Queueing Theory and Simulation, lecture 14

Nicky van Foreest

June 22, 2021

Contents

1 Status and Plan **1**

2 Open network of $M/M/c$ stations **1**

3 On $\lambda = \gamma + \lambda P$. **3**

4 Summary of the course **5**

1 Status and Plan

1.1 Plan

- Past: Exact models and N policies
- Now:
  - An open network of $M/M/c$ queues
  - Fixed points of the (vector) equation $\gamma = \lambda + \gamma P$ and $v = c + Pv$.
- Next: you are going to work for the exam

2 Open network of $M/M/c$ stations

2.1 Model

- New jobs arrive from the 'outside world' to station i as a Poisson process with rate γ_i.
- Job service times are $\text{Exp}(\mu)$.
- After service jobs are routed from station i to station j with probability P_{ij}, or leave the network with probability P_{i0}.
- The matrix P is known as a routing matrix.
- There are M stations
- Assumption (we address this later in more detail) the system is stable, i.e., all new jobs eventually leave.

2.2 What we like to know

- The expected total sojourn time $\mathsf{E}[J]$.
- The stationary distribution of the jobs over the stations.

$$p(n) = \mathsf{P}[L_1 = n_1, \ldots, L_M = n_M]$$

2.3 Traffic equations

- What arrives at station i eventually leaves: $\delta_i = \lambda_i$.
- Outflow: Let $\lambda_i, i = 1, \ldots, M$ be the out-flows of stations $i, 1, \ldots, M$.
- In-flows at station i: $\gamma_i + \sum_j \lambda_j P_{ji}$
- Inflow equals outflow station i:

$$\lambda_i = \gamma_i + \sum_{j=0}^{M} \lambda_j P_{ji}. \tag{1}$$

- Total outflow of network: $|\gamma| = \sum_i \gamma_i$.
- Solve for λ in $\lambda = \gamma + \lambda P \implies \lambda = \gamma(I - P)^{-1}$.
- $\rho_i = \lambda_i \mathsf{E}[S_i]$, $\rho_i < 1$ for stability.

2.4 Sojourn times and visit ratios

- $\mathsf{E}[L_i] = \lambda_i \mathsf{E}[J_i]$.
-

$$|\gamma| \mathsf{E}[J] = \mathsf{E}[L] = \sum_i \mathsf{E}[L_i] = \sum_i \lambda_i \mathsf{E}[J_i] \tag{2}$$

$$\mathsf{E}[J] = \sum_i \frac{\lambda_i}{|\gamma|} \mathsf{E}[J_i]. \tag{3}$$

- $\lambda_i/|\gamma|$ is the visit ratio of Station i.
- Little's law applied twice!

2.5 All stations are $M/M/c$ queues

- For the $M/M/c$ queue, interdeparture times are Exp(λ), see the exercises in the book.
- Take any station j. Assume its departure process is Poisson λ_j.
- What moves to station i is the *thinned* Poisson process $\lambda_j P_{ji}$.
- The inflow at station i is the *merged* stream of external arrivals (Poisson γ_i) and the departures of the other stations (Poisson with rates $\lambda_j, j = 1, \ldots, M$.
- Merging Poisson streams results in a Poisson stream, hence station i receives a Poisson stream

2.6 Sojourn time for nework with $M/M/1$ queues

- $$\mathsf{E}[J] = \sum_i \frac{\lambda i}{|\gamma|} \mathsf{E}[J_i] = \sum_i \frac{\lambda i}{|\gamma|} \frac{\mathsf{E}[S_i]}{1-\rho_i}. \tag{4}$$

- Think again about why we like to know $\mathsf{E}[J]$.
  - $\mathsf{E}[J]$ is average time between investment in raw materials and receiving a payment from a customer.
  - $\mathsf{E}[J]$ is time a customer likes to know when placing an order.

2.7 Product form solution

- It can be proven that for an open network of $M/M/c$ queues that

$$p(n) = \mathsf{P}[L_1 = n_1, \ldots, L_M = n_M] = \Pi_{i=1}^M p(n_i). \tag{5}$$

- That is: the queues are independent $M/M/c$ queues!
- In case station i is an $M/M/1$ queue

$$p(n_i) = \mathsf{P}[L_i = n_i] = (1-\rho_i)\rho_i^{n_i}. \tag{6}$$

- See the book (exercises) for a proof for two $M/M/1$ in tandem.

3 On $\lambda = \gamma + \lambda P$.

3.1 Motivation

- $\lambda = \gamma + \lambda P$ are the traffic equations for queueing networks
- For N-policies, the recursion for T has the form

$$T(q) = c + \alpha T(q+1) + \beta T(q-1) \tag{7}$$
$$\implies T = c + PT, \tag{8}$$
$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & \ldots \\ \beta & 0 & \alpha & 0 & \ldots \\ 0 & \beta & 0 & \alpha & 0 \\ \vdots & & \ddots & \ddots & \ddots \end{pmatrix}, \tag{9}$$

- Similar recursion for V.

3.2 Solving $\lambda = \gamma + \lambda P$.

- Use recursion:

$$\lambda = \gamma + \lambda P \tag{10}$$
$$= \gamma + (\gamma + \lambda P)P = \gamma + \gamma P + \lambda P^2 \tag{11}$$
$$= \gamma \sum_{m=0}^{n} P^m + \lambda P^{n+1} \tag{12}$$
$$\to \gamma \sum_{m=0}^{\infty} P^m \text{ as } n \to \infty, \tag{13}$$

if $P^n \to 0$ as $n \to \infty$.

3.3 Powers of P

- Suppose that P is diagolizable. Then there exist eigenvectors assembled in a matrix V such that
  - $VP = \Lambda V$,
  - $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_M)$.
- Hence,

$$P^2 = V^{-1}\Lambda V \cdot V^{-1}\Lambda V = V^{-1}\Lambda^2 V, \tag{14}$$
$$P^n = V^{-1}\Lambda^n V \tag{15}$$
$$\Lambda^n = \text{diag}(\lambda_1^n, \ldots, \lambda_M^n). \tag{16}$$

3.4 Eigenvalues of P

- Assume that the network is transient: $P_{i0}^M > 0$. This means that after at most M jumbs from any station to another, it is possible to leave the network.
- What is the longest network you can make? A tandem network: a job has to visit all M stations before being able to leave the network.
- Under the condition of transience, we use Gershgorin's theorem (see book, really neat and simple theorem) to show that $|\lambda_i| < 1$ for all i.
- Hence, $\Lambda^n \to 0$ as $n \to \infty$.

3.5 Sum of P^n

- Hence:

$$\sum_{i=0}^{\infty} P^n = V^{-1} \sum_{i=0}^{\infty} \Lambda^i V \tag{17}$$
$$= V^{-1}\text{diag}(1/(1-\lambda_1), \ldots, 1/(1-\lambda_M))V. \tag{18}$$

- Note that this is numerically not the most efficient way to solve $(I - P)^{-1}$.

4 Summary of the course

4.1 Topics and most important concepts

- Simulation(construction) of QS, recursions
- Approximation for $G/G/c$, Sakasegawa's formula
- Sample path (simulation) analysis:
  - Renewal reward
  - Level crossing
  - PASTA
  - Little's law
- Analysis of exact models $M/M/1$, $M^X/M/1$, $M/G/1$: Polaczek-Khinchine formula
- Queuing control: recursion
- Open networks: stationary distribution of Markov chains.

4.2 One journey finishes, but another takes off

- Queueing systems are examples of stochastic processes and Markov chains
- The analysis N policies is an example of optimal stopping theory.
- Optimal stopping theory has applications to electrical networks, medicine, partial differential equations. In particular:
  - Finance, option theory.
  - Multi-armed bandits, web site optimization
  - Reinforcement learning (Automatic driving cars, games, and so on)
- So study hard on QTS to get on board for the extended ride.