

Finite buffer fluid networks with overflows

Citation for published version (APA):

Fleuren, S. T. G., Lefeber, A. A. J., & Nazarathy, J. (2012). Finite buffer fluid networks with overflows. In *Proceedings of the 6th International Workshop on Applied Probability (IWAP2012), 11-14 June 2012, Jerusalem, Israel* (pp. 1-41).

Document status and date:

Published: 01/01/2012

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Netherlands Organisation for Scientific Research

Finite buffer fluid networks with overflows

Stijn Fleuren, Erjen Lefeber, Yoni Nazarathy

IWAP 2012, Jerusalem



TU / **e**

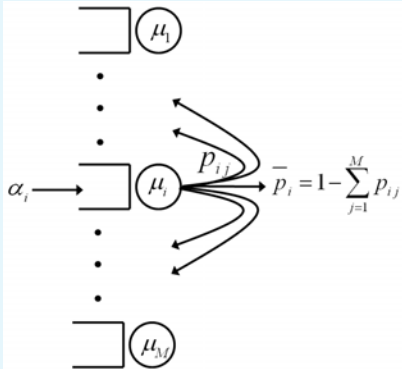
Technische Universiteit
Eindhoven
University of Technology

June 11, 2012

Where innovation starts

Jackson (1957); Goodman, Massey (1984); Chen, Mandelbaum (1991)

Jackson 1957



Problem data

α, μ, P

Traffic equations (stable case)

$$\lambda_j = \alpha_j + \sum_{j=1}^N \lambda_j p_{ji}$$

$$\lambda = \alpha + P' \lambda$$

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

Theorem (Jackson 1957)

Given an $(M/M/1)^N$ system where every node can be filled and drained, let $\lambda = [\lambda_1, \dots, \lambda_N]'$ be the solution of the throughput equation

$$\lambda = \alpha + P'\lambda$$

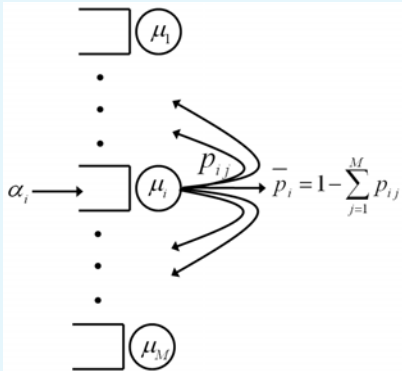
If $\rho_i = \lambda_i/\mu_i$ and $\rho_i < 1$ for all i , then

$$\lim_{t \rightarrow \infty} P(X_1(t) = n_1, \dots, X_N(t) = n_N) = \prod_{i=1}^N (1 - \rho_i) \rho_i^{n_i}$$

for all integers $n_i \geq 0$.

Jackson (1957); Goodman, Massey (1984); Chen, Mandelbaum (1991)

Jackson 1957



Problem data

α, μ, P

Traffic equations (general)

$$\lambda_j = \alpha_j + \sum_{j=1}^N \min(\lambda_j, \mu_j) p_{ji}$$

$$\lambda = \alpha + P' \min(\lambda, \mu)$$

Theorem (Goodman, Massey 1984)

Given an $(M/M/1)^N$ system where every node can be filled and drained, let $\lambda = [\lambda_1, \dots, \lambda_N]'$ be the solution of the throughput equation

$$\lambda = \alpha + P' \min(\lambda, \mu)$$

If $\rho_i = \lambda_i / \mu_i$ and $U = \{i \mid \rho_i < 1\}$, then

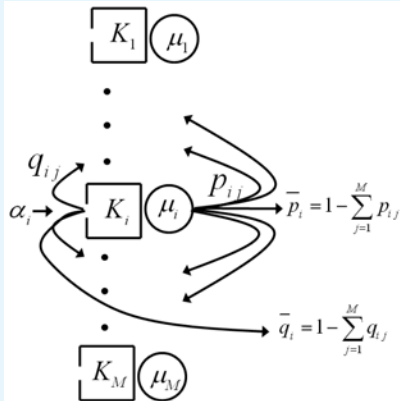
$$\lim_{t \rightarrow \infty} P(X_i(t) = n_i; i \in U) = \prod_{i \in U} (1 - \rho_i) \rho_i^{n_i}$$

for all integers $n_i \geq 0$ with $i \in U$. Moreover, if $j \notin U$ then

$$\lim_{t \rightarrow \infty} P(X_j(t) = n) = 0$$

for all integers $n \geq 0$

Network



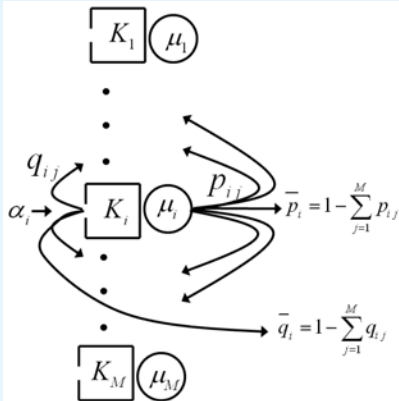
Problem data

α, μ, P, Q, K

Our contribution (in progress)

- ▶ Limiting traffic equations
- ▶ Efficient algorithm for unique solution
- ▶ Limiting deterministic trajectories
- ▶ Limiting sojourn time distribution

Network



Problem data

α, μ, P, Q, K

Our contribution (in progress)

- ▶ Limiting traffic equations
- ▶ Efficient algorithm for unique solution
- ▶ Limiting deterministic trajectories
- ▶ Limiting sojourn time distribution

Scaling yields a fluid system

7/19

A sequence of systems: $N = 1, 2, \dots$

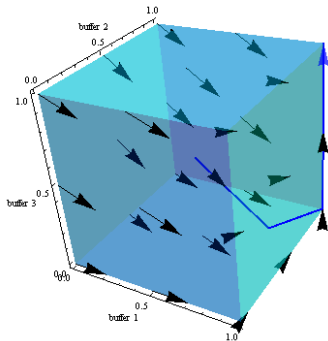
$$\alpha^N = N\alpha$$

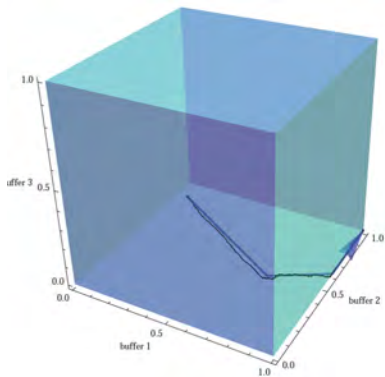
$$\mu^N = N\mu$$

$$K^N = NK$$

Make the jobs **fast** and the buffers **big** by taking $N \rightarrow \infty$.

The proposed limiting model is a deterministic fluid system





$$\lim_{N \rightarrow \infty} \sup_t \left\{ \left| \frac{X^N(t)}{N} - x(t) \right| \right\} = 0$$

outflow rate: $\min(\lambda, \mu)$

overflow rate: $\lambda - \min(\lambda, \mu) = \max(0, \lambda - \mu)$

Traffic equations

$$\lambda_i = \alpha_i + \sum_{j=1}^N \min(\lambda_j, \mu_j) p_{ji} + \sum_{j=1}^N \max(0, \lambda_j - \mu_j) q_{ji}$$

or

$$\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$$

Question

How to (efficiently) solve traffic equations for given α, μ, P, Q, K ?

outflow rate: $\min(\lambda, \mu)$

overflow rate: $\lambda - \min(\lambda, \mu) = \max(0, \lambda - \mu)$

Traffic equations

$$\lambda_i = \alpha_i + \sum_{j=1}^N \min(\lambda_j, \mu_j) p_{ji} + \sum_{j=1}^N \max(0, \lambda_j - \mu_j) q_{ji}$$

or

$$\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$$

Question

How to (efficiently) solve traffic equations for given α, μ, P, Q, K ?

outflow rate: $\min(\lambda, \mu)$

overflow rate: $\lambda - \min(\lambda, \mu) = \max(0, \lambda - \mu)$

Traffic equations

$$\lambda_i = \alpha_i + \sum_{j=1}^N \min(\lambda_j, \mu_j) p_{ji} + \sum_{j=1}^N \max(0, \lambda_j - \mu_j) q_{ji}$$

or

$$\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$$

Question

How to (efficiently) solve traffic equations for given α, μ, P, Q, K ?

$$\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$$

Let $w = \lambda - \min(\lambda, \mu)$ and $z = \mu - \min(\lambda, \mu)$. Then $\lambda = w - z + \mu$ and

$$w \geq 0$$

$$z \geq 0$$

$$w'z = 0$$

Furthermore we obtain for the traffic equation

$$w - z + \mu = \alpha + P'(\mu - z) + Q'w$$

$$(I - Q')w = \alpha - (I - P')\mu + (I - P')z$$

$$w = \underbrace{(I - Q')^{-1}(\alpha - (I - P')\mu)}_q + \underbrace{(I - Q')^{-1}(I - P')}_M z$$

$$\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$$

Let $w = \lambda - \min(\lambda, \mu)$ and $z = \mu - \min(\lambda, \mu)$. Then $\lambda = w - z + \mu$ and

$$w \geq 0$$

$$z \geq 0$$

$$w'z = 0$$

Furthermore we obtain for the traffic equation

$$w - z + \mu = \alpha + P'(\mu - z) + Q'w$$

$$(I - Q')w = \alpha - (I - P')\mu + (I - P')z$$

$$w = \underbrace{(I - Q')^{-1}(\alpha - (I - P')\mu)}_q + \underbrace{(I - Q')^{-1}(I - P')}_M z$$

LCP

LCP(q, M): Find z, w such that

$$w - Mz = q \qquad w, z \geq 0 \qquad w'z = 0$$

For our system: $q = (I - Q')^{-1}(\alpha - (I - P')\mu)$, $M = (I - Q')^{-1}(I - P')$

Theorem

LCP(q, M) has **unique solution for all q** iff M is a **P -matrix**, i.e. determinants of all $2^N - 1$ principal submatrices are positive

Observation

No polynomial time algorithm (yet) exists for solving the P -matrix LCP

LCP

LCP(q, M): Find z, w such that

$$w - Mz = q \qquad w, z \geq 0 \qquad w'z = 0$$

For our system: $q = (I - Q')^{-1}(\alpha - (I - P')\mu)$, $M = (I - Q')^{-1}(I - P')$

Theorem

LCP(q, M) has **unique solution for all q** iff **M is a P -matrix**, i.e. determinants of all $2^N - 1$ principal submatrices are positive

Observation

No polynomial time algorithm (yet) exists for solving the P -matrix LCP

Algorithm of Goodman and Massey (1984)

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu)$

Observation: If we would know the stable and unstable nodes, we can solve for λ .

Step 1: Assume all queues are unstable, i.e. output rate μ_i , and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an over-estimate.

Let $I(1) = \{i \mid \lambda_i(1) < \mu_i\}$ denote the set of stable nodes.

Step 2: Assume nodes $i \notin I(1)$ are unstable and solve for the arrival rate: $\lambda(2) \leq \lambda(1)$.

Repeat until: $I(n) = I(n + 1)$.

Algorithm of Goodman and Massey (1984)

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu)$

Observation: If we would know the stable and unstable nodes, we can solve for λ .

Step 1: Assume all queues are unstable, i.e. output rate μ_i , and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an over-estimate.

Let $I(1) = \{i \mid \lambda_i(1) < \mu_i\}$ denote the set of stable nodes.

Step 2: Assume nodes $i \notin I(1)$ are unstable and solve for the arrival rate: $\lambda(2) \leq \lambda(1)$.

Repeat until: $I(n) = I(n + 1)$.

Algorithm of Goodman and Massey (1984)

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu)$

Observation: If we would know the stable and unstable nodes, we can solve for λ .

Step 1: Assume all queues are unstable, i.e. output rate μ_i , and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an over-estimate.

Let $I(1) = \{i \mid \lambda_i(1) < \mu_i\}$ denote the set of stable nodes.

Step 2: Assume nodes $i \notin I(1)$ are unstable and solve for the arrival rate: $\lambda(2) \leq \lambda(1)$.

Repeat until: $I(n) = I(n + 1)$.

Algorithm of Goodman and Massey (1984)

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu)$

Observation: If we would know the stable and unstable nodes, we can solve for λ .

Step 1: Assume all queues are unstable, i.e. output rate μ_i , and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an over-estimate.

Let $I(1) = \{i \mid \lambda_i(1) < \mu_i\}$ denote the set of stable nodes.

Step 2: Assume nodes $i \notin I(1)$ are unstable and solve for the arrival rate: $\lambda(2) \leq \lambda(1)$.

Repeat until: $I(n) = I(n + 1)$.

Algorithm of Goodman and Massey (1984)

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu)$

Observation: If we would know the stable and unstable nodes, we can solve for λ .

Step 1: Assume all queues are unstable, i.e. output rate μ_i , and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an over-estimate.

Let $I(1) = \{i \mid \lambda_i(1) < \mu_i\}$ denote the set of stable nodes.

Step 2: Assume nodes $i \notin I(1)$ are unstable and solve for the arrival rate: $\lambda(2) \leq \lambda(1)$.

Repeat until: $I(n) = I(n + 1)$.

Algorithm of Goodman and Massey (1984)

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu)$

Observation: If we would know the stable and unstable nodes, we can solve for λ .

Step 1: Assume all queues are unstable, i.e. output rate μ_i , and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an over-estimate.

Let $I(1) = \{i \mid \lambda_i(1) < \mu_i\}$ denote the set of stable nodes.

Step 2: Assume nodes $i \notin I(1)$ are unstable and solve for the arrival rate: $\lambda(2) \leq \lambda(1)$.

Repeat until: $I(n) = I(n + 1)$.

Algorithm

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$

Step 1: Assume all nodes have infinite capacity, i.e. $K_i = \infty$ and no overflow, and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an under-estimate.

Let $J(1) = \{j \mid \lambda_j(1) > \mu_j\}$ denote the overflowing nodes.

Step 2: Assume nodes $j \notin J(1)$ have infinite capacity and solve for the arrival rate: $\lambda(2) \geq \lambda(1)$.

Repeat until: $J(n) = J(n + 1)$.

Number of iterations

Worst case: $O(N^2)$.

Practice (max 800 nodes): $O(\log(N))$

Algorithm

Worst case: $O(N^5)$.

Practice: $O(N^3 \log(N))$

Algorithm

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$

Step 1: Assume all nodes have infinite capacity, i.e. $K_i = \infty$ and no overflow, and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an under-estimate.

Let $J(1) = \{j \mid \lambda_j(1) > \mu_j\}$ denote the overflowing nodes.

Step 2: Assume nodes $j \notin J(1)$ have infinite capacity and solve for the arrival rate: $\lambda(2) \geq \lambda(1)$.

Repeat until: $J(n) = J(n + 1)$.

Number of iterations

Worst case: $O(N^2)$.

Practice (max 800 nodes): $O(\log(N))$

Algorithm

Worst case: $O(N^5)$.

Practice: $O(N^3 \log(N))$

Algorithm

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$

Step 1: Assume all nodes have infinite capacity, i.e. $K_i = \infty$ and no overflow, and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an under-estimate.

Let $J(1) = \{j \mid \lambda_j(1) > \mu_j\}$ denote the overflowing nodes.

Step 2: Assume nodes $j \notin J(1)$ have infinite capacity and solve for the arrival rate: $\lambda(2) \geq \lambda(1)$.

Repeat until: $J(n) = J(n + 1)$.

Number of iterations

Worst case: $O(N^2)$.

Practice (max 800 nodes): $O(\log(N))$

Algorithm

Worst case: $O(N^5)$.

Practice: $O(N^3 \log(N))$

Algorithm

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$

Step 1: Assume all nodes have infinite capacity, i.e. $K_i = \infty$ and no overflow, and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an under-estimate.

Let $J(1) = \{j \mid \lambda_j(1) > \mu_j\}$ denote the overflowing nodes.

Step 2: Assume nodes $j \notin J(1)$ have infinite capacity and solve for the arrival rate: $\lambda(2) \geq \lambda(1)$.

Repeat until: $J(n) = J(n + 1)$.

Number of iterations

Worst case: $O(N^2)$.

Practice (max 800 nodes): $O(\log(N))$

Algorithm

Worst case: $O(N^5)$.

Practice: $O(N^3 \log(N))$

Algorithm

Problem: Solve $\lambda = \alpha + P' \min(\lambda, \mu) + Q' \max(0, \lambda - \mu)$

Step 1: Assume all nodes have infinite capacity, i.e. $K_i = \infty$ and no overflow, and solve for arrival rate: $\lambda(1)$.

Observation: $\lambda(1)$ is at worst an under-estimate.

Let $J(1) = \{j \mid \lambda_j(1) > \mu_j\}$ denote the overflowing nodes.

Step 2: Assume nodes $j \notin J(1)$ have infinite capacity and solve for the arrival rate: $\lambda(2) \geq \lambda(1)$.

Repeat until: $J(n) = J(n + 1)$.

Number of iterations

Worst case: $O(N^2)$.

Practice (max 800 nodes): $O(\log(N))$

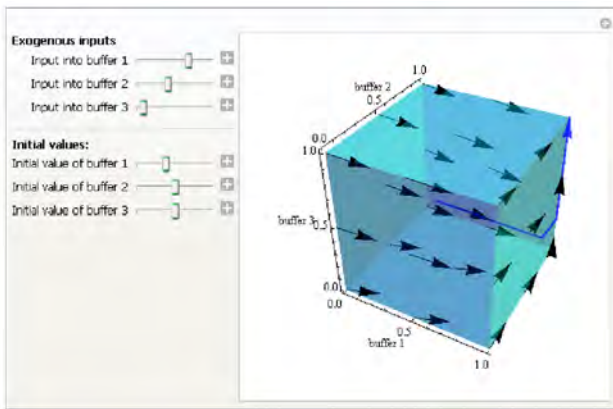
Algorithm

Worst case: $O(N^5)$.

Practice: $O(N^3 \log(N))$

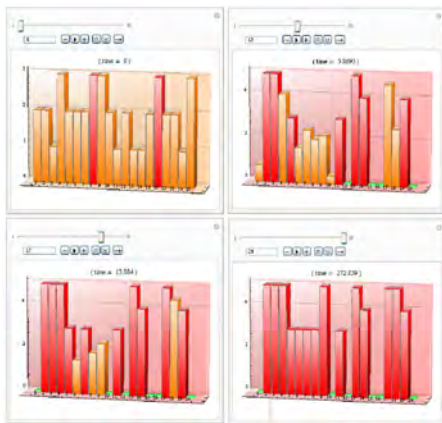
Algorithm can also be used for determining transient behavior

See also <http://demonstrations.wolfram.com/DynamicsOfADeterministicOverflowFluidNetwork/>



Algorithm can also be used for determining transient behavior

See also <http://demonstrations.wolfram.com/DynamicsOfADeterministicOverflowFluidNetwork/>



Definition

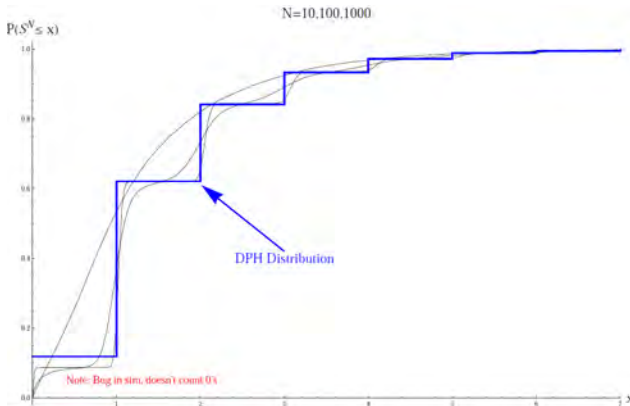
Sojourn time: time in system of customer arriving to steady state FCFS system

Definition

S^N : sojourn time of customer in N th scaled system

Problem

We want to find the limiting distribution of S^N , i.e. $P(S^N \leq x)$, for $N \rightarrow \infty$.



Observation

Sojourn times scale to a discrete distribution

$$F = \{1, \dots, s\}$$

$$\lambda_i > \mu_i \text{ for } i \in F$$

$$\bar{F} = \{s + 1, \dots, N\}$$

$$\lambda_i < \mu_i \text{ for } i \in \bar{F}$$

Observation

Time through $i \in F \approx NK_i/(N\mu_i) = K_i/\mu_i$, time through $i \notin F \approx 1/(N\mu_i) \approx 0$.

For job at entrance of buffer $i \in F$:

- ▶ enters buffer i w.p. $\approx \mu_i/\lambda_i$
- ▶ routed to entrance of buffer j w.p. $\approx (1 - \mu_i/\lambda_i)q_{ij}$
- ▶ leaves system w.p. $\approx (1 - \mu_i/\lambda_i)\bar{q}_i$

Job at entrance of buffer $i \in \bar{F}$:

- ▶ routed according to P almost immediately

$$F = \{1, \dots, s\}$$

$$\lambda_i > \mu_i \text{ for } i \in F$$

$$\bar{F} = \{s + 1, \dots, N\}$$

$$\lambda_i < \mu_i \text{ for } i \in \bar{F}$$

Observation

Time through $i \in F \approx NK_i/(N\mu_i) = K_i/\mu_i$, time through $i \notin F \approx 1/(N\mu_i) \approx 0$.

For job at entrance of buffer $i \in F$:

- ▶ enters buffer i w.p. $\approx \mu_i/\lambda_i$
- ▶ routed to entrance of buffer j w.p. $\approx (1 - \mu_i/\lambda_i)q_{ij}$
- ▶ leaves system w.p. $\approx (1 - \mu_i/\lambda_i)\bar{q}_i$

Job at entrance of buffer $i \in \bar{F}$:

- ▶ routed according to P almost immediately

$$F = \{1, \dots, s\}$$

$$\lambda_i > \mu_i \text{ for } i \in F$$

$$\bar{F} = \{s + 1, \dots, N\}$$

$$\lambda_i < \mu_i \text{ for } i \in \bar{F}$$

Observation

Time through $i \in F \approx NK_i/(N\mu_i) = K_i/\mu_i$, time through $i \notin F \approx 1/(N\mu_i) \approx 0$.

For job at entrance of buffer $i \in F$:

- ▶ enters buffer i w.p. $\approx \mu_i/\lambda_i$
- ▶ routed to entrance of buffer j w.p. $\approx (1 - \mu_i/\lambda_i)q_{ij}$
- ▶ leaves system w.p. $\approx (1 - \mu_i/\lambda_i)\bar{q}_i$

Job at entrance of buffer $i \in \bar{F}$:

- ▶ routed according to P almost immediately

$$F = \{1, \dots, s\}$$

$$\lambda_i > \mu_i \text{ for } i \in F$$

$$\bar{F} = \{s + 1, \dots, N\}$$

$$\lambda_i < \mu_i \text{ for } i \in \bar{F}$$

Observation

Time through $i \in F \approx NK_i/(N\mu_i) = K_i/\mu_i$, time through $i \notin F \approx 1/(N\mu_i) \approx 0$.

For job at entrance of buffer $i \in F$:

- ▶ enters buffer i w.p. $\approx \mu_i/\lambda_i$
- ▶ routed to entrance of buffer j w.p. $\approx (1 - \mu_i/\lambda_i)q_{ij}$
- ▶ leaves system w.p. $\approx (1 - \mu_i/\lambda_i)\bar{q}_i$

Job at entrance of buffer $i \in \bar{F}$:

- ▶ routed according to P almost immediately

$$F = \{1, \dots, s\}$$

$$\lambda_i > \mu_i \text{ for } i \in F$$

$$\bar{F} = \{s + 1, \dots, N\}$$

$$\lambda_i < \mu_i \text{ for } i \in \bar{F}$$

Observation

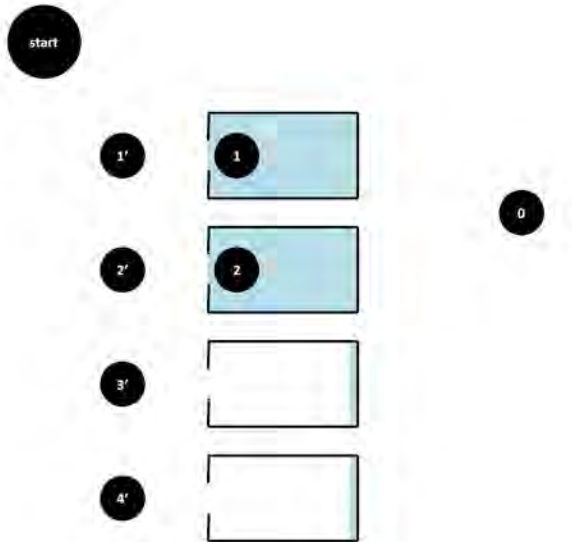
Time through $i \in F \approx NK_i/(N\mu_i) = K_i/\mu_i$, time through $i \notin F \approx 1/(N\mu_i) \approx 0$.

For job at entrance of buffer $i \in F$:

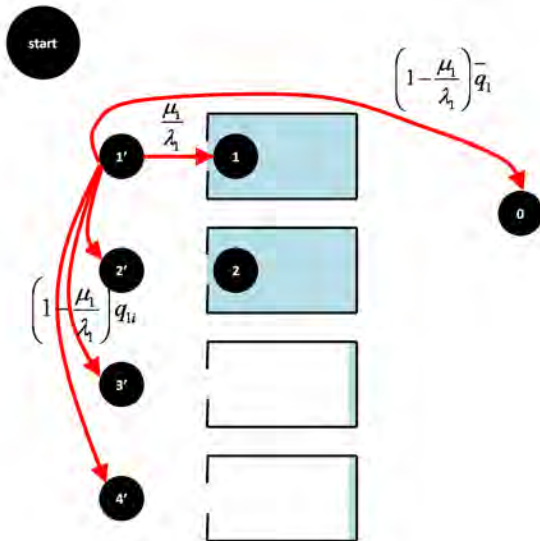
- ▶ enters buffer i w.p. $\approx \mu_i/\lambda_i$
- ▶ routed to entrance of buffer j w.p. $\approx (1 - \mu_i/\lambda_i)q_{ij}$
- ▶ leaves system w.p. $\approx (1 - \mu_i/\lambda_i)\bar{q}_i$

Job at entrance of buffer $i \in \bar{F}$:

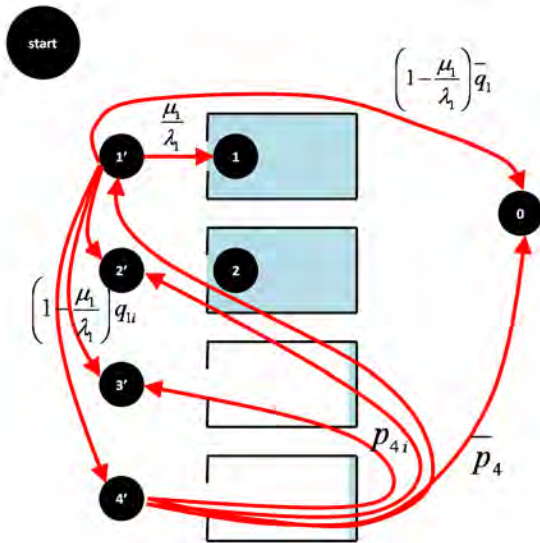
- ▶ routed according to P almost immediately



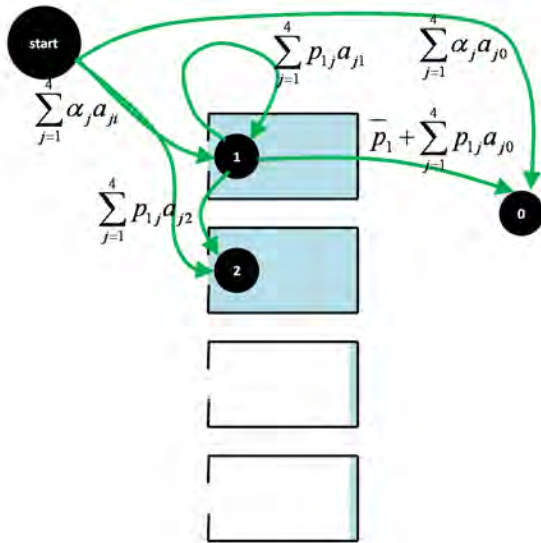
Fast chain on
 $\{0, 1, 2, 1', 2', 3', 4'\}$



Fast chain on
 $\{0, 1, 2, 1', 2', 3', 4'\}$



a_{ij} absorption probability in $j \in \{0, 1, 2\}$ starting in i'



Slow chain on
 $\{0, 1, 2\}$, transitions
based on fast chain.

Finite buffer networks with overflows.

Contributions

- ▶ Limiting traffic equations
- ▶ Efficient algorithm for unique solution
- ▶ Limiting deterministic trajectories
- ▶ Limiting sojourn time distribution

Future work

Work on limits (Chen and Mandelbaum, 1991)