**IEOR 6711: Stochastic Models I**

**Professor Whitt, Tuesday, December 4, 2012**

# More on Queueing Networks

## 1 Overview

There is a substantial body of literature on stochastic networks. The two principle kinds are queueing networks and loss networks; see recent books by Chen and Yao [4] and Serfozo [13] for a full-length account, with many references. See Asmussen [2] for a fast account. For heavy-traffic limits (not to be discussed), see Chen and Yao [4] and Chapter 14 of $W^2$ [17] and references therein.

Some of the older classics also remain important; e.g., see Kelly [8, 9], Sauer and Chandy [12], Walrand [14] and Whittle [18].

## 2 Reversibility and Open Queueing Networks (OQN's)

See §6 of the CTMC notes. Kelly [8] is very nicely written.

## 3 Closed Queueing Networks (CQN's)

### 3.1 The Basic Model

Now we consider the CQN analog of the OQN with single-server queues discussed in §6 and Theorem 6.9. We obtain such an open network by having no flow into or out of the network. There is a fixed number of customers or jobs in the system.

Thus we have no external arrival processes and make the Markovian routing among the $m$ queues be according to an irreducible DTMC with transition matrix $P$. As for the OQN, we assume that each queue has a single server, unlimited waiting room, the first-come first-served (FCFS) service discipline and has exponential service times with rate $\mu_i$ at queue $i$. (The results extend to the case in which the service rate depends on the number of jobs at that queue, i.e., we could have $\mu_i(n_i)$ be the service rate at queue $i$, where $n_i$ is the number there. But we work with the more elementary perspective.)

From a **modeling perspective**, it is natural to wonder whether an open model or a closed model is more appropriate. In some sense, an open model is more natural, because in most actual systems, customers or jobs arrive from outside, move around and then eventually depart when all required service is complete. When closed models are used, people think of a new job replacing an old one when it has completed its required service. Which model is appropriate depends on what is thought to be subject to control.

In an OQN, the throughput is determined by the external arrival rates. The total arrival rate from outside necessarily equals the total departure rate from the network in steady state. The total arrival rate can be considered the throughput. Thus, with an OQN, the throughput is an independent variable (directly specified by the modeler) and the expected number of customers in the system is a dependent variable (calculated from the model analysis). In contrast, with a CQN, the fixed network population is an independent variable (directly specified by the modeler), while the throughput is a dependent variable (calculated from the model analysis).

In computer systems people have overwhelmingly preferred the CQN perspective, because they think of the system population is something natural to control, whereas the throughput should be determined. See $W^2$ [15] for a discussion of OQN and CQN models. The CQN perspective can be achieved with OQN's by using the fixed population mean (FPM) method, discussed in [15]. The idea of the FPM method is to choose an external arrival rate that makes the overall mean population equal to a desired target value. With a CQN, the total population is fixed at some value $N$. In contrast, with an OQN model using the FPM, the total population in steady state is random with a mean equal to $N$. Which approach is desired may depend on the actual variability of the system population.

### 3.2 Solving the CQN Model

For the CQN, the traffic rate equations again are central. Since there are no external arrival processes, instead of (6.12) and (6.13) we have the **CQN traffic rate equations**

$$\lambda_j = \sum_{i=1}^{m} \lambda_i P_{i,j} \quad \text{for} \quad 1 \le j \le m, \tag{1}$$

or equivalently, in matrix notation,

$$\Lambda = \Lambda P, \tag{2}$$

where $\Lambda \equiv (\lambda_1, \ldots, \lambda_m)$ is a $1 \times m$ row vector and $P$ is the usual $m \times m$ Markov transition matrix.

We immediately observe that the CQN traffic equations in (1) and (2) has infinitely many positive solutions, because this equation agrees with the fundamental DTMC equation $\pi = \pi P$. Hence, the unique stationary probability vector $\pi$ is one such solution. However, here there is no reason why the flow rates should sum to 1. Any positive multiple of $\pi$ would be another solution. From the DTMC theorem, those are the only solutions. We do see that the vector $\pi$ does give the relative rates. The ratio of the flow rate through queues $i$ and $j$ necessarily is $\lambda_i/\lambda_j$. We can just use $\pi$ with this interpretation. So now we think of $\pi$ as one invariant measure for the routing matrix $P$ in (1).

Let $\mathbf{n} \equiv (n_1, n_2, \ldots, n_m)$ be a state vector for the CTMC containing the $m$ queues. The CTMC in turn has a rate matrix assigning rate $\mu_i P_{i,j}$ for a service completion from queue $i$ combined with that customer moving to queue $j$ whenever $n_i > 0$. For example, the rate of transitions from the state $(n_1, n_2, n_3, \ldots, n_m)$ to $(n_1 - 1, n_2 + 1, n_3 \ldots, n_m)$ is $\mu_1 P_{1,2}$ whenever $n_1 > 0$.

If we initially ignore the population constraint

$$\sum_{i=1}^{m} n_i = N, \tag{3}$$

then the following can be shown to be a stationary measure for the CQN:

$$f(n) \equiv f((n_1, n_2, \ldots, n_m)) = \prod_{i=1}^{m} \rho_i^{n_i}, \tag{4}$$

where

$$\rho_i \equiv \frac{\lambda_i}{\mu_i} \tag{5}$$

for $\lambda \equiv (\lambda_1, \lambda_2, \ldots, \lambda_m)$ a solution to (1) or (2). However, the population constraint is not satisfied; we have $f(n) > 0$ for $n$ not satisfying (3).

We can apply Kelly's lemma to show that a stationary distribution is indeed given by (4). As stated above, the rate of transitions from the state $(n_1, n_2, n_3, \ldots, n_m)$ to $(n_1 - 1, n_2 + 1, n_3 \ldots, n_m)$ is $\mu_1 P_{1,2}$ whenever $n_1 > 0$. Similarly, for the reverse-time CTMC, the rate of transitions from the state $(n_1 - 1, n_2 + 1, n_3 \ldots, n_m)$ to $(n_1, n_2, n_3, \ldots, n_m)$ is $\mu_2 \overleftarrow{P}_{1,2}$. After defining $\overleftarrow{P}_{i,j}$ such that $\lambda_i \overleftarrow{P}_{i,j} = \lambda_j P_{j,i}$ for all $i$ and $j$. The service rates are the same: $\overleftarrow{\mu}_i = \mu_i$ for all $i$. With that construction, we achieve

$$f(n)Q_{n,n'} = f(n')\overleftarrow{Q}_{n',n} \quad \text{for all} \quad n \quad \text{and} \quad n', \tag{6}$$

where $n$ and $n'$ are both $m$-dimensional state vectors (reachable from each other). For example, suppose that $n = (n_1, n_2, n_3, \ldots, n_m)$ and $n' = (n_1 - 1, n_2 + 1, n_3 \ldots, n_m)$. Then (9) becomes

$$\begin{aligned} f(n)Q_{n,n'} &= f(n)\mu_1 P_{1,2} = f(n')\overleftarrow{Q}_{n',n} \\ &= f(n')\mu_2 \overleftarrow{P}_{2,1} = \frac{f(n')\mu_2\lambda_1 P_{1,2}}{\lambda_2} \end{aligned} \tag{7}$$

which can be seen to be valid because, first, the two $P_{1,2}$ terms cancel and, second,

$$\frac{f(n)}{f(n')} = \frac{\mu_2\lambda_1}{\lambda_2\mu_1}. \tag{8}$$

Next, we can obtain the desired stationary measure satisfying the population constraint (3) by restricting the CTMC to the subset of the state space where (3) holds. That subset can be seen to form a closed subset of the state space for this CTMC. With our transition probabilities, the total population never changes. For any such CTMC, the truncation of the original stationary distribution is a stationary distribution on the closed subset.

Thus, we know that a stationary probability vector for the CQN with the population constraint is

$$\alpha_n \equiv \alpha_{(n_1,n_2,\ldots,n_m)} \equiv \frac{f(n)}{G(N)}, \tag{9}$$

where $f(n)$ is given by (4) and

$$G(N) \equiv \sum_{n:n\in S(N)} f(n) \equiv \sum_{(n_1,\ldots,n_m):(n_1,\ldots,n_m)\in S(N)} f(n) \tag{10}$$

and

$$S(N) \equiv \{(n_1, \ldots, n_m) : n_i \geq 0 \quad \text{for all} \quad i \quad \text{and} \quad n_1 + n_2 + \cdots + n_m = N\}. \tag{11}$$

### 3.3 Connection to OQN's

We can approach the CQN using the OQN. Suppose that we are given a CQN with an irreducible routing matrix $P$. Necessarily $P_{2,1} > 0$, possibly after relabeling the queues. Create an associated OQN by making two changes: (i) introduce a Poisson external arrival process at queue 1 with small rate $\epsilon > 0$ and (ii) let all flow that would have gone from queue 2 to queue 1 leave the network.

The resulting OQN has total external arrival rate $\epsilon$ and thus total departure rate $\epsilon$. We now solve the OQN. For sufficiently small $\epsilon$, we will have $\rho_i \equiv \lambda_i/\mu_i < 1$ at each queue in the OQN. Under that condition, the OQN will have a unique stationary probability vector $\alpha$ satisfying $\alpha Q = 0$. In particular, we will have

$$f_{OQN}(n) \equiv f_{OQN}((n_1, \ldots, n_m)) = \prod_{i=1}^{m} (1 - \rho_i)\rho_i^{n_i} \tag{12}$$

However this unique invariant measure for the OQN is also an invariant measure for the CQN.

Of course, (12) is not the same as (4) because there is the extra term

$$A \equiv \prod_{i=1}^{m}(1-\rho_i), \tag{13}$$

in (12), but $A$ in (13) is a constant, so *both* (4) and (12) are stationary measures for the CQN CTMC, by the same argument given above. This reasoning also leads to formulas (9)-(11) above.

### 3.4 The Convolution (Buzen) Algorithm

We can compute the normalization constant or partition function $\pi(S_N))$ in (10) by a recursive algorithm, called the convolution algorithm or the Buzen algorithm. Let the queues be labeled from 1 to $m$. Inst3ead of the notation $G(N)$, now use new notation that counts the number of queues, with some given initial ordering. Let

$$G(N,m) = \sum_{\{\mathbf{n}:n_1+n_2+\cdots+n_m=N\}} \prod_{i=1}^{m} \rho_i^{n_i}, \tag{14}$$

where

$$\rho_i \equiv \frac{\lambda_i}{\mu_i}, \quad 1 \le i \le m, \tag{15}$$

with $\lambda \equiv (\lambda_1, \ldots, \lambda_m)$ a solution to (1).

**Theorem 3.1** (*Buzen's convolution algorithm*) *For any $N > 1$ and $M > 1$, the normalization constant $G(N,m)$ defined in* (14)*can be solved by the two-dimensional recursion*

$$G(N,m) = g(N,m-1) + \rho_m G(N-1,m). \tag{16}$$

**Proof.** The first term on the right in (16) covers the case in which $n_m = 0$, while the second case on the right covers the complementary case in which $n_m > 0$. ∎

### 3.5 Generating Functions of the Normalization Constant

An alternative way to compute the normalization constant is to numerically invert its generating function. The generating functions have a remarkably simple form. For the basic model with normalization constant $G(N)$ in (10), its generating function is defined as

$$\hat{G}(z) \equiv \sum_{N=0}^{\infty} G(N) z^N. \tag{17}$$

Instead of applying the convolution algorithm, we could calculate $G(N)$ by numerically inverting the generating function in (17). For the basic model, we have the following result. We now follow [5] and denote the total population by $K$ and the number of queues by $q$

**Proposition 3.1** (*generating function for a single-chain single-class CQN of single-server queues*) *For a single-chain single-class CQN with $q$ single-server queues,*

$$\hat{G}(\mathbf{z}) \equiv \sum_{K=0}^{\infty} G(K) z^K = \prod_{i=1}^{q} (1-\rho_i z^i)^{-1}. \tag{18}$$

**Proof.** Substituting in $G(K)$ from (22) into the special case of (23), we obtain

$$\begin{aligned}
\hat{G}(\mathbf{z}) &\equiv \sum_{N=0}^{\infty} G(K) z^K = \sum_{K=0}^{\infty} \left( \sum_{\mathbf{n} \in S(K)} f(\mathbf{n}) \right) z^K \\
&= \sum_{K=0}^{\infty} \left( \sum_{\mathbf{n} \in S(K)} \prod_{i=1}^{q} \rho_i^{n_i} \right) z^K \\
&= \sum_{K=0}^{\infty} \left( \sum_{\mathbf{n} \in S(K)} \prod_{i=1}^{q} \rho_i^{n_i} z^{n_i} \right) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_q=0}^{\infty} \left( \prod_{i=1}^{q} \rho_i^{n_i} z^{n_i} \right) \\
&= \prod_{i=1}^{q} \left( \sum_{n_1=0}^{\infty} \cdots \sum_{n_q=0}^{\infty} \rho_i^{n_i} z^{n_i} \right) = \prod_{i=1}^{q} (1 - \rho_i z)^{-1} \qquad (19)
\end{aligned}$$

as claimed. ∎

### 3.6 Multi-Chain Multi-Class Models

We next introduce more general models. See pp. 949-950 of Choudhury et al. [5] (class handout) and p. 91 of [12]. Now we have **customer classes** as well as **queues**. The state is the vector of possible numbers $n_{(r,i)}$ of customers of class $r$ at queue $i$. The set of all pairs $(r, i)$, called *stages*, is the set $C$. The **routing matrix** is a transition matrix inducing a DTMC on the set of stages. We assume that the set of stages $C$ can be partitioned into $p$ mutually exclusive closed communication classes $C_j$. Each communication class is called a **chain**. We stipulate the population in each communication class $C_j$ is fixed at $K_j$.

Let $p(\mathbf{n}) \equiv \alpha(\mathbf{n}) \equiv p(n_{r,i} : (r, i) \in C)$ be the stationary and steady-state probability vector. It has a product-form with normalization, i.e.,

$$p(\mathbf{n}) = \frac{f(\mathbf{n})}{G(\mathbf{K})}, \qquad (20)$$

where

$$G(\mathbf{K}) = \sum_{\mathbf{n} \in S(\mathbf{K})} f(\mathbf{n}) \qquad (21)$$

and

$$S(\mathbf{K}) = \{\mathbf{n} \equiv (n_{(r,i)} : (r, i) \in C) : \sum_{(r,i) \in C_j} n_{r,i} = K_j, \quad 1 \leq j \leq p\}. \qquad (22)$$

### 3.7 Generating Functions in Multi-Chain Multi-Class Models

We now turn to the more general multi-chain multi-class model. Again see p. 950 of Choudhury et al. [5] (class handout). See [3, 10, 11] for additional background. See [5, 7] for the numerical inversion algorithm to compute the normalization constant (partition function $G(\mathbf{K})$. The generating function is defined by

$$\hat{G}(\mathbf{z}) = \sum_{K_1=0}^{\infty} \cdots \sum_{K_p=0}^{\infty} G(\mathbf{K}) \prod_{j=1}^{p} z_j^{K_j} \qquad (23)$$

for $G(\mathbf{K})$ in (22).

The general expression for the generating function is taken from [3].

**Theorem 3.2** (*generating function for a p-chain multi-class CQN of q single-server queues and $N-q$ infinite-server queues*)

*For a p-chain multi-class CQN with q single-server queues and $N-q$ infinite-server queues (indexed by $q+1,\ldots,N$),*

$$\hat{G}(\mathbf{z}) \equiv \sum_{K=0}^{\infty} G(K) z^K = \frac{\exp\left(\sum_{j=1}^{p} \rho_{j,0} z^j\right)}{\prod_{i=1}^{q}\left(1-\sum_{j=1}^{p} \rho'_{j,i} z_j\right)}. \tag{24}$$

*where $\rho_{j,0} \equiv \sum_{i=q+1}^{N} \sum_{(r,i)\in C_j} \rho_{r,i}$ and $\rho'_{j,i} \equiv \sum_{(r,i)\in C_j} \rho'_{r,i}$.*

To give the essential reasoning, we give the details for additional special cases.

**Proposition 3.2** (*generating function for a single-chain single-class CQN of infinite-server queues*) *For a single-chain single-class CQN with q infinite-server queues,*

$$\hat{G}(\mathbf{z}) \equiv \sum_{K=0}^{\infty} G(K) z^K = e^{\rho z}, \tag{25}$$

*where $\rho \equiv \rho_1 + \cdots + \rho_q$.*

**Proof.** Substituting in $G(K)$ from (22) into the special case of (23), we obtain

$$\begin{aligned}
\hat{G}(\mathbf{z}) &\equiv \sum_{K=0}^{\infty} G(K) z^K = \sum_{K=0}^{\infty} \left( \sum_{\mathbf{n}\in S(K)} f(\mathbf{n}) \right) z^K \\
&= \sum_{K=0}^{\infty} \left( \sum_{\mathbf{n}\in S(K)} \prod_{i=1}^{q} \frac{\rho_i^{n_i}}{n_i!} \right) z^K \\
&= \sum_{K=0}^{\infty} \left( \sum_{\mathbf{n}\in S(K)} \prod_{i=1}^{q} \frac{(\rho_i z)^{n_i}}{n_i!} \right) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_q=0}^{\infty} \left( \prod_{i=1}^{q} \frac{(\rho_i z)^{n_i}}{n_i!} \right) \\
&= \prod_{i=1}^{q} \left( \sum_{n_1=0}^{\infty} \cdots \sum_{n_q=0}^{\infty} \frac{(\rho_i z)^{n_i}}{n_i!} \right) = \prod_{i=1}^{q} e^{\rho_i z} = e^{\rho_1 z + \cdots + \rho_q z} = e^{\rho z}
\end{aligned} \tag{26}$$

as claimed. ∎

We now consider a two-class case.

**Proposition 3.3** (*generating function for a 2-chain multi-class CQN of single-server queues*) *For a 2-chain multi-class CQN with q single-server queues,*

$$\hat{G}(\mathbf{z}) \equiv \sum_{K_1=0}^{\infty} \sum_{K_2=0}^{\infty} G(K_1, K_2) z_1^{K_1} z_2^{K_2} = \prod_{i=1}^{q} \left(1 - \rho'_{1,i} z_1 - \rho'_{2,i} z_2\right)^{-1}. \tag{27}$$

*where $\rho'_{j,i} \equiv \sum_{(r,i)\in C_j} \rho'_{r,i}$*

**Proof.** Substituting in $G(K)$ from (22) into the special case of (23), we obtain

$$\begin{aligned}
\hat{G}(\mathbf{z}) &\equiv \hat{G}(z_1,z_2) = \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty} G(K_1,K_2) z_1^{K_1} z_2^{K_2} \\
&= \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty}\left(\sum_{\mathbf{n}\in S(K_1,K_2)} f(\mathbf{n})\right) z_1^{K_1} z_2^{K_2} \\
&= \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty}\left(\sum_{(n_1,n_2)\in S(K_1,K_2)}\left(\prod_{i=1}^{q} n_i! \prod_{r} \frac{\rho_{r,i}'^{n_{r,i}}}{n_{r,i}!}\right)\right) z_1^{K_1} z_2^{K_2} \\
&= \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty}\left(\sum_{(n_1,n_2)\in S(K_1,K_2)}\left(\prod_{i=1}^{q} n_i! \prod_{(r,i)\in C_1} \frac{(\rho_{r,i}' z_1)^{n_{r,i}}}{n_{r,i}!} \prod_{(r,i)\in C_2} \frac{(\rho_{r,i}' z_2)^{n_{r,i}}}{n_{r,i}!}\right)\right) \\
&= \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty}\left(\sum_{(n_1,n_2)\in S(K_1,K_2)}\left(\prod_{i=1}^{q} \frac{n_i!}{n_{1,i}!n_{2,i}!} \prod_{(r,i)\in C_1} \frac{n_{1,i}!(\rho_{r,i}' z_1)^{n_{r,i}}}{n_{r,i}!} \prod_{(r,i)\in C_2} \frac{n_{2,i}!(\rho_{r,i}' z_2)^{n_{r,i}}}{n_{r,i}!}\right)\right) \\
&= \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty}\left(\sum_{(n_1,n_2)\in S(K_1,K_2)}\left(\prod_{i=1}^{q} \frac{n_i!}{n_{1,i}!n_{2,i}!}(\rho_{1,i} z_1)^{n_{1,i}}(\rho_{2,i} z_2)^{n_{2,i}} \times \right.\right. \\
&\qquad \left.\left. \times \prod_{(r,i)\in C_1} \frac{n_{1,i}!(\rho_{r,i}' z_1)^{n_{r,i}}}{(\rho_{1,i} z_1)^{n_{1,i}} n_{r,i}!} \prod_{(r,i)\in C_2} \frac{n_{2,i}!(\rho_{r,i}' z_2)^{n_{r,i}}}{(\rho_{2,i} z_2)^{n_{2,i}} n_{r,i}!}\right)\right) \\
&= \sum_{K_1=0}^{\infty}\sum_{K_2=0}^{\infty}\left(\sum_{(n_1,n_2)\in S(K_1,K_2)}\left(\prod_{i=1}^{q} \frac{n_i!}{n_{1,i}!n_{2,i}!} \frac{(\rho_{1,i} z_1)^{n_{1,i}}(\rho_{2,i} z_2)^{n_{2,i}}}{(\rho_{1,i} z_1 + \rho_{2,i} z_2)^{n_i}} (\rho_{1,i} z_1 + \rho_{2,i} z_2)^{n_i} \times \right.\right. \\
&\qquad \left.\left. \times \prod_{(r,i)\in C_1} \frac{n_{1,i}!(\rho_{r,i}' z_1)^{n_{r,i}}}{(\rho_{1,i} z_1)^{n_{1,i}} n_{r,i}!} \prod_{(r,i)\in C_2} \frac{n_{2,i}!(\rho_{r,i}' z_2)^{n_{r,i}}}{(\rho_{2,i} z_2)^{n_{2,i}} n_{r,i}!}\right)\right) \\
&= \prod_{i=1}^{q}\sum_{n_i=0}^{\infty} (\rho_{1,i}' z_1 + \rho_{2,i}' z_2)^{n_i} \\
&= \prod_{i=1}^{q} \left(1 - \rho_{1,i}' z_1 - \rho_{2,i}' z_2\right)^{-1}. \qquad (28)
\end{aligned}$$

as claimed. ∎

### 3.8 Quasi-Reversibility

See pp. 90-98 of Walrand [14]

## 4 Loss Networks

See [6, 9, 16].

### 4.1 The Model

## References


[1] Abate, J., G. L. Choudhury, G. L., W. Whitt. 1999. An introduction to numerical transform inversion and its application to probability models. in *Computational Probability*, W. Grassman (ed.), Kluwer, Boston, 257–323.

[2] Asmussen, S. 2003. *Applied Probability and Queues*, second edition, Springer.

[3] Bertozzi, A. and J. McKenna. 1993. Multidimensional residues, generating functions and their application to queueing networks. In *SIAM Review* 35, 239–268.

[4] Chen, H. and D. D. Yao. 2001. *Fundamentals of Queueing Networks: Performance, Asymptotics and Optimization*, Springer.

[5] Choudhury, G. L., K. K. Leung and W. Whitt. 1995. Calculating Normalization Constants of Closed Queueing Networks by Numerically Inverting Their Generating Functions. J. ACM 42, 935–970.

[6] Choudhury, G. L., K. K. Leung and W. Whitt. 1995b. An Algorithm to Compute Blocking Probabilities in Multi-Rate Multi-Class Multi-Resource Loss Models. Advances in Appl. Prob. 27, 1104–1143.

[7] Choudhury, G. L., D. L. Lucantomi and W. Whitt. 1994. Multi-Dimensional Transform Inversion with Applications to the Transient M/G/1 Queue. *Annals of Applied Probability* 4, 719–740.

[8] Kelly, F. P. 1979. *Reversibility and Stochastic Networks*, Wiley.

[9] Kelly, F. P. 1991. Loss networks. *Ann. Appl. Prob.* 1, 319–378.

[10] Kogan, Y. 2002. Asymptotic Expansions for Large Closed and Loss Queueing Networks. *Math. Prob. Engr.* 8, 323–348.

[11] Pemantle, R. and M. C. Wilson. 2008. Twenty Combinatorial Examples of Asymptotics Derived from Multivariate Generating Functions. *SIAM Review* 50, 199–272.

[12] Sauer, A. H. and K. M. Chandi. 1981. *Computer Systems Performance Modeling*, Prentice Hall.

[13] Serfozo, R. F. 1999. *Introduction to Stochastic Networks*, Springer.

[14] Walrand, J. 1988. *An Introduction to Queueing Networks*, Prentice Hall.

[15] Whitt, W. 1984. Open and Closed Models of Networks of Queues. *AT&T Bell laboratories Tech. J.* 63, 1911–1979.

[16] Whitt, W. 1985. Blocking When Service is Required from Several Facilities Simultaneously. *AT&T Technical Journal* 64, 1807–1856.

[17] Whitt, W. 2002. *Stochastic-Process Limits*, Springer.

[18] Whittle, P. 1986. *Systems in Stochastic Equilbirium*, Wiley.