Balance equations are

\[ \lambda \pi(0,0) = \mu_2 \pi(0,1) \]

\[ (\lambda + \mu_2) \pi(0,i) = \mu_2 \pi(0,i + 1) + \mu_1 \pi(1,i - 1) \]

\[ (\lambda + \mu_1) \pi(i,0) = \mu_2 \pi(i,1) + \lambda \pi(i - 1,0) \]

\[ (\lambda + \mu_1 + \mu_2) \pi(i,j) = \mu_2 \pi(i,j + 1) + \mu_1 \pi(i + 1,j - 1) + \lambda \pi(i - 1,j) \]
Solution is

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

which can be demonstrated by substituting into balance equations. Here \( \rho_i = \lambda/\mu_i \).
Burke's Theorem: Departure process of \( M/M/1 \) queue is Poisson with rate \( \lambda \) independent of arrival process.

\[ \Rightarrow \text{can analyze feed forward networks using above result plus splitting and merging properties of Poisson process.} \]
Given \( k \) queue feed forward QN with Poisson arrivals exponential service times, and

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

\( n_i \geq 0; 1 \leq i \leq k; \)
Let $N = \sum_{1\leq i \leq k} N_i$, then

$$E[N] = \sum_{i=1}^{k} E[N_i]$$

$$= \sum_{i=1}^{k} \frac{\lambda_i}{(\mu_i - \lambda_i)}$$

Little’s result gives $E[T] = E[N]/\lambda$
Open QN with feedback

- $k$ queues
- exponential service times, with mean $1/\mu_i$ for queue $i$
- external Poisson arrival processes to nodes with rate $r_i$ to node $i$
- $p_{i,j}$ - prob. job leaving queue $i$ goes to queue $j$,

$$\sum_{j=1}^{k} p_{i,j} < 1 \Rightarrow P(\text{job leaves system after } i) = 1 - \sum_{j=1}^{k} p_{i,j}$$
- $\lambda_j$ job arrival rate to queue $j$

$$\lambda_j = r_j + \sum_{i=1}^{k} \lambda_i p_{i,j} \quad 1 \leq j \leq k$$

- $N_j$ steady state number at node $j$

$$\pi(n_1, \ldots, n_k) = P(N_1 = n_1, \ldots, N_k = n_k),$$

$$n_i \geq 0; 1 \leq i \leq k$$
Open QN with Feedback

Flows not generally described by Poisson processes

Q: what is $\pi(n_1, \ldots, n_k)$?

Jackson's Theorem for open queueing networks.

$$\pi(n_1, \ldots, n_k) = \prod_{i=1}^{k} (1 - \rho_i) \rho_i^{n_i},$$

$$n_i \geq 0; 1 \leq i \leq k$$

Each queue behaves like $M/M/1$ queue in isolation!

Proof: substitute into balance equations.
Example

\[ r_1 = 4 \quad \mu_1 = 8 \]

\[ r_2 = 5 \quad \mu_2 = 10 \]
Example

\[ r_1 = 4 \quad \mu_1 = 8 \]

\[ r_2 = 5 \quad \mu_2 = 10 \]

\[
\begin{align*}
\lambda_1 &= 4 + \frac{\lambda_2}{4} \\
\lambda_2 &= 5 + \frac{\lambda_1}{2}
\end{align*}
\]

\[ \Rightarrow \lambda_1 = 6, \quad \lambda_2 = 8 \]
\[
\pi(n_1, n_2) = \frac{1}{4} \left( \frac{3}{4} \right)^{n_1} \frac{1}{5} \left( \frac{4}{5} \right)^{n_2}
\]

\[
E[N] = \sum_{i=1}^{2} E[N_i] = \sum_{i=1}^{2} \lambda_i / (\mu_i - \lambda_i)
\]

\[
= 3 + 4 = 7
\]

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

\[ r_1 = 4 \quad \mu_1 = 8 \]

\[ r_2 = 5 \quad \mu_2 = 10 \]
Example continued

- Define $T^{(i)}$ to be response time for a job that enters at queue $i$

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

Results hold when nodes are
- multiple server nodes (M/M/c),
- infinite server nodes with arbitrary service time distr.
- finite buffer nodes (M/M/c/K) (careful about interpretation of results),
- PS 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 $p_{i,j}$, $1 \leq i, j \leq M$
- visit ratios, $\{v_i\}$. If $v_1 = 1$, then $v_i$ is mean number of visits to queue $i$ between visits to queue 1

\[
v_1 = 1
\]

\[
v_i = \sum_{j=1}^{M} v_j p_{j,i} \quad i = 2, \ldots, M
\]
\( \gamma_i: \text{ thruput of queue } i, \)
\[ \frac{\gamma_i}{\gamma_j} = \frac{v_i}{v_j}, \quad 1 \leq i, j \leq M \]
can think of \( v_i \) as relative thruput
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 $p_{i,j}$, $1 \leq i,j \leq M$
- Node throughputs, $\{\gamma_i\}$.

$$\gamma_i = \sum_{j=1}^{M} \gamma_j p_{j,i}, \quad i = 1, \ldots, M$$
Modeling Closed QNs

- model as continuous time Markov chain; state is $(N_1(t), \ldots, N_M(t))$ where $N_i(t)$ is the number of jobs at station $i$.
- $(N_1, \ldots, N_M) = \lim_{t \to \infty} (N_1(t), \ldots, N_M(t))$
- $\pi(n_1, \ldots, n_M) = P(N_1 = n_1, \ldots, N_M = n_M)$
Example
Transition rate diagram
Balance equations

\[ \sum_{i=1}^{M} 1\{n_i > 0\}(1 - p_{ii})\pi(n_1, ..., n_M)\mu_i = \]

\[ \sum_{i=1}^{M} \sum_{j \neq i} 1\{n_j > 0\}\mu_i p_{ij} \times \]

\[ \pi(n_1, ..., n_{i+1}, ..., n_{j-1}, ..., n_M) \]
Steady State Solution

Theorem (Gordon and Newell).

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

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

Theorem (Gordon and Newell).

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

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

Consider previous 3 queue example, \( \mu_1 = 2, \mu_2 = 3, \mu_3 = 2, \)
\( p_{1,2} = 3/4, p_{1,3} = 1/4 \). Then \( v_1 = 1, v_2 = 3/4, v_3 = 1/4 \) and

\[ \pi(\vec{n}) = \frac{1}{G(N)} \frac{1}{(1/2)^{n_1}} \frac{1}{(1/4)^{n_2}} \frac{1}{(1/8)^{n_3}} \]
Example
Comments:

- quantity \( D_i = v_i / \mu_i \) sometimes referred to as mean service demand at node \( i \) between visits to node 1
Extensions

In addition to FCFS single server queues, network can include FCFS multiple server queues (expo. service times), infinite server queues (general service times), PS single and multiple server queues (general service times), state dependent server queues

Solution is

\[
\pi(\vec{n}) = \frac{1}{G(N)} \prod_{i=1}^{M} h_i(n_i) \quad \vec{n} \geq \vec{0}; \quad \sum_{i=1}^{M} n_i = N
\]
$$h_i(n) \text{ given as}$$

**single server queue**

$$h_i(n) = \left( \frac{v_i}{\mu_i} \right)^n \quad n = 0, 1, \ldots$$

**queue with** $c_i$ **servers**

$$h_i(n) = \begin{cases} 
1/n!(v_i/\mu_i)^n, & 0 \leq n \leq c_i \\
1/(c_i!c_i^{n-c_i})(v_i/\mu_i)^n, & c_i < n
\end{cases}$$

**infinite server queue**

$$h_i(n) = \frac{1}{n!} \left( \frac{v_i}{\mu_i} \right)^n \quad 0 \leq n$$
state dependent sever \((\mu_i(k), \ k \geq 1)\)

\[ h_i(n) = \prod_{k=1}^{n} \frac{v_i}{\mu_i(k)} \quad 0 \leq n \]

\[ h_i(0) = 1 \text{ in all cases} \]
Algorithms for Computing Performance Metrics

- direct method for computing $G(N)$
- convolution algorithm for computing $G(N)$
- mean value analysis algorithm (MVA) for directly computing $E[N_i], E[W_i], E[T_i], \gamma_i$
Direct approach

\[ G(N) = \sum_{\text{all feasible } \bar{n}^i = 1} \prod_{i=1}^{M} h_i(n_i) \]

Problem is that no. of states is \( \binom{N + M - 1}{N} \) which grows exponentially in \( M \) and \( N \)
Convolution Algorithm

Notation

\[ S(n,m) = \{(n_1, \ldots, n_m) \mid n_i \geq 0; 1 \leq i \leq m; \sum_{i=1}^{m} n_i = n \} \]

\[ H_m(\vec{n}) = \prod_{i=1}^{m} h_i(n_i), \quad |\vec{n}| = m, \ n_i \geq 0, \ 1 \leq i \leq m \]

\[ g(n,m) = \sum_{\vec{n} \in S(n,m)} H_m(\vec{n}) \]

where

\[ G(N) = g(N,M) \]
General Case

\[ g(n, m) = \sum_{k=0}^{n} \sum_{\bar{n} \in S(n, m)} H_m(\bar{n}) \]

\[ = \sum_{k=0}^{n} \sum_{\bar{n} \in S(n, m)} \prod_{j=1}^{m} h_j(n_j) \]

\[ = \sum_{k=0}^{n} h_m(k) \sum_{\bar{n} \in S(n, m)} \prod_{j=1}^{m-1} h_j(n_j) \]

\[ = \sum_{k=0}^{n} h_m(k) \sum_{\bar{n} \in S(n-k, m-1)} H_{m-1}(\bar{n}) \]

\[ = \sum_{k=0}^{n} h_m(k) g(n-k, m-1) \]
**General Case**

Initial condition

\[ g(n, 0) = \begin{cases} 
1, & n = 0, \\
0, & n = 1, \ldots 
\end{cases} \]

Computational complexity: \( O(MN^2) \), where cost of computing \( h_m(k) \) taken as one
Single server nodes

\[ g(n, m) \]

\[ = \sum_{\tilde{n} \in S(n, m), n_m > 0} H_m(\tilde{n}) + \sum_{\tilde{n} \in S(n, m), n_m = 0} H_m(\tilde{n}) \]

\[ = \sum_{\tilde{n} \in S(n, m), n_m > 0} \left( \frac{v_m}{\mu_m} \right)^n m^{-1} \prod_{i=1}^{m-1} h_i(n_i) + \sum_{\tilde{n} \in S(n, m-1)} H_{m-1}(\tilde{n}) \]

\[ = \frac{v_m}{\mu_m} \sum_{\tilde{n} \in S(n, m), n_m > 0} \left( \frac{v_m}{\mu_m} \right)^{n-1} m^{-1} \prod_{i=1}^{m-1} h_i(n_i) + g(n, m-1) \]

\[ = \frac{v_m}{\mu_m} \sum_{\tilde{n} \in S(n-1, m)} \left( \frac{v_m}{\mu_m} \right)^n m^{-1} \prod_{i=1}^{m-1} h_i(n_i) + g(n, m-1) \]
In the case that all nodes are single server nodes, computational complexity is \(O(MN)\)

\[
= \frac{v_m}{\mu_m} \sum_{\tilde{n} \in S(n-1,m)} H_m(\tilde{n}) + g(n, m - 1)
\]

\[
= \frac{v_m}{\mu_m} g(n - 1, m) + g(n, m - 1)
\]
In the case that all nodes are single server nodes, computational complexity is $O(MN)$
Performance Metrics

\[ P(N_M = n) - \text{general case:} \]

\[ P(n_M = n) = \frac{1}{G(N)h_M(n)g(N-n,M-1)} , \quad n=0,\ldots,N \]

Note: to obtain \( P(N_i = n) \), it is necessary to reorder queues with \( i \)-th queue as last queue and redo convolution algorithm
Single server case:

\[ P(N_M \geq n) = \frac{1}{G(N)} \sum_{\tilde{n} \in S(n, m) \atop n_m \geq n} H_m(\tilde{n}) \]

\[ = \frac{1}{G(N)} \left( \frac{v_m}{\mu_m} \right)^n \sum_{\tilde{n} \in S(n, m) \atop n_m \geq 0} \]

\[ = \frac{G(N - n)}{G(N)} \left( \frac{v_m}{\mu_m} \right)^n \]
Note that this holds for any single server node $i$,

$$P(N_i \geq n) = \frac{G(N - n)}{G(N)} \left(\frac{v_i}{\mu_i}\right)^n$$
Other metrics

General case: We state without proof that the above expression holds for multiple server, infinite server and state dependent server nodes as well.

\[
E[N_i]:
\]
\[
E[N_i] = \sum_{n=1}^{N} nP(N_i = n)
\]

\[
E[T_i]:
\]
\[
E[T_i] = E[N_i]/\gamma_i
\]
Other metrics

Throughput (single server case):

\[
\gamma_i = U_i \mu_i \\
= \frac{v_i}{\mu_i} \frac{G(N - 1)}{G(N)} \mu_i \\
= v_i \frac{G(N - 1)}{G(N)}
\]
Example: Central Server Model

\[ v_1 = 0.1v_1 + v_2 + v_3 \]
\[ v_2 = 0.7v_1 \]
\[ v_3 = 0.2v_1 \]

Set \( v_1 = 1 \Rightarrow v_2 = 0.7, v_3 = 0.2; \frac{v_1}{\mu_1} = 1, \frac{v_2}{\mu_2} = 1, \frac{v_3}{\mu_3} = 2 \)
CPU util.

I/O device util.

System thruput
Avg. job response time ($E[T]$)
Mean Value Analysis (MVA) Algorithm

Key idea: a job that moves from one queue to another, at time of arrival to queue sees system with the same statistics as system with one less customer.

We give algorithm for solving network consisting of single server nodes (FCFS or PS) and infinite server nodes.
MVA

Notation.

- nodes 1 - $M_0$ single server nodes; nodes $M_0 + 1$ thru $M$ are infinite server nodes

Consider system with population of $n$ jobs

- $\overline{N}_i(n)$ - average number of jobs at node $i$
- $\overline{T}_i(n)$ - average response time at node $i$
- $\gamma_i^{(n)}$ - throughput of node $i$
MVA algorithm

0. $\bar{N}_i(0) = 0, \quad 1 \leq i \leq M$ \hspace{1cm} \textit{initialization}

for $n = 1$ to $N$ do

1. $\bar{T}_i(n) = \left[1 + \bar{N}_i(n-1)\right]/\mu_i, \quad 1 \leq i \leq M_0$

   $\bar{T}_i(n) = 1/\mu_i, \quad M_0 + 1 \leq i \leq M$

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

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

   $\bar{N}_i(n) = \gamma_i(n) \bar{T}_i(n), \quad 1 \leq i \leq M$
Example: file server

\[ v_1 = 1, \quad v_2 = 4, \quad v_3 = 3 \]
<table>
<thead>
<tr>
<th>$N$</th>
<th>$\bar{T}_1$</th>
<th>$\bar{N}_1$</th>
<th>$\bar{T}_2$</th>
<th>$\bar{N}_2$</th>
<th>$\bar{T}_3$</th>
<th>$\bar{N}_3$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2sec</td>
<td>.74</td>
<td>120ms.</td>
<td>.17</td>
<td>80ms.</td>
<td>.09</td>
<td>1/2.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.368 job/sec</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2sec</td>
<td>1.42</td>
<td>140ms</td>
<td>.4</td>
<td>87ms</td>
<td>.18</td>
<td>2/2.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.709 j/s</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2sec</td>
<td>2.03</td>
<td>168ms</td>
<td>.68</td>
<td>94ms</td>
<td>.29</td>
<td>3/2.952</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.02 j/s</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2sec</td>
<td>2.57</td>
<td>202ms</td>
<td>1.03</td>
<td>103ms</td>
<td>.4</td>
<td>4/3.117</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.28 j/s</td>
<td></td>
</tr>
</tbody>
</table>
Closed QNs w Multiple Classes of Jobs

- $M$ queues
- $K$ job classes with $N_k$ of class $k$: $\vec{N} = (N_1, \ldots, N_K)$
- $p_{i,j,k}$ - prob. that job enters queue $j$ given that it is of class $k$ and just departed queue $i$, $1 \leq i, j \leq M$, $1 \leq k \leq K$
- $v_{i,k}$ - average no. of visits of class $k$ job to queue $i$ between visits to queue 1; $v_{1,k} = 1$, $1 \leq k \leq K$

$$v_{i,k} = \sum_{j=1}^{M} v_{j,k} p_{i,j,k} \quad i = 2, \ldots, M; \ 1 \leq k \leq K$$

- $X_{i,k}$ - service time of class $k$ job at node $i$
- $N_{i,k}$ - no. of class $k$ jobs at node $i$: $\vec{N}_i = (N_{i,1}, \ldots, N_{i,K})$
Performance metrics

- joint queue length distributions
- avg queue lengths, avg response times
- throughputs, $\gamma_{i,k}$, $\gamma_i$, $i=1,...,M$; $k=1,...,K$
Multiclass QNs

Assumptions on service times at node $i$
- if FCFS, then $X_{i,k}$ must be exponential rv with mean $1/\mu_{i,k} \equiv 1/\mu_i$ independent of job class
- if PS, LCFSPR, or IS, then $X_{i,k}$ may have general distribution with mean $1/\mu_{i,k}$
BCMP Result

Closed QN has product form steady state solution.

\[
P(\vec{N}_1 = \vec{n}_1, \ldots, \vec{N}_K = \vec{n}_K) = \frac{1}{G(\vec{N})} \prod_{i=1}^{M} h_i(\vec{n}_i)
\]

\[
h_i(\vec{n}) = \left( \sum_{k=1}^{K} n_k! \right) \prod_{k=1}^{K} \frac{1}{n_k!} \left( \frac{\nu_{i,k}}{\mu_{i,k}} \right)^{n_k}
\]

FCFS, PS, LCFSPR

\[
h_i(\vec{n}) = \prod_{k=1}^{k} \frac{1}{n_k!} \left( \frac{\nu_{i,k}}{\mu_{i,k}} \right)^{n_k}
\]

IS
Changes to algorithms

- convolution algorithm (single server)

\[ g(\vec{n},m) = \sum_{\vec{l}=0}^{\vec{n}} h_m(\vec{l}) g(\vec{n} - \vec{l},m - 1) \]

- throughput of class k at node i

\[ \gamma_{i,k}(\vec{N}) = v_{i,k} \frac{G(\vec{N} - \vec{e}_k)}{G(\vec{N})} \]

where \( \vec{e}_k \) is vector of 0s and a single one in \( k \)-th position.
MVA, uses following equations

\[
\bar{T}_{i,k}(\vec{n}) = \frac{1}{\mu_{i,k}} \left( 1 + \sum_{s=1}^{K} \bar{N}_{i,s}(\vec{n} - \vec{e}_k) \right)
\]

\[
\gamma_k(\vec{n}) = \frac{n_k}{\sum_{j=1}^{M} v_{j,k} \bar{T}_{j,k}(\vec{n})}
\]

\[
\bar{N}_{i,k}(\vec{n}) = v_{i,k} \gamma_k(\vec{n}) \bar{T}_{i,k}(\vec{n})
\]

- same computational complexity as convolution
- there exists a third algorithm (RECAL) which adds one class at a time and achieves a complexity of \(O(K^{M+1})\) which we will not cover.
Open and Mixed QNs

Open QNs

- $M$ queues
- $K$ job classes
- Routing probabilities $\{p_{i,j,k}\}$ as before and arrival rates $r_{i,k}$ of class $k$ to queue $i$
- $\lambda_{i,k}$ arrival rate at queue $i$ of class $k$ jobs,
  \[ \lambda_{i,k} = r_{i,k} + \sum_{1 \leq j \leq M} \lambda_{j,k} p_{j,i,k} \]
- Service times $X_{i,k}$
under same assumptions as before + Poisson arrivals, we get same product form steady state solution. Differences are:

- use $\lambda_{i,k}$ instead of $v_{i,k}$ in $h_i()$

- Replace $1/G(N)$ with $\prod_{i=1}^{M} \pi_i(0)$ where $\pi_i(0)$ is steady state prob. that queue $i$ is empty when analyzed in isolation with Poisson arrivals with rates $\{\lambda_{i,k}\}$

- Mixed QNs: some classes are open and some closed. Can be solved using combination of the two solution techniques
Limitations

- exponential and homogeneity assumptions for FCFS
- exclusion of policies such as priority queueing
- parallelism not allowed
- simultaneous possession of resources not permitted
- finite buffers with blocking, or loss not permitted
- state dependent routing, load balancing not permitted
- some constraint on size of systems that can be handled
Approximation Methodologies

- aggregation
- surrogate delays
- decomposition
- MVA-based approximations

Common elements to last three: fixed point methods.
Aggregation Approach

Queueing Network A → Queueing Network B

Queueing Network A → \[ \mu(n) \]
Aggregation Theorem for Product Form Networks

- single class product form queueing network with $M'$ queues, $N$ customers, branching probabilities $\{p_{i,j}\}$, visit ratios $\{v_i\}$

- focus on first $M$ queues, $P(N_1 = n_1, \ldots, N_M = n_M)$ - joint queue length distribution ($M < M'$).
Consider queueing network containing $M + 1$ queues

- first $M$ identical to first $M$ in preceding network
- queue $M+1$ - FCFS, exponential service times, queue length dependent service rate $\mu_{M+1}(n)$
Branching probabilities

\[
p'_{i,j} = \begin{cases} 
  p_{i,j}, & i, j \leq M \\
  \sum_{k=M+1}^{M'} p_{i,k}, & i \leq M, j = M + 1 \\
  \frac{v_j - \sum_{k=1}^{M} v_k p_{k,j}}{\sum_{l=M+1}^{M'} v_l \sum_{k=1}^{M} p_{l,k}}, & i = M + 1, j \leq M \\
  0, & \text{otherwise}
\end{cases}
\]
Queue length dependent rate

- $\mu_{M+1}(n)$ - throughput of original QN with first $M$ queues removed, population $n$
queue 0 provides zero delay to all customers, i.e., $\mu_0 = \infty$
Aggregation Theorem
Aggregation theorem

Theorem.

\[ P(N_1 = n_1, \ldots, N_M = n_M) = P(N'_1 = n_1, \ldots, N'_M = n_M) \]

for all \( n_k \geq 0; \sum_{1 \leq k \leq M} n_k = N \)

\[ \square \] Proof. Left as an exercise to the reader.
queue 0 provides zero delay to all customers, i.e., \( \mu_0 = \infty \)
Application of Aggregation

- central server network
- product form - except CPU has two stage hyperexponential state distribution and FCFS
- large state Markov chain
- approximate by 2 queue network
  - CPU unchanged
  - I/Os replaced by ql dependent server queue, \( \mu_c(n) \)
- $\mu_c(n)$ - throughput of QN with CPU removed, population $n$
- two queue network modeled as small state MC
- solve with Tangram
Another application

- single network session
- Poisson arrivals
- end-to-end window flow control
  - fixed window size $W > 0$
  - ACKs signal packet receipt
- model as open tandem QN
  - Poisson arrivals $\lambda$
  - expo sts, FCFS
Model

- violates product form assumptions
- candidate for aggregation approximation
$\mu_c(n)$ - throughput of QN with token queue removed, population $n$

$$
\mu_c(n) = \begin{cases} 
TPUT(n), & n = 1, \ldots, W \\
TPUT(W), & n \geq W + 1 
\end{cases}
$$
Decomposition
Application of decomposition

- tandem system of $M$ finite capacity queues, $i = 1, \ldots, M$
- Poisson arrivals, $\lambda$
- exponential service time, $\mu_i$, $i = 1, \ldots, M$
- buffer size $B_i$, including customer in service, $i = 1, \ldots, M$
approximate as independent $M/M/1/B_i$ queues

$q(\lambda_i, \mu_i, B_i) = P(N_i = B_i)$, overflow prob at queue $i$

$\lambda_i$ - departure rate of queue $i - 1$, $i=2,\ldots,M$

arrival rate $\lambda$, $i = 1$

end-to-end loss probability

$$1 - \prod_{i=1}^{M} (1 - q(\lambda_i, \mu_i, B_i))$$

calculating $\lambda_i$s

$\lambda_1 = \lambda$

$\lambda_{i+1} = \lambda_i (1 - q(\lambda_i, \mu_i, B_i))$
MVA-based approach

- example: non-exponential service times, FCFS, single class QN

- $EX_i, EX^2_i$ first, second moments for station $i$

- basic equation

\[
\bar{T}_i(n) \approx (\bar{N}_i(n-1) + 1 - U_i(n-1))EX_i \\
+ U_i(n-1)EX^2_i/(2EX_i), \quad i = 1, \ldots, M
\]

- $U_i(n)$ - server $i$ utilization when population is $n$

\[
U_i(n) = \gamma_i(n)EX_i, \quad i = 1, \ldots, M
\]

- all other equations unchanged

- many, many applications
Surrogate delays
Finite queueing networks with blocking

- M queues in tandem
- Poisson arrivals, exponential service times
- Queues 2 - M with finite capacity queues, $B_i < \infty$
- Customers block at queue $i-1$ when queue $i$ full
  - Blocking after service (BAS)
  - Blocking before service
**Queues with blocking**

- Introduce surrogate delays to account for blocking

\[
\mu'_i = \frac{1}{\mu_i} + s_i
\]

- \( s_i = P(N_i \geq B_i) = (1 - \rho_i) \rho_i^{B_i}, \quad i = 1, M - 1 \)
- \( s_M = 0 \)
- \( \rho_i = \frac{\lambda}{\mu'_i} \)
- Need to solve a fixed point problem to determine \( \mu'_i \)
Summary

- several approaches to approximation
- hybrids possible
- many, many examples in the literature
- validation against measurement, simulation required