Capturing blocking and spillback in finite capacity queuing networks

Carolina Osorio

Transport and Mobility Laboratory, EPFL

May 2007
Outline

- finite capacity queuing network framework
- model description
- validation
- case study
Overall objectives

Current phase: define aggregate model
**Finite capacity networks**

**Aim:** estimate network performance

How can we model these networks?

**Approach:** queueing theory.
Queueing networks

- Jackson networks
  - infinite buffer size assumption
  - violated in practice

Between-queue correlation structure
- complex to grasp
- helps explain: blocking, spillbacks, deadlocks, chained events

If these events want to be acknowledged:

finite capacity queueing networks
Finite capacity queueing networks FCQN

Main application fields:
- software architectures performance prediction
- telecommunications
- manufacturing systems

More uncommon applications:
- pedestrian flow through circulation systems
- prisoner flow through a network of prisons with varying security levels
- hospital patient flow
Queueing: framework

- $c_i$ parallel servers
- $K_i$ total capacity: nb serveurs + queueing slots
- $\lambda_i$: average arrival rate
- $\mu_i$: average service rate
- $p_{ij}$: transition probabilities (routing)

- station (queue)
- job
FCQN methods

We can evaluate the main network performance measures using the joint stationary distribution, $\pi$.

$$\pi = (P(N_1 = n_1, ..., N_S = n_S), \ (n_1, ..., n_S) \in (S_1, ..., S_S))$$

1. **Closed form expression**
   - product-form dbn: (Jackson, BCMP)
   - small networks: two-station single server with either tandem or closed topology

For more general topology networks:

2. **Exact numerical evaluation**
3. **Approximation methods: decomposition methods**
Exact numerical methods

\[
\begin{aligned}
\pi Q &= 0 \\
\sum_{s \in S} \pi_s &= 1
\end{aligned}
\]

\(\pi\): stationary dbn of the network  
\(Q\): network transition rate matrix  
\(S\): state space

For each network state we define:

- all possible transitions to other states
- their corresponding rates

Disadvantages:

- **untractable**: limited to small networks  
- **not flexible**: changes in the configuration or topology: redefine \(Q\)

A more flexible approach: decomposition methods.
Decomposition methods

By decomposing we can aim at analysing:

- arbitrary topology and size

**Method description**

1. decompose the network into subnetworks
2. analyse each subnetwork independently: estimates of the marginal dbns
3. estimate the main performance measures

**Subnetwork**

- size: single queues
- analysis using global balance equations.
- obtain estimates of the marginal dbns
Current objective

Existing methods mainly concern

- single server + feed-forward network
- multiple server + tandem

For multiple server + arbitrary topology:

- revise queue capacities (endogenous)
- vary network topologies (analogy with closed form dbn networks)

Requires:

- approximations to ensure integrality of endogenous capacities
- aposteriori validation (e.g. check positivity)

unsuitable for an optimization framework
Current objective

- multiple server + arbitrary topology + BAS
- preserving initial network configuration (topology + capacities)
- explicitly model blocking events
Global balance equations

\[
\left\{ \begin{array}{l}
\pi(i)Q(i) = 0 \\
\sum_{s \in S(i)} \pi(i)_s = 1
\end{array} \right.
\]

\(\pi(i)\): stationary dbn of station \(i\)
\(Q(i)\): transition rate matrix
\(S(i)\): state space
State space

Upon arrival to a queue a job:
1. [queue]
2. is served
3. [blocked]
4. departs

State space of station $i$:

$$S_i = \{(A_i, B_i, W_i) \in \mathbb{N}^3, A_i + B_i \leq c_i, W_i \leq K_i - c_i\}$$

We want to estimate:

$$\pi(i) = \left(P((A_i, B_i, W_i) = (a, b, w)) \forall (a, b, w) \in S(i)\right)$$
Transition rates

$Q(i) : \begin{cases} \text{effective arrival rates} \\
\text{effective service rates}
\end{cases} \quad \text{stationary dbn of the subnetwork} \quad \updownarrow \quad \text{marginal stationary dbn of the network}

For a given station how can we estimate the

- effective arrival rates ?
- effective service rates ?

Main challenge and complexity lies in appropriately acknowledging the correlation between the stations i.e. in appropriately revising these structural parameters.
Transition rates

Upon arrival to a queue a job:
1. [queue]
2. is served
3. [blocked]
4. departs

Grasping the between station correlation implies appropriately estimating the transition rates between these states.
Transition rates

\(Q(i)\) is a function of:

- \(\lambda_i, \mu_i\): average arrival and service rate
- \(P^f_i\): average blocking probability
- \(\tilde{\mu}(i, b)\): average unblocking rate given that there are \(b\) blocked jobs

Consider station \(i\) which is in state \((A_i, B_i, W_i) = (a, b, w)\).

Then the possible transitions and their rates are:

<table>
<thead>
<tr>
<th>new state (l)</th>
<th>rate (q_{kl})</th>
<th>condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>((a, b, w + 1))</td>
<td>(\lambda_i)</td>
<td>(a + b \leq c_i &amp; w + 1 \leq K_i - c_i)</td>
</tr>
<tr>
<td>((a + 1, b, w))</td>
<td>(\lambda_i)</td>
<td>(a + b + 1 \leq c_i)</td>
</tr>
<tr>
<td>((a - 1, b, w))</td>
<td>(a\mu_i(1 - P^f_i))</td>
<td>(w = 0)</td>
</tr>
<tr>
<td>((a, b, w - 1))</td>
<td>(a\mu_i(1 - P^f_i))</td>
<td>(w \geq 1)</td>
</tr>
<tr>
<td>((a - 1, b + 1, w))</td>
<td>(a\mu_i P^f_i)</td>
<td>always possible</td>
</tr>
<tr>
<td>((a, b - 1, w))</td>
<td>(\tilde{\mu}(i, b))</td>
<td>(w = 0)</td>
</tr>
<tr>
<td>((a + 1, b - 1, w - 1))</td>
<td>(\tilde{\mu}(i, b))</td>
<td>(w \geq 1)</td>
</tr>
</tbody>
</table>

Let’s estimate these parameters ...
Average blocking probability

\[ P^f_i = \sum_j p_{ij} P(N_j = K_j) \]

where \( P(N_j = K_j) \) is the probability that station \( j \) is full.
Arrival rates

- $\lambda_i$: total arrival rate (includes potentially lost arrivals)
- $\lambda_i^{\text{eff}}$: the effective arrival rate (excludes lost arrivals)
- $\gamma_i$: external arrival rate

1) Loss model:

$$\lambda_i^{\text{eff}} = \lambda_i (1 - P(N_i = K_i))$$

where $N_i$ denotes the total number of jobs at station $i$ ($N_i = A_i + B_i + W_i$).

2) Flow conservation laws hold for the effective arrival rates:

$$\lambda_i^{\text{eff}} = \gamma_i (1 - P(N_i = K_i)) + \sum_j p_{ij} \lambda_j^{\text{eff}}$$

Inter-arrival times $\sim \epsilon(\lambda_i)$, i.i.d
Service and unblocking rates

When station $i$ is in state $(a, b, w)$:

1) service rate:
\[ a \text{ parallel servers } \Rightarrow \text{service rate: } a\mu_i. \]

2) unblocking rate:
if there are $b$ blocked jobs at station $i$:

\[
\text{how many parallel blocked queues are there?}
\]

\[
\textbf{aim: } a\mu_i \leftrightarrow \tilde{\mu}(i, b) = \phi(i, b) \tilde{\mu}_i^o
\]
Service and unblocking rates

**aim:** \( a \mu_i \leftrightarrow \tilde{\mu}(i, b) = \phi(i, b) \tilde{\mu}_i^o \)

- one station blocking : \( \tilde{\mu}_i^o \)
- \( d \) distinct destination stations : \( d\tilde{\mu}_i^o \)
  \( d \) virtual parallel queues

\( \phi(i, b) \) represents: the average number of blocking stations given that there are \( b \) blocked jobs at station \( i \)
Service and unblocking rates

- $\tilde{\mu}_i^{\circ}$ approach: average “inter-unblocking times” across destination stations

$$\frac{1}{\tilde{\mu}_i^{\circ}} = \sum_{j \in I^+} \frac{\lambda_{j}^{\text{eff}}}{\lambda_{i}^{\text{eff}} \hat{\mu}_j c_j}$$

- $\phi(i, b)$ approach: condition on the number of distinct stations that are blocking the $b$ jobs.

$$\frac{1}{\tilde{\mu}(i,b)} = \min(b, \text{card}(I^+)) \sum_{d=1} \frac{1}{d \tilde{\mu}_i^{\circ}} = \frac{1}{ \tilde{\mu}_i^{\circ}} \sum_{d=1} \frac{1}{d} \sum_{l_i \in L} b_l^{l_i j} \prod_{j \in I^+} \tilde{p}_{ij}$$

adding an assumption ...

$$\tilde{\mu}(i, b) = \tilde{\mu}_i^{\circ} \phi(i, b)$$

where $\phi(i, b)$ is now exogenous

- Service time $\sim \varepsilon(\mu_i)$, i.i.d
- Time between unblockings $\sim \varepsilon(\tilde{\mu}_i^{\circ})$, i.i.d
Aims were:

- decompose the network into single stations
- solve the global balance equations associated to each station:

\[
\begin{align*}
\pi(i)Q(i) &= 0 \\
\sum_{s \in S(i)} \pi(i)_s &= 1
\end{align*}
\]

- define \( S(i) \)
- estimate \( Q(i) = f(\lambda_i, \mu_i, P^f_i, \tilde{\mu}(i, b)) \)
- estimate the transition rates
Summary

\[ E(i) = \begin{cases} 
\pi(i)Q(i) = 0 \\
\sum_{s \in S(i)} \pi(i)_s = 1 \\
Q(i) = f(\lambda_i, \mu_i, P^f_i, \tilde{\mu}(i, b)) \\
\lambda^\text{eff}_i = \lambda_i(1 - P(N_i = K_i)) \\
\gamma_i(1 - P(N_i = K_i)) + \sum_j p_{ji}\lambda^\text{eff}_j \\
P^f_i = \sum_j p_{ij} P(N_j = K_j) \\
\tilde{\mu}(i, b) = \tilde{\mu}^\text{av}_i \phi(i, b) \\
\frac{1}{\mu^\text{eff}_i} = \sum_{j \in I^+} \frac{\lambda^\text{eff}_j \mu_j c_j}{\mu^\text{eff}_i} \\
\frac{1}{\mu_i} = \frac{1}{\mu^\text{eff}_i} + \frac{P^f_i}{\mu^\text{av}_i} \\
\frac{1}{\mu^\text{avg}_i} = \sum_{b \geq 1} \frac{P(B_i = b)}{P(B_i > 0)} \sum_{k=1}^b \frac{1}{b} \frac{1}{\mu(i, k)} \\
P(N_i = K_i) = \sum_{s \in X^i} \pi(i)_s \\
P(B_i = b) = \sum_{s=(.,b,.) \in S(i)} \pi(i)_s \\
P(B_i > 0) = 1 - \sum_{s=(.,0,.) \in S(i)} \pi(i)_s 
\end{cases} \]

- **Exogenous**: \{\mu_i, \gamma_i, p_{ij}, c_i, K_i, \phi(i, b)\}
- All other parameters are endogenous

**MATLAB fsolve**: route for systems of nonlinear equations.
Method validation

Validation versus:

- pre-existing decomposition methods
- simulation results on a set of small networks
- simulation results on a network of hospital rooms
Validation

Validation versus pre-existing methods


- Hillier and Boling. 1967. Finite queues in series with exponential or erlang service times. A numerical approach. *Operations research*
Validation [1]

Setting: triangular topology with single-server stations ($c_j = 1$)

\[
\forall i, c_i = 1, p_{12} = 1, \gamma_1 = 1, \gamma_2 = \gamma_3 = 0
\]

<table>
<thead>
<tr>
<th>scenario</th>
<th>$p_{12}$</th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$\gamma_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>1.4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1.3</td>
<td>1.6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1.4</td>
<td>1.8</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1.6</td>
<td>2.2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1.7</td>
<td>2.4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1.8</td>
<td>2.6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1.9</td>
<td>2.8</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Capturing blocking and spillback in finite capacity queuing networks – p.27/38
Validation [2]

Theoretical bound on the throughput Bell (1982):

\[
\mu_1 = 3, \mu_2 = 1, \ c_1 = c_2 = 1 \\
\gamma_1 = 1, \gamma_2 = 0
\]

<table>
<thead>
<tr>
<th>scenario</th>
<th>( K_1 - c_1 )</th>
<th>( K_2 - c_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

\( \gamma_1 \) would be represented as a diagram with nodes 1 and 2 connected by arrows indicating the flow from 1 to 2.
Validation vs. simulation results

\[ \forall i \quad c_i = K_i = 3, \quad \text{card}(S_i) = 10 \]

station index \( i \): 1 2 3 4 5 6 7 8 9
\[
\begin{array}{cccccccccc}
\gamma_i & - & 0.2 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
\mu_i & 0.3 & 0.3 & 0.3 & 0.1 & 0.01 & 0.014 & 0.1 & 0.4 & 0.5
\end{array}
\]

scenario \( \gamma \)
\[
\begin{array}{cccccc}
1 & 0.1 \\
2 & 0.2 \\
3 & 0.3 \\
4 & 0.4
\end{array}
\]

\[ \forall i \quad c_i = K_i = 3, \quad \text{card}(S_i) = 10 \]

\[ \forall i \quad c_i = K_i = 3, \quad \text{card}(S_i) = 10 \]

\[ \forall i \quad c_i = K_i = 3, \quad \text{card}(S_i) = 10 \]
Validation [3]
Validation [3]

Network C: $\pi(5)$

Capturing blocking and spillback in finite capacity queuing networks – p.31/38
Case study

**Hospital bed blocking**: recent demand for modeling and acknowledging this phenomenon:

- patient care and budgetary improvements (Cochran (2006), Koizumi (2005))
- flexibility responsiveness of the emergency and surgical admissions procedure (Mackay (2001)).

The existing analytic hospital network models are limited to:

- feed-forward topologies
- at most 3 units
HUG application

- **Network of interest**: network of operative and post-operative rooms in the HUG, Geneva University Hospital.

- **Dataset**
  - records of arrivals and transfers between hospital units
  - 25336 patient records
  - redunduncies in the dataset eliminated
  - used to estimate $\gamma, \mu, p_{ij}$

**Network model:**

<table>
<thead>
<tr>
<th>Unit</th>
<th>BO U</th>
<th>BO OPERA</th>
<th>BO ORL</th>
<th>IF CHIR</th>
<th>IF MED</th>
<th>IM MED</th>
<th>IM NEURO</th>
<th>REV OPERA</th>
<th>REV ORL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_i$</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>18</td>
<td>18</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

- beds $\leftrightarrow$ servers
- no waiting space $\leftrightarrow$ bufferless ($K_i = c_i$)
HUG application

\( \gamma \): avg external arrival rates

- observations: Oct 2nd 2004 - Oct 2nd 2005
- estimator: MLE (avg nb of occurrences)

\( \mu \): avg service rate

- estimator: MLE \( \left( \frac{1}{\text{LOS}} \right) \)
- Assumption: departure time includes no blocking

\( p_{ij} \): transition probabilities:

- frequency of each transition
## HUG application

<table>
<thead>
<tr>
<th></th>
<th>BOU</th>
<th>BO OPERA</th>
<th>BO ORL</th>
<th>IF CHIR</th>
<th>IF MED</th>
<th>IM MED</th>
<th>IM NEURO</th>
<th>REV OPERA</th>
<th>REV ORL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_i$</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>18</td>
<td>18</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>$\gamma_i$</td>
<td>0.392</td>
<td>0.502</td>
<td>0.246</td>
<td>0.059</td>
<td>0.176</td>
<td>0.025</td>
<td>0.013</td>
<td>0.155</td>
<td>0</td>
</tr>
<tr>
<td>$\mu_i$</td>
<td>0.317</td>
<td>0.255</td>
<td>0.335</td>
<td>0.013</td>
<td>0.015</td>
<td>0.014</td>
<td>0.015</td>
<td>0.22</td>
<td>0.518</td>
</tr>
</tbody>
</table>

\[
(p_{ij}) = \begin{pmatrix}
0 & 0 & 0 & 0.16 & 0.02 & 0 & 0 & 0.71 & 0 \\
0 & 0 & 0 & 0.07 & 0 & 0 & 0 & 0.84 & 0 \\
0 & 0 & 0 & 0.03 & 0.01 & 0 & 0 & 0 & 0.95 \\
0.18 & 0.01 & 0.03 & 0 & 0.03 & 0.01 & 0.11 & 0.03 & 0 \\
0.05 & 0.01 & 0.01 & 0.01 & 0 & 0.07 & 0 & 0 & 0 \\
0.02 & 0 & 0 & 0.01 & 0.1 & 0 & 0 & 0 & 0 \\
0.05 & 0 & 0.05 & 0.04 & 0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\
0 & 0 & 0 & 0.05 & 0 & 0 & 0.05 & 0.02 & 0
\end{pmatrix}
\]

- Number of unknowns/equations: 635
HUG application

validation of the results

Capturing blocking and spillback in finite capacity queuing networks – p.36/38
### Estimation results

<table>
<thead>
<tr>
<th>$c_i$</th>
<th>BO U</th>
<th>BO OPERA</th>
<th>BO ORL</th>
<th>IF CHIR</th>
<th>IF MED</th>
<th>IM MED</th>
<th>IM NEURO</th>
<th>REV OPERA</th>
<th>REV ORL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>18</td>
<td>18</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>$P(N_i = K_i)$</td>
<td>0.042</td>
<td>0.001</td>
<td>0.001</td>
<td>0.102</td>
<td>0.046</td>
<td>0.226</td>
<td>0.471</td>
<td>0.006</td>
<td>0.000</td>
</tr>
<tr>
<td>$P(N_i = 0)$</td>
<td>0.244</td>
<td>0.136</td>
<td>0.464</td>
<td>0.000</td>
<td>0.000</td>
<td>0.053</td>
<td>0.009</td>
<td>0.017</td>
<td>0.591</td>
</tr>
<tr>
<td>$p_i^f$</td>
<td>0.021</td>
<td>0.012</td>
<td>0.004</td>
<td>0.063</td>
<td>0.020</td>
<td>0.006</td>
<td>0.006</td>
<td>0.005</td>
<td>0.029</td>
</tr>
<tr>
<td>$\frac{1}{\mu_i}$ LOS</td>
<td>3.2510</td>
<td>3.9499</td>
<td>3.0067</td>
<td>78.0939</td>
<td>66.8836</td>
<td>71.7699</td>
<td>66.8884</td>
<td>4.5497</td>
<td>2.1668</td>
</tr>
<tr>
<td>$P(B_i &gt; 0)$</td>
<td>0.0399</td>
<td>0.0142</td>
<td>0.0055</td>
<td>0.1918</td>
<td>0.0400</td>
<td>0.0117</td>
<td>0.0105</td>
<td>0.0038</td>
<td>0.0559</td>
</tr>
</tbody>
</table>
Conclusions and current aims

Conclusions:

- a decomposition method allowing the analysis of FCQN
- explicitly models the blocking phase
- preserves network topology and configuration
- validation versus both pre-existing methods and simulation estimates shows encouraging results
- application on a real case study

Aims:

- come back to general framework: integrate with DES.