Analysis of Queueing Systems with Sample Paths and Simulation

Nicky D. van Foreest

October 30, 2017

Contents

1 Introduction

Motivation and Examples Queueing systems abound, and the analysis and control of queueing systems are major topics in the control, performance evaluation and optimization of production and service systems.

At my local supermarket, for instance, any customer that joins a queue of 4 of more customers get his/her shopping for free. Of course, there are some constraints: at least one of the cashier facilities has to unoccupied by a server and the customers in queue should be equally divided over the cashiers that are open. (And perhaps there are some further rules, of which I am unaware.) The manager that controls the occupation of the cashier positions is focused on keeping $\pi(4)+\pi(5)+\cdots$, i.e., the fraction of customers that see upon arrival a queue length exceeding 3, very small. In a sense, this is easy enough: just hire many cashiers. However, the cost of personnel may then outweigh the yearly average cost of paying the customer penalties. Thus, the manager's problem becomes to plan and control the service capacity in such a way that both the penalties and the personnel cost are small.

Fast food restaurants also deal with many interesting queueing situations. Consider, for instance, the making of hamburgers. Typically, hamburgers are made-to-stock, in other words, they are prepared before the actual demand has arrived. Thus, hamburgers in stock can be interpreted as customers in queue waiting for service, where the service time is the time between the arrival of two customers that buy hamburgers. The hamburgers have a typical lifetime, and they have to be scrapped if they remain on the shelf longer than some amount of time. Thus, the waiting time of hamburgers has to be closely monitored. Of course, it is easy to achieve zero scrap cost, simply by keeping no stock at all. However, to prevent lost-sales it is very important to maintain a certain amount of hamburgers on stock. Thus, the manager has to balance the scrap cost against the cost of lost sales. In more formal terms, the problem is to choose a policy to prepare hamburgers such that the cost of excess waiting time (scrap) is balanced against the cost of an empty queue (lost sales).

Service systems, such as hospitals, call centers, courts, and so on, have a certain capacity available to serve customers. The performance of such systems is, in part, measured by the total number of jobs processed per year and the fraction of jobs processed within a certain time between receiving and closing the job. Here the problem is to organize the capacity such that the sojourn time, i.e., the typical time a job spends in the system, does not exceed some threshold, and such that the system achieves a certain throughput, i.e., jobs served per year.

Clearly, all the above systems can be seen as queueing systems that have to be monitored and controlled to achieve a certain performance. The performance analysis of such systems can, typically, be characterized by the following performance measures:

1. The fraction of time $p(n)$ that the system contains n customers. In particular, $1-p(0)$, i.e., the fraction of time the system contains jobs, is important, as this is a measure of the time-average occupancy of the servers, hence related to personnel cost.

2. The fraction of customers $\pi(n)$ that 'see upon arrival' the system with n customers. This measure relates to customer perception and lost sales, i.e., fractions of arriving customers

that do not enter the system.

3. The average, variance, and/or distribution of the waiting time.

4. The average, variance, and/or distribution of the number of customers in the system.

Here the system can be anything that is capable of holding jobs, such as a queue, the server(s), an entire court house, patients waiting for an MRI scan in a hospital, and so on.

It is important to realize that a queueing system can, typically, be decomposed into *two subsystems*, the queue itself and the service system. Thus, we are concerned with three types of waiting: waiting in queue, i.e., *queueing time*, waiting while being in service, i.e., the *service time*, and the total waiting time in the system, i.e., the *sojourn time*.

Organization In these notes we will be primarily concerned with making models of queueing systems such that we can compute or estimate the above performance measures. Part of our work is to derive analytic models. The benefit of such models is that they offer structural insights into the behavior of the system and scaling laws, such as that the average waiting time scales (more or less) linearly in the variance of the service times of individual customers. However, these models have severe shortcomings when it comes to analyzing real queueing systems, in particular when particular control rules have to be assessed. Consider, for example, the service process at a check-in desk of KLM. Business customers and economy customers are served by two separate queueing systems. The business customers are served by one server, server A say, while the economy class customers by three servers, say. What would happen to the sojourn time of the business customers if server A would be allowed to serve economy class customers when the business queue is empty? For the analysis of such cases simulation is a very useful and natural approach.

In the first part of these notes we concentrate on the analysis of *sample paths of a queueing process*. We assume that a typical sample path captures the 'normal' stochastic behavior of the system. This sample-path approach has two advantages. In the first place, most of the theoretical results follow from very concrete aspects of these sample paths. Second, the analysis of sample-paths carries over right away to simulation. In fact, simulation of a queueing system offers us one (or more) sample paths, and based on such sample paths we derive behavioral and statistical properties of the system. Thus, the performance measures defined for sample paths are precisely those used for simulation. Our aim is not to provide rigorous proofs for all results derived below; for the proofs and further background discussion we refer to El-Taha and Stidham Jr. [1998]. As a consequence we tacitly assume in the remainder that results derived from the (long-run) analysis of a particular sample path are equal to their 'probabilistic counterpart'.

In the second part we construct algorithms to analyze open and closed queueing networks. Many of the sample path results developed for the single-station case can be applied to these networks. As such, theory, simulation and algorithms form a nicely round out part of work. For this part we refer to the book of Prof. Zijm; the present set of notes augments the discussion there.

Exercises I urge you to make *all* exercises in this set of notes. Many exercises require many of the tools you learned previously in courses on calculus, probability, and linear algebra. Here you can see them applied. Moreover, many of these tools will be useful for other, future, courses. Thus, the investments made here will pay off for the rest of your (student) career. Moreover, the exercises are meant to *illustrate* the material and to force you to *think* about it. Thus, the main text does not contain many examples; the exercises form the examples.

You'll notice that many of these problems are quite difficult, often not because the problem itself is difficult, but because you need to combine a substantial amount of knowledge all at the same time. All this takes time and effort. Next to this, I did not include the exercises with the intention that you would find them easy. The problems should be doable, but hard.

The solution manual is meant to prevent you from getting stuck and to help you increase your knowledge of probability, linear algebra, programming (analysis with computer support), and queueing in particular. Thus, read the solutions very carefully.

As a guideline to making the exercises I recommend the following approach. First read the notes. Then attempt to make a exercises for 10 minutes or so by yourself. If by that time you have not obtained a good idea on how to approach the problem, check the solution manual. Once you have understood the solution, try to repeat the arguments *with the solution manual closed*.

Further reading The following books are very useful to extend one's knowledge of probability theory and queueing systems:

1. Capiński and Zastawniak [2003]
2. Tijms [1994] and/or Tijms [2003]
3. El-Taha and Stidham Jr. [1998]
4. Bolch et al. [2006]

Acknowledgments I would like to acknowledge dr. J.W. Nieuwenhuis for our many discussions on the formal aspects of queueing theory and prof. dr. W.H.M. Zijm for allowing me to use the first few chapters of his book.

2 Single-Station Queueing Systems

In this chapter, we start with a discussion of the exponential distribution and the related Poisson process, as these concepts are perhaps the most important building blocks of queueing theory. With these concepts we can specify the arrival and service processes of customers, so that we can construct queueing processes and define performance measures to provide insight into the (transient and average) behavior of queueing processes. As it turns out, these constructions can be easily implemented as computer programs, thereby allowing to use simulation to analyze queueing systems. We then continue with developing models for various single-station queueing systems in steady-state, which is, in a sense to be discusses later, the long-run behavior of a stochastic system.[1] In the analysis we use sample-path arguments to count how often certain events occur as functions of time. Then we define probabilities in terms of limits of fractions of these counting processes. Another useful aspect of sample-path analysis is that the definitions for the performance measures are entirely constructive, hence by leaving out the limits, they provide expressions that can be right away used in statistical analysis of (simulations of) queueing systems. Level-crossing arguments will be of particular importance as we use these time and again to develop recursions by which we can compute steady-state probabilities of the queue length or waiting time process.

[1] This statement is, admittedly, vague, to say the least.

2.1 Exponential Distribution

Theory and Exercises

As we will see in the sections to come, the modeling and analysis of any queueing system involves the specification of the (probability) distribution of the time between consecutive arrival epochs of jobs, or the specification of the distribution of the number of jobs that arrive in a certain interval. For the first case, the most common distribution is the exponential distribution, while for the second it is the Poisson distribution. For these reasons we start our discussion of the analysis of queueing system with these two exceedingly important distributions. In the ensuing sections we will use these distributions time and again.

As mentioned, one of the most useful models for the interarrival times of jobs assumes that the sequence $\{X_i\}$ of interarrival times is a set of *independent and identically distributed (i.i.d.)* random variables. Let us write X for the generic random time between two successive arrivals. For many queueing systems, measurements of the interarrival times between consecutive arrivals show that it is reasonable to model an interarrival X as an *exponentially distributed* random variable, i.e.,

$$\mathsf{P}\{X \leq t\} = 1 - e^{-\lambda t} := G(t)$$

The constant λ is often called the *rate*. The reason behind this will be clarified once we relate the exponential distribution to the Poisson process in Section 2.2. In the sequel we often write $X \sim \exp(\lambda)$ to mean that X is exponentially distributed with rate λ.

Exercise 2.1.1. If the random variable $X \sim \exp(\lambda)$, show that its mean $\mathsf{E}\{X\} = \frac{1}{\lambda}$.

Exercise 2.1.2. If $X \sim \exp(\lambda)$, show that its second moment $\mathsf{E}\{X^2\} = \frac{2}{\lambda^2}$.

Exercise 2.1.3. If $X \sim \exp(\lambda)$, show that the *variance* $\mathsf{V}\{X\} = \lambda^{-2}$.

The above exercises can also be easily solved with the concept of *moment generating function* which is defined for a random variable $X \geq 0$ as

$$M_X(t) = \mathsf{E}\left\{e^{tX}\right\}.$$

Exercise 2.1.4. If X is an exponentially distributed random variable with parameter λ, show that its moment generating function $M_X(t) = \lambda/(\lambda - t)$.

Exercise 2.1.5. Use the moment generating function to show that

$$\mathsf{E}\{X\} = M'(0) = \frac{1}{\lambda}, \qquad \mathsf{E}\left\{X^2\right\} = M''(0) = \frac{2}{\lambda^2},$$

where $M'(t) = \frac{\mathrm{d}}{\mathrm{d}t} M(t)$.

Define the *square coefficient of variation (SCV)* as

$$C_a^2 = \frac{\mathsf{V}\{X\}}{(\mathsf{E}\{X\})^2}. \tag{2.1}$$

As will become clear later, the SCV is a very important concept in queueing theory. Memorize it as a measure of *relative variability*.

Exercise 2.1.6. Prove that when X is exponentially distributed, $C_a^2 = 1$.

Let us show with simulation how the exponential distribution originates. Consider N people that regularly visit a shop. We assume that we can characterize the interarrival times $\{X_k^i, k = 1,2,\ldots\}$ of customer i by some distribution function, for instance the uniform distribution. Then, with $A_0^i = 0$ for all i,

$$A_k^i = A_{k-1}^i + X_k^i = \sum_{j=1}^{n} X_j^i,$$

is the arrival moment of the kth visit of customer i. Now the shop owner 'sees' the superposition of the arrivals of all customers. One way to compute the arrival moments of all customers together as seen by the shop is to put all the numbers $\{A_k^i, k = 1,\ldots,n, i = 1,\ldots,N\}$ into one set, and sort these numbers in increasing order. This results in the (sorted) set of arrival times $\{A_k, k = 1,2,\ldots\}$ at the shop. Taking $A_0 = 0$,

$$X_k = A_k - A_{k-1},$$

must be the interarrival time between the $k-1$th and kth visit to the shop. Like this, starting from interarrival times of individual customers we can construct interarrival times as seen by the shop.

Exercise 2.1.7. Let A_i be the arrival time of customer i and set $A_0 = 0$. Assume that the interarrival times $\{X_i\}$ are i.i.d. with exponential distribution with mean $1/\lambda$ for some $\lambda > 0$. Prove that

$$f_{A_i}(t) = \lambda e^{-\lambda t} \frac{(\lambda t)^{i-1}}{(i-1)!}$$

is the density of $A_i = X_1 + X_2 + \cdots + X_i = \sum_{k=1}^{i} X_k$ with $i \geq 1$.

Exercise 2.1.8. Assume that the interarrival times $\{X_i\}$ are i.i.d. and $X_i \sim \exp(\lambda)$. Let $A_i = X_1 + X_2 + \cdots + X_i = \sum_{k=1}^{i} X_k$ with $i \geq 1$. Use the density of A_i or the moment generating function of A_i to show that

$$\mathsf{E}\{A_i\} = \frac{i}{\lambda},$$

that is, the expected time to see i jobs is i/λ.

To plot the *empirical distribution function*, or the *histogram*, of the interarrival times at the shop, we need to count the number of interarrival times smaller than time t for any t. For this, we introduce the *indicator function*:

$$\mathbb{1}_A = \begin{cases} 1, & \text{if the event } A \text{ is true,} \\ 0, & \text{if the event } A \text{ is false.} \end{cases}$$

Define then the emperical distribution of the interarriva

$$\mathsf{P}_{nN}\{X \leq t\} = \frac{1}{nN} \sum_{k=1}^{nN} \mathbb{1}_{X_k \leq t},$$

where $\mathbb{1}_{X_k \leq t} = 1$ if $X_k \leq t$ and $\mathbb{1}_{X_k \leq t} = 0$ if $X_k > t$.

Let us now compare the probability density as obtained for several simulation scenarios to the density of the exponential distribution, i.e., to $\lambda e^{-\lambda t}$. As a first example, take $N = 1$ customer and let the computer generate $n = 100$ uniformly distributed numbers on the set $[4, 6]$.

Thus, the time between two visits of this customer is somewhere between 4 and 6 hours, and the average interarrival times $\mathsf{E}\{X\} = 5$. In a second simulation we take $N = 3$ customers, and in the third, $N = 10$ customers. The emperical distributions are shown, from left to right, in the three panels in Figure 2.1. The continuous curve is the graph of $\lambda e^{-\lambda x}$ where $\lambda = N/\mathsf{E}\{X\} = N/5$. (In Eq. (2.26) we show that when 1 person visits the shop with an average interarrival time of 5 hours, it must be that the arrival rate is $1/\mathsf{E}\{X\} = 1/5$. Hence, when N customers visit the shop, each with an average interarrival time of 5 hours, the total arrival rate as seen by the shop must be $N/5$.) As a second example, we extend the simulation to $n = 1000$ visits to the shop, see Figure 2.2. In the third example we take the interarrival times to be normally distributed times with mean 5 and $\sigma = 1$, see Figure 2.3.

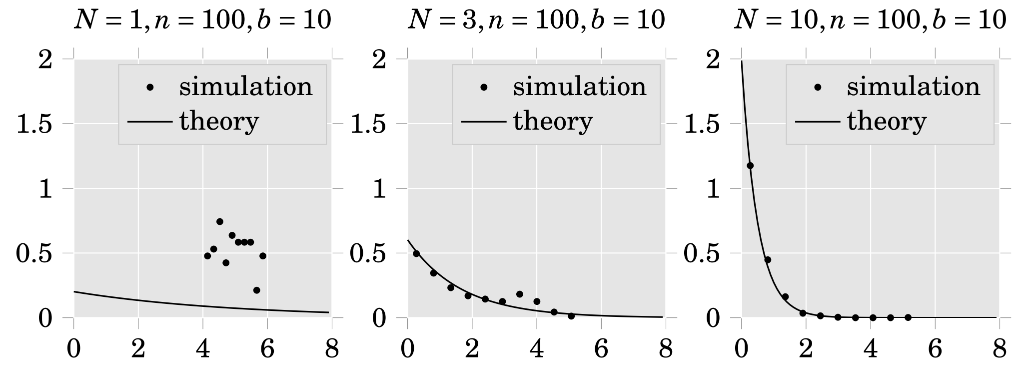


Figure 2.1: The interarrival process as seen by the shop owner. Observe that the density $\lambda e^{-\lambda x}$ intersects the y-axis at level $N/5$, which is equal to the arrival rate when N persons visit the shop. The parameter $n = 100$ is the simulation length, i.e., the number of visits per customer, and $b = 10$ is number of bins to collect the data.

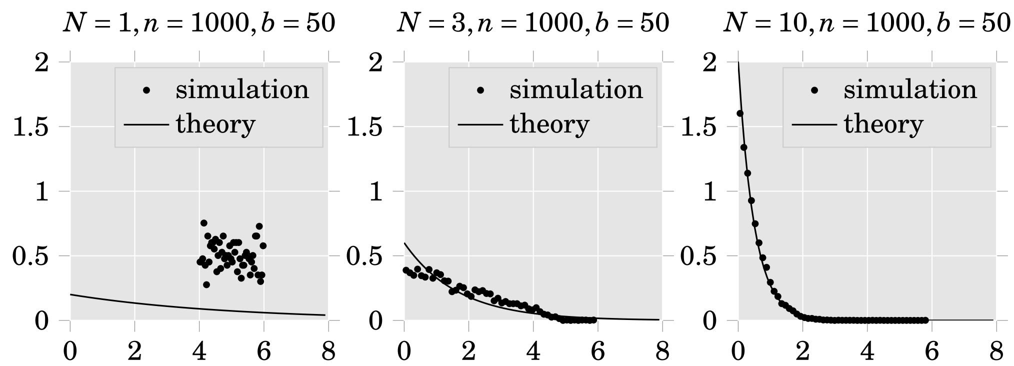


Figure 2.2: Each of the N customer visits the shop at uniformly distributed interarrival times, but now the number of visits is $n = 1000$.

Exercise 2.1.9. Try to make Figure 2.2 with simulation.

As the graphs show, even when the customer population consists of 10 members, each visiting the shop with an interarrival time that is quite 'far' from exponential, the distribution of the interarrival times as observed by the shop is very well approximated by an exponential

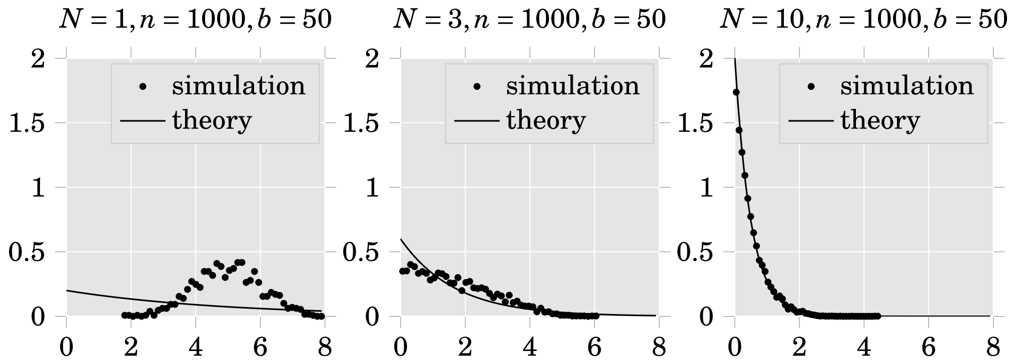


Figure 2.3: Each of the N customer visits the shop with normally distributed interarrival times with $\mu = 5$ and $\sigma = 1$.

distribution. Thus, for a real shop, with many thousands of customers, or a hospital, call center, in fact for nearly every system that deals with random demand, it seems reasonable to use the exponential distribution to model interarrival times. In conclusion, the main conditions to use an exponential distribution are: 1) arrivals have to be drawn from a large population, and 2) each of the arriving customers decides, independent of the others, to visit the system.

Another reason to use the exponential distribution is that an exponentially distributed random variable is *memoryless*, that is, it satisfies the property that

$$\mathsf{P}\{X > t + h | X > t\} = \mathsf{P}\{X > h\}.$$

In words, the probability that X is larger than some time $t + h$, conditional on it being larger than a time t, is equal to the probability that X is larger than h. Thus, no matter how long we have been waiting for the next arrival to occur, the probability that it will occur in the next h seconds remains the same. This property seems to be vindicated also in practice: suppose that a patient with a broken arm just arrived at the emergency room of a hospital, what does that tell us about the time the next patient will be brought in? Not much, as most of us will agree.

Exercise 2.1.10. (This exercise should help you recall the concept of conditional probability.) We have to give one present to one of three children. As we cannot divide the present into parts, we decide to let 'fate decide'. That is, we choose a random number in the set $\{1,2,3\}$. The first child that guesses the number wins the present. Show that the probability of winning the present is the same for each child.

Exercise 2.1.11. Show that an exponentially distributed random variable is memoryless.

Exercise 2.1.12. If $X \sim \exp(\lambda)$ and $S \sim \exp(\mu)$ and X and S are independent, show that

$$Z = \min\{X, S\} \sim \exp(\lambda + \mu),$$

hence $\mathsf{E}\{Z\} = (\lambda + \mu)^{-1}$.

Exercise 2.1.13. If $X \sim \exp(\lambda)$, $S \sim \exp(\mu)$ and independent, show that

$$\mathsf{P}\{X \le S\} = \frac{\lambda}{\lambda + \mu}.$$

Exercise 2.1.14. A machine serves two types of jobs. The processing time of jobs of type i, $i = 1,2$, is exponentially distributed with parameter μ_i. The type T of job is random and independent of anything else, and such that $\mathsf{P}\{T = 1\} = p = 1 - q = 1 - \mathsf{P}\{T = 2\}$. (An example is a desk serving men and women, both requiring different average service times, and p is the probability that the customer in service is a man.) Show that the expected processing time and variance are given by

$$\begin{aligned}\mathsf{E}\{X\} &= p\,\mathsf{E}\{X_1\} + q\,\mathsf{E}\{X_2\} \\ \mathsf{V}\{X\} &= p\,\mathsf{V}\{X_1\} + q\,\mathsf{V}\{X_2\} + pq(\mathsf{E}\{X_1\} - \mathsf{E}\{X_2\})^2.\end{aligned}$$

Interestingly, we see that even if $\mathsf{V}\{X_1\} = \mathsf{V}\{X_2\} = 0$, $\mathsf{V}\{X\} > 0$ if $\mathsf{E}\{X_1\} \neq \mathsf{E}\{X_2\}$. Bear this in mind; we will use these ideas later when we discuss the effects of failures on the variance of service times of jobs.

Hints

Hint 2.1.1:

$$\mathsf{E}\{X\} = \int_0^\infty t\,\mathrm{d}F(t) = \int_0^\infty t f(t)\,\mathrm{d}t = \int_0^\infty t\lambda e^{-\lambda t}\,\mathrm{d}t = \frac{1}{\lambda},$$

where $f(t) = \lambda e^{-\lambda t}$ is the density function of the distribution function F of X.

Hint 2.1.2:

$$\mathsf{E}\{X^2\} = \int_0^\infty t^2\lambda e^{-\lambda t}\,\mathrm{d}t = \frac{2}{\lambda^2}.$$

Hint 2.1.3: Use, and memorize, the very practical formula

$$\mathsf{V}\{X\} = \mathsf{E}\{X\}^2 - (\mathsf{E}\{X\})^2.$$

Hint 2.1.4:

$$M_X(t) = \mathsf{E}\{\exp(tX)\} = \int_0^\infty e^{tx} f(x)\,\mathrm{d}x.$$

Hint 2.1.7: Check the result for $i = 1$ by filling in $i = 1$ (just to be sure that you have read the formula right), and compare the result to the exponential density. Then write $A_i = \sum_{i=1}^k X_k$, and compute the moment generating function for A_i and use that the interarrival times X_i are independent. Use the moment generating function of X_i.

Hint 2.1.8: Use that $\int_0^\infty (\lambda t)^i e^{-\lambda t}\,\mathrm{d}t = \frac{i!}{\lambda}$. Another way would be to use that, once you have the moment generating function of some random variable X, $\mathsf{E}\{X\} = \frac{\mathrm{d}}{\mathrm{d}t} M(t)|_{t=0}$.

Hint 2.1.10: For the second child, condition on the event that the first does not chose the right number. Use the definition of conditional probability:

$$\mathsf{P}\{A|B\} = \frac{\mathsf{P}\{AB\}}{\mathsf{P}\{B\}},$$

provided $\mathsf{P}\{B\} > 0$.

Hint 2.1.11: Condition on the event $X > t$.

Hint 2.1.12: Use that if $Z = \min\{X,S\} > x$, it then must be that $X > x$ and $S > x$. Then use independence of X and S.

Hint 2.1.13: Define the joint distribution of X and S and carry out the computations, or use conditioning, or use the result of the previous exercise.

Hint 2.1.14: Let X be the processing (or service) time at the server, and X_i the service time of a type i job. Then,

$$X = \mathbb{1}_{T=1}X_1 + \mathbb{1}_{T=2}X_2,$$

where $\mathbb{1}$ is the indicator function, that is, $\mathbb{1}_A = 1$ if the event A is true, and $\mathbb{1}_A = 0$ if A is not true.

Solutions

Solution 2.1.1:

$$\begin{aligned}
\mathsf{E}\{X\} &= \int_0^\infty t\lambda e^{-\lambda t}\,\mathrm{d}t, \quad \text{density is } \lambda e^{-\lambda t} \\
&= \lambda^{-1}\int_0^\infty u e^{-u}\,\mathrm{d}u, \quad \text{by change of variable } u = \lambda t, \\
&= -\lambda^{-1}\, t e^{-t}\big|_0^\infty + \lambda^{-1}\int_0^\infty e^{-t}\,\mathrm{d}t \\
&= -\lambda^{-1}\, e^{-t}\big|_0^\infty = \frac{1}{\lambda}.
\end{aligned}$$

Solution 2.1.2:

$$\begin{aligned}
\mathsf{E}\left\{X^2\right\} &= \int_0^\infty t^2\lambda e^{-\lambda t}\,\mathrm{d}t \\
&= \lambda^{-2}\int_0^\infty u^2 e^{-u}\,\mathrm{d}u, \quad \text{by change of variable } u = \lambda t, \\
&= -\lambda^{-2}\, t^2 e^{-t}\big|_0^\infty + 2\lambda^{-2}\int_0^\infty t e^{-t}\,\mathrm{d}t \\
&= -2\lambda^{-2}\, t e^{-t}\big|_0^\infty + 2\lambda^{-2}\int_0^\infty e^{-t}\,\mathrm{d}t \\
&= -2\lambda^{-2}\, e^{-t}\big|_0^\infty \\
&= 2/\lambda^2.
\end{aligned}$$

Solution 2.1.3: By the previous problems, $\mathsf{E}\left\{X^2\right\} = 2/\lambda^2$ and $\mathsf{E}\{X\} = \lambda$.

Solution 2.1.4:

$$\begin{aligned}
M_X(t) = \mathsf{E}\{\exp(tX)\} &= \int_0^\infty e^{tx}\,\mathrm{d}F(x) = \int_0^\infty e^{tx} f(x)\,\mathrm{d}x = \int_0^\infty e^{tx}\lambda e^{-\lambda x}\,\mathrm{d}x \\
&= \lambda\int_0^\infty e^{(t-\lambda)x}\,\mathrm{d}x = \frac{\lambda}{\lambda - t}.
\end{aligned}$$

Solution 2.1.5: $M'(0) = \lambda/(\lambda - t)^2$. Hence, $M'(0) = 1/\lambda$. And, $M''(t) = 2\lambda/(\lambda - t)^3$, hence $\mathsf{E}\left\{X^2\right\} = M''(1) = 2\lambda/\lambda^3 = 2\lambda^{-2}$.

Solution 2.1.6: By the previous problems, $\mathsf{V}\{X\} = 1/\lambda^2$ and $\mathsf{E}\{X\} = 1/\lambda$.

Solution 2.1.7: One way to find the distribution of A_i is by using the moment generating function $M_{A_i}(t) = \mathsf{E}\left\{e^{tA_i}\right\}$ of A_i. Let X_i be the interarrival time between customers i and $i-1$, and $M_X(t)$ the associated moment generating function. Using the i.i.d. property of the $\{X_i\}$,

$$\begin{aligned} M_{A_i}(t) &= \mathsf{E}\left\{e^{tA_i}\right\} = \mathsf{E}\left\{\exp\left(t\sum_{j=1}^{i} X_j\right)\right\} \\ &= \prod_{j=1}^{i} \mathsf{E}\left\{e^{tX_j}\right\} = \prod_{j=1}^{i} M_{X_j}(t) = \prod_{j=1}^{i} \frac{\lambda}{\lambda - t} = \left(\frac{\lambda}{\lambda - t}\right)^i. \end{aligned}$$

From a table of moment generation functions it follows immediately that $A_i \sim \Gamma(i, \lambda)$, i.e., A_i is Gamma distributed.

Solution 2.1.8:

$$\mathsf{E}\{A_i\} = \int_0^\infty t f_{A_i}(t)\,\mathrm{d}t = \int_0^\infty t\lambda e^{-\lambda t}\frac{(\lambda t)^{i-1}}{(i-1)!}\,\mathrm{d}t.$$

Thus,

$$\mathsf{E}\{A_i\} = \frac{1}{(i-1)!}\int_0^\infty e^{-\lambda t}(\lambda t)^i\,\mathrm{d}t = \frac{i!}{(i-1)!\lambda} = \frac{i}{\lambda},$$

where we used the hint.

What if we would use the moment generating function, as derived by the previous exercise?

$$\begin{aligned} \mathsf{E}\{A_i\} &= \left.\frac{\mathrm{d}}{\mathrm{d}t} M_{A_i}(t)\right|_{t=0} \\ &= \left.\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\lambda}{\lambda - t}\right)^i\right|_{t=0} \\ &= \left. i\left(\frac{\lambda}{\lambda - t}\right)^{i-1} \frac{\lambda}{(\lambda - t)^2}\right|_{t=0} \\ &= \left.\frac{i}{\lambda}\left(\frac{\lambda}{\lambda - t}\right)^{i-1}\right|_{t=0} \\ &= \frac{i}{\lambda}. \end{aligned}$$

Finally, the simplest way to obtain the answer is to use that the expectation is a linear operator, i.e., $\mathsf{E}\{X+Y\} = \mathsf{E}\{X\} + \mathsf{E}\{Y\}$ for any r.v. X and Y. Then,

$$\mathsf{E}\{A_i\} = \mathsf{E}\left\{\sum_{k=1}^{i} X_k\right\} = i\,\mathsf{E}\{X\} = \frac{1}{\lambda}.$$

Solution 2.1.9: The source code can be found in `progs/converge_to_exp.py`.

Solution 2.1.10: The probability that the first child to guess also wins is 1/3. What is the probability for child number two? Well, for him/her to win, it is necessary that child one does not win and that child two guesses the right number of the remaining numbers. Assume, without loss of generality that child 1 chooses 3 and that this is not the right number. Then

$$\begin{aligned}
&\mathsf{P}\{\text{Child 2 wins}\}\\
&=\mathsf{P}\{\text{Child 2 guesses the right number and child 1 does not win}\}\\
&=\mathsf{P}\{\text{Child 2 guesses the right number}|\text{ child 1 does not win}\}\cdot\mathsf{P}\{\text{Child 1 does not win}\}\\
&=\mathsf{P}\{\text{Child 2 makes the right guess in the set }\{1,2\}\}\cdot\frac{2}{3}\\
&=\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}.
\end{aligned}$$

Similar conditional reasoning gives that child 3 wins with probability 1/3.

Solution 2.1.11: By the definition of conditional probability

$$\mathsf{P}\{X>t+h|X>t\}=\frac{\mathsf{P}\{X>t+h,X>t\}}{\mathsf{P}\{X>t\}}=\frac{\mathsf{P}\{X>t+h\}}{\mathsf{P}\{X>t\}}=\frac{e^{-\lambda(t+h)}}{e^{-\lambda t}}=e^{-\lambda h}=\mathsf{P}\{X>h\}.$$

As an aside, it can be shown that only exponential random variables have the memoryless property. The proof of this fact requires quite some work; we refer the reader to the literature if s/he want to check this, see e.g. Yushkevich and Dynkin [1969, Appendix 3].

Solution 2.1.12: Use that X and S are independent to get

$$\begin{aligned}
\mathsf{P}\{Z>x\}&=\mathsf{P}\{\min\{X,S\}>x\}=\mathsf{P}\{X>x\text{ and }S>x\}=\mathsf{P}\{X>x\}\mathsf{P}\{S>x\}\\
&=e^{-\lambda x}e^{-\mu x}=e^{-(\lambda+\mu)x}.
\end{aligned}$$

Solution 2.1.13: There is more than one way to show that $\mathsf{P}\{X\leq S\}=\lambda/(\lambda+\mu)$.

Method 1. Observe first that X and S, being exponentially distributed, both have a density. Moreover, as they are independent, the joint density takes the form

$$f_{X,S}(x,y)=f_X(x)f_S(y)=\lambda\mu e^{-\lambda x}e^{-\mu y}.$$

With this,

$$\begin{aligned}
\mathsf{P}\{X\leq S\}&=\mathsf{E}\{\mathbb{1}_{X\leq S}\}\\
&=\int_0^\infty\int_0^\infty \mathbb{1}_{x\leq y}f_{X,S}(x,y)\,\mathrm{d}y\,\mathrm{d}x\\
&=\lambda\mu\int_0^\infty\int_0^\infty \mathbb{1}_{x\leq y}e^{-\lambda x}e^{-\mu y}\,\mathrm{d}y\,\mathrm{d}x\\
&=\lambda\mu\int_0^\infty e^{-\mu y}\int_0^y e^{-\lambda x}\,\mathrm{d}x\,\mathrm{d}y\\
&=\mu\int_0^\infty e^{-\mu y}(1-e^{-\lambda y})\,\mathrm{d}y\\
&=\mu\int_0^\infty (e^{-\mu y}-e^{-(\lambda+\mu)y})\,\mathrm{d}y\\
&=\mu\int_0^\infty (e^{-\mu y}-e^{-(\lambda+\mu)y})\,\mathrm{d}y\\
&=1-\frac{\mu}{\lambda+\mu}
\end{aligned}$$

Method 2. Start with the law of total probability in the discrete case:

$$\mathsf{E}\{A\} = \sum_{i=1}^{k} \mathsf{E}\{A \,|\, B_i\}\,\mathsf{P}\{B_i\},$$

where $B_1, \ldots, B_k$ forms a set of events such that $\cup_{i=1}^{k} B_i$ is equal to the sample space, $B_i \cap B_j = \varnothing$ and $\mathsf{P}\{B_i\} > 0$. Now in Capiński and Zastawniak [2003] it is shown that this can be extended to random variables X and Y with a densities f_X, f_Y. The result is

$$\mathsf{E}\{Y\} = \int_0^\infty \mathsf{E}\{Y \,|\, X = x\}\, f_X(x)\,\mathrm{d}x.$$

Here, conceptually, $f_X(x)\,\mathrm{d}x \approx \mathsf{P}\{X \in \mathrm{d}x\}$ plays the role of $\mathsf{P}\{B_i\}$ in the discrete case above. With this notion, and observing that $f_S(s) = \mu e^{-\mu s}\,\mathrm{d}s$,

$$\mathsf{P}\{X \le S\} = \mathsf{E}\{\mathbb{1}_{X \le S}\} = \int_0^\infty \mathsf{E}\{\mathbb{1}_{X \le S} | S = s\}\, \mu e^{-\mu s}\,\mathrm{d}s.$$

Now, $\mathsf{E}\{\mathbb{1}_{X \le S} | S = s\}$ is a tricky object, as $\mathsf{P}\{S = s\} = 0$, so that $\mathsf{E}\{\mathbb{1}_{X \le S} \,| S = s\}$ *cannot* be defined as

$$\frac{\mathsf{E}\{\mathbb{1}_{X \le s}\}}{\mathsf{P}\{S = s\}}.$$

The way to proceed is to consider the conditional probability density function of X given that $S = s$, which is defined as

$$f_{X|S}(x|s) = \frac{f_{X,S}(x,s)}{f_Y(s)},$$

where, as before, $f_{X,S}(x,s)$ is the joint density of X and S. With this, we can properly define

$$\mathsf{P}\{X \le S | S = s\} = \mathsf{E}\{\mathbb{1}_{X \le S} | S = s\} = \int_0^\infty \mathbb{1}_{x \le s} f_{X|S}(x|s)\,\mathrm{d}x.$$

Using the definition of $f_{X|S}(x|s)$ and the independence of X and S it follows that

$$f_{X|S}(x|s) = \frac{f_{X,S}(x,s)}{f_Y(s)} = \frac{\lambda e^{-\lambda x} \mu e^{-\mu s}}{\mu e^{-\mu s}} = \lambda e^{-\lambda x}$$

from which we get that

$$\begin{aligned}
\mathsf{E}\{\mathbb{1}_{X \le S} | S = s\} &= \int_0^\infty \mathbb{1}_{x \le s} f_{X|S}(x|s)\,\mathrm{d}x \\
&= \int_0^\infty \mathbb{1}_{x \le s} \lambda e^{-\lambda x}\,\mathrm{d}x \\
&= \int_0^s \lambda e^{-\lambda x}\,\mathrm{d}x \\
&= 1 - e^{-\lambda s}.
\end{aligned}$$

And then, that is,

$$\begin{aligned}
\mathsf{P}\{X \le S\} &= \int \mathsf{P}\{X \le S | S = s\}\, \mu e^{-\mu s}\,\mathrm{d}s \\
&= \mu \int_0^\infty \mathsf{E}\{\mathbb{1}_{X \le S} | S = s\}\, e^{-\mu s}\,\mathrm{d}s = \mu \int_0^\infty (1 - e^{-\lambda s}) e^{-\mu s}\,\mathrm{d}s.
\end{aligned}$$

Straightforward integration gives the final result.

Solution 2.1.14: With the hint,

$$
\begin{aligned}
\mathsf{E}\{X\} &= \mathsf{E}\{\mathbb{1}_{T=1}X_1\} + \mathsf{E}\{\mathbb{1}_{T=2}X_2\} \\
&= \mathsf{E}\{\mathbb{1}_{T=1}\}\,\mathsf{E}\{X_1\} + \mathsf{E}\{\mathbb{1}_{T=2}\}\,\mathsf{E}\{X_2\}, \text{ by the independence of } T, \\
&= \mathsf{P}\{T=1\}/\mu_1 + \mathsf{P}\{T=2\}/\mu_2 \\
&= p/\mu_1 + q/\mu_2 \\
&= p\,\mathsf{E}\{X_1\} + q\,\mathsf{E}\{X_2\}.
\end{aligned}
$$

(The next derivation may seem a bit long, but the algebra is standard. I include all steps so that you don't have to use pen and paper yourself if you want to check the result.) Next, using that

$$\mathbb{1}_{T=1}\,\mathbb{1}_{T=2} = 0 \text{ and } \mathbb{1}_{T=1}^2 = \mathbb{1}_{T=1},$$

we get

$$
\begin{aligned}
\mathsf{V}\{X\} &= \mathsf{E}\left\{X^2\right\} - (\mathsf{E}\{X\})^2 \\
&= \mathsf{E}\left\{(\mathbb{1}_{T=1}X_1 + \mathbb{1}_{T=2}X_2)^2\right\} - \left(\frac{p}{\mu_1} + \frac{q}{\mu_2}\right)^2 \\
&= \mathsf{E}\left\{\mathbb{1}_{T=1}X_1^2 + \mathbb{1}_{T=2}X_2^2\right\} - \left(\frac{p}{\mu_1} + \frac{q}{\mu_2}\right)^2 \\
&= p\,\mathsf{E}\left\{X_1^2\right\} + q\,\mathsf{E}\left\{X_2^2\right\} - \left(\frac{p}{\mu_1} + \frac{q}{\mu_2}\right)^2 \\
&= p\,\mathsf{V}\{X_1\} + p(\mathsf{E}\{X_1\})^2 + q\,\mathsf{V}\{X_2\} + q(\mathsf{E}\{X_2\})^2 - \left(\frac{p}{\mu_1} + \frac{q}{\mu_2}\right)^2 \\
&= p\,\mathsf{V}\{X_1\} + \frac{p}{\mu_1^2} + q\,\mathsf{V}\{X_2\} + \frac{q}{\mu_2^2} - \left(\frac{p}{\mu_1} + \frac{q}{\mu_2}\right)^2 \\
&= p\,\mathsf{V}\{X_1\} + q\,\mathsf{V}\{X_2\} + \frac{p}{\mu_1^2} + \frac{q}{\mu_2^2} - \frac{p^2}{\mu_1^2} - \frac{q^2}{\mu_2^2} - \frac{2pq}{\mu_1\mu_2} \\
&= p\,\mathsf{V}\{X_1\} + q\,\mathsf{V}\{X_2\} + \frac{p(1-p)}{\mu_1^2} + \frac{q(1-q)}{\mu_2^2} - \frac{2pq}{\mu_1\mu_2} \\
&= p\,\mathsf{V}\{X_1\} + q\,\mathsf{V}\{X_2\} + \frac{pq}{\mu_1^2} + \frac{qp}{\mu_2^2} - \frac{2pq}{\mu_1\mu_2} \\
&= p\,\mathsf{V}\{X_1\} + q\,\mathsf{V}\{X_2\} + pq(\mathsf{E}\{X_1\} - \mathsf{E}\{X_2\})^2.
\end{aligned}
$$

2.2 Poisson Distribution

In this section we provide a derivation of, and motivation for, the Poisson process, and clarify its relation with the exponential distribution at the end.

Consider a machine that fails occasionally. Let us write $N(s,t)$ for the number of failures occurring during a time interval of $(s,t]$. We assume, without loss of generality, that repairs are instantaneous. Clearly, as we do not know in advance how often the machine will fail, we model $N(s,t)$ as a random variable for all times s and t.

Our first assumption is that the failure behavior of the machine does not significantly change over time. Then it is reasonable to assume that the expected number of failure is proportional to the length of the interval T. Thus, it is reasonable to assume that there exists some constant λ such that

$$\mathsf{E}\{N(s,t)\} = \lambda(t-s) \tag{2.2}$$

The constant λ is often called the *arrival rate*, or failure rate in this case.

The second assumption is that $\{N(s,t), s \leq t\}$ has *stationary* and *independent increments*. Stationarity means that the distribution of the number of arrivals are the same for all intervals of equal length. Formally, $N(s_1,t_1)$ has the same distribution as $N(s_2,t_2)$ if $t_2 - s_2 = t_1 - s_1$. Independence means, roughly speaking, that knowing that $N(s_1,t_1) = n$, does not help to make any predictions about $N(s_2,t_2)$ if the intervals (s_1,t_1) and (s_2,t_2) have no overlap.

To find the distribution of $N(0,t)$, let us split the interval $[0,t]$ into n sub-intervals, all of equal length, and ask: 'What is the probability that the machine will fail in some given sub-interval.' By our second assumption, the failure behavior is constant over time. Therefore, the probability p to fail in each interval should be equal. Moreover, if n is large, p must be small, for otherwise (2.2) could not be true. As a consequence, if the time intervals are very small, we can safely neglect the probability that two or more failures occur in one such tiny interval.

As a consequence, then, we can model the occurrence of a failure in some period i as a Bernoulli distributed random variable B_i such that $\mathsf{P}\{B_i = 1\}$ and $\mathsf{P}\{B_i = 0\} = 1 - \mathsf{P}\{B_i = 1\}$, and we assume that $\{B_i\}$ are independent. The total number of failures $N_n(t)$ that occur in n intervals is then binomially distributed

$$\mathsf{P}\{N_n(t) = k\} = \binom{n}{k} p^k (1-p)^{n-k}. \tag{2.3}$$

In the exercises we ask you to use this to motivate that, in some appropriate sense, $N_n(t)$ converges to $N(t)$ for $n \to \infty$ such that

$$\mathsf{P}\{N(t) = k\} = e^{-\lambda t} \frac{(\lambda t)^k}{k!}. \tag{2.4}$$

We say that $N(t)$ is *Poisson distributed* with rate λ, and write $N(t) \sim P(\lambda t)$.

Moreover, if there are no failures in some interval $[0,t]$, then it must be that $N(t) = 0$ and the interarrival time $X_1 = A_1 - 0$ (as $A_0 = 0$) must be larger than t. Therefore,

$$\mathsf{P}\{X_1 > t\} = \mathsf{P}\{N(t) = 0\} = e^{-\lambda t} \frac{(\lambda t)^0}{0!} = e^{-\lambda t}.$$

The relations we discussed above are of paramount importance in the analysis of queueing process. We summarize this by a theorem.

Theorem 2.2.1. A counting process $\{N(t)\}$ is a Poisson process with rate λ if and only if the inter-arrival times $\{X_i\}$, i.e., the times between consecutive arrivals, are i.i.d. and $\mathsf{P}\{X_1 \leq t\} = 1 - e^{-\lambda t}$. In other words, $X_i \sim \exp(\lambda) \Leftrightarrow N(t) \sim P(\lambda t)$

Exercises

Exercise 2.2.1. Show that $\mathsf{E}\{N_n(t)\} = \sum_{i=1}^{n} \mathsf{E}\{B_i\} = np$.

Exercise 2.2.2. What is the difference between $N_n(t)$ and $N(t)$?

Exercise 2.2.3. Show how the binomial distribution (2.3) converges to the Poisson distribution (2.4) if $n \to \infty$, $p \to 0$ such that $np = \lambda$.

Exercise 2.2.4. If the inter-arrival times $\{X_i\}$ are i.i.d. and exponentially distributed with mean $1/\lambda$, prove that the number $N(t)$ of arrivals during interval $[0,t]$ is Poisson distributed.

Exercise 2.2.5. Show that if $N(t) \sim P(\lambda t)$, we have for small h,

1. $\mathsf{P}\{N(h) = n \,|\, N(0) = n\} = 1 - \lambda h + o(h)$
2. $\mathsf{P}\{N(h) = n+1 \,|\, N(0) = n\} = \lambda h + o(h)$
3. $\mathsf{P}\{N(h) \geq n+2 \,|\, N(0) = n\} = o(h)$,

where $o(h)$ is a function $f(h)$ such $f(h)/h \to 0$ as $h \to 0$.

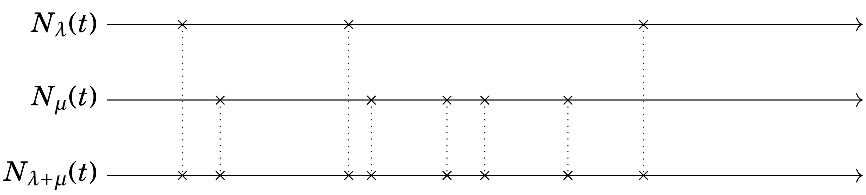
Exercise 2.2.6. Assume a timer fires at times $0 = T_0 < T_1 < T_2 < \cdots$, such that $T_k - T_{k-1} \sim \exp(\lambda)$. Define $N(t) = \sum_{k=0}^{\infty} k \, \mathbb{1}_{T_k \leq t < T_{k+1}}$, where $\mathbb{1}$ is the *indicator function*, that is, $\mathbb{1}_A = 1$ if the event A is true, and $\mathbb{1}_A = 0$ if A is not true. What is the distribution of $N(t)$?

Exercise 2.2.7. (I added this question to clarify the next question, you can skip it for year 16/17.) Show that for a Poisson process N,

$$\mathsf{P}\{N(0,s] = 1 \,|\, N(0,t] = 1\} = \frac{s}{t}.$$

Thus, if you know that $N(0,t] = 1$, the arrival is distributed uniformly on the interval $[0,t]$.

Exercise 2.2.8. (Merged Poisson streams form a new Poisson process with the sum of the rates.) Assume that $N_a(t) \sim \mathrm{P}(\lambda t)$, $N_s(t) \sim \mathrm{P}(\mu t)$ and independent. Show that $N_a(t) + N_s(t) \sim \mathrm{P}((\lambda + \mu)t)$. The figure below provides a graphical representation of merging (also called superposition) of streams.



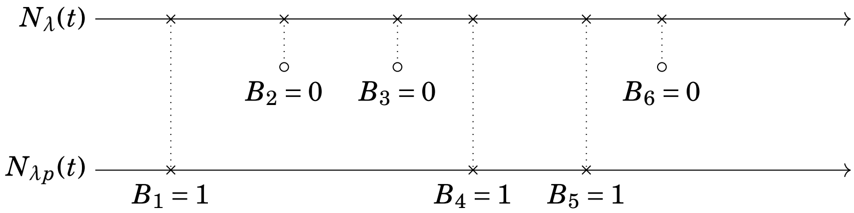
Exercise 2.2.9. Assume that $N_a(t) \sim \mathrm{P}(\lambda t)$, $N_s(t) \sim \mathrm{P}(\mu t)$ and independent. Use that $N_a(t) + N_s(t) \sim \mathrm{P}((\lambda + \mu)t)$ to conclude

$$\mathsf{P}\{N_a(h) = 1, N_s(h) = 0 | N_a(h) + N_s(h) = 1\} = \frac{\lambda}{\lambda + \mu}.$$

Note that the right-hand-side does not depend on h, hence it holds for any time h, whether it is small or not.

Exercise 2.2.10. Assume that $N_a(t) \sim \mathrm{P}(\lambda t)$, $N_s(t) \sim \mathrm{P}(\mu t)$ and independent. What is actually the meaning of the event $\{N_a(h) = 1, N_s(h) = 0\} \cap \{N_a(h) + N_s(h) = 1\}$?

Exercise 2.2.11. (A Bernoulli-thinned Poisson process is still a Poisson process) Consider a Poisson process. Split the process in the following way. When a job arrives, throw a coin that lands heads with probability p and tails with $q = 1 - p$. When the coin lands heads, call the job of type 1, otherwise of type 2. Another way of thinning is by modeling the stream of people passing a shop as a Poisson process with rate λ. With probability p a person decides, independent of anything else, to enter the shop, c.f. the figure below. The crosses at the upper line are passersby in the street. The crosses at the lower line are the customers that enter the shop. The outcome of the kth throw of a coin is indicated by B_k. Since $B_1 = 1$ the first passerby turns into a customer entering the shop; the second passerby does not enter as $B_2 = 0$, and so on. Like this, the stream of passerby's is thinned with Bernoulli distributed random variables.



The concept of thinning is particularly useful to analyze queueing networks. Suppose the departure stream of a machine is split into two substreams, e.g., a fraction p of the jobs move on another machine and the rest $(1 - p)$ of the jobs leave the system. Then we can model the arrival stream at the second machine as a thinned stream (with probability p) of the departures of the first machine.

Show that the Poisson process obtained by thinning the original process is $\sim P(\lambda t p)$ for any t.

Exercise 2.2.12. Show that $\mathsf{E}\{N(t)\} = \lambda t$ and $\mathsf{V}\{N(t)\} = \lambda t$. Why is the SCV of $N(t)$ equal to $1/\lambda t$? Conclude that the relative variability of $N(t)$ becomes smaller as t becomes larger. Observe that the SCV of a Poisson distributed random variable is not the same as the SCV of an exponentially distributed random variable.

Hints

Hint 2.2.1: Recall that $\mathsf{E}\{X + Y\} = \mathsf{E}\{X\} + \mathsf{E}\{Y\}$. Just as a reminder, $\mathsf{E}\{XY\} \neq \mathsf{E}\{X\}\mathsf{E}\{Y\}$ in general. Only when X and Y are uncorrelated (which is implied by independence), the product of the expectations is the expection of the products.

Hint 2.2.3: First find p, n, λ and t are such that the rate at which event occur in both processes are the same. Then consider the binomial distribution and use the standard limit $(1 - x/n)^n \to e^{-x}$ as $n \to \infty$.

Hint 2.2.4: Realize that $\mathsf{P}\{N(t) = k\} = \mathsf{P}\{A_k \leq t\} - \mathsf{P}\{A_{k+1} \leq t\}$.

Hint 2.2.5: Think about the meaning of the formula $\mathsf{P}\{N(h) = n \mid N(0) = n\}$. It is a conditional probability that should be read like this: given that up to time 0 we have seen n arrivals (i.e, $N(0) = n$), what is the probability that just a little later (h) the number of arrivals is still n, i.e, $N(h) = n$? Then use the definition of the Poisson distribtuion to compute this probability. Finally, use Taylor's expension of e^x to see that $e^x = 1 + x + o(x)$ for $|x| << 1$. Furthermore, use that $\sum_{i=2}^{\infty} x^i/i! = \sum_{i=0}^{\infty} x^i/i! - x - 1 = e^x - x - 1$.

Hint 2.2.7: Use Bayes' law for conditional probability. Observe that

$$\{N(0,s] + N(s,t] = 1\} \cap \{N(0,s] = 1\} = \{1 + N(s,t] = 1\} \cap \{N(0,s] = 1\} = \{N(s,t] = 0\} \cap \{N(0,s] = 1\}.$$

Hint 2.2.8: Use a conditioning argument or use probability generating functions (i.e. $\mathsf{E}\{z^X\} = \sum_k z^k \mathsf{P}\{X = k\}$. In particular, for conditioning, use that $\mathsf{P}\{AB\} = \mathsf{P}\{A \mid B\} \mathsf{P}\{B\}$. More generally, if the set A can be split into disjoint sets B_i, i.e, $A = \bigcup_{i=1}^n B_i$, then

$$\mathsf{P}\{A\} = \sum_{i=1}^n \mathsf{P}\{AB_i\} = \sum_{i=1}^n \mathsf{P}\{A \mid B_i\} \mathsf{P}\{B_i\},$$

where we use the conditioning formula to see that $\mathsf{P}\{AB_i\} = \mathsf{P}\{A \mid B_i\} \mathsf{P}\{B_i\}$. Now choose practical sets B_i.

Hint 2.2.9: Suppose we write $N(t) = N_a(t) + N_s(t)$. Then

$$\mathsf{P}\{N_a(h) = 1, N_s(h) = 0 | N(h) = 1\}$$

is the probability that $N_a(h) = 1$ and $N_s(h) = 0$ *given* that $N(t) = 1$. In other words, the question is find out that, given one of the two processes 'fired', what is the probability that N_a was the one that 'fired'.

Hint 2.2.11: Condition on the total number of arrivals $N(t) = m$ up to time t. Realize that the probability that a job is of type 1 is Bernoulli distributed, hence when you consider m jobs in total, the number of type 1 jobs is binomially distributed.

Again use that if the set A can be split into disjoint sets B_i, i.e, $A = \bigcup_{i=1}^n B_i$, then

$$\mathsf{P}\{A\} = \sum_{i=1}^n \mathsf{P}\{A \mid B_i\} \mathsf{P}\{B_i\}.$$

Now choose practical sets B_i.

You might also consider the random variable

$$Y = \sum_{i=1}^N Z_i,$$

with $N \sim P(\lambda)$ and $Z_i \sim B(p)$. Show that the moment generating function of Y is equal to the moment generating function of a Poisson random variable with parameter λp.

Hint 2.2.12: You might use generating functions here.

Solutions

Solution 2.2.1:

$$\mathsf{E}\{N_n(t)\} = \mathsf{E}\left\{\sum_{i=1}^{n} B_i\right\} = \sum_{i=1}^{n} \mathsf{E}\{B_i\} = n\,\mathsf{E}\{B_i\} = np.$$

Solution 2.2.2: $N_n(t)$ is a binomially distributed random variable with parameters n and p. The maximum value of $N_n(t)$ is n. The random variable $N(t)$ models the number of failures that can occur during $[0,t]$. As such it is not necessarily bounded by n. Thus, $N_n(t)$ and $N(t)$ cannot represent the same random variable.

Solution 2.2.3: Now we like to relate $N_n(t)$ and $N(t)$. It is clear that we at least want the expectations to be the same, that is, $np = \lambda t$. This implies that

$$p = \frac{\lambda t}{n},$$

so that

$$\mathsf{P}\{N_n(t) = k\} = \binom{n}{k}\left(\frac{\lambda t}{n}\right)^k \left(1 - \frac{\lambda t}{n}\right)^{n-k}.$$

To see that

$$\lim_{n\to\infty} \binom{n}{k}\left(\frac{\lambda t}{n}\right)^k \left(1 - \frac{\lambda t}{n}\right)^{n-k} = e^{-\lambda t}\frac{(\lambda t)^k}{k!}.$$

use that

$$\begin{aligned}
\binom{n}{k}\left(\frac{\lambda t}{n}\right)^k \left(1 - \frac{\lambda t}{n}\right)^{n-k} &= \frac{n!}{k!(n-k)!}\left(\frac{\lambda t}{n}\frac{n}{n-\lambda t}\right)^k \left(1-\frac{\lambda t}{n}\right)^n \\
&= \frac{(\lambda t)^k}{k!}\left(\frac{n}{n-\lambda t}\right)^k \frac{n!}{n^k(n-k)!}\left(1-\frac{\lambda t}{n}\right)^n \\
&= \frac{(\lambda t)^k}{k!}\left(\frac{n}{n-\lambda t}\right)^k \frac{n}{n}\frac{n-1}{n}\cdots\frac{n-k+1}{n}\left(1-\frac{\lambda t}{n}\right)^n.
\end{aligned}$$

Observe now that, as λt is finite, $n/(n-\lambda t) \to 1$ as $n \to \infty$. Also for any finite k, $(n-k)/n \to 1$. Finally, we use that $(1+x/n)^n \to e^x$ so that , $\left(1-\frac{\lambda t}{n}\right)^n \to e^{-\lambda t}$. The rest is easy, so that, as $n \to \infty$, the above converges to

$$\frac{(\lambda t)^k}{k!}e^{-\lambda t}.$$

Solution 2.2.4: We want to show that

$$\mathsf{P}\{N(t) = k\} = e^{-\lambda t}\frac{(\lambda t)^k}{k!}.$$

Now observe that $\mathsf{P}\{N(t) = k\} = \mathsf{P}\{A_k \le t\} - \mathsf{P}\{A_{k+1} \le t\}$. Using the density of A_{k+1} as obtained previously and applying partial integration leads to

$$\begin{aligned}
\mathsf{P}\{A_{k+1} \le t\} &= \lambda \int_0^t \frac{(\lambda s)^k}{k!}e^{-\lambda s}\,\mathrm{d}s \\
&= \lambda \frac{(\lambda s)^k}{k!}\frac{e^{-\lambda s}}{-\lambda}\bigg|_0^t + \lambda\int_0^t \frac{(\lambda s)^{k-1}}{(k-1)!}e^{-\lambda s}\,\mathrm{d}s \\
&= -\frac{(\lambda t)^k}{k!}e^{-\lambda t} + \mathsf{P}\{A_k \le t\}
\end{aligned}$$

We are done.

Solution 2.2.5: 1. $\mathsf{P}\{N(h) = n | N(0) = n\} = \mathsf{P}\{N(h) = 0\} = e^{-\lambda h}(\lambda h)^0/0! = e^{-\lambda h} = 1 - \lambda h + o(h)$, as follows from a standard argument in analysis that $e^{-\lambda h} = 1 - \lambda h + o(h)$ for h small.

2. $\mathsf{P}\{N(h) = n+1 | N(0) = n\} = \mathsf{P}\{N(h) = 1\} = e^{-\lambda h}(\lambda h)^1/1! = (1 - \lambda h + o(h))\lambda h = \lambda h - \lambda^2 h^2 + o(h) = \lambda h + o(h)$.

3.

$$\begin{aligned}
\mathsf{P}\{N(h) \ge n+2 | N(0) = n\} &= \mathsf{P}\{N(h) \ge 2\} \\
&= e^{-\lambda h} \sum_{i=2}^{\infty} \frac{(\lambda h)^i}{i!} = e^{-\lambda h} \left(\sum_{i=0}^{\infty} \frac{(\lambda h)^i}{i!} - \lambda h - 1 \right) \\
&= e^{-\lambda h}(e^{\lambda h} - 1 - \lambda h) = 1 - e^{-\lambda h}(1 + \lambda h) \\
&= 1 - (1 - \lambda h + o(h))(1 + \lambda h) = 1 - (1 - \lambda^2 h^2 + o(h)) \\
&= \lambda^2 h^2 + o(h) = o(h).
\end{aligned}$$

We can also use the results of the previous parts to see that

$$\begin{aligned}
\mathsf{P}\{N(h) \ge n+2 | N(0) = n\} &= \mathsf{P}\{N(h) \ge 2\} = 1 - \mathsf{P}\{N(h) < 2\} \\
&= 1 - \mathsf{P}\{N(h) = 0\} - \mathsf{P}\{N(h) = 1\} \\
&= 1 - (1 - \lambda h) + o(h)) - (\lambda h + o(h)) \\
&= o(h).
\end{aligned}$$

Solution 2.2.6: $N(t) \sim \mathrm{P}(\lambda t)$.

Solution 2.2.7:

$$\begin{aligned}
\mathsf{P}\{N(0,s] = 1 \,|\, N(0,t] = 1\} &= \frac{\mathsf{P}\{N(0,s] = 1, N(0,t] = 1\}}{\mathsf{P}\{N(0,t] = 1\}} \\
&= \mathsf{P}\{N(0,t] = 1 \,|\, N(0,s] = 1\} \frac{\mathsf{P}\{N(0,s] = 1\}}{\mathsf{P}\{N(0,t] = 1\}} \\
&= \mathsf{P}\{N(0,s] + N(s,t] = 1 \,|\, N(0,s] = 1\} \frac{e^{-\lambda s}\lambda s}{e^{-\lambda t}\lambda t} \\
&= \mathsf{P}\{1 + N(s,t] = 1 \,|\, N(0,s] = 1\} e^{-\lambda(s-t)} \frac{s}{t} \\
&= \mathsf{P}\{N(s,t] = 0\} e^{-\lambda(s-t)} \frac{s}{t} = e^{-\lambda(t-s)} e^{-\lambda(s-t)} \frac{s}{t} = \frac{s}{t}.
\end{aligned}$$

Solution 2.2.8: First we show how to use conditioning.

$$\mathsf{P}\{N_a(t) + N_s(t) = n\} = \sum_{i=0}^{n} \mathsf{P}\{N_a(t) + N_s(t) = n | N_a(t) = i\} \mathsf{P}\{N_a(t) = i\}$$

Now, if $N_a(t) = i$, then

$$N_a(t) + N_s(t) = n \iff i + N_s(t) = n \iff N_s(t) = n - i.$$

Thus,

$$
\begin{aligned}
\mathsf{P}\{N_a(t)+N_s(t)=n\} &= \sum_{i=0}^{n} \mathsf{P}\{N_s(t)=n-i\}\,\mathsf{P}\{N_a(t)=i\} \\
&= \sum_{i=0}^{n} \frac{(\mu t)^{n-i}}{(n-i)!}\frac{(\lambda t)^i}{i!}e^{-(\mu+\lambda)t} \\
&= e^{-(\mu+\lambda)t}\sum_{i=0}^{n} \frac{(\mu t)^{n-i}}{(n-i)!}\frac{(\lambda t)^i}{i!} \\
&= e^{-(\mu+\lambda)t}\frac{1}{n!}\sum_{i=0}^{n}\binom{n}{i}(\mu t)^{n-i}(\lambda t)^i \\
&= \frac{((\mu+\lambda)t))^n}{n!}e^{-(\mu+\lambda)t}.
\end{aligned}
\tag{2.5}
$$

Now with the probability generating functions.

$$
\begin{aligned}
M_a(z) = \mathsf{E}\left\{z^{N_a(t)}\right\} &= \sum_{k=0}^{\infty} z^k\,\mathsf{P}\{N_a(t)=k\} \\
&= \sum_{k=0}^{\infty} z^k e^{-\lambda t}\frac{(\lambda t)^k}{k!} \\
&= e^{-\lambda t}\sum_{k=0}^{\infty}\frac{(z\lambda t)^k}{k!} \\
&= \exp(\lambda t(z-1)).
\end{aligned}
$$

Use this, and the similar expression for $N_s(t)$ and independence to see that

$$
M(z) = \mathsf{E}\left\{z^{N(t)}\right\} = \mathsf{E}\left\{z^{N_a(t)}\right\}\mathsf{E}\left\{z^{N_s(t)}\right\} = \exp((\lambda+\lambda)t(z-1)).
$$

Finally, since the generating function uniquely characterizes the distribution, and since the above expression has the same form as $M_1(z)$ but with $\lambda+\mu$ replacing λ, we can conclude that $N(t) \sim P((\lambda+\mu)t)$.

Solution 2.2.9: With the above:

$$
\begin{aligned}
&\mathsf{P}\{N_a(h)=1, N_s(h)=0 | N_a(h)+N_s(h)=1\} \\
&= \frac{\mathsf{P}\{N_a(h)=1, N_s(h)=0, N_a(h)+N_s(h)=1\}}{\mathsf{P}\{N_a(h)+N_s(h)=1\}} \\
&= \frac{\mathsf{P}\{N_a(h)=1, N_s(h)=0\}}{\mathsf{P}\{N_a(h)+N_s(h)=1\}} \\
&= \frac{\mathsf{P}\{N_a(h)=1\}\,\mathsf{P}\{N_s(h)=0\}}{\mathsf{P}\{N_a(h)+N_s(h)=1\}} \\
&= \frac{\lambda h \exp -\lambda h \exp -\mu h}{((\lambda+\mu)h)\exp(-(\lambda+\mu)h)} \\
&= \frac{\lambda h \exp(-(\lambda+\mu)h)}{((\lambda+\mu)h)\exp(-(\lambda+\mu)h)} \\
&= \frac{\lambda}{\lambda+\mu}
\end{aligned}
$$

Solution 2.2.10: This means that, given that an event occurred, the event was an arrival, i.e., N_a was the first.

Solution 2.2.11: Suppose that N_1 is the thinned stream, and N the total stream. Then

$$\begin{aligned}
\mathsf{P}\{N_1 = k\} &= \sum_{n=k}^{\infty} \mathsf{P}\{N_1 = k, N = n\} = \sum_{n=k}^{\infty} \mathsf{P}\{N_1 = k \,|\, N = n\} \mathsf{P}\{N = n\} \\
&= \sum_{n=k}^{\infty} \mathsf{P}\{N_1 = k \,|\, N = n\} e^{-\lambda} \frac{\lambda^n}{n!} = \sum_{n=k}^{\infty} \binom{n}{k} p^k (1-p)^{n-k} e^{-\lambda} \frac{\lambda^n}{n!} \\
&= e^{-\lambda} \sum_{n=k}^{\infty} \frac{p^k (1-p)^{n-k}}{k!(n-k)!} \lambda^n = e^{-\lambda} \frac{(\lambda p)^k}{k!} \sum_{n=k}^{\infty} \frac{(\lambda(1-p))^{n-k}}{(n-k)!} \\
&= e^{-\lambda} \frac{(\lambda p)^k}{k!} \sum_{n=0}^{\infty} \frac{(\lambda(1-p))^n}{n!} = e^{-\lambda} \frac{(\lambda p)^k}{k!} e^{\lambda(1-p)} \\
&= e^{-\lambda p} \frac{(\lambda p)^k}{k!}.
\end{aligned}$$

We see that the thinned stream is Poisson with parameter λp. (For notational ease, we left out the t, otherwise it is $P(\lambda t p)$.

Now consider $Y = \sum_{i=1}^{N} Z_i$. Suppose that $N = n$, so that n arrivals occurred. Then we throw n coins with success probability p. It follows that Y is indeed a thinned Poisson random variable. Model the coins as a generic Bernoulli distributed random variable Z. We first need

$$\mathsf{E}\left\{e^{sZ}\right\} = e^0 \mathsf{P}\{Z = 0\} + e^s \mathsf{P}\{Z = 1\} = (1-p) + e^s p.$$

Suppose that $N = n$, then since the Z_i are i.i.d.,

$$\mathsf{E}\left\{e^{s\sum_{i=1}^n Z_i}\right\} = \left(\mathsf{E}\left\{e^{sZ}\right\}\right)^n = \left(1 + p(e^s - 1)\right)^n$$

Then, using conditioning on N,

$$\begin{aligned}
\mathsf{E}\left\{e^{sY}\right\} &= \mathsf{E}\left\{\mathsf{E}\left\{e^{s\sum_{i=1}^n Z_i} \,\middle|\, N = n\right\}\right\} = \mathsf{E}\left\{\mathsf{E}\left\{(1 + p(e^s - 1))^n \,|\, N = n\right\}\right\} \\
&= \sum_{n=0}^{\infty} \left(1 + p(e^s - 1)\right)^n e^{-\lambda} \frac{\lambda^n}{n!} = e^{-\lambda} \sum_{n=0}^{\infty} \frac{(1 + p(e^s - 1))^n \lambda^n}{n!} \\
&= e^{-\lambda} e^{\lambda(1 + p(e^s - 1))} = e^{\lambda p(e^s - 1)}.
\end{aligned}$$

Thus, Y has the same moment generating function as a Poisson distributed random variable with parameter λp. Since moment-generating functions specify the distribution uniquely, $Y \sim P(\lambda p)$.

Solution 2.2.12: I expect that this is easy for you by now. The exercise is just meant to help you recall this.

When a random variable N is Poisson distributed with parameter λ,

$$\begin{aligned}
\mathsf{E}\{N\} &= \sum_{n=0}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!} \\
&= \sum_{n=1}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!}, \text{ since the term with } n=0 \text{ cannot contribute} \\
&= e^{-\lambda}\lambda \sum_{n=1}^{\infty} \frac{\lambda^{n-1}}{(n-1)!} \\
&= e^{-\lambda}\lambda \sum_{n=0}^{\infty} \frac{\lambda^n}{n!}, \text{ by a change of variable} \\
&= e^{-\lambda}\lambda e^{\lambda} \\
&= \lambda.
\end{aligned}$$

Similarly, using that $\mathsf{V}\{N\} = \mathsf{E}\{N^2\} - (\mathsf{E}\{N\})^2$,

$$\begin{aligned}
\mathsf{E}\{N^2\} &= \sum_{n=0}^{\infty} n^2 e^{-\lambda} \frac{\lambda^n}{n!} \\
&= e^{-\lambda} \sum_{n=1}^{\infty} n \frac{\lambda^n}{(n-1)!} \\
&= e^{-\lambda} \sum_{n=0}^{\infty} (n+1) \frac{\lambda^{n+1}}{n!} \\
&= e^{-\lambda}\lambda \sum_{n=0}^{\infty} n \frac{\lambda^n}{n!} + e^{-\lambda}\lambda \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} \\
&= \lambda^2 + \lambda.
\end{aligned}$$

The Poisson *process* $\{N(t)\}$ is a much more complicated object than the Poisson distributed random variable $N(t)$. The process contains it is an *uncountable set* of random variables, whereas $N(t)$ is just *one* random variable. However, for a fixed t the element $N(t)$ is a random variable that is Poisson distributed with parameter λt. Using the above, the answer of the question follows immediately.

With generating functions we get the result without too much effort. Suppose N is a Poisson distributed random variable. Then

$$\begin{aligned}
\phi(z) &= \mathsf{E}\left\{z^N\right\} \\
&= \sum_{k=0}^{\infty} z^k \mathsf{P}\{N=k\} \\
&= \sum_{k=0}^{\infty} z^k \frac{\lambda^k}{k!} e^{-\lambda} \\
&= e^{-\lambda} \sum_{k=0}^{\infty} \frac{(z\lambda^k}{k!} \\
&= e^{-\lambda} e^{z\lambda} \\
&= e^{(z-1)\lambda}.
\end{aligned}$$

Observe that $\phi'(z) = \mathsf{E}\left\{N z^N\right\}$, so that $\phi'(1) = \mathsf{E}\{N\}$; also $\phi''(z) = \mathsf{E}\left\{N(N-1)z^{N-2}\right\}$, hence $\phi''(1) = \mathsf{E}\{N(N-1)\}$. With this,

$$\begin{aligned}
\mathsf{E}\{N\} &= \phi'(1) = \lambda, \\
\mathsf{E}\left\{N^2\right\} &= \phi''(1) + \mathsf{E}\{N\} = \lambda^2 + \lambda, \\
\mathsf{V}\{N\} &= \mathsf{E}\left\{N^2\right\} - (\mathsf{E}\{N\})^2 = \lambda, \\
SCV &= \frac{\mathsf{V}\{N(t)\}}{(\mathsf{E}\{N(t)\})^2} = \frac{\lambda t}{(\lambda t)^2} = \frac{1}{(\lambda t)^2}.
\end{aligned}$$

Clearly, as t increases, $1/\lambda t$ decreases.

Here is a point of confusion for some students. The SCV of an exponentially distributed random variable is 1; the SCV of the related Poisson process $N_\lambda(t)$ is *not* identically 1 for all t.

2.3 Kendall's Notation to Characterize Queueing Processes

As will become apparent in Sections 2.4 and 2.5, the construction of any queueing process involves three main elements: the distribution of the interarrival times between consecutive jobs, the distribution of the service times of the individual jobs, and the number of servers present to process jobs. In this characterization it is implicit that the interarrival times form a set of i.i.d. (independent and identically distributed) random variables, the service times are also i.i.d., and finally, the interarrival times and service times are mutually independent.

To characterize the type of queueing process it is common to use the *abbreviation* $A/B/c/K$ where A is the distribution of the interarrival times, B the distribution of the services, c the number of servers, and K the size of the system. In this notation it is assumed that jobs are served in first-in-first-out (FIFO) order; FIFO scheduling is also often called first-come-first-serve (FCFS).

Let us illustrate the shorthand $A/B/c/K$ with some examples:

- $M/M/1$: the distribution of the interarrival times is Memory-less, hence exponential, the service times are also Memoryless, and there is 1 server. As K is unspecified, it is assumed to be infinite.

- $M/M/c$: A *multi-server* queue with c servers in which all servers have the same capacity. Jobs arrive according to a Poisson process and have exponentially distributed processing times.

- $M(n)/M(n)/1$: the interarrival times are exponential, just as the service times, but the rates of the arrival and service processes may depend on the queue length n.

- $M/M/c/K$: interarrival times and process times are exponential, and the *system capacity* is K jobs. Thus, the queue can contain at most $K-c$ jobs.[2]

- $M/M/c/c$: in this system the number of servers is the same as the system capacity, thus the queue length is always zero. This queueing system is useful to model the determine the number of beds in a hospital; the beds act as servers.

[2]Sometimes, the K stands for the capacity in the queue, not the entire system. The notation differs among authors. Due to the lack of consistency, I might also not be consistent; be warned.

- $M^X/M/1$: Customers arrive with exponentially distributed interarrival times. However, each customer brings in a number of jobs, known as a batch. The number of jobs in each batch is distributed as the random variable X. Thus, the arrival process of work is *compound Poisson*.
- $M/G/1$: the interarrival times are exponentially distributed, the service times can have any *G*eneral distribution (with finite mean), and there is 1 server.
- $M/G/\infty$: exponential interarrival times, service times can have any distribution, and there is an unlimited supply of servers. This is also known as an *ample* server. Observe that in this queueing process, jobs actually never have wait in queue; upon arrival there is always a free server available.
- $M/D/1-LIFO$. Now job service times are *D*eterministic, and the service sequence is last-in-first-out (LIFO).
- $G/G/1$: generally distributed interarrival and service times, 1 server.

In the sequel we will use Kendall's notation frequently to distinguish the different queueing models. Ensure that you familiarize yourself with this notation.

Exercises

Exercise 2.3.1. What are some advantages and disadvantages of using the Shortest Processing Time First (SPTF) rule to serve jobs?

Hints

Hint 2.3.1: Look up the relevant definitions on wikipedia or Hall [1991]

Solutions

Solution 2.3.1:

- Advantage: SPTF minimizes the number of jobs in queue. Thus, if you want to keep the shop floor free of jobs (Work In Progress, WIP), then this is certainly a good rule.
- Disadvantage: large jobs get near to terrible waiting times, and the variance of the waiting time increases. Thus, the C_s^2 is larger than under FIFO. Also, SPTF does not take duedates into account, thus giving a reliable duedate quotation to a customer is hard (near to impossible.)

2.4 Construction of Discrete-Time Queueing Processes

In this section we discuss a case as this provides real-life motivation to analyze queueing systems. After describing the case, we we develop a set of recursions by which we can construct queueing systems in discrete time. As it turns out, this case is too hard to analyze by mathematical means, so that simulation is the best way forward. Interestingly, the structure of the simulation is very simple. As such, simulation is an exceedingly convincing tool to communicate the results of an analysis of a queueing system to managers (and the like).

At a mental health department five psychiatrists do intakes of future patients to determine the best treatment process for the patients. There are complaints about the time patients have to wait for their first intake; the desired waiting time is around two weeks, but the realized waiting time is sometimes more than three months. The organization considers this is to be unacceptably long, but... what to do about it?

To reduce the waiting times the five psychiatrists have various suggestions.

1. Not all psychiatrists have the same amount of time available per week to do intakes. This is not a problem during weeks that all are present. However, psychiatrists tend to take holidays, visit conferences, and so on. So, if the psychiatrist with the most intakes per week would go on leave, this might affect the behavior of the queue length considerably. This raises the question about the difference in allocation of capacity allotted to the psychiatrists. What are the consequences on the distribution and average of the waiting times if they would all have the same weekly capacity?

2. The psychiatrists tend to plan their holidays after each other, to reduce the variation in the service capacity. What if they would synchronize their holidays, to the extent possible, rather than spread their holidays?

3. Finally, suppose the psychiatrists would do 2 more intakes per week in busy times and 2 less in quiet weeks. Assuming that the system is stable, i.e., the service capacity exceeds the demand, then on average the psychiatrists would not do more intakes, i.e., their workload would not increase, but the queue length may be controlled better.

To evaluate the effect of these suggestions on reducing the queueing dynamics we develop a simple simulator and a number of plots. The simulation of a queueing system involves the specification of its behavior over time. The easiest method to construct queueing processes, hence to develop simulations, is to 'chop up' time in periods and develop recursions for the behavior of the queue from period to period. Note that the length of such a period depends on the case for which the model is developed. For instance, to study queueing processes at a supermarket, a period can consist of 5 minutes, while for a production environment, e.g., a job shop, it can be a day, or even a week.

Using fixed sized periods has it advantages as it does not require to specify specific inter-arrival times or service times of individual customers. Only the number of arrivals in a period and the number of potential services need to be specified, which is useful since in many practical settings, e.g., production environments, it is easier to provide data in these terms then in terms of inter-arrival and service times. It is, however, necessary to make some careful choices about timing.

Let us *define*

$$\begin{aligned} a_k &= \text{number of jobs that arrive at period } k, \\ c_k &= \text{number of jobs that can be served during period } k, \\ d_k &= \text{number of jobs that depart the queue } \textit{during} \text{ period } k, \\ Q_k &= \text{number of jobs in queue at the } \textit{end} \text{ of period } k. \end{aligned} \tag{2.6}$$

In the sequel we also call a_k the *batch* of job arrivals in period k. The definition of a_k is a bit subtle: we may assume that the arriving jobs arrive either at the start or at the end of the period. In the first case, the jobs can be served during period k, in the latter case, they *cannot* be served during period k.

Since Q_{k-1} is the queue length at the end of period $k-1$, it must also be the queue length at the start of period k. Assuming that jobs that arrive in period k cannot be served in period k, the number of customers that can depart from the queue in period k is

$$d_k = \min\{Q_{k-1}, c_k\}, \tag{2.7a}$$

since only the jobs that are present at the start of the period, i.e, Q_{k-1}, can be served if the capacity exceeds the queue length. Now that we know the number of departures, the queue at the end of period k is given by

$$Q_k = Q_{k-1} - d_k + a_k. \tag{2.7b}$$

As an example, suppose that $c_k = 7$ for all k, and $a_1 = 5$, $a_2 = 4$ and $a_3 = 9$; also $Q_0 = 8$. Then, $d_1 = 7$, $Q_1 = 8-7+5 = 6$, $d_2 = 6$, $Q_2 = 6-6+4 = 4$, $d_3 = 4$, $Q_3 = 4-4+9 = 9$, and so on.

Of course we are not going to carry out these computations by hand. Typically we use company data about the sequence of arrivals $\{a_k\}_{k=1,2,\ldots}$ and the capacity $\{c_k\}_{k=1,\ldots}$ and feed this data into a computer to compute the recursions (2.7). If we do not have sufficient data we make a probability model for these data and use the computer to generate random numbers with, hopefully, similar characteristics as the real data. At any rate, from this point on we assume that it is easy, by means of computers, to obtain numbers $a_1, \ldots, a_n$ for $n \gg 1000$, and so on.

We now show how to adapt (2.6) and (2.7) to the case introduced above.

As a first step we model the arrival process of patients as a Poisson process, c.f., Section 2.2. The duration of a period is taken to be a week. The average number of arrivals per period, based on data of the company, was slightly less than 12 per week; in the simulation we set it to $\lambda = 11.8$ per week. We model the capacity in the form of a matrix such that row i corresponds to the weekly capacity of psychiatrist i:

$$C = \begin{pmatrix} 1 & 1 & 1 & \ldots \\ 1 & 1 & 1 & \ldots \\ 1 & 1 & 1 & \ldots \\ 3 & 3 & 3 & \ldots \\ 9 & 9 & 9 & \ldots \end{pmatrix}$$

Thus, psychiatrists 1, 2, and 3 do just one intake per week, the fourth does 3, and the fifth does 9 intakes per week. The sum over column k is the total service capacity for week k of all psychiatrists together.

With the matrix C it is simple to make other capacity schemes. A more balanced scheme would be like this:

$$C = \begin{pmatrix} 2 & 2 & 2 & \ldots \\ 2 & 2 & 2 & \ldots \\ 3 & 3 & 3 & \ldots \\ 4 & 4 & 4 & \ldots \\ 4 & 4 & 4 & \ldots \end{pmatrix},$$

We next include the effects of holidays on the capacity. This is easily done by setting the capacity of a certain psychiatrist to 0 in a certain week. Let's assume that just one psychiatrist is on leave in a week, each psychiatrist has one week per five weeks off, and the psychiatrists' holiday schemes rotate. To model this, we set $C_{1,1,} = C_{2,2} = \cdots = C_{1,6} = C_{2,7} = \cdots = 0$, i.e.,

$$C = \begin{pmatrix} 0 & 2 & 2 & 2 & 2 & 0 & \dots \\ 2 & 0 & 2 & 2 & 2 & 2 & \dots \\ 3 & 3 & 0 & 3 & 3 & 3 & \dots \\ 4 & 4 & 4 & 0 & 4 & 4 & \dots \\ 4 & 4 & 4 & 4 & 0 & 4 & \dots \end{pmatrix},$$

Hence, the total average capacity must be $4/5 \cdot (2+2+3+4+4) = 12$ patients per week. The other holiday scheme—all psychiatrists take holiday in the same week–corresponds to setting entire columns to zero, i.e., $C_{i,5} = C_{i,10} = \cdots = 0$ for week 5, 10, and so on. Note that all these variations in holiday schemes result in the same average capacity.

Now that we have modeled the arrivals and the capacities, we can use the recursions (2.7) to simulate the queue length process for the four different scenarios proposed by the psychiatrists, unbalanced versus balanced capacity, and spread out holidays versus simultaneous holidays. The results are shown in Figure 2.4. It is apparent that suggestions 1 and 2 above do not significantly affect the behavior of the queue length process.

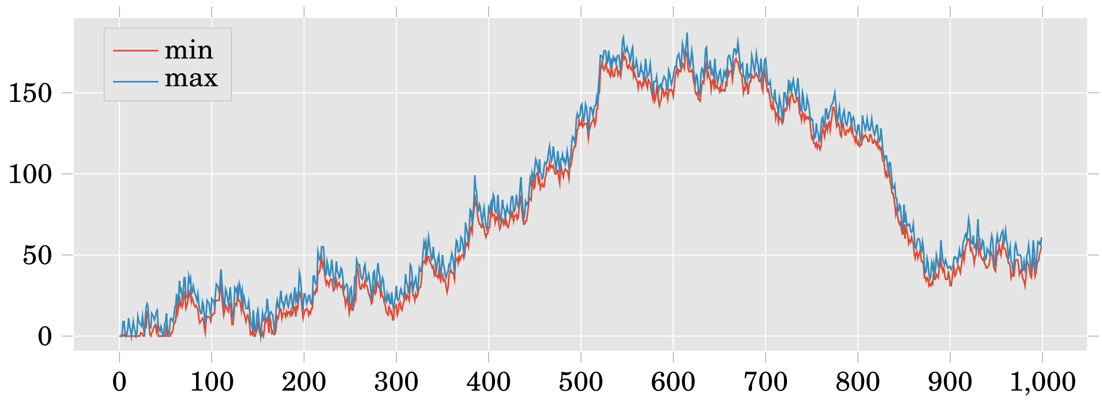


Figure 2.4: Effect of capacity and holiday plans. We plot for time point the maximum and the minimum queue length for each of the policies. We see that the effect of each of these policies is, for all practical purposes, negligible.

Now we consider suggestion 3, which comes down to doing more intakes when it is busy, and do less when it is quiet. A simple rule to implement this is by considering last week's queue Q_{n-1}: if $Q_{n-1} < 12$, i.e., the service capacity of one week, then do e intakes less. Here, $e = 1$ or 2, or perhaps a larger number; it corresponds to the amount of control we want to exercise. When $Q_{n-1} > 24$, i.e., larger than two weeks of intakes, do e intakes more. Let's consider three different control levels, $e = 1$, $e = 2$, and $e = 5$; thus in the last case all psychiatrists do one extra intake. The previous simulation shows that it is safe to disregard the holiday plans, so just assume a flat service capacity of 12 intakes a week.

Figure 2.5 shows a striking difference indeed. The queue does not explode any more, and already taking $e = 1$ has a large influence.

From the simulation experiment we learn that changing holiday plans or spreading the work over multiple servers, i.e., psychiatrists, does not significantly affect the queueing behavior. However, controlling the service rate as a function of the queue length improves the situation quite dramatically.

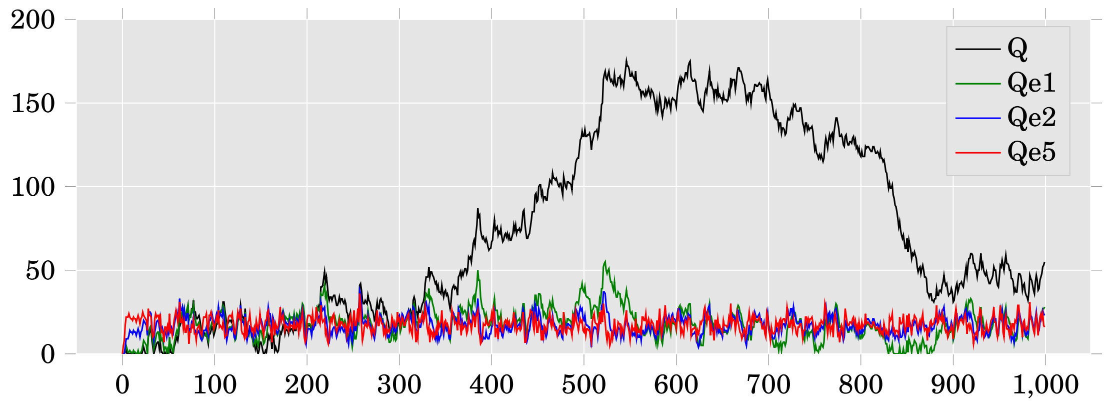


Figure 2.5: Controlling the number of intakes. Clearly, adapting the service rate 'does wonders' to control the queue length.

Thus, we made models for the arrival and service processes and then developed a simple recursion of the type (2.7) to *construct* a queueing process. Finally, for the analysis, we made plots of these processes.

In more abstract terms the study of queueing system is focused on studying the probabilistic properties of the queueing length process and related concepts such as waiting time, server occupancy, fraction of customers lost, and so on. Once we have constructed the queueing process we can compute all performance measures of relevance, such as the average waiting time, c.f. Section 2.8. If it turns out that the performance of the system is not according to what we desire, we can change parts of the system with the aim to improve the situation and assess the effect of this change. For instance, if the average waiting time is too long, we might add service capacity with the aim to reduce the service times, hence reduce the average waiting time. With simulation it is easy to study the effect of, hence evaluate, such decisions.

Observe that, even with these simple recursions, we can obtain considerable insight into this, otherwise, very complicated controlled queueing process. (If the reader doubts the value of simulation, s/he should try to develop other mathematical methods to analyze multi-server queueing system with vacations, of which this is an example. Warning, do not even attempt: you'll fail as it is most probably too hard.) The simplicity of the recursions is deceitful, but with these simple recursions we can analyze many practical queueing situations. Together with students the author applied it numerous times, for instance,

- Should a certain hospital invest in a new MRI scanner to reduce waiting times?
- When to switch on and off a tin bath at an electronics component factory?
- What is the effect of reducing the number of jobs in a paint factory?
- Post parcel routing in a post sorting center.
- Controlling inventories at various companies.
- Throughput time analysis at courts.

And so on, and so on. The recursions are indeed astonishingly useful. We therefore urge the reader to practice with this type of queueing modeling. The exercises below provide ample material for this purpose.

In passing we remark that yet more powerful simulations can be carried out with, so-called, event-based simulations. We refer to [Hall, 1991, Section 4.5] for a general description of this technique. The author applied this to quite complicated queueing networks:

- Analysis of the packaging process of a large beer factory.
- Performance analysis of large telecommunication networks
- Optimization of truck routing and queueing for a sugar factory.

Due to lack of time, we decided not to include this.

The reader should understand from the above case that, once we have the recursions, we can analyze the system and make plots to evaluate suggestions for improvement. Thus, getting the recursions is crucial to construct, i.e., model, queueing processes. For this reason, most of the exercises below focus on obtaining recursions for many different queueing systems.

A comment is required about the modeling exercises below. It may be that the recursions you find are not identical to the recursions in the solution; the reason is that the assumptions you make might not be equal the ones I make. I don't quite know how get out of this paradoxical situation. In a sense, to completely specify the model, we need the recursions. However, if the problem statement would contain the recursions, there would be nothing left to practice anymore. Another way is to make the problem description five times as long, but this is also undesirable. So, let's be pragmatic: the aim is that you practice with modeling, and that you learn from the solutions. If you obtain *reasonable* recursions, but they are different from mine, then your answer is just as good.

Exercises

Exercise 2.4.1. What are the consequences of setting $d_k = \min\{Q_{k-1} + a_k, c_k\}$ rather than the definition (2.7a)?

Exercise 2.4.2. (Queue with Blocking) Consider a queueing system under daily review, i.e., at the end of the day the queue length is measured. We assume that at the end of the day no jobs are still in service. We assume that jobs that arrive at day n cannot be served in day n. The queue length cannot exceed level K. Formulate a set of recursions to cover this case.

Exercise 2.4.3. (Yield loss) A machine produces items, but a fraction p of the items produced in a period turn out to be faulty, and have to be made anew. Develop a set of recursions to cover this case.

Exercise 2.4.4. (Rework) A machine produces items, but a fraction p of the items do not meet the quality requirements after the first service but need some extra service time but less than an entirely new arriving job. Make a model to analyze this case. Compare this case with the yield loss problem above.

Let's assume that the repair of a faulty requires half of the work of a new job, and that the faulty jobs are processed with priority over the new jobs. (There are of course many different policies to treat rework; here we make this assumption. For your interest: another possibility is that faulty items are processed at the end of the day. Yet another possibility is that faulty items are collected until there are N, say, and then the entire batch of N is repaired.)

Exercise 2.4.5. (Cost models) A single-server queueing station processes customers. At the start of a period the server capacity is chosen, so that for period k the capacity is c_k. Demand that arrives in a period can be served in that period. It costs β per unit time per unit processing capacity to operate the machine, i.e., to have it switched on. There is also a cost h per unit time per job in the system. Make a cost model to analyze the long-run average cost for this case.

Exercise 2.4.6. (N-policies) A machine can switch on and off. If the queue length hits N, the machine switches on, and if the system becomes empty, the machine switches off. It costs K to switch on the machine. There is also a cost β per unit time while the machine is switched on, and it costs h per unit time per customer in the system. Make a cost model.

Exercise 2.4.7 (use=false)**.** (Queue with setups) One server serves two parallel queues, one at a time. After serving a batch of 20 jobs of one queue the server moves to the other queue. The change of queue requires one period setup time.

Exercise 2.4.8. How would you model (in terms of recursions) a server whose capacity depends on the queue length? Consider, as an example, a rule such that the server only works if the queue is larger than a threshold t.

Exercise 2.4.9. (Fair queueing) One server serves two queues. Each queue receives service capacity in proportion to its queue length. Derive a set of recursions to analyze this situation.

Exercise 2.4.10. (Priority queuing) Another interesting situation is a system with two queues served by one server, but such that one queue, queue A gets priority over the other queue. Again find a set of recursions to describe this case.

Exercise 2.4.11. (Queues with reserved service capacity) Consider a single-server that serves two parallel queues. Each queue receives a minimal service capacity every period. Reserved capacity unused for one queue can be used to serve the other queue. Any extra capacity beyond the reserved capacity is given to queue A with priority. Formulate a set of recursions to analyze this situation.

Exercise 2.4.12. (Queue with protected service capacity, lost capacity) Consider a single-server that serves two parallel queues. Each queue receives a minimal service capacity every period. Reserved capacity unused for one queue cannot be used to serve the other queue. Any extra capacity beyond the reserved capacity is given to queue A with priority. Formulate a set of recursions to analyze this situation.

Let r_A be the reserved capacity for queue A, and likewise for r_B. We assume of course that $c_k \geq r_A + r_B$, for all k.

Exercise 2.4.13. (Tandem networks) Consider a production network with two production stations in tandem, that is, the jobs processed by station A are in the next period to the downstream Station B. Extend the recursions of (2.7) to simulate this situation.

Exercise 2.4.14. (A tandem queue with blocking) Consider a production network with two production stations in tandem with blocking: when intermediate queue, i.e., the queue front of Station B, exceeds some level M, then station A has to stop producing, and when $Q_k^B < M$ station A is not allowed to produce more than the intermediate queue can contain. Extend the recursions of (2.7) to simulate this situation.

Exercise 2.4.15. (Merging departure streams) Consider another production situation with two machines, A and B say, that send their products to Station C. Derive a set of recursion relations to simulate this system.

Exercise 2.4.16. (Estimating the lead time distribution.) Take $d_k = \min\{Q_{k-1} + a_k, c_k\}$, and assume that jobs are served in FIFO sequence. Find an expression for the shortest possible waiting time $W_-(k)$ of a job that arrives at time k, and an expression for the largest possible waiting time $W_+(k)$

Exercise 2.4.17. (Merging incoming streams) *You can skip this problem for the midterm as I changed the answer quite a bit after the lecture. It is still interesting though.* Consider a single-server queue that servers two customer 'streams' in a FIFO discipline. Thus, both streams enter one queue that is served by the server. Let $\{a_k^a\}$ be the number of arrivals of stream a in period k and $\{a_k^b\}$ be the number of arrivals of stream b. Find a set of recursions by which it becomes possible to analyze the waiting time distribution of each of the streams. Assume that the service capacity c is constant for all periods, and that jobs that arrive in period k can also be served in period k.

Exercise 2.4.18. (Splitting streams) Consider a machine (a paint mixing machine) that produces products for two separate downstream machines A and B (two different paint packaging machines), each with its own queue. Suppose we want to analyze the queue in front of station A. For this we need to know the arrivals to station A, that is, the departures of the mixing station that go to station A. Provide a set of recursions to simulate this system.

Exercise 2.4.19. (Inventory control) The recursions used in the exercises above can also be applied to analyze inventory control policies. Consider a production system that can produce maximally M_k items per week during normal working hours, and maximally N_k items during extra (weekend and evening hours). Let, for period k,

$$\begin{aligned}
D_k &= \text{Demand in week } k, \\
S_k &= \text{Sales, i.e., number of items sold, in week } k, \\
r_k &= \text{Revenue per item sold in week } k, \\
X_k &= \text{Number of items produced in week } k \text{ during normal hours}, \\
Y_k &= \text{Number of items produced in week } k \text{ during extra hours}, \\
c_k &= \text{Production cost per item during normal hours}, \\
d_k &= \text{Production cost per item during extra hours}, \\
h_k &= \text{holding cost per item, due at the end of week } k, \\
I_k &= \text{On hand inventory level at the end of week } k.
\end{aligned}$$

Management needs a production plan that specifies for the next T weeks the number of items to be produced per week. Formulate this problem as an LP problem, taking into account the inventory dynamics.

Exercise 2.4.20. Implement Eqs. 2.7 in a computer program and simulate a simple single-server queueing system.

Hints

Hint 2.4.16: Consider a numerical example. Suppose $Q_{k-1} = 20$. Suppose that the capacity is $c_k = 3$ for all k. Then a job that arrives in the kth period, must wait at least 20/3 (plus rounding) periods before it can leave the system. Now generalize this numerical example.

Hint 2.4.19: Formulate the decision variables/controls, the objective and the constraints.

Solutions

Solution 2.4.1: The assumption is that the jobs arrive at the start of period k, before service in period k starts, rather than at the end of the period. Therefore the arrivals at period k can also be served during period k.

Solution 2.4.2: All jobs that arrive such that the queue become larger than K must be dropped. Thus, the accepted arrivals a'_n in week n are such that $a'_n = \min\{a_n, K - Q_{n-1}\}$, where a_n are all arrivals in period n. The rest of the recursions remain the same.

Solution 2.4.3: The amount produced in period k is d_k. Thus, pd_k is the amount lost, neglecting rounding errors for the moment. Thus, pd_k items have to be fed back to the system in the next period to be remade. Therefore the total amount of arrivals in period $k+1$ is $a'_{k+1} = a_{k+1} + pd_k$, i.e., the external arrivals plus the extra items. Now use the standard recursions but with the $\{a'_k\}$ rather than $\{a_k\}$.

Can you use these recursions to show that the long-run average service capacity $n^{-1}\sum_{i=1}^{n} c_i$ must be larger than $\lambda(1+p)$?

If you like you can incorporate time-dependent failure rates $\{p_k\}$ too. Whether this makes practical sense depends on the context of course.

Solution 2.4.4: Suppose again that a fraction p is faulty. Since these faulty items require less processing time than a new job, the service capacity c_k, i.e., the number of jobs that can be processed in period k, is a bit bigger; part of the capacity is spent on new jobs but another part is spent on the faulty jobs. By the assumptions above, the repair of a faulty requires half of the work of a new job, and the faulty jobs are processed with priority over the new jobs. Assume queue A contains the faulty items, and queue B the new jobs. Then the recursions become:

$$
\begin{aligned}
d_{k,A} &= \min\{Q_{k-1,A}, 2c_k\}, \text{ as faulty jobs require half of the processing time)}\\
c_{k,B} &= c_k - d_{k,A}/2,\\
d_{k,B} &= \min\{Q_{k-1,B}, c_{k,B}\},\\
Q_{k,A} &= Q_{k-1,A} + a_{k,A} - d_{k,A},\\
Q_{k,B} &= Q_{k-1,B} + a_{k,B} - d_{k,B}.
\end{aligned}
$$

Solution 2.4.5: First consider the dynamics of the queue. Since the capacity is chosen at the start of the period:

$$
\begin{aligned}
d_k &= \min\{Q_{k-1} + a_k, c_k\}\\
Q_k &= Q_{k-1} + a_k - d_k.
\end{aligned}
$$

The cost to operate the server during period k is βc_k. Thus, the total cost up to some time T for the server must be $\beta\sum_{k=1}^{T} c_k$. In period k we also have to pay hQ_k, since h is the cost per customer per period in the system. Thus, the long-run average cost is

$$\frac{1}{T}\sum_{k=1}^{T}\left(\beta c_k + hQ_k\right).$$

It is an interesting problem to find a policy that minimizes (the expectation of) this cost. The policy is such that the capacity for period k can be chosen based on the queue length Q_{k-1} and *estimates* of the demands $\hat{d}_k, \hat{d}_{k+1}, \ldots$. This problem is not easy, as far as I can see.

Solution 2.4.6: First we need to implement the N-policy. For this we need an extra variable to keep track of the state of the server. Let $I_k = 1$ if the machine is on in period k and $I_k = 0$ if it is off. Then $\{I_k\}$ must satisfy the relation

$$I_{k+1} = \begin{cases} 1 & \text{if } Q_k \geq N, \\ I_k & \text{if } 0 < Q_k < N, \\ 0 & \text{if } Q_k = 0, \end{cases}$$

and assume that $I_0 = 0$ at the start, i.e., the machine if off. Thus, we can write:

$$I_{k+1} = \mathbb{1}_{Q_k \geq N} + I_k \mathbb{1}_{0<Q_k<N} + 0 \cdot \mathbb{1}_{Q_k=0}.$$

With I_k it follows that $d_k = \min\{Q_{k-1}, I_k c_k\}$, from which Q_k follows, and so on.

The machine cost for period k is βI_k, because only when the machine is on we have to pay β, and the queueing cost is hQ_k. To determine the total switching cost is harder as we need to determine how often the machine has been switched on up to time T. Observe that the machine is switched on in period k if $I_{k-1} = 0$ and $I_k = 1$. Thus, whenever $I_k - I_{k-1} = 1$ the machine is switched on, when $I_k - I_{k-1} = 0$ the state of the machine remains the same, and if $I_k - I_{k-1} = -1$ the machine is switched off. In other words $\max\{I_k - I_{k-1}, 0\}$ captures what we need. The total cost up to time T becomes:

$$\sum_{k=1}^{T}\left(\beta I_k + hQ_k + K\max\{I_k - I_{k-1}, 0\}\right).$$

Solution 2.4.7:

Solution 2.4.8: One model could be to let the server only switch on when the queue is larger than some threshold t, and when the server is on, it works at rate c per period. In that case, $c_k = c\,\mathbb{1}_{Q_{k-1}>t}$.

Solution 2.4.9: Let c_k^i be the capacity allocated to queue i in period k. The fair rule gives that

$$c_k^1 = \frac{Q_{k-1}^1}{Q_{k-1}^1 + Q_{k-1}^2} c = c - c_k^2.$$

Then,

$$d_k^1 = \min\{Q_{k-1}^1, c_k^1\},$$
$$Q_k^1 = Q_{k-1}^1 + a_k^1 - d_k^1,$$

and likewise for the other queue.

Solution 2.4.10: The rules below implement a strict priority rule for jobs type A, i.e., jobs sent into queue A.

$$
\begin{aligned}
d_{k,A} &= \min\{Q_{k-1,A}, c_k\}, \\
c_{k,B} &= c_k - d_{k,A}, \\
d_{k,B} &= \min\{Q_{k-1,B}, c_{k,B}\}, \\
Q_{k,A} &= Q_{k-1,A} + a_{k,A} - d_{k,A}, \\
Q_{k,B} &= Q_{k-1,B} + a_{k,B} - d_{k,B}.
\end{aligned}
$$

As an aside, another interesting rule to distribute the capacity c_k over the queues could be based on the principle of *equal division of the contested sum*. This principle is based on game theoretic ideas. Aumann and Maschler applied this principle to clarify certain division rules discussed in the Talmud to divide the legacy among a number of inheritors, each having a different claim size.

Solution 2.4.11: First determine how much capacity queue B minimally needs in period k:

$$c_{k,B} = \min\{Q_{k-1,B}, r_B\}$$

Observe that, since $c_k \geq r_A + r_B$, this rule ensures that queue A receives at least its reserved capacity r_A.

Since queue A is served with priority, we first give all capacity, except what queue B minimally needs, to queue A:

$$d_{k,A} = \min\{Q_{k-1,A}, c_k - c_{k,B}\}.$$

And then we can give any left over capacity to queue B, if needed.

$$d_{k,B} = \min\{Q_{k-1,B}, c_k - d_{k,A}\}.$$

An example is the weekly capacity offered by a psychiatrist at a hospital. Part of the weekly capacity is reserved/allocated/assigned to serve certain patients groups. For instance, each week the psychiatrist does at most five intakes of new patients, provided there are any, and the rest of the capacity is used to treat other patients. The existing patients can also be divided in different groups, each receiving a minimal capacity. It there less patients of some group, then the capacity can be planned/given to other patient groups.

Solution 2.4.12: Queue A can use all capacity, except what is reserved for queue B:

$$d_{k,A} = \min\{Q_{A,k-1}, c_k - r_B\}.$$

Observe that, since $c_k \geq r_A + r_B$, this rule ensures that queue A receives at least its reserved capacity r_A.

Queue B cannot receive more than $c_k - r_A$, since r_A is allocated to queue A, and if queue A does not use all of r_A, then the surplus is lost. Also, queue B cannot get more than $c_k - d_{k,A}$ as this is what remains after serving queue A. Thus, letting $c_{k,B} = \min\{c_k - r_A, c_k - d_{k,A}\} = c_k - \max\{r_A, d_{k,A}\}$, we see that for queue B:

$$d_{k,B} = \min\{Q_{B,k-1}, c_{k,B}\}.$$

An example can be the operation room of a hospital. There is a weekly capacity, part of the capacity is reserved for emergencies. It might not be possible to assign this reserved capacity to other patient groups, because it should be available at all times for emergency patients. A result of this is that unused capacity is lost.

In practice it may not be as extreme as in the model, but still part of the unused capacity is lost. 'Use it, or lose it', is what often, but not always, applies to service capacity.

Solution 2.4.13: Let a_k be the external arrivals at station A. Then:

$$\begin{aligned} d_k^A &= \min\{Q_{k-1}^A, c_k^A\}, \\ Q_k^A &= Q_{k-1}^A - d_k^A + a_k. \end{aligned} \tag{2.8}$$

The departures of the first station during period k are the arrivals at station B at the end of period k, i.e., $a_k^B = d_k^A$. Thus,

$$\begin{aligned} a_k^B &= d_k^A, \\ d_k^B &= \min\{Q_{k-1}^B, c_k^B\}, \\ Q_k^B &= Q_{k-1}^B - d_k^B + a_k^B. \end{aligned} \tag{2.9}$$

Solution 2.4.14:

$$\begin{aligned} d_k^A &= \min\{Q_{k-1}^A, c_k^A, M - Q_{k-1}^B\}, \\ Q_k^A &= Q_{k-1}^A - d_k^A + a_k, \\ a_k^B &= d_k^A, \\ d_k^B &= \min\{Q_{k-1}^B, c_k^B\}, \\ Q_k^B &= Q_{k-1}^B - d_k^B + a_k^B. \end{aligned} \tag{2.10}$$

This is a bit subtle: since there is room $M - Q_{k-1}^B$ at the intermediate buffer and $d_k^A \leq M - Q_{k-1}^B$, we know that in the worst case, i.e., when $c_k^B = 0$, still $Q_k^B = Q_{k_1}^B + d_k^A$. Thus, we are sure that the queue length of the intermediate queue will not exceed M.

There is still a small problem: What if for the first initial periods $M < Q_{k-1}^B$. Then $M - Q_{k-1}^B <$ 0 and then by the specification above, $d_k^A < 0$. This is not what we want. Therefore,

$$d_k^A = \min\{Q_{k-1}^A, c_k^A, \max\{M - Q_{k-1}^B, 0\}\}.$$

Solution 2.4.15: Realize that Stations A and B have their own arrivals.

$$\begin{aligned} d_k^A &= \min\{Q_{k-1}^A, c_k^A\}, \\ Q_k^A &= Q_{k-1}^A - d_k^A + a_k^A, \\ d_k^B &= \min\{Q_{k-1}^B, c_k^B\}, \\ Q_k^B &= Q_{k-1}^B - d_k^B + a_k^B, \\ a_k^C &= d_k^A + d_k^B, \\ d_k^C &= \min\{Q_{k-1}^C, c_k^C\}, \\ Q_k^C &= Q_{k-1}^C - d_k^C + a_k^C. \end{aligned} \tag{2.11}$$

Solution 2.4.16: Let's tag the first customer that arrives in period k. This tagged customer sees Q_{k-1} customer in the system, hence the tagged customer's service can only start after all Q_{k-1} customers have been served. Now, if $Q_{k-1} - c_k > 0$, there are still people in front of the tagged customer. In fact, as long as $c_k + c_{k+1} + \cdots + c_m < Q_{k-1}$ the number of customers in front of the tagged customer are still in the system.

We can also tag the last customer that arrives in period k. This customer will certainly have left if m is such that $c_k + \cdot + c_m > Q_{k-1} + a_k$.

In formulas the above comes down to the following. A job that arrives in period k cannot be served before period

$$W_{-,k} := \max\left\{m : \sum_{i=k}^{k+m} c_i < Q_{k-1}\right\},$$

and it must have been served before period

$$W_{+,k} := \min\left\{m : \sum_{i=k}^{k+m} c_i \geq Q_k\right\}.$$

Thus, the waiting time of jobs arriving in period k must lie in the interval $[W_{-,k}, W_{+,k}]$.

Solution 2.4.17: The behavior of the queue length process is easy:

$$\begin{aligned} d_k &= \min\{Q_{k-1} + a_k^a + a_k^b, c\}, \\ Q_k &= Q_{k-1} + a_k^a + a_k^b - d_k. \end{aligned}$$

To determine the waiting times, observe that any arrival in period k, independent of the stream, has to wait until all jobs at the start of the period in queue, i.e., $Q_{k-1} - d_k$, are cleared; note that we assume here that the jobs served in period k depart at the start of the interval. Thus, the minimal waiting time is $W_{k,-} = \lfloor Q_{k-1}/c \rfloor$. Similarly, the maximal waiting time is $W_{k,+} = \lfloor (Q_{k-1} + a_k^a + a_k^b)/c \rfloor$.

The remaining problem is to make a model to 'distribute' the time between $W_{k,-}$ and $W_{k,+}$ over the two streams.

A simple model is to assume that the waiting time is averaged over the jobs. Then each job perceives a waiting time of

$$\frac{W_{k,-} + W_{k,+}}{2}.$$

Another way is to give priority to a customers, but only for the jobs that arrive in this period. (Hence, this is different from the priority queue. There priority customers can overtake customers that arrived earlier. In the present case this is not allowed, jobs of type a that arrive in period k cannot overtake b jobs that arrived prior to period k.) Making this completely explicit (so that the recursion can be fed to the computer) requires a bit of work however. It is important to understand the trick we will discuss now because we will use it to model queueing systems with batching. Observe that the first job of the a stream only has to wait for $W_{k,-}$, the second job must wait $W_{k,-} + 1$, and so on. Thus, the waiting time W_k^a for the a_k^a items is such that

$$W_{k,-}^a := \lfloor Q_{k-1}/c \rfloor \leq W_k^a \leq \lfloor (Q_{k-1} + a_k^a)/c \rfloor =: W_{k,+}^a.$$

Similarly, for the b jobs must be

$$W_{k,-}^b := \lfloor (Q_{k-1} + a_k^a)/c \rfloor \leq W_k^b \leq \lfloor (Q_{k-1} + a_k^a + a_k^b)/c \rfloor =: W_{k,+}^b.$$

Note that $W^a_{k,+} = W^b_{k,-}$. It is then sensible to set

$$
\begin{aligned}
W^a_k &= \frac{W^a_{k,-} + W^a_{k,+}}{2}, \\
W^b_k &= \frac{W^b_{k,-} + W^b_{k,+}}{2}.
\end{aligned}
$$

I currently lack time to check whether the above does not contain any off-by-one errors. For future work, implement it in a simulator...

Solution 2.4.18: 1. Realize that the recursions of Eq (2.7) applied to the queueing situation at the first machine provide us with the total number of departures d_k during period k. However, it does not tell us about the type of these departures. Thus, to compute the queue in front of station A, we need do know the number of departures of type A, rather than the total number of departures of the first station.

2. It is reasonable that the number of jobs of type A in queue at the first station is equal to

$$Q_k \frac{\lambda_A}{\lambda_A + \lambda_B}.$$

It is therefore reasonable to assume that the capacity c_k of the first station is also shared in this proportion to type A and B jobs. Thus, the number of departures to station A is

$$d_k(A) = \frac{\lambda_A}{\lambda_A + \lambda_B} \min\{Q_{k-1}, c_k\}.$$

The rest of the recursions is very similar to what we did in earlier exercises.

Solution 2.4.19: The decision variables are X_k, Y_K and S_k (note, it is not necessary to meet all demand: the production cost and profit may vary per period.) The objective is

$$\max \sum_{k=1}^{T} (r_k S_k - c_k X_k - d_k Y_k - h_k I_k).$$

The constraints are

$$
\begin{aligned}
&0 \le S_k \le D_k, \\
&0 \le X_k \le M_k, \\
&0 \le Y_k \le N_k, \\
&I_k = I_{k-1} + X_k + Y_k - S_k. \\
&I_k >= 0.
\end{aligned}
$$

Solution 2.4.20: Here is an example in python; please read the code to see how I approach the problem. The code is really easy, as it is nearly identical to the mathematical specification. You don't have to memorize the specific syntax of the code, of course, but it is (at least I find it) interesting to see how little code is actually necessary to set things up.

Below I fix the seed of the random number generator to ensure that I always get the same results from the simulator. The arrival process is Poisson, and the number of services is fixed to 21 per period. I use `Q = np.zeros_like(a)` to make an array of the same size as the number of arrival a initially set to zeros. The rest of the code is nearly identical to the formulas in the text.

```
import numpy as np

np.random.seed(3) # fix the seed

labda = 20
mu = 21
```

These are the number of arrivals in each period.

```
a = np.random.poisson(labda, 10)
a
```

```
array([21, 17, 14, 10, 22, 22, 17, 17, 19, 21])
```

The number of potential services

```
c = mu * np.ones_like(a)
c
```

```
array([21, 21, 21, 21, 21, 21, 21, 21, 21, 21])
```

Now for the queueing recursions:

```
Q = np.zeros_like(a)
d = np.zeros_like(a)
Q[0] = 10  # initial queue length

for k in range(1, len(a)):
    d[k] = min(Q[k - 1], c[k])
    Q[k] = Q[k - 1] - d[k] + a[k]
```

These are the departures for each period:

```
d
```

```
array([ 0, 10, 17, 14, 10, 21, 21, 19, 17, 19])
```

The queue lengths

```
Q
```

```
array([10, 17, 14, 10, 22, 23, 19, 17, 19, 21])
```

```
loss = (Q > 20)
loss
```

```
array([False, False, False, False,  True,  True, False, False, False,
True], dtype=bool)
```

```
loss.sum()
```

```
3
```

Here I define loss as the number of periods in which the queue length exceeds 20. Of course, any other threshold can by taken. Counting the number of such periods is very easy in python: `(Q>20)` gives all entries of Q such $Q > 20$, the function `sum()` just adds them.

Now all statistics:

```
d.mean()
```

```
14.800000000000001
```

```
Q.mean()
```

```
17.199999999999999
```

```
Q.std()
```

```
4.3772137256478576
```

```
(Q > 20).sum()
```

```
3
```

Since this is a small example, the mean number of departures, i.e., `d.mean()`, is not equal to the arrival rate λ. Likewise for the computation of the mean, variance, and other statistical functions.

Now I am going to run the same code, but for a larger instance.

```
num = 1000
a = np.random.poisson(labda, num)
c = mu * np.ones_like(a)
Q = np.zeros_like(a)
d = np.zeros_like(a)
Q[0] = 10  # initial queue length

for k in range(1, len(a)):
    d[k] = min(Q[k - 1], c[k])
    Q[k] = Q[k - 1] - d[k] + a[k]
```

```
d.mean()
```

```
20.178000000000001
```

```
Q.mean()
```

```
28.420000000000002
```

```
Q.std()
```

```
10.155865300406461
```

```
(Q > 30).sum()/num * 100
```

```
34.200000000000003
```

I multiply with 100 to get a percentage. Clearly, many jobs see a long queue, the mean is already some 24 jobs. For this arrival rate a service capacity of $\mu = 21$ is certainly too small.

Next, an example with the same arrival pattern but with a Poisson distributed number of jobs served per day, so the service rate is still 21, but the period capacity c is random variable with Poisson distribution with $P(21)$

```
c = np.random.poisson(mu, num)
Q = np.zeros_like(a)
d = np.zeros_like(a)
Q[0] = 10  # initial queue length

for k in range(1, len(a)):
    d[k] = min(Q[k - 1], c[k])
    Q[k] = Q[k - 1] - d[k] + a[k]
```

```
d.mean()
```

```
20.148
```

```
Q.mean()
```

```
42.518000000000001
```

```
Q.std()
```

```
22.841096208369684
```

```
(Q > 30).sum()/num * 100
```

```
62.399999999999999
```

Compared to the case with constant service capacity, i.e., $c = 21$ in each period, we see that the average waiting time increases, just as the number of periods in which the queue length exceeds 30. Clearly, variability in service capacity does not improve the performance of the queueing system, quite on the contrary. In later sections we will see why this is so.

You can try to implement some of the other cases of the exercises in this section and analyze them in the same way. Hopefully you understand from this discussion that once you have the recursions to construct/simulate the queueing process, you are 'in business'. The rest is easy: make plots, do some counting, i.e., assemble statistics, vary parameters for sensitivity analysis, and so on.

2.5 Construction of the G/G/1 Queueing Process in Continuous Time

In the previous section we considered time in discrete 'chunks', minutes, hours, days, and so on. For given numbers of arrivals and capacity per period we use a set of recursions (2.7) to compute the departures and queue length per period. Another way to construct a queueing system is to consider inter-arrival times between consecutive customers and the service times each of these customers require. With this we obtain a description of the queueing system in continuous time. The goal of this section is to develop a set of recursions for the $G/G/1$ single-server queue.

Assume we are given, as basic data, the *arrival process* $\{A(t); t \geq 0\}$: the number of jobs that arrived during $[0,t]$. Thus, $\{A(t); t \geq 0\}$ is a *counting process*.

From this arrival process we can obtain various other interesting concepts, such as the arrival times of individual jobs. Specially, if we know that $A(s) = k-1$ and $A(t) = k$, then the arrival time A_k of the kth job must lie somewhere in $(s,t]$. Thus, from $\{A(t)\}$, we can define

$$A_k = \min\{t : A(t) \geq k\}, \tag{2.12}$$

and set $A_0 = 0$. Once we have the set of arrival times $\{A_k\}$, the *inter-arrival times* $\{X_k, k = 1,2,\dots\}$ between consecutive customers can be constructed as

$$X_k = A_k - A_{k-1}. \tag{2.13}$$

Often the basic data consists of the inter-arrival times $\{X_k; k = 1,2,\dots\}$ rather than the arrival times $\{A_k\}$ or the number of arrivals $\{A(t)\}$. Then we construct the arrival times as

$$A_k = A_{k-1} + X_k,$$

with $A_0 = 0$. From the arrival times $\{A_k\}$ we can, in turn, construct the arrival process $\{A(t)\}$ as

$$A(t) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k \le t}, \tag{2.14a}$$

where $\mathbb{1}$ is the indicator function. Thus, in the above we count all arrivals that occur up to time t. Another, equivalent, way to define $A(t)$ is

$$A(t) = \max\{k : A_k \le t\}. \tag{2.14b}$$

Clearly, we see that from the inter-arrival times $\{X_k\}$ it is possible to construct $\{A_k\}$ and $\{A(t)\}$, and the other way around, from $\{A(t)\}$ we can find $\{A_k\}$ and $\{X_k\}$. Figure 2.6 shows these relations graphically. To memorize it may be helpfull to write it like this:

$$\begin{aligned} A_k : \mathbb{N} \to \mathbb{R}, &\quad \text{job id (integer) to arrival time (real number),} \\ A(t) : \mathbb{R} \to \mathbb{N}, &\quad \text{time (real number) to number of jobs (integer).} \end{aligned}$$

To compute the departure times $\{D_k\}$ we proceed in stages. The first stage is to construct the *waiting time in queue* $\{W_{Q,k}\}$ as seen by the arrivals. In Figure 2.6 observe that the waiting time of the kth arrival must be equal to the waiting time of the $k-1$th customer plus the amount of *service time* required by job $k-1$ minus the time that elapses between the arrival of job $k-1$ and job k, unless the server becomes idle between jobs $k-1$ and k. In other words,

$$W_{Q,k} = [W_{Q,k-1} + S_{k-1} - X_k]^+, \tag{2.15}$$

where $[x]^+ = \max\{x, 0\}$. If we set $W_{Q,0} = 0$, we can compute $W_{Q,1}$ from this formula, and then $W_{Q,2}$ and so on.

The time job k *leaves the queue and moves on to the server* is

$$D_{Q,k} = A_k + W_{Q,k},$$

because a job can only move to the server after its arrival plus the time it needs to wait in queue. Note that we here explicitly use the FIFO assumption.

Right after the job moves from the queue to the server, its service starts. Thus, $D_{Q,k}$ is the epoch at which the service of job k starts. After completing its service, the job leaves the system. Hence, the *departure time of the system* is

$$D_k = D_{Q,k} + S_k.$$

The *sojourn time*, or *waiting time in the system*, is the time a job spends in the entire system. With the above relations we see that

$$W_k = D_k - A_k = D_{Q,k} + S_k - A_k = W_{Q,k} + S_k, \tag{2.16}$$

where each of these equations has its own interpretation.

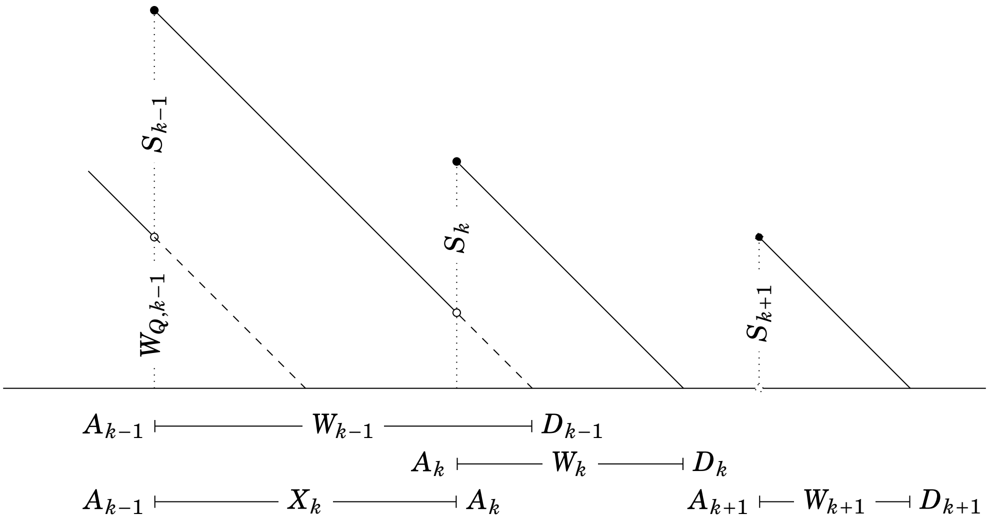


Figure 2.6: Construction of the $G/G/1$ queue in continuous time. The sojourn time W_k of the kth job is sum of the work in queue $W_{Q,k}$ at its arrival epoch A_k and its service time S_k; its departure time is then $D_k = A_k + W_k$. The waiting time of job k is clearly equal to $W_{k-1} - X_k$. We also see that job $k+1$ arrives at an empty system, hence its sojourn time $W_{k+1} = S_{k+1}$. Finally, the virtual waiting time process is shown by the lines with slope -1.

A bit of similar reasoning gives another recursion for W_k:

$$\begin{aligned} W_{Q,k} &= [W_{k-1} - X_k]^+, \\ W_k &= W_{Q,k} + S_k = [W_{k-1} - X_k]^+ + S_k. \end{aligned} \tag{2.17}$$

from which follows a recursion for D_k

$$D_k = A_k + W_k. \tag{2.18}$$

This in turn specifies the departure process $\{D(t)\}$ as

$$D(t) = \max\{k; D_k \leq t\} = \sum_{k=1}^{\infty} \mathbb{1}_{D_k \leq t}.$$

Once we have the arrival and departure processes it is easy to compute the *number of jobs in the system* at time t as

$$L(t) = A(t) - D(t) + L(0), \tag{2.19}$$

where $L(0)$ is the number of jobs in the system at time $t = 0$; typically we assume that $L(0) = 0$. Thus, if we were to plot $A(t)$ and $D(t)$ as functions of t, then the difference $L(t)$ between the graphs of $A(t)$ and $D(t)$ tracks the number in the system, see Figure 2.7.

The number in the system as seen by the kth arrival is defined as

$$L_k = L(A_k), \tag{2.20}$$

since the kth job arrives at time A_k and $\{L(s)\}$ is right-continuous.

Observe that in a queueing system, jobs can be in queue or in service. For this reason we distinguish between the number in the system $L(t)$, the number in queue $L_Q(t)$, and the number

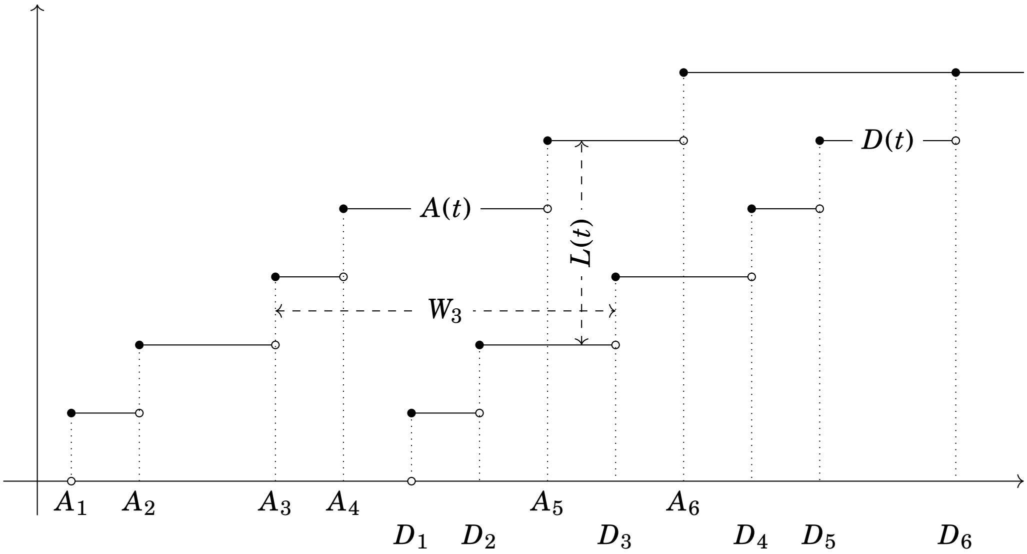


Figure 2.7: Relation between the arrival process $\{A(t)\}$, the departure process $\{D(t)\}$, the number in the system $\{L(t)\}$ and the waiting times $\{W_k\}$.

of jobs in service $L_s(t)$. If we know $D_Q(t)$, i.e. the number of jobs that departed from the queue up to time t, then

$$L_Q(t) = A(t) - D_Q(t)$$

must be the number of jobs in queue. The above expressions for $L(t)$ and $L_Q(t)$ then show that the number in service must be

$$L_s(t) = D_Q(t) - D(t) = L(t) - L_Q(t).$$

Finally, the *virtual waiting time process* $\{V(t)\}$ is the amount of waiting that an arrival would see if it would arrive at time t. To construct $\{V(t)\}$, we simply draw lines that start at points (A_k, W_k) and have slope -1, unless the line hits the x-axis, in which case the virtual waiting time remains zero until the next arrival occurs. Thus, the lines with slope -1 in Figure 2.6 show (a sample path of) the virtual waiting time.

Observe that, just as in Section 2.4, we have obtained a set of recursions by which we can run a simulation of a queueing process of whatever length we need, provided we have a sequence of have inter-arrival times $\{X_k\}$ and service times $\{S_k\}$. A bit of experimentation with computer programs such as R or python will reveal that this is easy.

Exercises

Exercise 2.5.1. 1. Assume that $X_1 = 10$, $X_2 = 5$, $X_3 = 6$ and $S_1 = 17$, $S_2 = 20$ and $S_3 = 5$, compute the arrival times, waiting times in queue, the sojourn times and the departure times for these three customers.

Exercise 2.5.2. Suppose that $X_k \in \{1,3\}$ such that $\mathsf{P}\{X_k = 1\} = \mathsf{P}\{X_k = 3\}$ and $S_k \in \{1,2\}$ with $\mathsf{P}\{S_k = 1\} = \mathsf{P}\{S_k = 2\}$. If $W_{Q,0} = 3$, what are the distributions of $W_{Q,1}$ and $W_{Q,2}$?

Exercise 2.5.3. If $S \sim U[0,7]$ and $X \sim U[0,10]$, where $U[I]$ stands for the uniform distribution concentrated on the interval I, compute $\mathsf{P}\{S - X \leq x\}$.

Exercise 2.5.4. What are the meanings of $A_{A(t)}$ and $A(A_n)$?

Exercise 2.5.5. Would it make sense to define $A(t) = \min\{k : A_k \geq t\}$?

Exercise 2.5.6. Define $L_Q(t)$ as the number of job in queue, and $L_s(t)$ as the number of jobs in service. Likewise, let $D_Q(t)$ be the number of jobs that departed from the queue up to time t.

1. Why don't we need separate notation for $D_s(t)$, the number of jobs that departed from the server?
2. Is $D_Q(t) \leq D(t)$ or $D_Q(t) \geq D(t)$?
3. Why is $L(t) = L_Q(t) + L_s(t)$?
4. Express $L_Q(t)$ and $L_s(t)$ in terms of $A(t)$, $D_Q(t)$ and $D(t)$.
5. Consider a multi-server queue with m servers. Suppose that at some t it happens that $D_Q(t) - D_s(t) < m$ even though $A(t) - D_s(t) > m$. How can this occur?

Exercise 2.5.7. Show that $L(t) = A(t) - D(t)$ implies that

$$L(t) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k \leq t < D_k}.$$

Show also that if $L(A_k) > 0$, i.e., the system contains at least one job at the time of the kth arrival, then $A_k \leq D_{k-1}$, i.e., job k arrives before job $k-1$ departs.

Exercise 2.5.8. Can you derive the following (algorithmic efficient) procedure to compute the number of jobs in the system as seen by arrivals,

$$L_k = L_{k-1} + 1 - \sum_{i=k-1-L_{k-1}}^{k-1} \mathbb{1}_{D_i < A_k}?$$

Why do we take $i = k-1-L_{k-1}$ in the sum, and not $i = k-2-L_{k-1}$?

Exercise 2.5.9. Provide a specification of the virtual waiting time process $\{V(t)\}$ for all t.

Exercise 2.5.10. Another set of recursions to compute the arrival and departure times for a given set of inter-arrival and service times is the following:

$$\begin{aligned} A_k &= A_{k-1} + X_k, \\ D_k &= \max\{A_k, D_{k-1}\} + S_k. \end{aligned} \tag{2.21}$$

Now the computation of the waiting times is trivial: $W_k = D_k - A_k$.

1. Why do the recursions Eq. (2.21) work?
2. (Very difficult) Extend the above recursions to a situation in which one queue is served by two servers.
3. (Impossible?) Extend the above recursions to a situation in which one queue is served by $m > 2$ servers.

Exercise 2.5.11. Implement the above recursions in excel or some other computer program such as R, python, or julia, and check the results of the previous exercise.

Exercise 2.5.12. (Multiple queues, for the interested) Suppose one server serves two queues, such that jobs in queue A are served with priority over jobs in queue B. Assume that service is not-preemptive. Can you develop a similar set of recursions for each of the queues, or, otherwise, an algorithm?

Exercise 2.5.13. (LIFO queue) Can you develop a set of recursions or algorithm for a LIFO (Last-In-First-Out) queue?

Hints

Hint 2.5.2: Use that $W_{Q,1} = [W_{Q,0} + S_1 - X_1]^+$ and likewise for $W_{Q,2}$.
This first year probability. It should be elementary for you. Ensure you can do it.

Hint 2.5.3: This appears to be trivial..., but I didn't find it easy... In fact, I found it a major headache, hence it is good to try yourself before looking at the answer. Check your probability book if you don't know what to do.

Hint 2.5.5: Compare this to the definition in (2.14).

Hint 2.5.7: Use Boolean algebra. Write, for notational ease, $A = \mathbb{1}_{A_k \leq t}$ and $\bar{A} = 1 - A = \mathbb{1}_{A_k > t}$, and define something similar for D. Then show that $A - D = A\bar{D} - \bar{A}D$, and show that $\bar{A}D = 0$. Finally sum over k.

Hint 2.5.9: Make a plot of the function $A_{A(t)} - t$. What is the meaning of $V(A_{A(t)})$? What is $V(A_{A(t)} + A_{A(t)} - t$?

Solutions

Solution 2.5.1: The intent of this exercise is to make you familiar with the notation.
BTW, such simple test cases are also very useful to test computer code. The numbers in the exercise are one such simple case. You can check the results by hand; if the results of the simulator are different, there is a problem.

Solution 2.5.2: First find the distribution of $Y_k := S_{k-1} - X_k$ so that we can write $W_{Q,k} = [W_{Q,k-1} + Y_k]^+$. Use independence of $\{S_k\}$ and $\{X_k\}$:

$$\mathsf{P}\{Y_k = -2\} = \mathsf{P}\{S_{k-1} - X_k = -2\} = \mathsf{P}\{S_{k-1} = 1, X_k = 3\} = \mathsf{P}\{S_{k-1} = 1\}\,\mathsf{P}\{X_k = 3\} = \frac{1}{4}.$$

Dropping the dependence on k for ease, we get

$$\begin{aligned}
\mathsf{P}\{Y = -2\} &= \mathsf{P}\{S - X = -2\} = \mathsf{P}\{S = 1, X = 3\} = \mathsf{P}\{S = 1\}\,\mathsf{P}\{X = 3\} = \frac{1}{4} \\
\mathsf{P}\{Y = -1\} &= \mathsf{P}\{S = 2\}\,\mathsf{P}\{X = 3\} = \frac{1}{4} \\
\mathsf{P}\{Y = 0\} &= \mathsf{P}\{S = 1\}\,\mathsf{P}\{X = 1\} = \frac{1}{4} \\
\mathsf{P}\{Y = 1\} &= \mathsf{P}\{S = 2\}\,\mathsf{P}\{X = 1\} = \frac{1}{4}.
\end{aligned}$$

With this

$$\begin{aligned}
\mathsf{P}\{W_{Q,1}=1\} &= \mathsf{P}\{W_{Q,0}+Y_0=1\} = \mathsf{P}\{3+Y_0=1\} = \mathsf{P}\{Y_0=-2\} = \frac{1}{4} \\
\mathsf{P}\{W_{Q,1}=2\} &= \mathsf{P}\{3+Y=2\} = \mathsf{P}\{Y=-1\} = \frac{1}{4} \\
\mathsf{P}\{W_{Q,1}=3\} &= \mathsf{P}\{3+Y=3\} = \mathsf{P}\{Y=0\} = \frac{1}{4} \\
\mathsf{P}\{W_{Q,1}=4\} &= \mathsf{P}\{3+Y=4\} = \mathsf{P}\{Y=1\} = \frac{1}{4}.
\end{aligned}$$

And, then

$$\begin{aligned}
\mathsf{P}\{W_{Q,2}=1\} = \mathsf{P}\{W_{Q,1}+Y=1\} &= \sum_{i=1}^4 \mathsf{P}\{W_{Q,1}+Y=1 \,|\, W_{Q,1}=i\}\,\mathsf{P}\{W_{Q,1}=i\} \\
&= \sum_{i=1}^4 \mathsf{P}\{i+Y=1 \,|\, W_{Q,1}=i\}\frac{1}{4} = \sum_{i=1}^4 \mathsf{P}\{Y=1-i \,|\, W_{Q,1}=i\}\frac{1}{4} \\
&= \frac{1}{4}\sum_{i=1}^4 \mathsf{P}\{Y=1-i\} = \frac{1}{4}(\mathsf{P}\{Y=0\}+\mathsf{P}\{Y=-1\}+\mathsf{P}\{Y=-2\}) = \frac{3}{16}.
\end{aligned}$$

Typing the solution becomes boring..., let's use the computer.

```
from lea import Lea

W = 3
S = Lea.fromVals(1, 2)
X = Lea.fromVals(1, 3)

# This is WQ1
W = Lea.fastMax(W + S - X, 0)
W

1 : 1/4
2 : 1/4
3 : 1/4
4 : 1/4


# This is WQ2
W = Lea.fastMax(W + S - X, 0)
W

0 : 3/16
1 : 3/16
2 : 4/16
3 : 3/16
4 : 2/16
5 : 1/16
```

Great! Our handiwork matches with the computer's results.

Solution 2.5.3: Let us write $f(x)$ for the density of $\mathsf{P}\{S-X\le x\}$, i.e., $\mathsf{P}\{S-X\le x\} = \int_{-\infty}^x f(y)\,\mathrm{d}y$. (A formal point, why does $\mathsf{P}\{S-X\le x\}$ actually have a density? You can skip this detail if you are not interested in mathematical details, nevertheless, it is interesting to think about.) With conditioning,

$$f(x) = \int_0^{10} \mathsf{P}\{S - X = x | X = u\} \mathsf{P}\{X \in \mathrm{d}u\}$$

This notation is, admittably, quite subtle (and to get it right requires much more work than we can do in this course). Intuitively, $\mathsf{P}\{X \in \mathrm{d}u\}$ means the (infinitesimal) probability that $X \in [u, u + \mathrm{d}u]$. Since X is uniform on $[0,10]$, $\mathsf{P}\{X \in [s,t]\} = (t-s)/10$ for $s,t \in [0,10]$ and $s < t$. Now, taking this literaly, let $s = u$ and $t = u + \mathrm{d}u$, it follows that,

$$\mathsf{P}\{X \in [u, u + \mathrm{d}u]\} = \frac{u + \mathrm{d}t - u}{10} = \frac{\mathrm{d}u}{10}.$$

Using this in the above expression for $f(x)$ gives

$$f(x) = \frac{1}{10} \int_0^{10} \mathsf{P}\{S - X = x | X = u\} \, \mathrm{d}u.$$

Now use that S and X are independent so that

$$\begin{aligned}
\mathsf{P}\{S - X = x | X = u\} &= \frac{\mathsf{P}\{S - X = x, X = u\}}{\mathsf{P}\{X = u\}} = \frac{\mathsf{P}\{S - u = x, X = u\}}{\mathsf{P}\{X = u\}} \\
&= \frac{\mathsf{P}\{S - u = x\} \mathsf{P}\{X = u\}}{\mathsf{P}\{X = u\}} = \mathsf{P}\{S = u - x\}.
\end{aligned}$$

Observe that we just use the formulas we are used to from conditioning on sets with positive probability. Recall, $\mathsf{P}\{A|B\} = \mathsf{P}\{AB\}/\mathsf{P}\{B\}$ only when $\mathsf{P}\{B\} > 0$. But here $\mathsf{P}\{X = x\} = 0$ for all x, since x consists of a single point and X is uniformly distributed. Thus, we divide by 0! To get the mathematical details right, requires quite a bit more work than we can do in this course. Interesingly, in this case it works, as $\mathsf{P}\{X = u\}$ appears in the numerator and denominator, hence can be cancelled. In general, be careful when using such shortcuts; it can go wrong easily.

With this,

$$f(x) = \frac{1}{10} \int_0^{10} \mathsf{P}\{S - u = x\} \, \mathrm{d}u = \frac{1}{10} \int_0^{10} \mathsf{P}\{S = x + u\} \, \mathrm{d}u.$$

Next, interpret $\mathsf{P}\{S = y\}$ as the density of $\mathsf{P}\{S \leq y\}$ so that $\mathsf{P}\{S = y\} = \mathbb{1}_{y \in [0,7]}/7$ Therefore, while using that in the above integral that x is fixed,

$$\mathsf{P}\{S = x + u\} = \frac{1}{7} \mathbb{1}_{-x \leq u \leq 7-x},$$

from which

$$\begin{aligned}
f(x) &= \frac{1}{70} \int_0^{10} \mathbb{1}_{-x \leq u \leq 7-x} \, \mathrm{d}u \\
&= \frac{1}{70} \int_{-\infty}^{\infty} \mathbb{1}_{0 \leq u \leq 10} \, \mathbb{1}_{-x \leq u \leq 7-x} \, \mathrm{d}u \\
&= \frac{1}{70} \int_{-\infty}^{\infty} \mathbb{1}_{\max\{0,-x\} \leq u \leq \min\{10,7-x\}} \, \mathrm{d}u.
\end{aligned}$$

First we make some simple observations. If $x > 7$ then $\min\{10, 7-x\} < 0$. This is smaller then $\max\{0, -x\}$. Thus, $f(x) = 0$ for $x \geq 7$. Likewise, when $x \leq -10$, $f(x) = 0$. Also, when $x \in [-3, 0]$ the max and min overlap. Thus, all in all,

$$f(x)=\begin{cases}0, & \text{if } x\le -10,\\ \frac{x+10}{70} & \text{if } x\in[-10,-3],\\ \frac{1}{10} & \text{if } x\in[-3,0],\\ \frac{7-x}{70} & \text{if } x\in[0,7],\\ 0, & \text{if } x>7,\end{cases}$$

The above is the answer; but, given the amount of effort I had to put into getting it, I want to check it. For this I used Wolframalpha. You can skip the rest of the solution below if you are not interested in how to do this (I will not ask this at the exam). However, to expand your skill set, I advise you to take notice of it anyway. Besides that, Wolframalpha is a great site. Please check it out if you haven't done so up to now.

So, I went to Wolframalpha, and this is what what I typed:

```
\int_{-\infty}^\infty Boole[0<=u<=10] Boole[-x<=u<=7-x] du,
```

so, once you know LaTeX you can use wolframalpha. Wolframalpha turned it to

```
integral\_(-infinity)^infinity Boole[0<=u<=10] Boole[-x<=u<=7-x] du
```

For your convenience, I also include the following code from Wolframalpa

```
Integrate[Boole[Max[0, -x] <= u <= Min[10, 7 - x]], {u, -Infinity, Infinity}]
```

I got the same results as Wolframalpa, but certainly not for free.

Solution 2.5.4: $A(t)$ is the number of arrivals during $[0,t]$. Suppose that $A(t)=n$. This nth job arrived at time A_n. Thus, $A_{A(t)}$ is the arrival time of the last job that arrived before or at time t. In a similar vein, A_n is the arrival time of the nth job. Thus, the number of arrivals up to time n, i.e., $A(A_n)$, must be n.

Solution 2.5.5: Suppose $A_3=10$ and $A_4=20$. Take $t=15$. Then $\min\{k : A_k\ge 15\}=4$ since $A_3<t=15<A_4$. On the other hand $\max\{k : A_k\le t\}=3$.

Solution 2.5.6:

1. Because $D_s(t)=D(t)$. Once customers leave the server, their service is completed, and they leave the queueing system.
2. All customers that left the system must have left the queue. Thus, $D_Q(t)\ge D(t)$.
3. Jobs in the system are in queue or in service.
4. $L_Q(t)=A(t)-D_Q(t)$. $L_S(t)=D_Q(t)-D(t)$. This is in line with the fact that $L(t)=L_Q(t)+L_s(t)=A(t)-D(t)$.
5. In that case, there are servers idling while there are still customers in queue. If such events occur, we say that the server is not work-conservative.

Solution 2.5.7:

1.

$$\begin{aligned}L(t)&=A(t)-D(t)\\ &=\sum_{k=1}^{\infty}\mathbb{1}_{A_k\le t}-\sum_{k=1}^{\infty}\mathbb{1}_{D_k\le t}\\ &=\sum_{k=1}^{\infty}[\mathbb{1}_{A_k\le t}-\mathbb{1}_{D_k\le t}].\end{aligned}$$

Write for the moment $A = \mathbb{1}_{A_k \le t}$ and $\bar{A} = 1 - A = \mathbb{1}_{A_k > t}$, and likewise for D. Now we can use Boolean algebra to see that $\mathbb{1}_{A_k \le t} - \mathbb{1}_{D_k \le t} = A - D = A(D + \bar{D}) - D = AD + A\bar{D} - D = A\bar{D} - D(1 - A) = A\bar{D} - D\bar{A}$. But $D\bar{A} = 0$ since $D\bar{A} = \mathbb{1}_{D_k \le t}\mathbb{1}_{A_k > t} = \mathbb{1}_{D_k \le t < A_k}$ which would mean that the arrival time A_k of the kth job would be larger than its departure time D_k. As $A\bar{D} = \mathbb{1}_{A_k \le t < D_k}$

$$\begin{aligned} L(t) &= \sum_{k=1}^{\infty} [\mathbb{1}_{A_k \le t} - \mathbb{1}_{D_k \le t}] \\ &= \sum_{k=1}^{\infty} \mathbb{1}_{A_k \le t < D_k}. \end{aligned}$$

Boolean algebra is actually a really nice way to solve logical puzzles. If you are interested you can find some examples on my homepage.

2. In a sense, the claim is evident, for, if the system contains a job when job k arrives, it cannot be empty. But if it is not empty, then at least the last job that arrived before job k, i.e., job $k - 1$, must still be in the system. That is, $D_{k-1} \ge A_k$. A more formal proof proceeds along the following lines. Using that $A(A_k) = k$ and $D(D_{k-1}) = k - 1$,

$$\begin{aligned} L(A_k) > 0 &\Leftrightarrow A(A_k) - D(A_k) > 0 \Leftrightarrow k - D(A_k) > 0 \Leftrightarrow k > D(A_k) \\ &\Leftrightarrow k - 1 \ge D(A_k) \Leftrightarrow D(D_{k-1}) \ge D(A_k) \Leftrightarrow D_{k-1} \ge A_k, \end{aligned}$$

where the last relation follows from the fact that $D(t)$ is a counting process, hence monotone non-decreasing.

Solution 2.5.8: Let $L_k = L(A_k-)$, i.e., the number of jobs in the system as 'seen by' job k. It must be that $L_k = k - 1 - D(A_k)$. To see this, assume first that no job has departed when job k arrives. Then job k must see $k - 1$ jobs in the system. In general, if at time A_k the number of departures is $D(A_k)$, then the above relation for L_k must hold. Applying this to job $k - 1$ we get that $L_{k-1} = k - 2 - D(A_{k-1})$.

For the computation of L_k we do not have to take the departures before A_{k-1} into account as these have already been 'incorporated in' L_{k-1}. Therefore,

$$L_k = L_{k-1} + 1 - \sum_{i=k-1-L_{k-1}}^{k-1} \mathbb{1}_{D_i < A_k}.$$

Suppose $L_{k-1} = 0$, i.e., job $k - 1$ finds an empty system at its arrival and $D_{k-1} > A_k$, i.e., job $k - 1$ is still in the system when job k arrives. In this case, $L_k = 1$, which checks with the formula. Also, if $L_{k-1} = 0$ and $D_{k-1} < A_k$ then $L_k = 0$. This also checks with the formula.

The reason to start at $k - 1 - L_{k-1}$ is that the number in the system as seen by job k is $k - 1 - D(A_k)$ (not $k - 2 - D(A_k)$). Hence, the jobs with index from $k - 1 - L_{k-1}, k - L_{k-1}, \ldots, k - 1$, could have left the system between the arrival of job $k - 1$ and job k.

Solution 2.5.9: There is a funny way to do this. Recall from a previous exercise that if $A(t) = n$, then A_n is the arrival time of the nth job. Thus, the function $A_{A(t)}$ provides us with arrival times as a function of t. When $t = A_{A(t)}$, i.e., when t is the arrival time of the $A(t)$th job, we set $V(t) = V(A_{A(t)}) = W_{A(t)}$ i.e., the virtual waiting time at the arrival time $t = A_{A(t)}$ is equal to the waiting time of the $A(t)$th job. Between arrival moments, the virtual waiting time decreases with slope 1, until it hits 0. Thus,

$$V(t) = [V(A_{A(t)}) + (A_{A(t)} - t)]^+ = [W_{A(t)} + (A_{A(t)} - t)]^+$$

The notation may be a bit confusing, but it is in fact very simple. Take some t, look back at the last arrival time before time t, which is written as $A_{A(t)}$. (In computer code these times are easy to find.) Then draw a line with slope -1 from the waiting time that the last arrival saw.

Solution 2.5.10: 1. Of course, the service of job k cannot start before it arrives. Hence, it cannot leave before $A_k + S_k$. Therefore it must be that $D_k \geq A_k + S_k$. But the service of job k can also not start before the previous job, i.e. job $k-1$, left the server. Thus job k cannot start before D_{k-1}. To clarify it somewhat further, define S'_k as the earliest start of job k. Then it must be that $S'_k = \max\{A_k, D_{k-1}\}$—don't confuse the earliest start S'_k and the service time S_k—and $D_k = S'_k + S_k$.

2. I found this not easy, to say the least... The problem is that in a multi-server queueing systems, unlike for single-server queues, jobs can overtake each other: a small job that arrives after a very large job can still leave the system earlier. After trying for several hours, I obtained an inelegant method. A subsequent search on the web helped a lot. The solution below is a modification of N. Krivulin, 'Recursive equations based models of queueing systems'.

   The recursions for the two-server system are this:

$$\begin{aligned}
A_k &= A_{k-1} + X_k, \\
C_k &= \max\{A_k, D_{k-2}\} + S_k, \\
M_k &= \max\{M_{k-1}, C_k\}, \\
D_{k-1} &= \min\{M_{k-1}, C_k\}.
\end{aligned}$$

   Here, C_k is the completion time of job k, and $\{D_k\}$ is a sorted list of departure times. Thus, D_k is the kth departure time; recall this is not necessarily equal to the completion time C_k of the kth job (as jobs may overtake each other). To understand the other equations, we reason like this. By construction, $C_k > D_{k-m}$ (as $S_k > 0$). Therefore, when we arrived at time C_k, $(k-m)$ jobs must have departed. Moreover, by construction, M_k tracks the latest completion time of all k jobs, hence, M_{k-m+1} is the latest completion time of the first $k-m+1$ jobs. Therefore, if $C_k > M_{k-1}$, job k must leave later than the latest of the jobs in $\{1,2,\ldots,k-1\}$. Hence, the latest departure time of the jobs in $\{1,2,\ldots,k-1\}$ jobs must be M_{k-1}. If however, $C_k < M_{k-1}$, then job k leaves earlier than the latest of the jobs in $\{1,2,\ldots,k-1\}$. As $C_k > D_{k-2}$, it must be that $C_k > M_{k-2}$, because D_{k-2} is latest departure of the jobs in $\{1,2,\ldots,k-2\}$, and this is also equal to M_{k-2}. As a consequence, if $C_k < M_{k-1}$, job k is also the first job that leaves after D_{k-2} (provided of course that $C_{k+1} < C_k$). Thus, all in all $D_{k-1} = \min\{M_{k-1}, C_k\}$.

3. Similar reasoning for the $G/G/m$ queue seems to lead to the following.

$$\begin{aligned}
A_k &= A_{k-1} + X_k, \\
C_k &= \max\{A_k, D_{k-m}\} + S_k, \\
M_k &= \max\{M_{k-m+1}, C_k\}, \\
D_{k-m+1} &= \min\{M_{k-m+1}, C_k\},
\end{aligned}$$

but it is not correct. Can you find a counter example?

Solution 2.5.11: You can check the files on `github` in the `progs` directory.

- `mm1.py` is the python implementation
- `mm1.R` is the R implementation
- `mm1.jl` is the julia implementation

Take your pick, and start playing with it. These examples are meant to be simple to understand, not necessarily super efficient.

Solution 2.5.12: This question came up in class. I don't think this can be constructed as a straightforward recursion, and a search on the web lead to nothing. In case you can find a recursion, please let me know.

While a straightforward recursion may not exist, it is not really difficult to code this queueing discipline. The key idea is to put all jobs into one queue, but sort the elements in the queue in order of priority. Every time the server becomes empty, it checks the head of the queue. If the queue is empty, it wait until the next arrival occurs. Otherwise, it starts the service of the first job in line. When a new job arrives, then identify its priority first, and then put it at the end of the jobs of the same priority.

This type of sorting is known as lexicographic sorting, which is what we do when we build a dictionary. First we sort words in order of first letter, then the second letter, and so on. In case of sorting the queue, we first have to sort in order of priority, then, within the class of jobs with the same priority, sort in ascending order of arrival time.

In python the code is like this, where I include the FIFO case for comparison. As always, it is not obligatory to memorize the code.

```
class Fifo(Queue):
    def __init__(self):
        self.queue = SortedSet(key = lambda job: job.arrivalTime)

class Priority(Queue): # a priority queue
    def __init__(self, numServers = 1):
        self.queue = SortedSet(key = lambda job: job.p)
```

Interestingly, once we have a proper environment to carry out simulations, changing the queueing discipline is a one-liner!.

Solution 2.5.13: Again, I don't know a good set of recursions, but an algorithm is straightforward. Rather then sorting the jobs in queue in ascending order of arrival time, just sort them in descending order of arrival time.

```
class Lifo(Queue):
    def __init__(self):
        self.queue = SortedSet(key = lambda job: -job.arrivalTime)
```

Another interesting queueing rule is to sort in increasing job size;

```
  class SPTF(Queue): # shortest processing time first
    def __init__(self):
        self.queue = SortedSet(key = lambda job: job.serviceTime)
```

Do you see how sort in descending order of job size (it just a matter of putting a minus sign at the right place)?

2.6 Queueing Processes as Regulated Random Walks

In the construction of queueing processes as set out in Section 2.4 we are given two sequences of i.i.d. random variables: the number of arrivals $\{a_k\}$ per period and the service capacities $\{c_k\}$. Assuming that jobs can be served in the period they arrive, the departure and queue length processes are generated by the recursions

$$\begin{aligned} Q_k &= [Q_{k-1} + a_k - c_k]^+, \\ d_k &= Q_{k-1} + a_k - Q_k, \end{aligned} \tag{2.22}$$

where $[x]^+ := \max\{x, 0\}$. Observe now that the relation for Q_k shares a resemblance to a random walk $\{Z_k, k = 0, 1, \ldots\}$ with Z_k given by

$$Z_k = Z_{k-1} + a_k - c_k.$$

Clearly, $\{Z_k\}$ is 'free', i.e., it can take positive and negative values, but $\{Q_k\}$ is restricted to the non-negative integers. In this section we show how to build the queueing process $\{Q_k\}$ from the random walk $\{Z_k\}$ by a device called a *reflection map*. Besides the elegance of this construction, the construction leads to spectacularly fast simulations. Moreover, we can use the probabilistic tools that have been developed for the random walk to analyze queueing systems. One example is the distribution of the time until an especially large queue length is reached; these times can be formulated as *hitting times* of the random walk. Another example is the average time it takes to clear a large queue.

In an exercise below we derive that Q_k satisfies the relation

$$Q_k = Z_k - \min_{0 \le i \le k} Z_i \wedge 0, \tag{2.23}$$

where Z_k is defined by the above random walk and $a \wedge b = \min\{a, b\}$ [3]. This recursion leads to really interesting graphs, see Figures 2.8 and 2.9. We consider two examples. Let $a_k \sim B(0.3)$, i.e., a_k is Bernoulli-distributed with success parameter $p = 0.3$, i.e., $\mathsf{P}\{a_k = 1\} = 0.3 = 1 - \mathsf{P}\{a_k = 0\}$, and $c_k \sim B(0.4)$. As a second example, we take $a_k \sim B(0.49)$ and construct the random walk as

$$Z_k = Z_{k-1} + 2a_k - 1.$$

Thus, if $a_k = 1$, the random walk increases by one step, while if $a_k = 0$, the random walk decreases by one step.

Now that we have seen that random walks can be converted into queueing systems, we study in an exercise below the transient distribution, i.e., the distribution as a function of time, of the random walk $\{Z_k\}$. The aim of this exercise is to show that there is no simple function by which we can compute the transient distribution of this relatively simple random walk. Since a queueing process is typically a more complicated object (as it is a constrained random walk), our hopes to finding anything simple for the transient analysis of the $M/M/1$ queue should not be too high. And the $M/M/1$ is but the simplest queueing system; other queueing systems will be more complicated yet. We therefore give up the analysis of such transient queueing systems and we henceforth contend ourselves with the analysis of queueing systems in the limit as $t \to \infty$. This of course warrants two questions: what type of limit is actually meant here,

[3] For further detail, refer to Baccelli and Massey [1988].

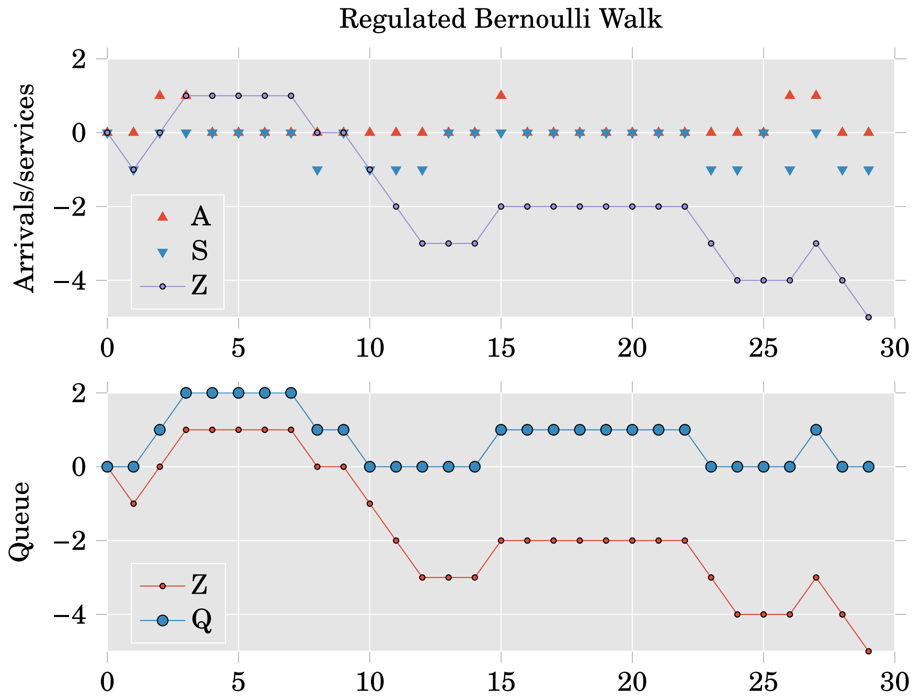


Figure 2.8: The upper panel shows a graph of the random walk Z. An upward pointing triangle corresponds to an arrival, a downward triangle to a potential service. The lower panel shows the queueing process $\{Q_k\}$ as a random walk with reflection.

and is such limiting situation reached in any reasonable time? We address these questions subsequently.

The *long-run limiting behavior* of a queueing system is an important topic by itself. The underlying question is what happens if we simulate the system for a long time. For instance, does there exist a random variable Q such that $Q_k \to Q$ in some sense? The answer to this question is in the affirmative, provided some simple stability conditions are satisfied, see Section 2.7. However, it requires a considerable amount of mathematics to make this procedure precise. To sketch what has to be done, first, we need to define $\{Q_k\}$ as random variables in their own right. Note that up to now we just considered each Q_k as a *number*, i.e., a measurement or simulation of the queue length time of the kth period. Defining Q_k as a random variable is not as simple as the definition of, for instance, the number of arrivals $\{a_k\}$; these random variables can be safely *assumed* to be i.i.d. However, the queue lengths $\{Q_k\}$ are certainly not i.i.d., but, as should be apparent from Eq. (2.17), they are *constructed* in terms of recursions. Next, based on these recursions, we need to show that the sequence of distribution functions $\{G_k\}$ associated with the random variables $\{Q_k\}$ converges to some limiting distribution function G, say. Finally, it is necessary to show that it is possible to construct a random variable Q that has G as its distribution function. In this sense, then, we can say that $Q_k \to Q$. The random variable Q is known as the *steady-state limit* of the sequence of random variables $\{Q_k\}$, and the distribution G of Q is known as the *limiting* or *stationary distribution* of $\{Q_k\}$.

In these notes we sidestep all these fundamental issues, as the details require measure theory and more advanced probability theory than we can deal with in this course. However, it can all be made precise.

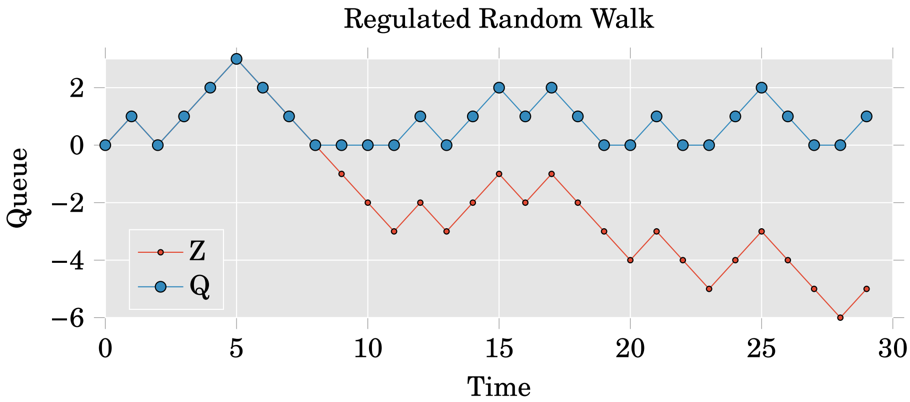


Figure 2.9: Another example of a reflected random walk.

Next, as an example, we address the rate of convergence of the sequence of waiting times $\{W_{Q,k}\}$ to a limiting random variable W_Q, where $W_{Q,k}$ is constructed according to the recursion Eq. (2.15). Suppose that $X_k \sim U\{1,2,4\}$ and $S_k \sim U\{1,2,3\}$. Starting with $W_{Q,0} = 5$ we use Eq. (2.15) to compute the *exact* distribution of $W_{Q,k}$ for $k = 1,2,\ldots,20$, c.f., the left panel in Figure 2.10. We see that when $k = 5$, the 'hump' of $\mathsf{P}\{W_{Q,5} = x\}$ around $x = 5$ is due the starting value of $W_{Q,0} = 5$. However, for $k > 10$ the distribution of $W_{Q,k}$ hardly changes, at least not visually. Apparently, the convergence of the sequence of distributions of $W_{Q,k}$ is rather fast. In the middle panel we show the results of a set of *simulations* for increasing simulation length, up to $N = 1000$ samples. Here the *empirical distribution* for the simulation is defined as

$$\mathsf{P}\{W_Q \leq x\} = \frac{1}{n} n \sum_{k=1}^{n} \mathbb{1}_{W_{Q,k} \leq x},$$

where $W_{Q,k}$ is obtained by simulation. As should be clear from the figure, the simulated distribution also seems to converge quite fast to some limiting function. Finally, in the right hand panel we compare the densities as obtained by the exact method and simulation with $n = 1000$. Clearly, for all practical purposes, these densities can be treated as the same.

The combination of the fast convergence to the steady-state situation and the difficulties with the transient analysis validates, to some extent, that most queueing theory is concerned with the analysis of the system in *stationarity*. The study of queueing systems in stationary state will occupy us for the rest of the book.

Exercises

Exercise 2.6.1. What is the difference between the recursive schemes of (2.22) and (2.7)? Explain in the former scheme the formula for d_k.

Exercise 2.6.2. Show that

$$Q_k = Z_k - \min_{0 \leq i \leq k} Z_i \wedge 0,$$

where Z_k is defined by the above random walk.

Exercise 2.6.3. (Transient behavior of the free $M/M/1$ queue.) The free $M/M/1$ queue is a queueing process that can become negative. Thus, it is just the random walk

$$Z(t) = Z(0) + N_\lambda(t) - N_\mu(t),$$

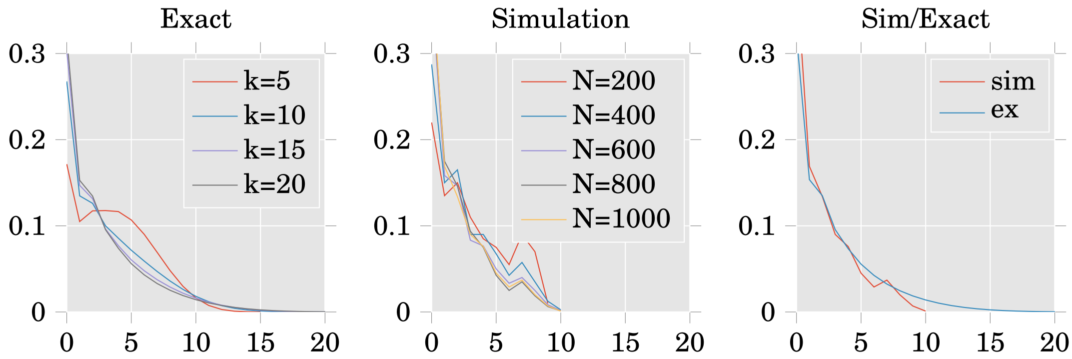


Figure 2.10: The density of $W_{Q,k}$ for $k = 5, 10, 15, 20$ computed by an exact method as compared the density obtained by simulation of different run lengths $N = 200, 400, \ldots 1000$. The right panel compares the exact density of $W_{Q,20}$ to the density obtained by simulation for $N = 1000$.

where the arrival process is a Poisson process $N_\lambda(t)$ and the departure process is a Poisson process $N_\mu(t)$. Show that

$$\mathsf{P}_m\{Z(t) = n\} = e^{-(\lambda+\mu)t} \left(\frac{\lambda}{\mu}\right)^{(n-m)/2} \sum_{k=0}^{\infty} \frac{(t\sqrt{\lambda\mu})^{2k+m-n}}{k!(k+m-n)!},$$

where $\mathsf{P}_m\{\cdot\}$ means that the random walk starts at m, i.e., $Z(0) = m$[4].

Now this summation is known as a modified Bessel function, but this does not make us, in some sense, much wiser. In fact, we use this result as a deterrent to study transient aspects of queueing system.

Exercise 2.6.4. In Eq. (2.24) the summation runs from 0 to ∞ while in the computation of $\mathsf{P}\{N_a(t)+N_b(t) = n\}$ in Eq. (2.5) the summation runs from 0 to n. Why are the summation limits different in these two equations?

Exercise 2.6.5. Can you relate the construction of $\{Q_k\}$ as a discrete-time random walk to a continuous-time construction of the $M/M/1$ queue?

Exercise 2.6.6. Suppose that $X_k \in \{1,3\}$ such that $\mathsf{P}\{X_k = 1\} = \mathsf{P}\{X_k = 3\}$ and $S_k \in \{1,2\}$ with $\mathsf{P}\{S_k = 1\} = \mathsf{P}\{S_k = 2\}$. Write a computer program to see how fast the distributions of $W_{Q,k}$ converge to a limiting distribution function.

Exercise 2.6.7. Validate the results of Figure 2.10 with simulation.

Hints

Hint 2.6.1: When are the arrivals in slot k available for service in either of the systems?

[4]The summation includes negative factorials when $k+m-n < 0$. The tacit assumption is to take $n! \in \{\pm\infty\}$ for $n \in \mathbb{Z}_-$. Another way to get around this problem is to take $k = \max\{0, m-n\}$.

Hint 2.6.2: Note first that from the expression for Z_k, $a_k - c_k = Z_k - Z_{k-1}$. Use this to get $Q_k = [Q_{k-1} + Z_k - Z_{k-1}]^+$. Subtract Z_k, use recursion and use subsequently,

$$
\begin{aligned}
&\max\{\max\{a,b\},c\} = \max\{a,b,c\},\\
&Q_0 = Z_0,\\
&\max\{-a,-b\} = -\min\{a,b\}.
\end{aligned}
$$

Hint 2.6.3: It is actually not hard, even though the expression looks hard. Use conditioning to see that $\mathsf{P}_m\{Z(t)=n\} = \mathsf{P}\{N_\mu(t) - N_\lambda(t) = m - n\}$. Then write out the definitions of the two Poisson distributions. Assemble terms. Then fiddle a bit with the terms to get $(t\sqrt{\lambda\mu})$.

Hint 2.6.5: Consider $Z(t) = Z(0) + N_\lambda(t) - N_\mu(t)$ at the embedded set of epochs at which either N_λ or N_μ changes. What is the probability that N_λ fires, given that one of the two Poisson processes fires? Relate this to $\{a_k\}$ and $\{c_k\}$.

Solutions

Solution 2.6.1: In the scheme

$$
\begin{aligned}
d_k &= \min\{Q_{k-1}, c_k\}\\
Q_k &= Q_{k-1} + a_k - d_k,
\end{aligned}
$$

the arrivals are assumed to arrive *at the end* of period k.

When we use the other scheme,

$$
\begin{aligned}
Q_k &= [Q_{k-1} + a_k - c_k]^+,\\
d_k &= Q_{k-1} + a_k - Q_k,
\end{aligned}
$$

We assume that the arrivals are available at the start of the kth slot. Observe that the number of jobs that enter the kth slot is $Q_{k-1} + a_k$. The number that remain at the end is Q_k. Thus, the number of jobs that depart must be the difference between what enters and what is left behind.

Solution 2.6.2: Note first that from the expression for Z_k, $a_k - c_k = Z_k - Z_{k-1}$. Using this in the recursion for Q_k, we get

$$Q_k = [Q_{k-1} + Z_k - Z_{k-1}]^+,$$

thus,

$$Q_k - Z_k = \max\{Q_{k-1} - Z_{k-1}, -Z_k\}.$$

From this, and using recursion, we see that

$$
\begin{aligned}
Q_k - Z_k &= \max\{\max\{Q_{k-2} - Z_{k-2}, -Z_{k-1}\}, -Z_k\}\\
&= \max\{Q_{k-2} - Z_{k-2}, -Z_{k-1}, -Z_k\}\\
&= \max\{Q_0 - Z_0, -Z_1, \ldots, -Z_k\}\\
&= \max\{0, -Z_1, \ldots, -Z_k\}\\
&= -\min\{0, Z_1, \ldots, Z_k\},
\end{aligned}
$$

where we subsequently use that

$$\begin{aligned}
&\max\{\max\{a,b\},c\} = \max\{a,b,c\},\\
&Q_0 = Z_0,\\
&\max\{-a,-b\} = -\min\{a,b\}.
\end{aligned}$$

Solution 2.6.3: With this we have a characterization of the queue length process as a function of time until it hits zero for the first time. What can we say about the distribution of $Q(t)$? With the above random walk,

$$\begin{aligned}
\mathsf{P}_m\{Z(t)=n\} &= \mathsf{P}\{m+N_\lambda(t)-N_\mu(t)=n\} = \mathsf{P}\{N_\lambda(t)-N_\mu(t)=n-m\}\\
&= \mathsf{P}\{N_\mu(t)-N_\lambda(t)=m-n\}\\
&= \sum_{k=0}^{\infty} \mathsf{P}\{N_\mu(t)=k-n+m \,|\, N_\lambda(t)=k\}\,\mathsf{P}\{N_\lambda(t)=k\}\\
&= \sum_{k=0}^{\infty} e^{-\mu t}\frac{(\mu t)^{k-n+m}}{(k-n+m)!} e^{-\lambda t}\frac{(\lambda t)^k}{k!}\\
&= e^{-(\lambda+\mu)t}\sum_{k=0}^{\infty}\frac{(\lambda t)^k(\mu t)^{k-n+m}}{k!(k-n+m)!}.
\end{aligned} \tag{2.24}$$

We can write this a bit simpler by noting that

$$\begin{aligned}
(\lambda t)^k(\mu t)^{k+m-n}t^{k+m-n} &= \lambda^k t^k \mu^{k+m-n}t^{k+m-n}\\
&= \lambda^k \mu^{k+m-n}(t\sqrt{\lambda\mu})^{2k+m-n}(\lambda\mu)^{-k+(n-m)/2}\\
&= (\lambda/\mu)^{(n-m)2}(t\sqrt{\lambda\mu})^{2k+m-n}.
\end{aligned}$$

With this,

$$\mathsf{P}\{Z(t)=n\} = e^{-(\lambda+\mu)t}\left(\frac{\lambda}{\mu}\right)^{(n-m)/2}\sum_{k=0}^{\infty}\frac{(t\sqrt{\lambda\mu})^{2k+m-n}}{k!(k+m-n)!}.$$

Solution 2.6.4: The process $\{N_a(t)+N_b(t)\}$ is a counting process, hence it is non-decreasing and has positive jumps. The process $\{Z(t)\}$ is a random walk; it can go up and down. Now, if $N_a(t)+N_b(t)=n$, then $N_b(t)\in\{0,\dots,n\}$, it can certainly not be smaller than 0 (as it is a counting process), and it can also be not more than n. On the other hand, if $Z(t)=n$, $N_\lambda(t)$ can be any value in $\mathbb{Z}$ as long as $N_\mu(t)=n+N_\lambda(t)$, provided $Z(0)=0$.

Solution 2.6.5: We use that merging two Poisson processes leads to another Poisson process and that a Poisson process can be split with Bernoulli random variables to form a thinned Poisson process. See one of the exercises of Section 2.2.

Consider the Poisson process $N_{\lambda+\mu}(t)$. This counting process has, for fixed t, the same distribution as $N_\lambda(t)+N_\mu(t)$. Let $\{A_k\}$ be the epochs at which the arrivals of $N_{\lambda+\mu}(t)$ occur. At A_k, throw a coin that lands heads with probability $\lambda/(\lambda+\mu)$ and tails with probability $\mu/(\lambda+\mu)$. Set $a_k=1$ and $c_k=0$ if the coin lands heads, and set $a_k=0$ and $c_k=1$ if it turns tails. The processes

$$\begin{aligned}
N_\lambda(t) &= \sum_{k=1}^{\infty} a_k \mathbb{1}_{A_k\le t},\\
N_\mu(t) &= \sum_{k=1}^{\infty} c_k \mathbb{1}_{A_k\le t},
\end{aligned}$$

are our original Poisson processes. Observe that with this procedure we have established a relation between the sequences $\{a_k\}$ and $\{c_k\}$ and the Poisson processes N_λ and N_μ.

The process $Z_k = Z_{k-1} + a_k - c_k$ is a random walk in discrete time and its reflection results in the discrete-time $M/M/1$ queue. The process $Z(t) = Z(0) + N_\lambda(t) - N_\mu(t)$, with N_λ and N_μ constructed based on the sequences $\{a_k\}$ and $\{c_k\}$, is the continuous-time random walk underlying the $M/M/1$ queue.

Solution 2.6.6: Here is an example with python. In R it must be equally simple. I compute the following

$$\max_x \{|\mathsf{P}\{W_{Q,k} \leq x\} - \mathsf{P}\{W_{Q,k-1} \leq x\}|\},$$

for x in the support of $W_{Q,k}$.

Let's print the difference (The Kolmogorov-Smirnov statistic) between the distributions of W_k and W_{k+1}.

```
from lea import Lea

S = Lea.fromVals(1,  2)
X = Lea.fromVals(1,  3)
U = S-X

W = Lea.fromVals(3)
for k in range(1, 10):
    W_new = Lea.fastMax(W + U, 0)
    m = max(abs(W_new.cdf(x)-W.cdf(x))            for x in range(W_new.
        support()[-1])
            )
    print(k, m)
    W = W_new
```

```
1 0.5
2 0.1875
3 0.125
4 0.08203125
5 0.0576171875
6 0.042236328125
7 0.03131103515625
8 0.02362060546875
9 0.018096923828125
```

So, after some 10 customers, the distribution hardly changes anymore.

Solution 2.6.7: You can study `waiting_time_simulation.py` at github if you like, but skipping it is OK. Trying to make the graph in your favorite programming language is fun.

2.7 Rate Stability and Utilization

In the analysis of any queueing process the first step should be to check the relations between the arrival, service and departure rates. The concept of rate is crucial because it captures our intuition that when, on the long run, jobs arrive faster than they can leave, the system must 'explode'. Thus, the first performance measures we need to estimate when analyzing a queueing system are the arrival and departure rate, and then we need to check that the arrival rate is smaller than the departure rate. In particular, the load, defined as the ratio of the arrival rate and service rate is of importance. In this section we define and relate these concepts. As a reminder, we keep the discussion in these notes mostly at an intuitive level, and refer to El-Taha and Stidham Jr. [1998] for proofs and further background.

We first formalize the *arrival rate* and *departure rate* in terms of the *counting processes* $\{A(t)\}$ and $\{D(t)\}$. The *arrival rate* is the long-run average number of jobs that arrive per unit time, i.e.,

$$\lambda = \lim_{t\to\infty} \frac{A(t)}{t}. \tag{2.25}$$

We remark in passing that this limit does not necessarily exist if $A(t)$ is some pathological function. If, however, the interarrival times $\{X_k\}$ are the basic data, and $\{X_k\}$ are i.i.d. and distributed as a generic random variable X with finite mean $\mathsf{E}\{X\}$, we can construct $\{A_k\}$ and $\{A(t)\}$ as described in Section 2.5; the strong law of large numbers guarantees that the above limit exists.

Observe that at time $t = A_n$, precisely n arrivals occurred. Thus, by applying the definition of $A(t)$ at the epochs A_n, we see that $A(A_n) = n$. Thus,

$$\frac{1}{n}\sum_{k=1}^{n} X_k = \frac{A_n}{n} = \frac{A_n}{A(A_n)}.$$

But since $A_n \to \infty$ if $n \to \infty$, it follows from Eq. (2.25) that the average interarrival time between two consecutive jobs is

$$\mathsf{E}\{X\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} X_k = \lim_{n\to\infty} \frac{A_n}{A(A_n)} = \lim_{t\to\infty} \frac{t}{A(t)} = \frac{1}{\lambda}, \tag{2.26}$$

where we take $t = A_n$ in the limit for $t \to \infty$. In words, the above states that the arrival rate λ is the inverse of the expected interarrival time.

The development of the departure times $\{D_k\}$ is entirely analogous to that of the arrival times; we leave it to the reader to provide the details. As a result we can define the *departure rate* as

$$\lim_{t\to\infty} \frac{D(t)}{t} = \gamma \tag{2.27}$$

Assume now that there is a single server. Let S_k be the required service time of the kth job to be served, and define

$$U_n = \sum_{k=1}^{n} S_k$$

as the total service time required by the first n jobs. With this, let

$$U(t) = \sup\{n : U_n \le t\}.$$

and define the *service* or *processing rate* as

$$\mu = \lim_{t\to\infty} \frac{U(t)}{t}.$$

In the same way as we derived that $\mathsf{E}\{X\} = 1/\lambda$, we obtain for the expected (or average) amount of service required by an individual job

$$\mathsf{E}\{S\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} S_k = \lim_{n\to\infty} \frac{U_n}{n} = \lim_{n\to\infty} \frac{U_n}{U(U_n)} = \lim_{t\to\infty} \frac{t}{U(t)} = \frac{1}{\mu}.$$

Now observe that, if the system is empty at time 0, it must be that at any time the number of departures must be smaller than the number of arrivals, i.e., $D(t) \leq A(t)$ for all t. Therefore,

$$\gamma := \lim_t \frac{D(t)}{t} \leq \lim_t \frac{A(t)}{t} = \lambda. \tag{2.28}$$

We call a system *(rate) stable* if

$$\lambda = \gamma,$$

in other words, the system is stable if, on the long run, jobs leave the system just as fast as they arrive. Of course, if $\lambda > \gamma$, the system length process $L(t) \to \infty$ as $t \to \infty$.

It is also evident that jobs cannot depart faster than they can be served, hence, $D(t) \leq U(t)$ for all t. Combining this with the fact that $\gamma \leq \lambda$, we get

$$\gamma \leq \min\{\lambda, \mu\}.$$

When $\mu \geq \lambda$ the above inequality reduces to $\gamma = \lambda$ for rate-stable systems. (It is interesting to prove this.) As it turns out, when $\mu = \lambda$ and the variance of the service times $\mathsf{V}\{S\} > 0$ or $\mathsf{V}\{X\} > 0$ then $\lim_t L(t)/t$ does not necessarily exist. For this reason we henceforth require that $\mu > \lambda$.

The concept of *load* or *utilization*, denoted by the symbol ρ, is fundamental. One way to define it is as the limiting fraction of time the server is busy, i.e.,

$$\rho = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{1}_{L(s)>0} \, \mathrm{d}s.$$

Interestingly, we can express this in terms of the arrival rate λ and service rate μ. Observe that

$$\sum_{k=1}^{A(t)} S_k \geq \int_0^t \mathbb{1}_{L(s)>0} \, \mathrm{d}s \geq \sum_{k=1}^{D(t)} S_k,$$

since t can lie half way a service interval and $A(t) \geq D(t)$. As $A(t) \to \infty$ as $t \to \infty$,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{A(t)} S_k = \lim_{t\to\infty} \frac{A(t)}{t} \frac{1}{A(t)} \sum_{k=1}^{A(t)} S_k = \lim_{t\to\infty} \frac{A(t)}{t} \cdot \lim_{t\to\infty} \frac{1}{A(t)} \sum_{k=1}^{A(t)} S_k = \lambda \mathsf{E}\{S\}.$$

Applying similar limits to the other inequality gives

$$\lambda \mathsf{E}\{S\} \geq \rho \geq \gamma \mathsf{E}\{S\}.$$

Hence, if $\gamma = \lambda$, $\rho = \lambda \mathsf{E}\{S\}$.

From the identies $\lambda^{-1} = \mathsf{E}\{X\}$ and $\mu^{-1} = \mathsf{E}\{S\}$, we get a further set of relations:

$$\rho = \lambda \mathsf{E}\{S\} = \frac{\lambda}{\mu} = \frac{\mathsf{E}\{S\}}{\mathsf{E}\{X\}}.$$

Thus, the load has also the interpretation as the rate at which jobs arrive multiplied by the average amount of work per job. Finally, recall that for a system to be rate-stable, it is necessary that $\mu > \lambda$, implying in turn that $\rho < 1$. The relation $\rho = \mathsf{E}\{S\}/\mathsf{E}\{X\} < 1$ then tells us that the average time it takes to serve a job must be less than the average time between two consecutive arrivals, i.e., $\mathsf{E}\{S\} < \mathsf{E}\{X\}$.

Exercises

Exercise 2.7.1. Define the departure time D_k of the kth job in terms of $\{D(t)\}$.

Exercise 2.7.2. Define the random variables $\{\tilde{X}_k, k = 1, \ldots\}$ as

$$\tilde{X}_k = S_{k-1} - X_k.$$

For stability of the queueing process it is essential that $\tilde{X}_k$ has negative expectation, i.e., $\mathsf{E}\{\tilde{X}_k\} = \mathsf{E}\{S_{k-1} - X_k\} < 0$. What is the conceptual meaning of this inequality?

Exercise 2.7.3. Define $\tilde{X}_k = S_{k-1} - X_k$. Show that $\mathsf{E}\{\tilde{X}_k\} < 0$ implies that $\lambda < \mu$.

Exercise 2.7.4. Show that $\mathsf{E}\{S\}/\mathsf{E}\{X\}$ is the fraction of time the server is busy.

Exercise 2.7.5. Show that $\mathsf{E}\{X - S\}/\mathsf{E}\{X\}$ is the fraction of time the server is idle.

Exercise 2.7.6. If $\mathsf{E}\{B\}$ is the expected busy time and $\mathsf{E}\{I\}$ is the expected idle time, show that

$$\mathsf{E}\{B\} = \frac{\rho}{1-\rho}\mathsf{E}\{I\}.$$

is the fraction of time the server is busy.

Exercise 2.7.7. Consider a queueing system with c identical servers (identical in the sense that each server has the same production rate μ). What would be a reasonable stability criterion for this system?

Exercise 2.7.8. Consider a paint factory which contains a paint mixing machine that serves two classes of jobs, A and B. The processing times of jobs of types A and B are constant and require t_A and t_B hours. The job arrival rate is λ_A for type A and λ_B for type B jobs. It takes a setup time of S_s hours to clean the mixing station when changing from paint type A to type B, and there is no time required to change from type B to A.

To keep the system (rate) stable, it is necessary to produce the jobs in batches, for otherwise the server, i.e., the mixing machine, spends a too large fraction of time on setups, so that $\mu < \lambda$. Thus, it is necessary to identify minimal batch sizes to ensure that $\mu > \lambda$. Motivate that the linear program below can be used to determine the minimal batch sizes.

$$\text{minimize } T$$

such that

$$\begin{aligned} T &= k_A t_A + S + k_B t_B, \\ \lambda_A T &< k_A, \\ \lambda_B T &< k_B. \end{aligned}$$

Exercise 2.7.9. Can you make an arrival process such that $A(t)/t$ does not have a limit?

Exercise 2.7.10. This exercise is meant to give the reader some idea about what needs to be done to put everything on solid ground. If you are not interested in the maths, you can skip the problem.

1. In Eq. (2.26) we replaced the limit with respect to n by a limit with respect to t. But why is this actually allowed? Use the notation $A_{A(t)}$ to show that all is OK.
2. Show that the function $t \to A(t)$ as defined by Eqs. (2.14) is right-continuous.

Exercise 2.7.11. Check with simulation that when $\lambda > \mu$ the queue length grows roughly linearly with slope $\lambda - \mu$. Thus, if $\rho > 1$, 'we are in trouble'.

Hints

Hint 2.7.1: Use the analogy with Eq. (2.12).

Hint 2.7.3: Remember that $\{X_k\}$ and $\{S_k\}$ are sequences of i.i.d. random variables. What are the implications for the expectations?

Hint 2.7.4: Let $T_1 > A_1$ be the first time after the arrival of job 1 that arrives at an empty system. (Observe that job 1 also arrives at an empty system.) Suppose, for ease of writing, that this job is the $n+1$th job, so that up to time T_1 the number of arrivals is n. Since the first job arrived at time $A_1 = X_1$, the first n jobs arrived during $[X_1, T_1)$. The total amount of service that arrived during this period is $\sum_{i=1}^n S_i$. What is the fraction of time the server has been busy in this cycle?

Hint 2.7.7: What is the rate in, and what is the service capacity?

Hint 2.7.8: Here are some questions to help you interpret this formulation.

1. What are the decision variables for this problem? In other words, what are the 'things' we can control/change?
2. What are the interpretations of $k_A t_A$, and $S + k_B t_B$?
3. What is the meaning of the first constraint? Realize that T represents one production cycle. After the completion of one such cycle, we start another cycle. Hence, the start of every cycle can be seen as a restart of the entire system.
4. What is the meaning of the other two constraints?
5. Why do we minimize the cycle time T?
6. Solve for k_A and k_B in terms of S, λ_A, λ_B and t_A, t_B.
7. Generalize this to m job classes and such that the cleaning time between jobs of class i and j is given by S_{ij}. (Thus, the setup times are sequence dependent.)

Hint 2.7.9: As a start, the function $\sin(t)$ has not a limit as $t \to \infty$. However, the time-average $\sin(t)/t \to 0$. Now you need to make some function whose time-average does not converge, hence it should grow fast, or fluctuate wilder and wilder.

Hint 2.7.10: Use that $A_{A(t)} \leq t < A_{A(t)+1}$. Divide by $A(t)$ and take suitable limits. BTW, such type of proof is used quite often to show that the existence of one limit implies, and is implied by, the existence of another type of limit.

Solutions

Solution 2.7.1:

$$D_k = \inf\{t; D(t) \geq k\}.$$

Solution 2.7.2: That the average time customers spend in service is smaller than the average time between the arrival of two subsequent jobs.

Solution 2.7.3: $0 > \mathsf{E}\{\tilde{X}_k\} = \mathsf{E}\{S_{k-1} - X_k\} = \mathsf{E}\{S_{k-1}\} - \mathsf{E}\{X_k\} = \mathsf{E}\{S\} - \mathsf{E}\{X\}$, where we use the fact that the $\{S_k\}$ and $\{X_k\}$ are i.i.d. sequences. Hence,

$$\mathsf{E}\{X\} > \mathsf{E}\{S\} \iff \frac{1}{\mathsf{E}\{S\}} > \frac{1}{\mathsf{E}\{X\}} \iff \mu > \lambda.$$

Solution 2.7.4: The fraction of time that the server has been busy during $[X_1, T_1)$ is

$$\frac{\sum_{i=1}^n S_i}{T_1 - X_1} = \frac{\sum_{i=1}^n S_i}{\sum_{i=1}^{n+1} X_i - X_1} = \frac{\sum_{i=1}^n S_i}{\sum_{i=2}^{n+1} X_i}$$

Now use the assumption that the $\{X_i\}$ and $\{S_i\}$ are sequences of i.i.d. random variables distributed as the generic random variables X and S, respectively. Then by taking expectations the above becomes

$$\frac{\mathsf{E}\{\sum_{i=1}^n S_i\}}{\mathsf{E}\{\sum_{i=2}^{n+1} X_i\}} = \frac{n\,\mathsf{E}\{S\}}{n\,\mathsf{E}\{X\}} = \frac{\mathsf{E}\{S\}}{\mathsf{E}\{X\}}.$$

These busy cycles occur over and over again. Thus, the long-run average fraction of time the server is busy must also be $\mathsf{E}\{S\}/\mathsf{E}\{X\}$. (For the die-hards, there is a subtle point here: the arrival epochs of the $G/G/1$ queue are not real renewal moments, hence the epochs at which the busy times start also do not form a sequence of renewal times. But then it is not true, in general, that the busy times $\{B_i\}$ have the same distribution, neither do the idles times $\{I_n\}$. Showing that in the limit all is OK requires a substantial amount of mathematics. The above claim is still true however.)

Solution 2.7.5: Since the fraction of idle time is 1 minus the fraction of busy time, it follows that $1 - \mathsf{E}\{S\}/\mathsf{E}\{X\} = (\mathsf{E}\{X\} - \mathsf{E}\{S\})/\mathsf{E}\{X\}$ is the idle time fraction.

Solution 2.7.6: Consider a busy cycle, that is, a cycle that starts with the first job that sees an empty system at upon arrival up to the time another job sees an empty system upon arrival. In such one cycle, the server is busy for an expected duration $\mathsf{E}\{B\}$. The total expected length of the cycle is $\mathsf{E}\{B\} + \mathsf{E}\{I\}$, since after the last job of the cycle left, the expected time until the next job is $\mathsf{E}\{I\}$.

Since ρ is utilization of the server,

$$\rho = \frac{\mathsf{E}\{B\}}{\mathsf{E}\{B\} + \mathsf{E}\{I\}}.$$

With a bit of algebra the result follows.

Solution 2.7.7: The criterion is that c must be such that $\lambda < c\mu$. (Thus, we interpret the number of servers as a *control*, i.e., a 'thing' we can change, while we assume that λ and μ cannot be easily changed.) To see this, we can take two different points of view. Imagine that the c servers are replaced by one server that works c times as fast. The service capacity of these two systems (i.e., the system with c servers and the system with one fast server) is the same, i.e., $c\mu$, where μ is the rate of one server. For the system with the fast server the load is defined as $\rho = \lambda/c\mu$, and for stability we require $\rho < 1$. Another way to see it is to assume that the stream of jobs is split into c smaller streams, each with arrival rate λ/c. In this case, applying the condition that $(\lambda/c)/\mu < 1$ per server leads to the same condition that $\lambda/(c\mu) < 1$.

Solution 2.7.8: Realize that the machine works in cycles. A cycle starts with processing k_A jobs of type A, then does a setup, and processes k_B jobs of type B, and then a new cycle starts again. The time it takes to complete one such cycle is $T = k_A t_A + S + k_B t_B$. The number of jobs of type A processed during one such cycle is, of course, k_A. Observe next that the average number of jobs that arrive during one cycle is $\lambda_A T$. We of course want that $\lambda_A T < k_A$, i.e., less jobs of type A arrive on average per cycle than what we can process.

Solution 2.7.9: If $N(t) = 3t^2$, then clearly $N(t)/t = 3t$. This does not converge to a limit.

Another example, let the arrival rate $\lambda(t)$ be given as follows:

$$\lambda(t) = \begin{cases} 1 & \text{if } 2^{2k} \leq t < 2^{2k+1} \\ 0 & \text{if } 2^{2k+1} \leq t < 2^{2(k+1)}, \end{cases}$$

for $k = 0, 1, 2, \ldots$. Let $N(t) = \lambda(t)t$. Then $A(t)/t$ does not have limit. Of course, these examples are quite pathological, and are not representable for 'real life cases' (Although this is also quite vague. What, then, is a real life case?)

For the mathematically interested, we seek a function for which its Cesàro limit does not exist.

Solution 2.7.10: Observing that $A_{A(t)}$ is the arrival time of the last job before time t and that $A_{A(t)+1}$ is the arrival time of the first job after time t:

$$A_{A(t)} \leq t < A_{A(t)+1} \Leftrightarrow \frac{A_{A(t)}}{A(t)} \leq \frac{t}{A(t)} < \frac{A_{A(t)+1}}{A(t)} = \frac{A_{A(t)+1}}{A(t)+1} \frac{A(t)+1}{A(t)}$$

Now $A(t)$ is a counting process such that $A(t) \to \infty$ as $t \to \infty$. Therefore, $\lim_t A_{A(t)}/A(t) = \lim_n A_n/n$. Moreover, it is evident that $\lim_t A_{A(t)+1}/(A(t)+1) = \lim_t A_{A(t)}/A(t)$, and that $(A(t)+1)/A(t) \to 1$ as $t \to \infty$. Thus it follows from the above inequalities that $\lim_n A_n/n = \lim_t t/A(t)$.

Hopefully this problem, and its solution, clarifies that even such small details require attention. If we want to make some progress with respect to developing some queueing theory, we have to skip most of the proofs and mathematical problems; we simply don't have enough time in this course to be concerned with all theorems and proofs.

For the right-continuity of $A(t)$, define $f(t) = 1\{A_1 \leq t\}$. Observe first that $f(t)$ is increasing, and $f(t) \in \{0, 1\}$. Thus, if $f(t) = 1$ then $f(u) = 1$ for all $u \geq t$, and if $f(t) = 0$ then $f(u) = 0$ for all $u \leq t$.

You may skip the rest of the prove below, but the above is essential to memorize; make a plot of $f(t)$, in particular the behavior around A_1 is important.

We need to prove, for right-continuity, that $f(u) \to f(t)$ as $u \downarrow t$. When $f(t) = 1$, $f(u) = 1$ for any $u > 1$, by the definition of $f(x)$. When $f(t) = 0$ we have to do a bit more work. Formally, we have to prove that, for fixed t and for all $\epsilon > 0$, there is a $\delta > 0$ such that $u \in (t, t+\delta) \Rightarrow |f(u) - f(t)| < \epsilon$. (Note the differences with the regular definition of continuity.) Since, by assumption, t is such that $f(t) = 0$, and $f \in \{0, 1\}$ we need to show that $f(u) = 0$ for $u \in (t, t+\delta)$. Now, clearly, if $f(t) = 0$ only if $t < A_1$. But, then for any $u \in (t, A_1)$, we have that $f(u) = 0$. Thus, taking $\delta = A_1 - t$ suffices.

The next step is to observe that $A(t)$ is a sum of right-continuous functions whose steps do not overlap since by assumption $0 < A_1 < A_2 < \cdots$. As A is (almost surely) finite sum of bounded, increasing and right-continuous functions, it is also right-continuous.

If you like, you can try to prove this last step too.

Solution 2.7.11: See my website.

2.8 (Limits of) Empirical Performance Measures

If the arrival and service processes are such that the queueing system is rate-stable, we can sensibly define other performance measures such as the average waiting time. In this section we define the second most important performance measures; the most important being the utilization ρ. We provide an overview of the relations between these performance measures in Figure 2.11.

With the construction of queueing processes in Section 2.5 we can compute the waiting time as observed by the first n, say, jobs. Thus, the average waiting time of the first n arrivals is given by $n^{-1}\sum_{k=1}^n W_k$. We therefore define the *expected waiting time* as

$$\mathsf{E}\{W\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n W_k, \tag{2.29}$$

and the expected time in queue as

$$\mathsf{E}\{W_Q\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n W_{Q,k}. \tag{2.30}$$

Note that these performance measures are limits of *emperical* measures. Note also that these statistics are as *observed by arriving jobs*: the first job has a waiting time W_1 at its arrival epoch, the second a waiting time W_2, and so on. For this reason we colloquially say that $\mathsf{E}\{W\}$ is the average waiting time as 'seen by arrivals'. The *distribution of the waiting times at arrival times* can be found by counting:

$$\mathsf{P}\{W \leq x\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n \mathbb{1}_{W_k\leq x}. \tag{2.31}$$

Finally, the (sample) *average number of jobs* in the system as seen by arrivals is given by

$$\mathsf{E}\{L\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n L_k, \tag{2.32}$$

where $L_k = L(A_k)$, i.e., the number in system at the arrival epoch of the kth job. The *distribution of* $\{L(t)\}$ *as seen by customers upon arrival*, is

$$\mathsf{P}\{L \leq m\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n \mathbb{1}_{L_k\leq m}. \tag{2.33}$$

A related set of performance measures follows by tracking the system's behavior over time and taking the *time-average*, rather than the average at sampling (observation) moments. Thus, if we simulate the queueing system up to time t, the *time-average number of jobs in the system* is given by

$$\mathscr{L}(t) = \frac{1}{t}\int_0^t L(s)\,\mathrm{d}s = \frac{1}{t}\int_0^t (A(s)-D(s))\,\mathrm{d}s, \tag{2.34}$$

where we use that $L(t) = A(t) - D(t)$ is the total number of jobs in the system at time t, c.f. Figure 2.7. Observe from the second equation that $\int_0^t L(s)\,\mathrm{d}s$ is the area enclosed between the graphs of $\{A(t)\}$ and $\{D(t)\}$. Assuming the limit exists for $t\to\infty$, we define

$$\mathsf{E}\{L\} = \lim_{t\to\infty} \frac{1}{t}\int_0^t L(s)\,\mathrm{d}s \tag{2.35}$$

Observe that, notwithstanding that the symbols are the same, this expectation need not be the same as (2.32), c.f. on of the exercises below. Next, define the following probability as the *time-average fraction of time the system contains at most m jobs*:

$$\mathsf{P}\{L \leq m\} = \lim_{t\to\infty} \frac{1}{t}\int_0^t \mathbb{1}_{L(s)\leq m}\,\mathrm{d}s. \tag{2.36}$$

Again, this probability need not be the same as what customers see upon arrival.

It is evident how to formulate the counterparts for the waiting time process $\{W(t)\}$.

Exercises

Exercise 2.8.1. Design a queueing system in which the load as seen the server is very different from what the customers see. Show that for such a queueing system typically

$$\lim_{t\to\infty} \frac{1}{t}\int_0^t L(s)\,\mathrm{d}s \neq \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n L_k.$$

where, by (2.20), $L_k = L(A_k)$ is the number in the system at the arrival epoch of the kth job. Hence, in general:

$$\lim_{t\to\infty} \int_0^t \mathbb{1}_{L(s)\leq m}\,\mathrm{d}s \neq \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^n \mathbb{1}_{L_k\leq m}.$$

Exercise 2.8.2.

1. Assume that $X_k = 10$ minutes and $S_k = 11$ minutes for all k, i.e., X_k and S_k are deterministic and constant. What are λ and μ? Compute A_k, W_k, D_k and L_k. How do these numbers behave as a function of k?.
2. Yet another simple case is to take $X_k = 10$ minutes and $S_k = 9$ minutes for all k. Answer the same questions as in the previous part.

Exercise 2.8.3. Consider a discrete-time model of a queueing system, as we developed in Section 2.4. In that case batches of jobs can arrive in a period, and the average number of customers that *see upon arrival* more than m customers in the system cannot be defined as (2.33). Provide a better definition.

Hints

Hint 2.8.1: Consider a queueing system with constant service and interarrival times.

Hint 2.8.3: Why is (2.33) not the same as the number of batches that see a queue length less than m?

Solutions

Solution 2.8.1: Take $X_k = 10$ and $S_k = 10-\epsilon$ for some tiny ϵ. Then $L(t) = 1$ nearly all of the time. In fact, $\mathsf{E}\{L\} = 1-\epsilon/10$. However, $L_k = 0$ for all k.

Solution 2.8.2:

1. $\lambda = 6$ per hour, and $\mu = 60/11$ per hour. Note that $\mu < \lambda$. $A_0 = 0$, $A_1 = 10$, $A_2 = 20$, etc., hence $A_k = 10k$. $W_{Q,0} = 0$, $W_{Q,1} = \max\{0+0-10, 0\} = 0$. $W_{Q,2} = \max\{0+11-10, 0\} = 1$. $W_{Q,3} = \max\{1+11-10, 0\} = 2$. Hence, $W_{Q,k} = k-1$ for $k \geq 1$. Thus, $W_k = k-1+11 = k+10$ for $k \geq 1$, and $D_k = 10k+k+10 = 11k+10$. Note that W_k increases linearly as a function of k. You can infer that, for the queue length to remain bounded, it is necessary that the service rate $\mu > \lambda$.

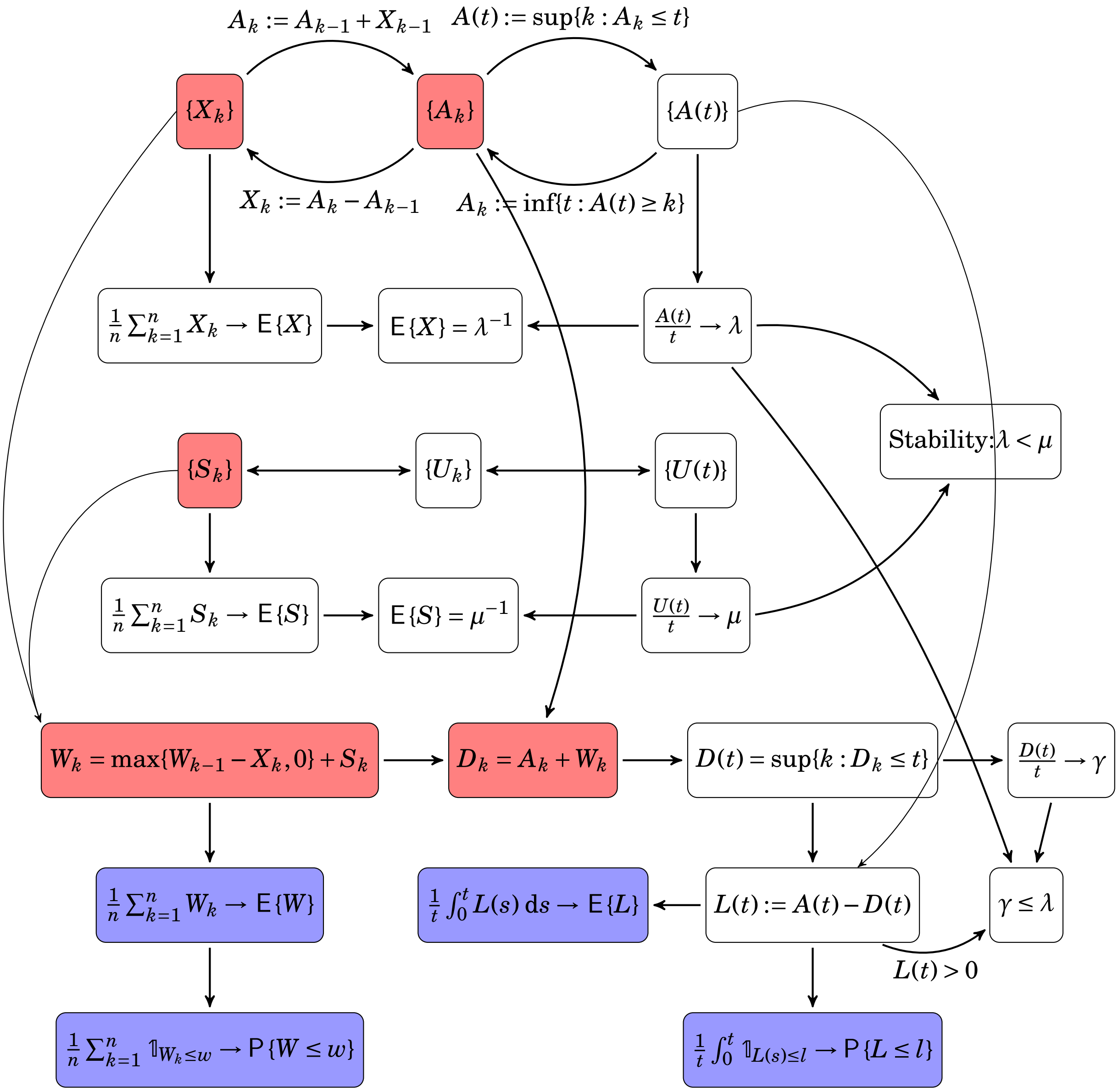


Figure 2.11: Here we sketch the relations between the construction of the $G/G/1$ queue from the primary data, i.e., the interarrival times $\{X_k; k \ge 0\}$ and the service times $\{S_k; k \ge 0\}$, and different performance measure.

2. Trivial.

Solution 2.8.3: Since we deal with a system in discrete time L_k is the queue length at the end of period k. Thus, $\sum_{k=1}^n \mathbb{1}_{L_k>m}$ counts the number of *periods* that the queue is larger than m. This is of course not the same as the number of *jobs* that see a queue larger than m; only when $a_k > 0$ the customers in a batch would see a queue $L_k > m$. Thus,

$$\sum_{k=1}^n \mathbb{1}_{L_k>m} \mathbb{1}_{a_k>0},$$

counts the number of batches.

Next, by assumption, a_k customers arrive during period k. The first of these customers sees a queue length of $L_{k-1} - d_k$, the second $L_{k-1} - d_k + 1$, and so on until the last customer who sees a queue length of $L_{k-1} - d_k + a_k - 1 = L_k - 1$. Thus, of all jobs the last sees the largest queue. Hence, if $L_k \leq m$, all customers of the batch see a queue less than m. If, however, $L_k > m$, then $L_k - m$ customers saw m or more jobs in the system.. Therefore, the fraction of arrivals that see a queue with m or more jobs is equal to

$$\frac{1}{A(n)} \sum_{k=1}^n (L_k - m) \mathbb{1}_{L_k>m}.$$

We could also define this a bit differently. Suppose that we don't want to distinguish between jobs in a batch, but simply want to say that if one job sees a long queue, all see a long queue. In that case,

$$\frac{1}{A(n)} \sum_{k=1}^n a_k \mathbb{1}_{L_k>m}.$$

Thus, when jobs arrive in batches, the definition of loss fraction requires some care; not all definitions need to measure the same.

2.9 Level-Crossing and Balance Equations

Consider a system at which customers arrive and depart in single entities, such as customers in a shop or jobs at some machine. If the system starts empty, then we know that the number $L(t)$ is the system at time t is equal to $A(t) - D(t)$, see Figure 2.12. Let us denote an arrival as an 'up-crossing' and a departure as a 'down-crossing'. Then, clearly $L(t)$ is the number of up-crossings up to time t minus the number of down-crossings up to time t. If $L(t)$ remains finite, or more generally $\lim_t L(t)/t = 0$, then it must be that

$$\lambda = \lim_t \frac{A(t)}{t} = \lim_t \frac{D(t) + L(t)}{t} = \lim_t \frac{D(t)}{t} + \lim_t \frac{L(t)}{t} = \gamma.$$

Hence, when $L(t)/t \to 0$, the *up crossing rate* $\lim_t A(t)/t = \lambda$ is equal to the *down-crossing rate* $\lim_t D(t)/t = \gamma$. We will generalize these notions of up- and downcrossing in this section to derive the *stationary*, also know as *long-run time average* or *steady-state*, distribution $p(n)$ that the system contains n jobs.

Let us say that the system is in *state n when it contains n jobs*, c.f. Figure 2.14, and the system *crosses* level n when the state changes from n to $n+1$, either 'from below' when an

$$\longrightarrow A(t) \longrightarrow \boxed{L(t) = A(t) - D(t)} \longrightarrow D(t) \longrightarrow$$

Figure 2.12: What goes in the box (i.e., $A(t)$) and has not yet left (i.e., $D(t)$) must be in the box, hence $L(t) = A(t) - D(t)$.

arrival occurs, or 'from above' when a departure occurs. The main result we derive in this section is the rate at which level n is crossed from below and above must match, i.e.,

$$\lambda(n)p(n) = \mu(n+1)p(n+1),$$

where $\lambda(n)$ is the arrival rate in state n, $p(n)$ the fraction of time the system is in state n and $\mu(n+1)$ the departure rate from $n+1$. To establish this we need a few definitions that are quite subtle and might seem a bit abstract, but below we will provide intuitive interpretations in terms of system KPIs. Once we have the proper definitions, the above result will follow straightaway.

Define[5]

$$A(n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k \leq t} \mathbb{1}_{L(A_k-)=n} \tag{2.37a}$$

as the number of arrivals up to time t that saw n customers in the system at their arrival, c.f. Figure 2.13,

$$Y(n,t) = \int_0^t \mathbb{1}_{L(s)=n} \, \mathrm{d}s \tag{2.37b}$$

as the total time the system contains n jobs during $[0,t]$, and

$$p(n,t) = \frac{1}{t}\int_0^t \mathbb{1}_{L(s)=n} \, \mathrm{d}s = \frac{Y(n,t)}{t}, \tag{2.37c}$$

as the fraction of time that $L(s) = n$ in $[0,t]$ (Recall that $L(t)$ is *right-continuous* in the definition of $A(n,t)$.)

Define also the limits:

$$\lambda(n) = \lim_{t\to\infty} \frac{A(n,t)}{Y(n,t)}, \quad p(n) = \lim_{t\to\infty} p(n,t),$$

as the *arrival rate in state* n and the *long-run fraction of time the system spends in state* n. To clarify the former definition, observe that $A(n,t)$ counts the number of arrivals that see n jobs in the system, while $Y(n,t)$ tracks the amount of time the system contains n jobs. Suppose that at T a job arrives that sees n in the system. Then $A(n,T) = A(n,T-)+1$, and this job arrives in an interval that it tracked by $Y(n,t)$, precisely because this job sees n in the system just prior to its arrival. Thus, just as $A(t)/t$ is the total number of arrivals during $[0,t]$ divided by t, $A(n,t)/Y(n,t)$ is the number of arrivals that see n divided by the time the system contains n jobs.

[5] It is common to write $f(x+) = \lim_{h\downarrow 0} f(x+h)$ for the right limit of a function f at x and $f(x-) = \lim_{h\downarrow 0} f(x-h)$ for the left limit. When $f(x-) = f(x+)$, f is continuous at x. Since in queueing systems we are concerned with processes with jumps, we need to be quite particular about left and right limits at jump epochs.

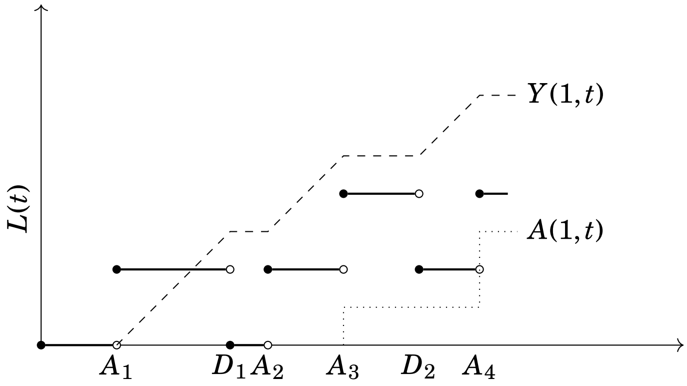


Figure 2.13: Plots of $Y(1,t)$ and $A(1,t)$. (For visual clarity, we subtracted ½ from $A(1,t)$, for otherwise its graph would partly overlap with the graph of L.)

Similarly, denoting by

$$D(n,t) = \sum_k \mathbb{1}_{D_k \leq t} \mathbb{1}_{L(D_k)=n}$$

the number of departures up to time t that *leave n customers behind*, we define

$$\mu(n+1) = \lim_{t\to\infty} \frac{D(n,t)}{Y(n+1,t)},$$

as *the departure rate from state* $n+1$. (It is easy to get confused here: to leave n jobs behind, the system must contain $n+1$ jobs just prior to the departure.) Figure 2.14 shows how $A(n,t)$ and $\lambda(n)$ relate to $D(n+1,t)$ and $\mu(n)$.

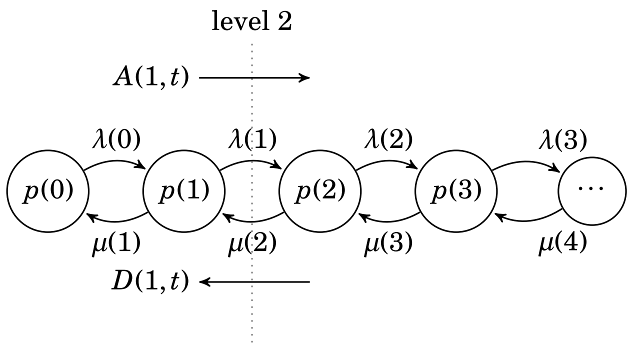


Figure 2.14: $A(1,t)$ counts the number of jobs up to time t that saw 1 job in the system; thus, right after the arrival of such a job the system contains 2 jobs. Similarly, $D(1,t)$ counts the number of departures that leave 1 job behind. Each time $A(1,t)$ increases by one or $D(1,t)$ decreases by one, level 2 (the dotted line separating states 1 and 2) is crossed. The number of times this level is crossed from below must be the same (plus or minus 1) the number of times it is crossed from above.

Observe that customers arrive and depart as single units. Thus, if $\{T_k\}$ is the ordered set of arrival and departure times of the customers, then $L(T_k) = L(T_k-) \pm 1$. But then we must also have that $|A(n,t) - D(n,t)| \leq 1$ (Think about this.). From this observation it follows immediately that

$$\lim_{t\to\infty}\frac{A(n,t)}{t} = \lim_{t\to\infty}\frac{D(n,t)}{t}. \tag{2.38}$$

With this equation we can obtain two nice and fundamental identities. The first we develop now; the second follows in Section 2.12.

The rate of jobs that 'see the system with n jobs' can be defined as $A(n,t)/t$. Taking limits we get

$$\frac{A(n,t)}{t} = \frac{A(n,t)}{Y(n,t)}\frac{Y(n,t)}{t} \to \lambda(n)p(n), \tag{2.39a}$$

where we use the above definitions for $\lambda(n)$ and $p(n)$. Similarly, the departure rate of jobs that leave n jobs behind is

$$\frac{D(n,t)}{t} = \frac{D(n,t)}{Y(n+1,t)}\frac{Y(n+1,t)}{t} \to \mu(n+1)p(n+1). \tag{2.39b}$$

Combining this with (2.38) we arrive at *the level-crossing equations*

$$\lambda(n)p(n) = \mu(n+1)p(n+1). \tag{2.40}$$

This result turns out to be exceedingly useful. Once we can specify $\lambda(n)$ and $\mu(n)$, we can compute the long-run fraction of time $p(n)$ that the system contains n jobs. To see this, rewrite the above into

$$p(n+1) = \frac{\lambda(n)}{\mu(n+1)}p(n). \tag{2.41}$$

Thus, if we have $p(n)$ we can compute $p(n+1)$, and so on. In other words, if $p(0)$ is known, than $p(1)$ follows, from which $p(2)$ follows, and so on. A straightaway iteration then leads to

$$p(n+1) = \frac{\lambda(n)\lambda(n-1)\cdots\lambda(0)}{\mu(n+1)\mu(n)\cdots\mu(1)}p(0). \tag{2.42}$$

Finally, to determine $p(0)$ we can use the fact that numbers $p(n)$ have to be normalized. Let the *normalization constant* be given by

$$G = 1 + \frac{\lambda(0)}{\mu(1)} + \frac{\lambda(0)\lambda(1)}{\mu(1)\mu(2)} + \cdots. \tag{2.43}$$

Then, $p(0) = G^{-1}$ and $p(n)$ follows from Eq. (2.42).

Let us now express a few important performance measures in terms of $p(n)$: the average number of items $\mathsf{E}\{L\}$ in the system and the fraction of time $\mathsf{P}\{L \leq n\}$ the system contains at least n jobs. As $L(s)$ counts the number of jobs in the system at time s (thus $L(s)$ is an integer),

$$L(s) = \sum_{n=0}^{\infty} n\, \mathbb{1}_{L(s)=n}.$$

With this we can write for the time-average number of jobs in the system

$$\mathcal{L}(t) = \frac{1}{t}\int_0^t \left(\sum_n n\, \mathbb{1}_{L(s)=n}\right) \mathrm{d}s = \sum_n \frac{n}{t}\int_0^t \mathbb{1}_{L(s)=n}\, \mathrm{d}s, \tag{2.44}$$

where we interchange the integral and the summation[6]. It then follows from Eq. (2.37c) that

$$\mathscr{L}(t) = \sum_n n\, p(n,t).$$

Finally, assuming that the limit $p(n,t) \to p(n)$ exists as $t \to \infty$ (and that the summation and limit can be interchanged in the above), it follows that

$$\mathscr{L}(t) \to \sum_{n=0}^{\infty} n p(n) = \mathsf{E}\{L\}, \text{ as } t \to \infty.$$

In words, $\mathscr{L}(t)$ converges to the long-run time average $\mathsf{E}\{L\}$ of the number in the system. In a loose sense we can say that $\mathsf{E}\{L\}$ is the average number in the system as perceived by the *server*. (Recall that this is not necessarily the same as what *arriving* jobs 'see'). Similarly, the probability that the system contains at least n jobs is

$$\mathsf{P}\{L \geq n\} := \sum_{i=n}^{\infty} p(i).$$

From the above we conclude that from the probabilities $p(n)$, more specifically, from a specification of $\lambda(n)$ and $\mu(n)$, we can compute numerous performance measures. Thus, determining $p(n)$ from Eqs. (2.41) and (2.43) for concrete queueing examples is the task of the next sections.

Before we embark on this task, we note that the level-crossing cannot always be used as we do here. The reason is that it is not always natural, or easy, to decompose the state space into two disjoint parts. For a more general approach, we focus on a single state and count how often this state is entered and left, c.f. Figure 2.15. Specifically, define

$$I(n,t) = A(n-1,t) + D(n,t),$$

as the number of times the queueing process enters state n either due to an arrival from state $n-1$ or due to a departure leaving n jobs behind. Similarly,

$$O(n,t) = A(n,t) + D(n-1,t),$$

counts how often state n is left either by an arrival or a departure to state $n-1$.

Of course, $|I(n,t) - O(n,t)| \leq 1$. Thus, from the fact that

$$\frac{I(n,t)}{t} = \frac{A(n-1,t)}{t} + \frac{D(n,t)}{t} \to \lambda(n-1)p(n-1) + \mu(n+1)p(n+1)$$

and

$$\frac{O(n,t)}{t} = \frac{A(n,t)}{t} + \frac{D(n-1,t)}{t} \to \lambda(n)p(n) + \mu(n)p(n)$$

we get that

$$\lambda(n-1)p(n-1) + \mu(n+1)p(n+1) = (\lambda(n) + \mu(n))p(n).$$

These equations hold for any $n \geq 0$ and are known as the *balance equations*. We will use these equations when studying queueing systems in which level crossing cannot be used, for instance for queueing networks.

Again, just by using properties, i.e., counting differences, that hold along any sensible sample path we obtain very useful statistical and probabilistic results.

[6] This is allowed as the integrand is non-negative. More generally, the interested reader should check Fubini's theorem.

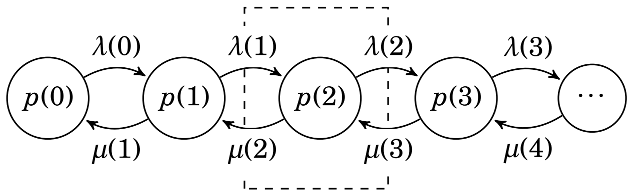


Figure 2.15: For the balance equations we count how often a box around a state is crossed from inside and outside. On the long run the entering and leaving rates should be equal. For the example here, the rate out is $p(2)\lambda(2) + p(2)\mu(2)$ while the rate in is $p(1)\lambda(1) + p(3)\mu(3)$.

Interpretation The definitions in (2.37) may seem a bit abstract, but they obtain an immediate interpretation when relating them to applications. To see this, we discuss two examples.

Consider the sorting process of post parcels at a distribution center of a post delivery company. Each day tens of thousands of incoming parcels have to be sorted to their final destination. In the first stage of the process, parcels are sorted to a region in the Netherlands. Incoming parcels are deposited on a conveyor belt. From the belt they are carried to outlets (chutes), each chute corresponding to a specific region. Employees take out the parcels from the chutes and put the parcels in containers. The arrival rate of parcels for a certain chute may temporarily exceed the working capacity of the employees, as such the chute serves as a queue. When the chute overflows, parcels are directed to an overflow container and are sorted the next day. The target of the sorting center is to deliver at least a certain percentage of the parcels within one day. Thus, the fraction of parcels rejected at the chute should remain small.

Suppose a chute can contain at most 20 parcels, say. Then, each parcel on the belt that 'sees' 20 parcels in its chute will be blocked. Let $L(t)$ be the number of parcels in the chute at time t. Then, $A(20,t)$ as defined in Eq. (2.37a) is number of *blocked parcels* up to time t, and $A(20,t)/A(t)$ is the fraction of rejected parcels. In fact, $A(20,t)$ and $A(t)$ are continuously tracked by the sorting center and used to adapt employee capacity to control the fraction of rejected parcels. Thus, in simulations, if one want to estimate loss fractions, $A(n,t)/A(t)$ is the most natural concept to consider.

For the second example, suppose there is a cost associated with keeping jobs in queue. Let w be the cost per job in queue per unit time so that the cost rate is nw when n jobs are in queue. But then is $wnY(n,t)$ the total cost up to time t to have n jobs in queue, hence the total cost up up to time t

$$C(t) = w \sum_{n=0}^{\infty} nY(n,t),$$

and the average cost is

$$\frac{C(t)}{t} = w \sum_{n=0}^{\infty} n \frac{Y(n,t)}{t} = w \sum_{n=0}^{\infty} np(n,t).$$

All in all, the concepts developed above have natural interpretations in practical queueing situations; they are useful in theory and in simulation, as they relate the theoretical concepts to actual measurements.

Exercises

Exercise 2.9.1. Should we take $L(A_k-) = n$ or $L(A_k)$ in the definition of $A(n,t)$?

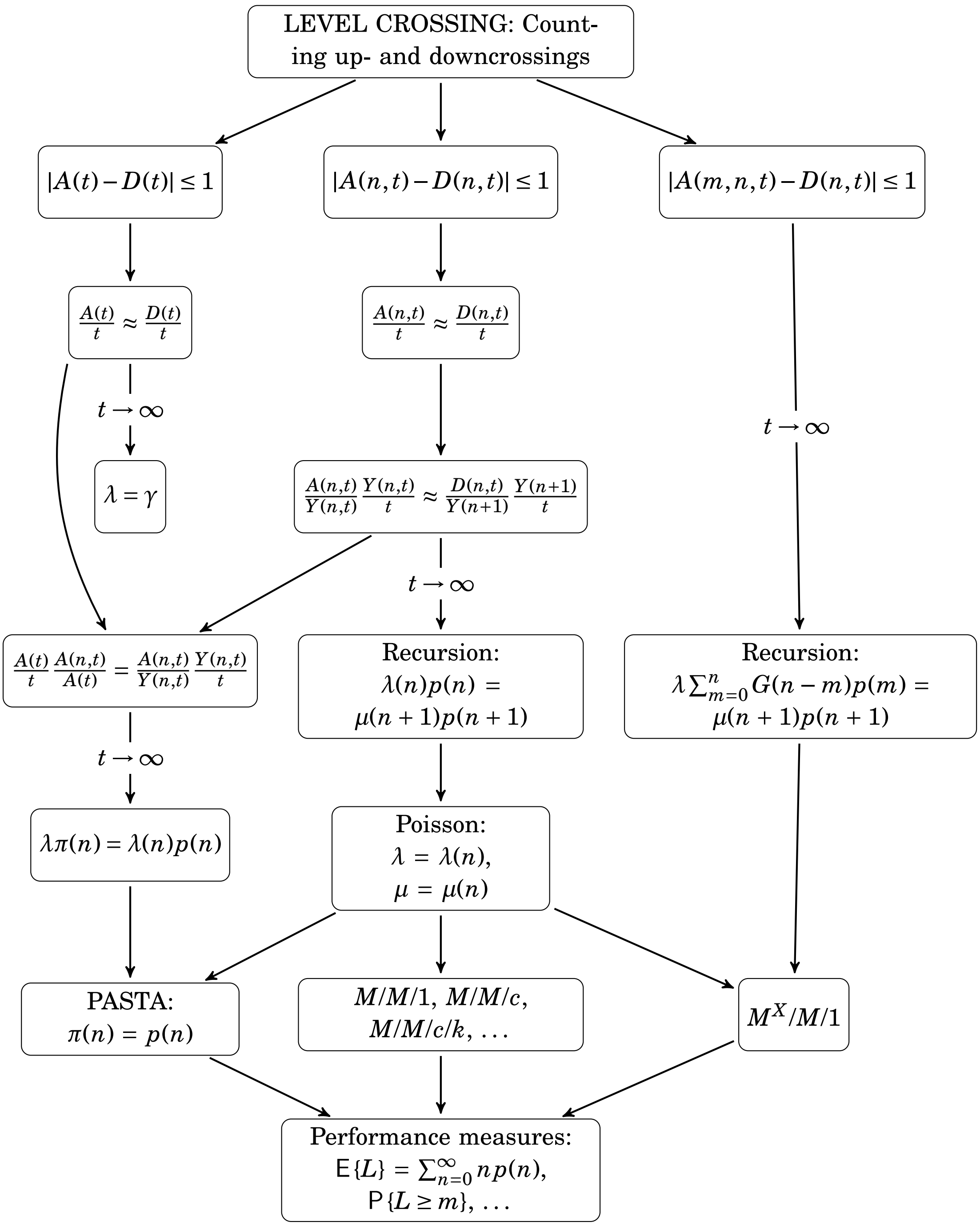


Figure 2.16: With level-crossing arguments we can derive a number of useful relations. This figure presents an overview of these relations that we derive in this and the next sections.

Exercise 2.9.2. Should we take $D(n-1,t)$ or $D(n,t)$ in the definition of $\mu(n)$?

Exercise 2.9.3. What is the difference between $A(n,t)$ and $A(t)$?

Exercise 2.9.4. Show that $A(n,t) \leq A(t)$.

Exercise 2.9.5. Why is $\lim_{t\to\infty} \frac{A(n,t)}{t} = 0$ if $\lambda > \gamma$, i.e., if the system is not rate-stable?

Exercise 2.9.6. Is it always true that $\lambda = \lim_t \frac{A(t)}{t} = \lim_t \frac{A(n,t)}{t}$.

Exercise 2.9.7. Consider the following (silly) queueing process. At times $0,2,4,\ldots$ customers arrive, each customer requires 1 unit of service, and there is one server. Find an expression for $Y(n,t)$.What acronym would describe this queueing situation?

Exercise 2.9.8. Consider a single server that serves one queue and serves only in batches of 2 jobs at a time (so never 1 job or more than 2 jobs), i.e., the $M/M^2/1/3$ queue. Single jobs arrive at rate λ and the interarrival times are exponentially distributed. The batch service times are also exponentially distributed with mean $1/\mu$. At most 3 jobs fit in the system. Make a drawing
Make a graph of the state-space and show, with arrows, the transitions that can occur.

Exercise 2.9.9. Use the graph of the previous question and a level-crossing argument to express the steady-state probabilities $p(n), n = 0,\ldots,3$ in terms of λ and μ.

Hints

Hint 2.9.7: For the acronym, observe that the service times and interarrival are *D*eterministic and there is one server. For the computation of $Y(n,t)$, make make a plot of $L(s)$ as a function of time for $n = 1$.

Hint 2.9.9: First balance the rates accross the levels. Then solve for $\pi(k)$.

Solutions

Solution 2.9.1: $L(t)$ is the number of customers in the system at time t. As such the function $t \to L(t)$ is *right-continuous*. The definition of $L(A_k-) = \lim_{t\uparrow A_k} L(t)$, i.e., the limit from the left. The customer therefore 'sees' $L(A_k-)$ just before he arrives.

Solution 2.9.2: $D(n-1,t)$ counts the departures that leave $n-1$ behind. Thus, just before the customer leaves, the system contains n customers.

Solution 2.9.3: $A(t)$ counts all customers that arrive up to time t, i.e., during $[0,t]$. Note that this *includes* time t. $A(n,t)$ counts the jobs that see n jobs in the system just before they arrive.

Solution 2.9.4: Observe that $\mathbb{1}_{A_k\leq t}\mathbb{1}_{L(A_k-)=n} = \mathbb{1}_{A_k\leq t}\mathbb{1}_{L(A_k-)=n} \leq \mathbb{1}_{A_k\leq t}$; the last inequality follows from the fact that $\mathbb{1}_{L(A_k-)=n} \leq 1$. In fact, most of the time, for a particular n, $\mathbb{1}_{L(A_k-)=n} = 0$. Therefore

$$A(n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k\leq t}\mathbb{1}_{L(A_k-)=n} \leq \sum_{k=1}^{\infty} \mathbb{1}_{A_k\leq t=A(t)}.$$

For any 'normal' queueing system, $A(t) > A(t,n)$, because the queue length fluctuates.

Solution 2.9.5: If $\lambda > \gamma$, then $L(t) \to \infty$. But then there must be last time, s say, that $L(s) = n+1$, and $L(t) > n+1$ for all $t > s$. Hence, after time s no job will see the system with n jobs. Thus $A(t,n) = A(s,n)$ for all $t > s$. This is a finite number, while $t \to \infty$, so that $A(n,t)/t \to 0$.

Solution 2.9.6: When $\gamma < \lambda$, λ can still be finite, but the server is simply too slow. In that case both limits can exists, the first limit is positive but the second is 0.

Note that $\gamma \leq \lambda$ (always): jobs cannot depart faster than they arrive. Hence, we do not have to consider the case $\gamma > \lambda$ in the above.

Solution 2.9.7: First, it is the $D/D/1$ queue, since there is one server and the interarrival times and service times are constant, i.e., *D*eterministic.

Next, to get $Y(n,t)$, observe that the system never contains more than 1 job. Hence, $Y(n,t) = 0$ for all $n \geq 2$. Then, make a plot of $L(s)$ for $n = 1$ as a function of time. function of time. Then we see that $Y(1,t) = \int_0^t \mathbb{1}_{L(s)=1}\,\mathrm{d}s$. Now observe that for our queueing system $L(s) = 1$ for $s \in [0,1)$, $L(s) = 0$ for $s \in [1,2)$, $L(s) = 1$ for $s \in [2,3)$, and so on, Thus, when $t < 1$, $Y(1,t) = \int_0^t \mathbb{1}_{L(s)=1}\,\mathrm{d}s = \int_0^t 1\,\mathrm{d}s = t$. When $t \in [1,2)$,

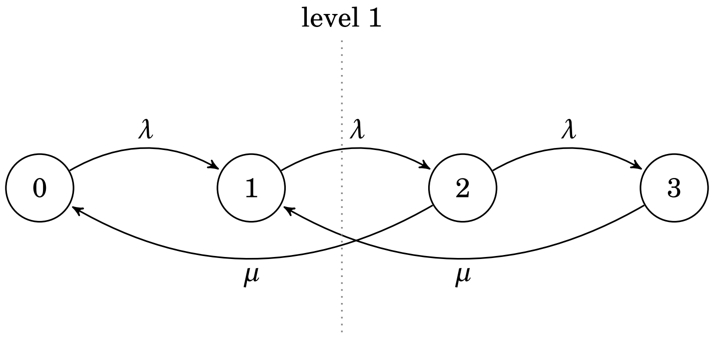
$$L(t) = 0 \implies \mathbb{1}_{L(t)=0} \implies Y(1,t) \text{ does not change.}$$

Continuing to $[2,3)$ and so on gives

$$Y(1,t) = \begin{cases} t & \leq t \in [0,1), \\ 1 & \leq t \in [1,2), \\ 1+(t-2) & \leq t \in [2,3), \\ 2 & \leq t \in [3,4), \\ 2+(t-4) & \leq t \in [4,5), \end{cases}$$

and so on. Since $Y(n,t) = 0$ for all $n \geq 2$, $L(s) = 1$ or $L(s) = 0$ for all s, therefore,

$$Y(0,t) = t - Y(1,t).$$



Solution 2.9.8:

Solution 2.9.9: With level crossing

$$\begin{aligned} \lambda p(0) &= \mu p(2) \\ \lambda p(1) &= \mu p(2) + \mu p(3), \quad \text{see level 1} \\ \lambda p(2) &= \mu p(3) \end{aligned}$$

Solving this in terms of $p(0)$ gives $p(2) = \rho p(0)$, $p(3) = \rho p(2) = \rho^2 p(0)$, and $p(1) = (\lambda+\mu)/\lambda p(0)$.

2.10 M/M/1 queue

In the $M/M/1$ queue, one server serves jobs arriving with exponentially distributed interarrival times and each requiring an exponentially distributed processing time. With Eq. (2.41), i.e., $\lambda(n)p(n) = \mu(n+1)p(n+1)$ we can derive a number of important results for this queueing process.

Recall from Section 2.6 that we can construct the $M/M/1$ queue as a reflected random walk where the arrivals are generated by a Poisson process $N_\lambda(t)$ and the departures (provided the number in the $L(t) > 0$) are generated according to the Poisson process $N_\mu(t)$. Since the rates of these processes do not depend on the state of the random walk, or the queue for that matter, $\lambda(n) = \lambda$ and $\mu(n) = \mu$ for all n. Thus, (2.41) reduces to

$$p(n+1) = \frac{\lambda(n)}{\mu(n+1)} p(n) = \frac{\lambda}{\mu} p(n) = \rho p(n),$$

where we use the definition of the load $\rho = \lambda/\mu$. Since this holds for any $n \geq 0$, it follows with recursion that

$$p(n+1) = \rho^{n+1} p(0).$$

Then, from the normalization condition

$$1 = \sum_{n=0}^{\infty} p(n) = p(0) \sum_{n=0}^{\infty} \rho^n = \frac{p(0)}{1-\rho},$$

so

$$p(0) = 1-\rho, \qquad\qquad p(n) = (1-\rho)\rho^n. \tag{2.45}$$

How can we use these equations? First, note that $p(0)$ must be the fraction of time the server is idle. Hence, the fraction of time the server is busy, i.e., the utilization, is

$$1 - p(0) = \rho = \sum_{n=1}^{\infty} p(n).$$

Here the last equation has the interpretation of the fraction of time the system contains at least 1 job. Next, in the exercises below we derive that

$$\mathsf{E}\{L\} = \frac{\rho}{1-\rho}. \tag{2.46}$$

and

$$\mathsf{P}\{L \geq n\} = \rho^n. \tag{2.47}$$

Let us interpret these expressions. First of all, we only need estimates of λ and μ to characterize the utilization of the server, the average queue length and the queue length distribution. In fact, only the ratio $\rho = \lambda/\mu$ is required. As such, assuming that interarrival times and services are exponentially distributed, by measuring the fraction of time the server is busy, we obtain an estimate for $1 - p(0)$, hence for ρ. With this, we can estimate $\mathsf{E}\{L\}$ and $\mathsf{P}\{L \geq n\}$. Next, the fact that $\mathsf{E}\{L\} \sim (1-\rho)^{-1}$ for $\rho \to 1$ implies that the average waiting time *increases asymptotically fast* to infinity when $\rho \to 1$. In practical terms, this means that striving for a high load results in (very) long average waiting times. As a consequence, this formula tells us that, in the design and operation of queueing systems, we should avoid the situation in which

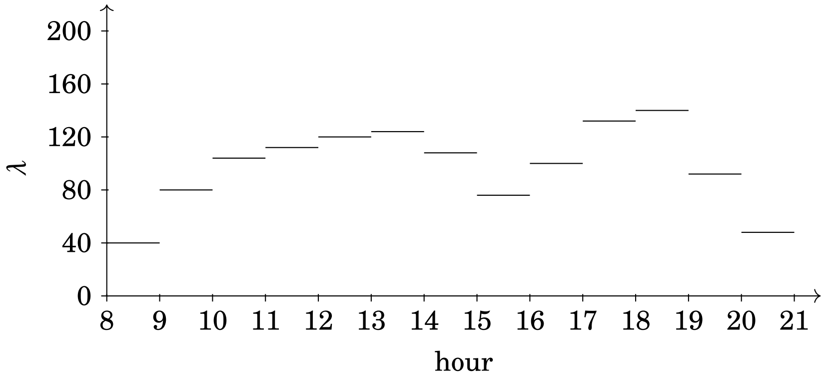


Figure 2.17: A demand profile of the arrival rate λ modeled as constant over each hour.

$\rho \approx 1$. Thus, we need to make a trade-off between the server utilization $1 - p(0) = \rho$ and the waiting time. In high load regimes, the server is used efficiently, but the queue lengths are excessive. In any sensible system, we should keep ρ quite a bit below 1, and accept lower utilization then might seem necessary.

The expression for $\mathsf{P}\{L \geq n\}$ shows that the probability that the queue length exceeds some threshold decreases exponentially fast (for any reasonable ρ). Moreover, if we make the simple assumption that customers decide to leave (or rather, not join) the system when the queue is longer than 9 say, then $\mathsf{P}\{L \geq 10\} = \rho^{10}$ is an estimator of the fraction of customers lost. Again, for the $M/M/1$ queue, the knowledge of ρ suffices to compute this estimate.

Supermarket Planning Let us consider the example of cashier planning of a supermarket to demonstrate how to use the tools we developed up to now. Out of necessity, our approach is a bit heavy-handed—Turning the example into a practically useful scheme requires more sophisticated queueing models and data assembly—but the present example contains the essential analytic steps to solve the planning problem.

The *service objective* is to determine the minimal service capacity such that the fraction of the time the queue length exceeds 10 is less than 1%.

The next step is to find the *relevant data*. For a supermarket this is relatively easy: the cash registers track all customers payments. Thus, we know the number of customers that left the shop, hence entered the shop. (We neglect the time customers spend in the shop.) Based on these data we can make a *demand profile*: the customer arrival rate per hour, c.f. Figure 2.17. It is also easy to find the service distribution from the cash registers. The first item scanned after a payment determines the start of a new service, and the payment closes the service. (As there is always a bit of time between the payment and the start of a new service we might add 15 seconds, say, to any service.)

To keep things simple, we model the arrival process as Poisson with an arrival rate that is constant during an hour and is specified by the demand profile, and we take the service time distribution as exponential with a mean of 1.5 minutes. We also *model* the behavior of the multi-server as a single fast server. Thus, we neglect any differences between a station with, for instance, 5 cashiers and a single server that works 5 times as fast as a normal cashier. As yet another simplification, we change the objective somewhat such that the number of jobs in the queueing system should not exceed 10. (Thus, the objective is no longer formulated in terms of queue lengths.)

To determine the required number of servers per hour let us consider the hour from 12 to 13. In the demand profile we see that the arrival rate in this hour is $\lambda_{12} = 120$ customers per hour. The objective $\mathsf{P}\{L > 10\} \leq 1\%$ translates into $\rho^{11} \leq 1\%$, hence, $\rho \leq 0.67$. Using that $\mathsf{E}\{S\} = 1.5$ minutes, we need the number of servers c to be such that

$$\frac{120}{60} 1.5 \frac{1}{c} = \rho < 0.66,$$

hence $c \approx 5$. Interestingly, for this model we only need to be concerned with $\lambda \mathsf{E}\{S\}$ to meet the objective. Therefore c is very simple to compute by the result

$$c \geq \lambda \frac{\mathsf{E}\{S\}}{0.66} = \lambda 2.3 \approx 2.5\lambda.$$

where λ is the arrival rate (per minute, *not* per hour). Thus, the conversion of the demand profile to a *load* profile is trivial: divide the hourly arrival rate by 60 and multiply by 2.5. The load profile tells us the minimal number of cashiers needed per hour.

Now we need to *cover the load profile with service shifts*. This is typically not easy since shifts have to satisfy all kinds of rules, e.g.: after 2 hours of work a cashier should take a break of at least 10 minutes; a shift length must be at least four hours, and not longer than 9 hours including breaks; when the shift is longer than 4 hours it needs to contain at least one break of 30 minutes; and so on. These shifts also have different costs, e.g.: shifts with hours after 18h are more expensive per hour; when the supermarket covers traveling costs, short shifts have higher marginal travel-ling costs; and so on.

The usual way to solve the covering problem is by means of an integer problem. First generate all (or a subset of the) allowed shift types with associated starting times. For instance, suppose only 4 shift plans are available

1. $++-++$
2. $+++-+$
3. $++-+++$
4. $+++-++$,

where a $+$ indicate a working hour and $-$ a break of an hour. Then generate shift types for each of these plans with starting times 8am, 9am, and on on, until the end of the day. Thus, a shift type is a shift plan that starts at a certain hour. Let x_i be the number of shift type i and c_i the cost of this type. Write $t \in s_i$ if hour t is covered by shift type i. Then the problem is given by

$$\min \sum_i c_i x_i.$$

such that

$$\sum_i x_i \mathbb{1}_{t \in s_i} \geq \frac{\lambda_t}{20}$$

for all hours t the shop is open and λ_t is the demand for hour t.

Conceptually it is easy how to make this approach more accurate and realistic. The queueing model is too simple, so as start we could replace it by the multi-server model (with exponential service times) we will discuss in Section 2.11. If the data tells us that the exponential distribution for the service times is very inaccurate, we might use the results of Section 2.20, however, this only provides us with the mean waiting times for the $M/G/c$ queue, not the queue length distribution, which is a drawback. (The analytic results for the queue lenght distribution of the $G/G/c$ queue are not so easy and also not very accurate. Whether it makes sense to prefer these expressions over the closed-form results of the $M/M/c$ queue is not so easy to say. It depends on the context.) Finally, the objective should depend on the number of cashiers that are open. Assuming that customers spread evenly over the queues, the objective is such that no queue is longer than 3 say. The objective is therefore such that the queue length should be smaller than 3 times the number of cashiers c. Finally, in many shops, the cashier capacity is made dependent on the queue length; when the queues are long, stockers open extra cashier positions until the first server becomes idle. Thus, as there is spare capacity, the factor 2.5λ can be made smaller, perhaps set to 2λ. However, this factor only changes the load profile, not the method to solve it.

Clearly, it does not require much effort to make the model very complicated, and it is easy to make it complicated to the extent that the model becomes useless. Mind: a good model is simple (but not too simple). So, notwithstanding the weaknesses of the above model, it is pretty powerful already.

Exercises

Exercise 2.10.1. Derive expression 2.46 for the average number of jobs in an $M/M/1$ queue.

Exercise 2.10.2. Derive expression 2.47 for the excess probability $\mathsf{P}\{L \geq n\}$, i.e., the probability that a long queue occurs.

Exercise 2.10.3 (use=false)**.** merge with inventory model of mnm1.tex file, or in the batchqueues.tex file.

Customers of, for example, a fast-food restaurant or a production facility, prefer to be served from stock. For this reason such companies often use a 'produce-up-to' policy: When the on-hand inventory I is equal or lower than some threshold r, the company produces items until the inventory level equals $r+1$ again. The level r is the known as the reorder level.

Suppose that customers arrive as a Poisson process with rate λ and the production times of single items are i.i.d. and exponentially distributed with parameter μ. Assume also that customers that cannot be served from on-hand stock are backlogged, that is, they wait until their item has been produced. What are the average on-hand inventory level, the average number of customer in backlog, and the fraction of customers that are backlogged?

If h is the cost per item per unit time in stock and b the cost per customer per unit time in backlog and $\bar{b}$ the cost per customer backlogged, what are the average costs of using a reorder level r?

What is the optimal reorder level r^*, i.e., the level that achieves minimal average cost?

Hints

Hint 2.10.2: $\mathsf{P}\{L \geq n\} = \sum_{k\geq n} p(k)$.

Hint 2.10.3: Realize that the inventory level $I(t)$ at time t can be modeled as $I(t) = r+1-L(t)$, where L is the number of jobs in an $M/M/1$ queue.

Solutions

Solution 2.10.1:

$$
\begin{aligned}
\mathsf{E}\{L\} &= \sum_{n=0}^{\infty} n p(n) \\
&= \sum_{n=0}^{\infty}\sum_{i=1}^{n} \mathbb{1}_{i\leq n} p(n) && n = \sum_{i=1}^{n} \mathbb{1}_{i\leq n} \\
&= \sum_{n=0}^{\infty}\sum_{i=1}^{\infty} \mathbb{1}_{i\leq n} p(n) && i > n \implies \mathbb{1}_{i\leq n} = 0 \\
&= \sum_{i=1}^{\infty}\sum_{n=0}^{\infty} \mathbb{1}_{i\leq n} p(n) && \text{Fubini} \\
&= \sum_{i=1}^{\infty}\sum_{n=i}^{\infty} p(n) && n < i \implies \mathbb{1}_{i\leq n} = 0 \\
&= \sum_{i=1}^{\infty}\sum_{n=i}^{\infty} (1-\rho)\rho^n && p(n) = (1-\rho)\rho^n \\
&= \sum_{i=1}^{\infty}\sum_{n=0}^{\infty} (1-\rho)\rho^{n+i} && n \to n+i \\
&= \sum_{i=1}^{\infty} (1-\rho)\rho^i \sum_{n=0}^{\infty} \rho^n && \rho^{n+i} = \rho^i\rho^n \\
&= \sum_{i=1}^{\infty} (1-\rho)\rho^i \frac{1}{1-\rho} \\
&= \sum_{i=1}^{\infty} \rho^i \\
&= \sum_{i=0}^{\infty} \rho^{i+1} && i \to i+1 \\
&= \rho\sum_{i=0}^{\infty} \rho^i \\
&= \frac{\rho}{1-\rho}.
\end{aligned}
$$

Note that, since the summands are positive, we can use Fubini's theorm to justify the interchange of the summations.

Solution 2.10.2:

$$
\begin{aligned}
\mathsf{P}\{L \geq n\} &= \sum_{k=n}^{\infty} p(k) = \sum_{k=n}^{\infty} p(0)\rho^k = (1-\rho)\sum_{k=n}^{\infty} \rho^k \\
&= (1-\rho)\rho^n \sum_{k=0}^{\infty} \rho^k = (1-\rho)\rho^n \frac{1}{1-\rho}.
\end{aligned}
$$

Solution 2.10.3:

2.11 M(n)/M(n)/1 Queue

As it turns out, many more single-server queueing situations can be modeled and analyzed by making a judicious choice of $\lambda(n)$ and $\mu(n)$ in (2.41), to wit, $\lambda(n)p(n) = \mu(n+1)p(n+1)$. For these queueing systems we just present the results. In the exercises we ask you to derive the formulas.

For the $M/M/1/K$, i.e., a system that cannot contain more than K jobs, take

$$\lambda(n) = \begin{cases} \lambda, & \text{if } n < K, \\ 0, & \text{if } n \geq K, \end{cases}$$
$$\mu(n) = \mu.$$

Then,

$$p(n) = \frac{\rho^n}{G}, \quad 0 \leq n \leq K, \tag{2.48a}$$
$$G = \frac{1-\rho^{K+1}}{1-\rho}, \tag{2.48b}$$
$$P_{\text{loss}} = \mathsf{P}\{L = K\} = \frac{\rho^K}{G} = \frac{1-\rho}{1-\rho^{K+1}}\rho^K. \tag{2.48c}$$

For the $M/M/c$ queue we can take

$$\lambda(n) = \lambda,$$
$$\mu(n) = \begin{cases} n\mu, & \text{if } n \leq c, \\ c\mu, & \text{if } n \geq c. \end{cases}$$

This model is also known as the *Erlang C*-formula. Define the load as

$$\rho = \frac{\lambda}{c\mu}. \tag{2.49a}$$

Then,

$$p(n) = \frac{1}{G}\frac{(c\rho)^n}{n!}, \quad n = 0, \ldots, c, \tag{2.49b}$$
$$p(n) = \frac{1}{G}\frac{c^c\rho^n}{c!}, \quad n = c, c+1, \ldots \tag{2.49c}$$
$$G = \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{(1-\rho)c!}, \tag{2.49d}$$
$$\mathsf{E}\{L_Q\} = \sum_{n=c}^{\infty}(n-c)p(n) = \frac{(c\rho)^c}{c!G}\frac{\rho}{(1-\rho)^2}, \tag{2.49e}$$
$$\mathsf{E}\{L_S\} = \sum_{n=0}^{c} np(n) + \sum_{n=c+1}^{\infty} cp(n) = \frac{\lambda}{\mu}. \tag{2.49f}$$

From this we can easily get the $M/M/c/c$ queue; here jobs cannot be in queue, only in service, and the system has c servers. This model is also known as the Erlang B-formula and is often used to determine the number of beds at hospitals, where the beds act as servers and the patients as jobs.

From the $M/M/c$ queue (or the $M/M/c/c$ queue) we can also obtain the $M/M/\infty$, i.e., a queueing system with ample servers. By taking the limit $c \to \infty$, note first that in (2.49d),

$$\frac{(c\rho)^c}{(1-\rho)c!} = \frac{(\lambda/\mu)^c}{(1-\rho)c!} \to 0, \quad \text{as } c \to \infty.$$

Hence

$$G = \sum_{n=0}^{c-1} \frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{(1-\rho)c!} \to \sum_{n=0}^{\infty} \frac{(c\rho)^n}{n!} = e^{\lambda/\mu}.$$

Next, for $n \geq c$:

$$p(n) = \frac{1}{G}\frac{c^c\rho^n}{c!} = \frac{\rho^n}{G}\frac{c^c}{c!} \to 0, \quad \text{as } c \to \infty,$$

since, from a standard limit $n^n/n! \to 0$ as $n \to \infty$. Therefore, for $n < c$:

$$p(n) = \frac{1}{G}\frac{(c\rho)^n}{n!} = \frac{1}{G}\frac{(\lambda/\mu)^n}{n!} \to e^{-\lambda/\mu}\frac{(\lambda/\mu)^n}{n!}, \quad \text{as } c \to \infty.$$

We see that the number of busy servers in the $M/M/\infty$ queue is Poisson distributed with parameter λ/μ, and $\mathsf{E}\{L\} = \mathsf{E}\{L_S\} = \lambda/\mu$. Observe that now λ/μ has no longer the interpretation of the fraction of time the server(s) are busy; it is the average number of busy servers.

We mention in passing—but do not prove it—that the same results also hold for the $M/G/\infty$ queue with $\lambda\,\mathsf{E}\{S\}$ rather than λ/μ.

Finally, we consider queues with *balking*, that is, queues in which customers leave when they find the queue too long at the moment they arrive. A simple example model with customer balking is given by x

$$\lambda(n) = \begin{cases} \lambda, & \text{if } n = 0, \\ \lambda/2, & \text{if } n = 1, \\ \lambda/4, & \text{if } n = 2, \\ 0, & \text{if } n > 2, \end{cases}$$

and $\mu(n) = \mu$.

Observe that in the example with balking we made a subtle implicit assumption; in Section 2.12 we elaborate on this assumption. To make the problem clear, note that balking customers *decide at the moment they arrive* to either join or leave; in other words, they decide based on what they 'see upon arrival'. In yet other words, they make decisions based on the state of the system at arrival moments, not on time-averages. However, the notion of $p(n)$ is a long-run *time-average*, and is typically not the same as what customers 'see upon arrival'. As a consequence, the performance measure $\mathsf{P}\{L \leq n\}$ is not necessarily in accordance with the perception of customers. To relate these two 'views', i.e., time-average versus observer-average, we need a new concept, *PASTA*, to be developed in in Section 2.12.

Exercises

Exercise 2.11.1. Consider the $M/M/2/2+1$ queue with arrival rate λ and service rate μ.

1. Derive the level-crossing equations for this queueing system.
2. Derive a simple and closed form expressions for the state probabilities in steady state.

Exercise 2.11.2.

1. Derive Eq. (2.48)
2. Show that as $K \to \infty$, the performance measures of the $M/M/1/K$ converge to those of the $M/M/1$ queue.

Exercise 2.11.3. (Multi-server queue with blocking) Consider the $M/M/c/c+K$ queue, try to derive the steady state probabilities $p(0), p(1), \ldots$. You do not have to compute the normalization constant G.

Exercise 2.11.4. Check that Eq.(2.49) for the $M/M/c$ queue reduces to the $M/M/1$ queue if you take $c = 1$.

Exercise 2.11.5. Derive Eq.(2.49) for the $M/M/c$ queue.

Exercise 2.11.6.

1. Show that the $M/M/c$ queue converges to the $M/M/\infty$ queue as $c \to \infty$.
2. Show that the $M/M\infty$ queue is stable for any finite λ.
3. Why is $\mathsf{E}\{L\} = \rho$ for the $M/M/\infty$ queue?

Exercise 2.11.7. (Hall 5.1) Give two examples of systems that ordinarily have a finite buffer size. Give two examples of systems that have a finite calling population.

Exercise 2.11.8. In what way is a queueing system with balking, at level b say, different from a queueing system with finite calling population of size b?

Exercise 2.11.9. (Systems with finite calling population)

1. Derive the steady state probabilities $p(n)$ for a single-server queue with a finite calling population with N members, i.e., jobs that are in service cannot arrive to the system.
2. Check the answer you obtained for the cases $N = 1$ and $N = 2$. Interpret the results.
3. Derive the steady state probabilities $p(n)$ for a queue with a finite calling population with N members and N servers, i.e., the number of servers in the queueing system is equal the size of the calling population.

Exercise 2.11.10.

What would be the difference between a multi-server queue and a single-server queue with a fast server? We can use the formulas for the $M/M/1$ queue and the $M/M/c$ queue to obtain some basic understanding of the difference. To this end, suppose we have an $M/M/3$ queue, with arrival rate $\lambda = 5$ per day and $\mu = 2$ per server, and we compare it to an $M/M/1$ with the same arrival rate but with a service rate of $\mu = 3 \cdot 2 = 6$.

When is it OK to approximate the $M/M/c$ queue by an $M/M/1$ queue with a fast server?

Hints

Hint 2.11.1: Think about what would be the appropriate model choices for $\lambda(n)$ and $\mu(n)$ and use the level-crossing equations $\lambda(n)p(n) = \mu(n+1)p(n+1)$. For instance, realize that $\lambda(3) = 0$: the system cannot contain more than 3 jobs, hence a state with 4 jobs must be impossible. We can achieve that by setting $\lambda(3) = 0$. For the service rate, how many servers are busy when the system contains 2 or more jobs? What does this say about $\mu(k)$ for $k = 2$ or $k = 3$.

Hint 2.11.2: Use that $\sum_{i=0}^{n} x^i = (1 - x^{n+1})/(1-x)$. BTW, is it necessary for this expression to be true that $|x| < 1$? What should you require for $|x|$ when you want to take the limit $n \to \infty$?

Hint 2.11.3: Use $\lambda(n)p(n) = \mu(n+1)p(n+1)$ and find suitable expressions for $\lambda(n)$ and $\mu(n+1)$.

Hint 2.11.4: Fill in $c = 1$. Realize that this is a check on the formulas.

Hint 2.11.5: Use $\lambda(n)p(n) = \mu(n+1)p(n+1) = \min\{c, n+1\}\mu p(n+1)$.

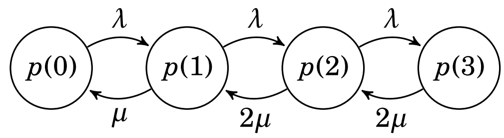
Hint 2.11.6: Use that for any x, $x^n/n! \to 0$ as $n \to \infty$.

Hint 2.11.9: Use $\lambda(n)p(n) = \mu(n+1)p(n+1)$, and realize that for this case $\lambda(n) = (N-n)\lambda$ and $\mu(n) = \mu$.

Hint 2.11.10: Implement the formulas for $\mathsf{E}\{L_Q(M/M/3)\}$ for the $M/M/3$ queue in some computer program (R, excel, python, whatever) and compare this to $\mathsf{E}\{L_Q(M/M/1)\}$ for the fast $M/M/1$ case. Make plots of $\mathsf{E}\{L_Q(M/M/3)\}$ and $\mathsf{E}\{L(M/M/3)\}$ as functions of ρ.

Solutions

Solution 2.11.1: 1. Use this figure. Make sure you understand why $\mu(2) = 2\mu$ and so on.



From this figure it follows right away that:

$$
\begin{aligned}
\lambda p(0) &= \mu p(1) \\
\lambda p(1) &= 2\mu p(2) \\
\lambda p(2) &= 2\mu p(3)
\end{aligned}
$$

2. From the above, with $\rho = \lambda/\mu$:

$$
\begin{aligned}
p(1) &= \rho p(0), \\
p(2) &= (\rho/2)p(1) = (\rho^2/2)p(0), \\
p(3) &= (\rho/2)p(2) = (\rho^3/4)p(0).
\end{aligned}
$$

Now we normalize to find $p(0)$. Thus, we want that:

$$
1 = p(0) + p(1) + p(2) + p(3) = p(0)\left(1 + \rho + \frac{\rho^2}{2} + \frac{\rho^3}{4}\right),
$$

hence,

$$
p(0) = (1 + \rho + \rho^2/2 + \rho^3/4)^{-1}.
$$

Solution 2.11.2: Note that

$$
1 = \sum_{i=0}^{K} p(i) = p(0) \sum_{i=0}^{K} \rho^i = p(0) \frac{1-\rho^{K+1}}{1-\rho}.
$$

Thus, for the normalization constant G we take

$$
G = \frac{1}{p(0)} = \frac{1-\rho^{K+1}}{1-\rho},
$$

and the result follows.

To take the limit $K \to \infty$—mind, not the limit $n \to \infty$—, write

$$G = \frac{1-\rho^{K+1}}{1-\rho} = \frac{1}{1-\rho} - \frac{\rho^{K+1}}{1-\rho}.$$

Since $\rho^{K+1} \to 0$ as $K \to \infty$ (recall, $\rho < 1$), we get

$$G \to \frac{1}{1-\rho},$$

as $K \to \infty$. Therefore $p(n) = \rho^n/G \to \rho^n(1-\rho)$, and the latter are the steady-state probabilities of the $M/M/1$ queue. Finally, if the steady state probabilities are the same, the performance measures (which are derived from $p(n)$) must be the same.

Solution 2.11.3: $\lambda(n) \equiv \lambda$ for all $n < c+K$. When $n = c+K$, $\lambda(n) = 0$, since then the system is full, and all arriving jobs will be dropped; in other words, there will still be jobs arriving to the system when $L = c+K$, but these jobs will be rejected, hence cannot generate a transition from state $c+K$ to $c+K+1$. When $n < c$, $\mu(n) = n\mu$ since only n servers are active/occupied when the system contains n jobs. When $n \geq c$, $\mu(n) = c\mu$. Thus, using $\rho = \lambda/(c\mu)$, for $n < c$,

$$p(n) = \frac{\lambda}{n\mu} p(n-1) = \frac{(\lambda/\mu)^n}{n!} p(0) = \frac{(c\rho)^n}{n!} p(0).$$

For $c \leq n \leq c+K$ and using the above to get $p(c-1)$:

$$\begin{aligned} p(n) &= \frac{\lambda}{c\mu} p(n-1) = \rho p(n-1) = \rho^2 p(n-2) = \dots \\ &= \rho^{n-c+1} p(c-1) = \rho^{n-c+1} \frac{(c\rho)^{c-1}}{(c-1)!} p(0) \\ &= \rho^n \frac{(c)^{c-1}}{(c-1)!} p(0) = \rho^n \frac{(c)^{c-1} c}{(c-1)! c} p(0) = \frac{c^c \rho^n}{c!} p(0). \end{aligned}$$

The normalization is trivial, numerically at least.

Solution 2.11.4: Take $c = 1$

$$p(0) = \frac{1}{G} \frac{(c\rho)^0}{0!} = \frac{1}{G}, \quad n = 0, \dots, 1-1 \tag{2.50}$$

$$p(n) = \frac{1}{G} \frac{c^c \rho^n}{c!} = \frac{1}{G} \frac{1^1 \rho^n}{1!} = \frac{\rho^n}{G}, \quad n = 1, 1+1, \dots \tag{2.51}$$

$$G = \sum_{n=0}^{c-1} \frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{(1-\rho)c!} = \sum_{n=0}^{0} \frac{\rho^0}{0!} + \frac{\rho}{(1-\rho)} = 1 + \frac{\rho}{1-\rho} = \frac{1}{1-\rho} \tag{2.52}$$

$$\mathsf{E}\{L_Q\} = \frac{(c\rho)^c}{c!G} \frac{\rho}{(1-\rho)^2} == \frac{\rho}{1/(1-\rho)} \frac{\rho}{(1-\rho)^2} = \frac{\rho^2}{1-\rho} \tag{2.53}$$

$$\mathsf{E}\{L_S\} = \sum_{n=0}^{c} n p(n) + \sum_{n=c+1}^{\infty} c p(n) = p(1) + 1 \sum_{n=2}^{\infty} p(n) = 1 - p(0) = \rho. \tag{2.54}$$

Everything is in accordance to the formulas we derived earlier for the $M/M/1$ queue.

Solution 2.11.5: First we use the hint to establish a generic relation for $p(n+1)$:

$$\begin{aligned}
p(n+1) &= \frac{\lambda}{\mu(n+1)}p(n) = \frac{\lambda}{\min\{c,n+1\}\mu}p(n) \\
&= \frac{1}{\min\{c,n+1\}}\frac{\lambda}{\mu}p(n) = \frac{1}{\min\{c,n+1\}}(c\rho)p(n) \\
&= \frac{1}{\min\{c,n+1\}\min\{c,n\}}(c\rho)^2 p(n-1) = \frac{1}{\Pi_{k=1}^{n+1}\min\{c,k\}}(c\rho)^{n+1}p(0).
\end{aligned}$$

Thus, if $n < c$:

$$p(n) = \frac{1}{\Pi_{k=1}^{n}\min\{c,k\}}(c\rho)^n p(0) = \frac{(c\rho)^n}{n!}p(0),$$

since $\min\{c,k\} = k$ when $k < c$. If $n \geq c$:

$$\begin{aligned}
p(n) &= \frac{1}{\Pi_{k=1}^{n}\min\{c,k\}}(c\rho)^n p(0) \\
&= \frac{1}{\Pi_{k=1}^{c-1}k\cdot\Pi_{k=c}^{n}c}(c\rho)^n p(0) \\
&= \frac{1}{(c-1)!c^{n-c+1}}c^n\rho^n p(0) \\
&= \frac{1}{c!c^{n-c}}c^n\rho^n p(0) = \\
&= \frac{c^c}{c!}\rho^n p(0).
\end{aligned}$$

For the normalization constant G set $p(0) = 1$ for the time being.

$$\begin{aligned}
G &= \sum_{n=0}^{\infty} p(n) = \sum_{n=0}^{c-1} p(n) + \sum_{n=c}^{\infty} p(n) \\
&= \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \sum_{n=c}^{\infty}\frac{c^c}{c!}\rho^n \\
&= \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \sum_{n=c}^{\infty}\frac{(c\rho)^c}{c!}\rho^{n-c} \\
&= \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!}\sum_{n=0}^{\infty}\rho^n \\
&= \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!(1-\rho)}.
\end{aligned}$$

Next,

$$\begin{aligned}
\mathsf{E}\{L_Q\} &= \sum_{n=c}^{\infty}(n-c)p(n) \\
&= \sum_{n=c}^{\infty}(n-c)\frac{c^c}{c!}\rho^n p(0) \\
&= \frac{c^c\rho^c}{Gc!}\sum_{n=c}^{\infty}(n-c)\rho^{n-c} \\
&= \frac{c^c\rho^c}{Gc!}\sum_{n=0}^{\infty}n\rho^n = \frac{c^c\rho^c}{Gc!}\frac{\rho}{(1-\rho)^2},
\end{aligned}$$

where, with our common trick (if we don't want to use generating functions),

$$\begin{aligned}\sum_{n=0}^{\infty} n\rho^n &= \sum_{n=0}^{\infty}\sum_{i=1}^{\infty} \mathbb{1}_{i\leq n}\rho^n = \sum_{i=1}^{\infty}\sum_{n=0}^{\infty} \mathbb{1}_{i\leq n}\rho^n \\ &= \sum_{i=1}^{\infty}\sum_{n=i}^{\infty}\rho^n = \sum_{i=1}^{\infty}\rho^i\sum_{n=0}^{\infty}\rho^n \\ &= \frac{1}{1-\rho}\sum_{i=1}^{\infty}\rho^i = \frac{\rho}{1-\rho}\sum_{i=0}^{\infty}\rho^i = \frac{\rho}{(1-\rho)^2}.\end{aligned}$$

Observe again that using indicators and Fubini's theorem (interchanging summations/integrals) makes the above computation painless. Realize, by the way, that

$$\sum_{n=0}^{\infty} np(n) = \sum_{n=1}^{\infty} np(n).$$

We next show that the expected number of jobs in service is given by

$$\mathsf{E}\{L_S\} = \sum_{n=0}^{c} np(n) + \sum_{n=c+1}^{\infty} cp(n).$$

This expression is not the easiest to start with. With a slight modification the entire derivation becomes easier. I also premultiply by the normalization constant G to get rid of it on the right hand side.

$$\begin{aligned}G\,\mathsf{E}\{L_S\} &= G\sum_{n=0}^{c} np(n) + \sum_{n=c+1}^{\infty} cp(n) \\ &= \sum_{n=1}^{c} n\frac{(c\rho)^n}{n!} + \sum_{n=c+1}^{\infty} c\frac{c^c\rho^n}{c!} = \sum_{n=1}^{c}\frac{(c\rho)^n}{(n-1)!} + \frac{c^{c+1}}{c!}\sum_{n=c+1}^{\infty}\rho^n \\ &= \sum_{n=0}^{c-1}\frac{(c\rho)^{n+1}}{n!} + \frac{(c\rho)^{c+1}}{c!}\sum_{n=0}^{\infty}\rho^n = c\rho\left(\sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!(1-\rho)}\right).\end{aligned}$$

Observe that the right hand side is precisely equal to ρcG, and hence,

$$\mathsf{E}\{L_S\} = c\rho = \frac{\lambda}{\mu}.$$

Solution 2.11.6: 1. The second term in (2.49d) is $(c\rho)^c/c! = (\lambda/\mu)^c/c!$. It is well known that $x^c/c! \to 0$ as $c \to \infty$.

2. No matter how many jobs are in service, there is always another free server available when a new job arrives. Thus, jobs never have to wait in queue, and only spend time in service. Since $\mathsf{E}\{S\} < \infty$ by assumption, jobs spend a finite time (with probability one) at a server.

3. Write $\rho = \lambda/\mu$. Then, from the formulas for the $M/M/\infty$ queue, it follows that $p(n) = e^{-\rho}\rho^n/n!$. Interestingly, we see that this is equal to $\mathsf{P}\{N = n\}$ where N is a Poisson r.v. with parameter ρ. Thus, the number in the system L is Poisson distributed with parameter ρ, thus $\mathsf{E}\{L\} = \rho$.

   Another way to see that $\mathsf{E}\{L\} = \rho$ is by noting that in the $M/M/\infty$ queue jobs do not interact with each other in the queue. When they arrive, there is always a free server available. Since work arrives at rate ρ, and all jobs are in service simultaneously, the average number of busy servers must also be ρ.

Solution 2.11.7: Finite buffer size. Formally the number of customers that fit into a shop is necessarily finite. This answer, however, is not intended. Typically, the number of customers in a restaurant is limited. Example 2: Sometimes call centers reject callers when the system is too busy.

A finite calling population occurs for instance at a factory with a number of machines. When a machine breaks down, it becomes a (repair) job at the repair department. Thus, a break down forms an arrival at the repair shop. The mechanics at the repair department form a set of parallel servers. Typically, the number of machines quite small, 10 or so, and when a machine is 'down', i.e., broken, it cannot break again. Hence, when 2, say, machines are in repair, the number of 'customers' that can arrive to the queueing system is only 8.

Solution 2.11.8: In a queueing system with balking, customers may decide to balk at a level b. Thus, whether only b customers are admitted to the system (i.e., blocked), or balk at level b, the effect is the same: the number of people in the system remains at or below b. However, a fraction of customer may already balk at lower levels, like in the example above, so that the arrival stream is 'thinned' due to balking customers. In that respect, a queueing system with balking behaves differently.

Solution 2.11.9: 1) Take $\lambda(n) = (N-n)\lambda$ and $\mu(n) = \mu$, and solve Eq. (2.41) and (2.43). Thus:

$$
\begin{aligned}
p(n+1) &= \frac{(N-n)\lambda}{\mu} p(n) = \rho(N-n)p(n) \\
&= \rho^2(N-n)(N-(n-1))p(n-1) \\
&= \rho^3(N-n)(N-(n-1))(N-(n-2))p(n-2) \\
&= \rho^{n+1}(N-n)(N-(n-1))\cdots(N-(0))p(0) \\
&= \rho^{n+1}\frac{N!}{(N-(n+1))!}p(0).
\end{aligned}
$$

Next, we need to normalize this. Observe that $p(N+1) = P(N+2) = \ldots = 0$ since there are just N customers, so that the system can never contain more than N customers. Thus, we want $p(0)$ to be such that

$$
1 = \sum_{n=0}^{N} p(n) = p(0) \sum_{n=0}^{N} \rho^n \frac{N!}{(N-n)!}
$$

We see from this that $p(0)$ times some constant must be 1. Hence, dividing by this constant, we get

$$
p(0) = \left(\sum_{n=0}^{N} \rho^n \frac{N!}{(N-n)!} \right)^{-1}.
$$

I asked WolframAlpha to simplify this, but the answer I got was not particularly revealing.

2)

3) Take $\lambda(n) = (N-n)\lambda$ and $\mu(n) = n\mu$. Then

$$
\begin{aligned}
p(n+1) &= \frac{\lambda(n)}{\mu(n+1)} p(n) = \frac{(N-n)\lambda}{(n+1)\mu} p(n) = \frac{(N-n)(N-(n-1))}{(n+1)n} \frac{\lambda^2}{\mu^2} p(n-1) \\
&= \frac{N!}{(N-(n+1))!} \frac{1}{(n+1)!} \rho^{n+1} p(0) = \binom{N}{n+1} \rho^{n+1} p(0).
\end{aligned}
$$

Hence, after normalization, i.e., requiring that $p(0)$ is such that $\sum_{n=0}^N p(n) = 1$, so that $p(0) = \left(\sum_{k=0}^N \rho^k \binom{N}{k}\right)^{-1}$, the final result becomes

$$p(n) = \frac{\rho^n \binom{N}{n}}{\sum_{k=0}^N \rho^k \binom{N}{k}}.$$

Solution 2.11.10: I am going to implement the formulas of Eq. (2.49) in python. First the results for the $M/M/3$ queue.

```
from math import exp, factorial

labda = 5
mu = 2
c = 3

rho = labda / mu / c
rho

0.8333333333333334

G = sum((c * rho)**n / factorial(n) for n in range(c))
G += (c * rho)**c / ((1 - rho) * factorial(c))
G

22.250000000000004

ELQ = (c * rho)**c / (factorial(c) * G) * rho / (1 - rho)**2
ELQ

3.511235955056181

ELS = rho * c
ELS

2.5

EL = ELQ + ELS
EL

6.011235955056181
```

Now for the $M/M/1$ queue:

```
labda = 5
c = 3
mu = 2*c

rho = labda / mu
rho

0.8333333333333334

ELQ = rho**2/(1-rho)
ELQ

4.166666666666668
```

```
ELS = rho
ELS
```

```
0.8333333333333334
```

```
EL = ELS + ELQ
EL
```

```
5.000000000000001
```

```
rho/(1-rho) # this must also be EL, just a check
```

```
5.000000000000002
```

Note the last check. As a rule, you should always compare your results with known results. BTW, that is one of the reasons I prefer to code the formulas instead of using a calculator. Testing with code is relatively easy, whereas with a calculator it is impossible (You simply can't check what you typed at the calculator.)

So, returning to the results, as expected, the number of jobs in queue is smaller for the $M/M/3$ queue, but the number in service is higher.

To put things in a larger perspective, see Figure 2.18 where we plot the ratio of the queue lengths and the system length as functions of ρ. We see, in case of high load, that $\mathsf{E}\{L_Q\}$ and $\mathsf{E}\{L\}$ are nearly the same for both systems. This is as expected: when the load is high, most jobs should be in the queue. Therefore, $\mathsf{E}\{L_Q\}/\mathsf{E}\{L\} \to 1$ as $\rho \to 1$. When ρ is small, the difference is quite large. This is also reasonable, because the service time in the fast $M/M/1$ is 3 times as small as the service time in the $M/M/3$ queue. Hence, as ρ is small, the time in the system is dominated by service time, as there is hardly any queueing time, if at all. Thus, there must be more jobs in the system on average in the $M/M/3$ queue than in the fast $M/M/1$ queue.

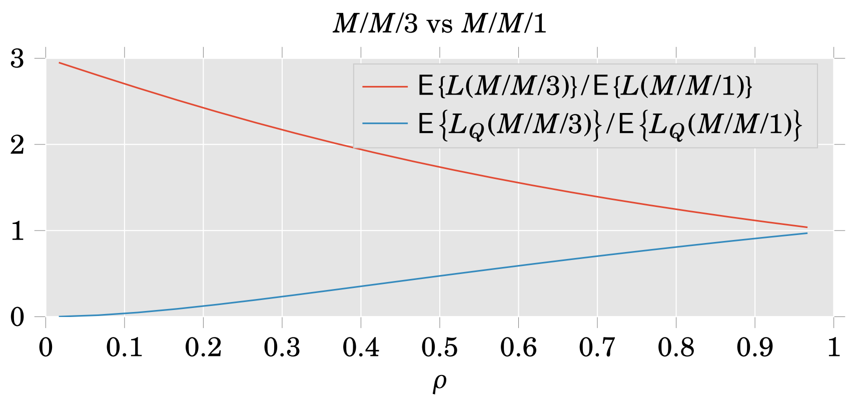


Figure 2.18: multi server versus single fast server

The code can be found on `github` in the `progs` directory.

2.12 Poisson Arrivals See Time Averages

Suppose the following limit exists:

$$\pi(n) = \lim_{m\to\infty} \frac{1}{m} \sum_{k=1}^{m} \mathbb{1}_{L(A_k-)=n},$$

where $\pi(n)$ is the long-run fraction of jobs that observe n customers in the system at the moment of arrival. It is natural to ask whether $\pi(n)$ and $p(n)$ are related, that is, whether what customers see upon arrival is related to the time-average behavior of the system. We will derive a condition in this section under which $\pi(n) = p(n)$.

We can make some progress with this question by rewriting $\pi(n)$ in the following way. Since $A(t) \to \infty$ as $t \to \infty$, it is reasonable that[7]

$$\begin{aligned}\pi(n) &= \lim_{t\to\infty} \frac{1}{A(t)} \sum_{k=1}^{A(t)} \mathbb{1}_{L(A_k-)=n} \\ &= \lim_{t\to\infty} \frac{1}{A(t)} \sum_{k=1}^{\infty} \mathbb{1}_{A_k \le t, L(A_k-)=n} \\ &= \lim_{t\to\infty} \frac{A(n,t)}{A(t)},\end{aligned} \tag{2.55}$$

where we use (2.37a) in the last row. But,

$$\frac{A(n,t)}{t} = \frac{A(n,t)}{A(t)} \frac{A(t)}{t} = \frac{A(t)}{t} \frac{A(n,t)}{A(t)} \to \lambda\pi(n), \quad \text{as } t \to \infty, \tag{2.56}$$

where we use Eq. (2.25) (i.e., $A(t)/t \to \lambda$). that Next, by Eq. (2.39),

$$\frac{A(n,t)}{t} = \frac{A(n,t)}{Y(n,t)} \frac{Y(n,t)}{t} \to \lambda(n)p(n), \quad \text{as } t \to \infty.$$

Thus

$$\lambda\pi(n) = \lambda(n)p(n), \tag{2.57}$$

from which follows our final result:

$$\lambda(n) = \lambda \iff \pi(n) = p(n),$$

So, why is this useful? Well, in words, it means that if the arrival rate does not depend on the state of the system, i.e., $\lambda = \lambda(n)$, the sample average $\pi(n)$ is equal to the time-average $p(n)$, i.e., $\pi(n) = p(n)$. But, when $\pi(n) = p(n)$, the customer perception at arrival moments, i.e., $\pi(n)$, is the same as the server perception, i.e., $p(n)$.

Of course, this property is not satisfied for any general queueing system, see the exercise below in which $X_k = 1$ hour for all k and $S_k = 59$ minutes. However, it is true when the arrival process is Poisson. This fact is typically called *PASTA*: Poisson Arrivals See Time Averages.

Thus, when customers arrive in accordance to a Poisson process (so that the inter-arrival times are exponentially distributed), it must be that $\pi(n) = p(n)$, hence, for the $M/M/1$ queue, we see that

$$\pi(n) = p(n) = (1-\rho)\rho^n.$$

With the above reasoning, we can also establish a relation between $\pi(n)$ and the statistics of the system as obtained by the departures, i.e., $\delta(n)$, to be defined presently. For this we turn again to Eq. (2.38), i.e., $|A(n,t) - D(n,t)| \le 1$. To obtain Eq. (2.40) we divided both sides of this equation by the time the system spends in a certain state. We can also use another form:

$$\frac{A(t)}{t}\frac{A(n,t)}{A(t)} = \frac{A(n,t)}{t} \approx \frac{D(n,t)}{t} = \frac{D(t)}{t}\frac{D(n,t)}{D(t)}.$$

[7] See below for the proof.

Taking limits at the left and right, we see again that the left hand becomes $\lambda\pi(n)$. For the right hand side, we use Eq. (2.27) and define, analogous to (2.55),

$$\delta(n) = \lim_{t\to\infty} \frac{D(n,t)}{D(t)}. \tag{2.58}$$

Thus, $\delta(n)$ is the long-run fraction of jobs that leave n jobs *behind*. Clearly, then, if the limits exist, the right hand side tends to $\gamma\delta(n)$ as $t \to \infty$. Hence, for (queueing) systems in which customers arrive and leave as single units, we have

$$\lambda\pi(n) = \gamma\delta(n). \tag{2.59}$$

Moreover, if the system is rate-stable, i.e., the output rate γ is equal to the input rate λ, we obtain

$$\lambda = \gamma \iff \pi(n) = \delta(n). \tag{2.60}$$

This means that the system as seen by arrivals, i.e., $\pi(n)$, is the same as what jobs leave behind, i.e., $\delta(n)$.

There is a subtle problem in (2.55) and the derivation of (2.56): $\pi(n)$ is defined as a limit over arrival epochs while in $A(n,t)/t$ we take the limit over time. Now the observant reader might ask why these limits should relate at all. The resolution lies in the *renewal reward theorem*. As this theorem is intuitive and very useful in its own right we state this theorem first, and then apply it to show (2.56). In Figure 2.19 we provide graphical motivation why this theorem is true; for the (simple) proof we refer to El-Taha and Stidham Jr. [1998].

Theorem 2.12.1 (Renewal Reward Theorem, $Y = \lambda X$)**.** Suppose the counting process $\{N(t), t \geq 0\}$ is such that $N(t)/t \to \lambda$ as $t \to \infty$, where $0 < \lambda < \infty$. Let $\{Y(t), t \geq 0\}$ be a non-decreasing right-continuous (deterministic) process. Define $X_k = Y(T_k) - Y(T_{k-1})$, $k \geq 1$, where T_k are the epochs at which N increases, i.e., $N(t) = \{k : T_k \leq t\}$. Then $Y(t)/t$ has a limit iff $n^{-1}\sum_k^n X_k$ has a limit in which case $Y = \lambda X$, i.e.,

$$\lim_t \frac{Y(t)}{t} = Y \iff \lim_n \frac{1}{n}\sum_k^n X_k = X \implies Y = \lambda X.$$

Now, to show that (2.55) is valid, define

$$\begin{aligned} Y(t) &:= A(n,t) = \sum_{k=1}^{A(t)} \mathbb{1}_{L(A_k-)=n} \\ X_k &:= Y(A_k) - Y(A_{k-1}) = A(n,A_k) - A(n,A_{k-1}) = \mathbb{1}_{L(A_k-)=n} \end{aligned}$$

Then,

$$X = \lim_{m\to\infty} \frac{1}{m}\sum_{k=1}^m X_k = \lim_{m\to\infty} \frac{1}{m}\sum_{k=1}^m \mathbb{1}_{L(A_k-)=n} = \pi(n).$$

Since $Y = \lim_t Y(t)/t = \lim_t A(n,t)/t$ it follows from the renewal reward theorem that

$$Y = \lambda X \implies \lim_t \frac{A(n,t)}{t} = \lambda X = \lambda\pi(n).$$

Thus, Eq. (2.56) follows from the renewal reward theorem.

In many (stochastic) systems the relation $Y = \lambda X$ is useful. As a further example, suppose customers pay on average an amount X and customers arrive at rate λ, then the system earns money at rate $\lambda X = Y$.

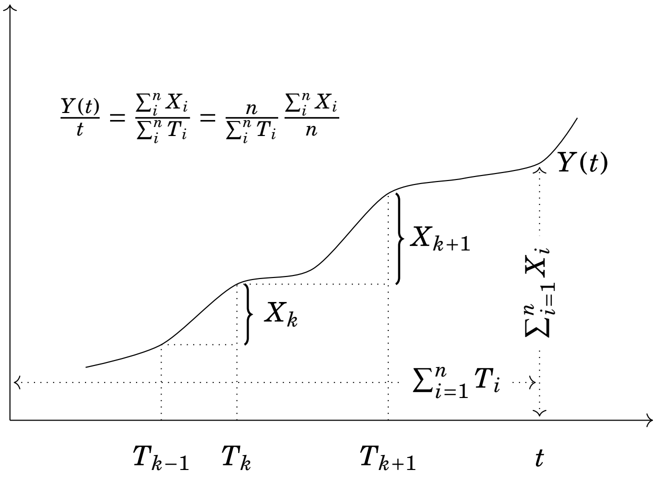


Figure 2.19: A graphical 'proof' of $Y = \lambda X$. Here $Y(t)/t \to Y$, $n/\sum_i^n T_i \to \lambda$ and $n^{-1}\sum_i^n X_i \to X$. (Observe that in the figure X_k does not represent an interarrival time; instead it corresponds to the increment of the (graph of) Y between two consecutive epochs T_{k-1} and T_k at which Y is observed.)

Exercise 2.12.1. Check that the conditions of the renewal reward theorem are satisfied in the above proof of (2.56). The counting process here is $\{A(t)\}$ and the epochs at which $A(t)$ increases are $\{A_k\}$. By assumption, $A_k \to \infty$, hence $A(t) \to \infty$ as $t \to \infty$. Moreover, by assumption $A(t)/t \to \lambda$. Also $A(n,t)$ is evidently non-decreasing and $A(n,t) \to \infty$ as $t \to \infty$.

Exercise 2.12.2. To see that $p(n)$ and $\pi(n)$ can be very different, consider a queueing system in which the inter-arrival times $\{X_i\}$ are all identical to 1 hour, and the service times are all identical to $S_i = 59$ minutes.

1. What are ρ, $p(0)$, $p(n)$ for $n \geq 2$. What is $\pi(n)$? What is the time average queue length $\mathsf{E}\{L\}$? What is the queue length as observed by customers?

2. Check Eq. (2.57) for this case.

3. Consider another (theoretical) queueing system in which each customer requires precisely 1 minute. At the start of each hour, 59 customers arrive. What is $p(n)$, what is $\pi(n)$? What is the time-average system length $\mathsf{E}\{L\}$?

Check the definitions of $Y(0,t)$, $Y(1,t)$, $A(0,t)$ and so on.

1. $A(t) = \lfloor t \rfloor$ provided the unit of t is in hours. $A(0,t) = A(t)$ and $A(n,t) = 0$ for $n > 0$. Thus,

$$
\begin{aligned}
A(0,t) &\approx t \\
A(1,t) &= 0 \\
Y(0,t) &= \frac{1}{60}\lfloor t \rfloor + (t - \lfloor t \rfloor)1\{t - \lfloor t \rfloor \in [59/60, 1)\} \approx \frac{1}{60}t, \\
Y(1,t) &= \frac{59}{60}\lfloor t \rfloor + (t - \lfloor t \rfloor)1\{t - \lfloor t \rfloor \in [0, 59/60)\} \approx \frac{59}{60}t. \\
\lambda(0) &= \lim_t \frac{A(0,t)}{Y(0,t)} = \lim_t \frac{t}{t/60} = 60, \\
\lambda(1) &= \lim_t \frac{A(1,t)}{Y(1,t)} = \lim_t \frac{0}{59/60t} = 0 \\
\lambda(n) &= 0, \quad n \geq 1. \\
\lambda &= \lim_t \frac{A(t)}{t} = 1 \\
p(0) &= \frac{1}{60}, \\
p(1) &= \frac{59}{60}, \\
\rho &= \frac{59}{60}, \\
\pi(0) &= 1.
\end{aligned}
$$

There is no queue so $\mathsf{E}\{L_Q\} = 0$, but $\mathsf{E}\{L\} = \rho$. The queue length as observed by customers is equal to 0, because jobs only arrive when the server is free.

2. We have to check that $\lambda(n)p(n) = \lambda\pi(n)$. First take $n = 0$. By the above: $\lambda(0)p(0) = 60 \cdot 1/60 = 1$, $\lambda\pi(0) = 1 \cdot 1 = 1$. For $n = 1$: $\lambda(1)p(1) = 0$. Since $\pi(1) = 0$ the result is again OK for $n = 1$. And so on .

3. Again $\rho = 59/60$.

   For $p(n)$: the fraction of time the system contains $n = 0$ jobs is $p(0) = 1/60 = 1 - \rho$. Also, as the service time of each job is precisely one minute, $p(n) = 1/60$ for $n = 1, \ldots, 59$.

   For $\pi(n)$, the first of the 59 jobs sees a free server, thus $\pi(0) = 1/59$. The second job of the 59 jobs sees one job in system (and this job is in service), hence $\pi(1) = 1/59$. More generally then, $\pi(n) = 1/59$ for $n = 0, \ldots 58$.

   Note in particular $p(59) = 1/60 \neq \pi(59) = 0$.

   Observe now that the average system length is very different from the previous example, even though $\rho = 59/60$ in both cases. Thus, knowledge of the load is not sufficient to make a statement about the average queue length.

   To compute the average queue length, as perceived by the customers, note that the first customer sees no queue, the second sees one customer in front, and so on. Thus, the average queue length as seen by customers is

$$\frac{0 + 1 + 2 + \cdots + 58}{59} = \frac{1}{59}\frac{58 \cdot 59}{2} = 29.$$

   The time average system length is $\mathsf{E}\{L\} = 59/60 + 58/60 + \cdots 1/60 + 0/60 = 59/2$.

These two examples show that the way customers arrive have an enormous impact on the queue length. Both queueing systems have the same utilization, i.e. 59/60, but the average queue lengths are not nearly the same, i.e. 0 versus 29.

Exercise 2.12.3. Is in general $\pi(n) \geq \delta(n)$? Use that $\lambda \geq \gamma$ always, and $\lambda = \gamma$ when the system is rate-stable. We have shown for one-step transition processes that $\lambda\pi(n) = \gamma\delta(n)$. Thus, $\pi(n) = \delta(n)\gamma/\lambda$. Since $\lambda \geq \gamma$, we have that $\pi(n) \leq \delta(n)$.

Exercise 2.12.4. Why is $\mu(n) = \mu$ for the $M/M/1$ queue? Think about the construction of the $M/M/1$ queue as a random walk, see Section 2.6.

The $M/M/1$-queue can be constructed as a reflection of the random walk $Z(t) = Z(0) + N_\lambda(t) - N_\mu(t)$. Clearly, downcrossings can only occur when N_μ fires. The rate at which the transitions of N_μ occur is constant, and, in particular, independent of the history of Z.

More specifically, for the interested, define $\sigma\{X(t) : t \in I\}$ as the σ-algebra generated by the stochastic procsses $\{X(t), t \in I\}$ on the index set I. Then, by construction of $\{N_\lambda(t)\}$ and $\{Z(t)\}$, we have that $\sigma\{Z(s) : s \in [0,t]\}$ and $\sigma\{N_\lambda(u) : u > t\}$ are independent.

Exercise 2.12.5. Show that

$$\lambda\pi(n) = \lambda(n)p(n) = \mu(n+1)p(n+1) = \gamma\delta(n).$$

What is the important condition for this to be true? Check all definitions of $Y(n,t)/t$ and so on. The important condition is that transitions occur as single steps. In other words, the relation is true for processes with *one-step* transitions, i.e, when $|A(n,t) - D(n,t)| \leq 1$. In that case,

$$\begin{aligned}
\frac{A(n,t)}{t} &= \frac{A(n,t)}{A(t)}\frac{A(t)}{t} \to \pi(n)\lambda \\
\frac{A(n,t)}{t} &= \frac{A(n,t)}{Y(n,t)}\frac{Y(n,t)}{t} \to \lambda(n)p(n) \\
\frac{D(n,t)}{t} &= \frac{D(n,t)}{Y(n+1,t)}\frac{Y(n+1,t)}{t} \to \mu(n+1)p(n+1) \\
\frac{D(n,t)}{t} &= \frac{D(n,t)}{D(t)}\frac{D(t)}{t} \to \delta(n)\gamma
\end{aligned}$$

2.13 Little's Law

Theory and Exercises

There is an important relation between the average time $\mathsf{E}\{W\}$ a job spends in the system and the long-run time-average number $\mathsf{E}\{L\}$ of jobs that is contained in the system. This is Little's law:

$$\mathsf{E}\{L\} = \lambda\,\mathsf{E}\{W\}. \tag{2.61}$$

Here we provide a sketch of the proof, c.f., El-Taha and Stidham Jr. [1998] for the details. In the forthcoming sections we will apply Little's law often. Part of the usefulness of Little's law is that it applies to all input-output systems, whether it is a queueing system or an inventory system or some much more general system.

We start with defining a few intuitively useful concepts. Clearly, from (2.34),

$$\mathscr{L}(t) = \frac{1}{t}\int_0^t L(s)\,\mathrm{d}s = \frac{1}{t}\int_0^t (A(s) - D(s))\,\mathrm{d}s$$

is the time-average of the number of jobs in the system during $[0,t]$. Observe once again from the second equation that $\int_0^t L(s)\,\mathrm{d}s$ is the area enclosed between the graphs of $A(s)$ and $D(s)$.

The waiting time of the kth job is the time between the moment the job arrives and departs, that is

$$W_k = \int_0^\infty \mathbb{1}_{A_k \leq s < D_k}\,\mathrm{d}s.$$

We can actually relate W_k to $\mathscr{L}(t)$, see Figure 2.7. Consider a departure time T at which the system is empty. Observe that $A(T) = D(T)$, as at time T all jobs that arrived up to T also have left. Thus, for all jobs $k \leq A(T)$ we have that $D_k \leq A(T)$, so that we can replace the integration bounds in the above expression for W_k by

$$W_k = \int_0^T \mathbb{1}_{A_k \leq s < D_k}\,\mathrm{d}s.$$

Moreover, if $s \leq T$,

$$L(s) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k \leq s < D_k} = \sum_{k=1}^{A(T)} \mathbb{1}_{A_k \leq s < D_k}.$$

Therefore,

$$\begin{aligned}
\int_0^T L(s)\,\mathrm{d}s &= \int_0^T \sum_{k=1}^{A(T)} 1\{A_k \leq s < D_k\}\,\mathrm{d}s \\
&= \sum_{k=1}^{A(T)} \int_0^T 1\{A_k \leq s < D_k\}\,\mathrm{d}s = \sum_{k=1}^{A(T)} W_k.
\end{aligned}$$

Thus, in words, the area between the graphs of $A(s)$ and $D(s)$ must be equal to the total waiting time spent by all jobs in the system until T. From this,

$$\mathscr{L}(T) = \frac{1}{T}\int_0^T L(s)\,\mathrm{d}s = \frac{A(T)}{T}\frac{1}{A(T)}\sum_{k=1}^{A(T)} W_k.$$

Finally, assuming there are an infinite number of times $0 \leq T_i < T_{i+1} < \cdots$, $T_i \to \infty$, at which $A(T_i) = D(T_i)$ and the following limits exist

$$\mathscr{L}(T_i) \to \mathsf{E}\{L\}, \qquad \frac{A(T_i)}{T_i} \to \lambda, \qquad \frac{1}{A(T_i)}\sum_{k=1}^{A(T_i)} W_k \to \mathsf{E}\{W\},$$

we obtain *Little's law*

$$\mathsf{E}\{L\} = \lambda\,\mathsf{E}\{W\}.$$

As a useful first application, consider a server of the $G/G/1$ queue as a system by itself. Assume the system is rate-stable, for otherwise the above limits do not exists. The arrival rate at the server is λ and the time a job remains in at the server is $\mathsf{E}\{S\}$. Thus, the averate number of jobs at the server is $\mathsf{E}\{L_S\} = \lambda\,\mathsf{E}\{S\}$. As $\lambda\,\mathsf{E}\{S\} = \rho$, we get $\mathsf{E}\{L_S\} = \rho$.

Remark 2.13.1. Sometimes (often?) students memorize Little's law in the wrong way. Thus, as an easy check, use the dimensions of the concepts: $\mathsf{E}\{L\}$ is an average *number*, λ is a *rate*, i.e., *numbers per unit time*, and $\mathsf{E}\{W\}$ is waiting *time*. *Checking the dimensions* in the formula prevents painfull mistakes.

Note also that $\mathsf{E}\{L\} \neq L(t)$; to clarify, the expected number of items in the system is not necessarily equal to the number of items in the system at some arbitrary time t. Thus, Little's law need not hold at all moments in time; it is a statement about *averages*.

Summary Perhaps a summary of the most useful results and concepts of the previous sections are in order here:

- Arrival rate, departure rate, rate stability: $\lambda = \gamma$
- PASTA: $\lambda\pi(n) = \lambda(n)p(n)$,
- Recursions $\lambda(n)p(n) = \mu(n+1)p(n+1)$,
- Renewal reward: $Y = \lambda X$
- Little's law: $\mathsf{E}\{L\} = \lambda\,\mathsf{E}\{W\}$.

Memorize these results. We will use them often in the sequel, but in other courses these results will also prove very useful.

Exercise 2.13.1. For a given single server queueing system the average number of customers in the system is $\mathsf{E}\{L\} = 10$, customers arrive at rate $\lambda = 5$ per hour and are served at rate $\mu = 6$ per hour.

1. What is the average time customers spend in the system?
2. Suppose at the moment you join the system, the number of customers in the system is 10. What is your expected time in the system?
3. Why are the answers between these two questions different?

Exercise 2.13.2. Which assumptions did we use to prove Little's law?

Exercise 2.13.3. Provide a graphical interpration of the proof of Little's law.

Hints

Hint 2.13.1: Think about the data that is given in either situation. Check the units when applying Little's law.

Hint 2.13.3: Make a drawing of $A(t)$ and $D(t)$ until time T, i.e., the first time the system is empty.

Solutions

Solution 2.13.1: 1. This was my initial answer (which is wrong): '$\mathsf{E}\{W\} = \lambda \mathsf{E}\{L\} = \lambda 10$'. Interestingly, I typed in Little's law in the wrong way... So, be aware! It's all too easy to make mistakes with Little's law.

This is correct:

$$\mathsf{E}\{W\} = \mathsf{E}\{L\}/\lambda = 10/\lambda = 10/5 = 2$$

and *not* $\mathsf{E}\{W\} = 10/\mu = 10/6 = 5/3$ hour.

2. The time you spend in the system is the expected remaining service time $\mathsf{E}\{S_r\}$, i.e., the time to serve the customer in service at the moment of arrival, plus $9/\mu$, i.e., the time to clear the queue of 9 customers (recall, 1 job is in service.)

3. In the second question, it is *given* that the system length is 10 at the moment of arrival. Now, L as 'seen' by a given customer upon arrival need not be the same as the time-average L.

Solution 2.13.2:
- $A(t)/t \to \lambda$ as $t \to \infty$, i.e., $A(t)/t$ has a limit as t converges to ∞.
- There exists a sequence of points $T_k, k = 0,1,2,\dots$ in time such that the server is idle.
- Either of the limits $\sum_k^n W_k/n = \sum_k^n S_k/n$ or $t^{-1}\int_0^t L(s)\,\mathrm{d}s$ exists, in which case the other exists.

Solution 2.13.3: The area enclosed between the graphs of $A(t)$ and $D(t)$ until T can be 'chopped up' in two ways: in the horizontal direction and the vertical direction. (Please make the drawing as you go along...) A horizontal line between $A(t)$ and $D(t)$ corresponds to the waiting time of a job, while a vertical line corresponds to the number of jobs in the system at time t. Now adding all horizontal lines (by integrating along the y-axis) makes up the total amount of waiting done by all the jobs until time T. On the other hand, adding the vertical lines (by integrating along the x-axis) is equal to the summation of all jobs in the system. Since the area is the same no matter whether you sum it in the horizontal or vertical direction:

$$\sum_{k=1}^{A(T)} W_k = \text{eclosed area} = \int_0^T (A(t) - D(t))\,dt.$$

Dividing both sides by $A(T)$ gives

$$\frac{1}{A(T)} \sum_{k=1}^{A(T)} W_k = \frac{1}{A(T)} \int_0^T (A(t) - D(t))\,dt.$$

Finally, observe that this equality holds between any two times T_i, T_{i+1}, where times $\{T_i\}$ are such that $A(T_i) = D(T_i)$. Then, as $T_i \to \infty$ which we assumed from the on-set, $\frac{1}{A(T_i)} \sum_{k=1}^{A(T_i)} W_k \to \mathsf{E}\{W\}$, and

$$\frac{T_i}{A(T_i)} \frac{1}{T_i} \int_0^{T_i} (A(t) - D(t))\,dt \to \lambda^{-1}\mathsf{E}\{L\}.$$

Hence, Little's law follows.

2.14 Some Useful Identities

Theory and Exercises

With the PASTA property and Little's law we can derive a number of useful and simple results for the $M/G/1$ queue. Recall, to use the PASTA, we need to assume that jobs arrive as a Poisson process.

The fraction of time the server is empty is $1-\rho = p(0)$. By PASTA, $\pi(0) = p(0)$, hence the fraction of customers that enter an empty system is also $1-\rho$.

Suppose that at A_k, i.e., the arrival epoch of the kth job, the server is busy. The remaining service time $S_{r,k}$ as seen by job k is the time between A_k and the departure epoch of the job in service. If the server is free at A_k, we set $S_{r,k} = 0$. Define the average *remaining service time* as seen at arrival epochs by the limit

$$\mathsf{E}\{S_r\} = \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} S_{r,k},$$

provided this limit exists. Note that $\mathsf{E}\{S_r\}$ includes the fraction of jobs that find the server idle. If we need the remaining service time for the jobs that see the server busy, use that a fraction ρ sees the server occupied upon arrival, while a fraction $1-\rho$ sees a free server. Therefore

$$\mathsf{E}\{S_r\} = \rho\,\mathsf{E}\{S_r \,|\, S_r > 0\} + (1-\rho)\cdot \mathsf{E}\{S_r \,|\, S_r = 0\} = \rho\,\mathsf{E}\{S_r \,|\, S_r > 0\}, \tag{2.62}$$

since, evidently, $\mathsf{E}\{S_r \,|\, S_r = 0\} = 0$. Observe that we used the PASTA property here.

Next, consider the waiting time in queue. It is evident that the expected waiting time for an arriving customer is the expected remaining service time plus the expected time in queue. The expected time in queue, in turn, must be equal to the expected number of customers in queue at an arrival epoch times the expected service time per customer, assuming that service times are i.i.d. If the arrival process is Poisson, it follows from PASTA that the average number of jobs in queue perceived by arriving customers is also the *time-average* number of jobs in queue $\mathsf{E}\{L_Q\}$. Thus, the expected time in queue is

$$\mathsf{E}\{W_Q\} = \mathsf{E}\{S_r\} + \mathsf{E}\{L_Q\}\,\mathsf{E}\{S\}. \tag{2.63}$$

Now, from Little's law we know that $\mathsf{E}\{L_Q\} = \lambda\,\mathsf{E}\{W_Q\}$. Using this,

$$\mathsf{E}\{W_Q\} = \mathsf{E}\{S_r\} + \lambda\,\mathsf{E}\{W_Q\}\,\mathsf{E}\{S\} = \mathsf{E}\{S_r\} + \rho\,\mathsf{E}\{W_Q\},$$

since $\rho = \lambda\,\mathsf{E}\{S\}$. But this gives for the $M/G/1$ queue that

$$\mathsf{E}\{W_Q\} = \frac{\mathsf{E}\{S_r\}}{1-\rho} = \frac{\rho}{1-\rho}\,\mathsf{E}\{S_r \,|\, S_r > 0\}, \tag{2.64}$$

where we use (2.62) in the last equation.

The average waiting time $\mathsf{E}\{W\}$ in the entire system, i.e., in queue plus in service, becomes

$$\mathsf{E}\{W\} = \mathsf{E}\{W_Q\} + \mathsf{E}\{S\} = \frac{\mathsf{E}\{S_r\}}{1-\rho} + \mathsf{E}\{S\}.$$

The situation can be significantly simplified for the $M/M/1$ queue as then the service times are also exponential, hence memoryless, implying that $\mathsf{E}\{S_r \,|\, S_r > 0\} = \mathsf{E}\{S\}$. Thus, for the

$M/M/1$ queue, and using (2.62) and the above formula, we get

$$\begin{aligned}\mathsf{E}\{W\} &= \mathsf{E}\{W_Q\} + \mathsf{E}\{S\} = \frac{\mathsf{E}\{S\}_r}{1-\rho} + \mathsf{E}\{S\} \\ &= \frac{\rho\,\mathsf{E}\{S\}}{1-\rho} + \mathsf{E}\{S\} = \frac{\mathsf{E}\{S\}}{1-\rho}.\end{aligned}$$

Another way to derive the above result is to conclude from PASTA that the expected number of jobs in the system at an arrival is $\mathsf{E}\{L\}$. Since all these jobs require an expected service time $\mathsf{E}\{S\}$, including the job in service by the memory-less property, the time in queue is $\mathsf{E}\{L\}\,\mathsf{E}\{S\}$. The time in the system is then the waiting time plus the service time, hence,

$$\mathsf{E}\{W\} = \mathsf{E}\{L\}\,\mathsf{E}\{S\} + \mathsf{E}\{S\} = \lambda\,\mathsf{E}\{W\}\,\mathsf{E}\{S\} + \mathsf{E}\{S\} = \rho\,\mathsf{E}\{W\} + \mathsf{E}\{S\}, \tag{2.65}$$

where we use Little's law $\mathsf{E}\{L\} = \lambda\,\mathsf{E}\{W\}$. Hence,

$$\mathsf{E}\{W\} = \frac{\mathsf{E}\{S\}}{1-\rho}.$$

Also,

$$\mathsf{E}\{W_Q\} = \mathsf{E}\{L\}\,\mathsf{E}\{S\} = \lambda\,\mathsf{E}\{W\}\,\mathsf{E}\{S\} = \rho\,\mathsf{E}\{W\} = \frac{\rho}{1-\rho}\,\mathsf{E}\{S\} = \frac{\rho^2}{1-\rho}\frac{1}{\lambda}, \tag{2.66}$$

which is consistent with our earlier result.

For the average queue length we use Little's law. Then

$$\begin{aligned}\mathsf{E}\{L\} &= \lambda\,\mathsf{E}\{W\} = \frac{\lambda\,\mathsf{E}\{S\}}{1-\rho} = \frac{\rho}{1-\rho}, \\ \mathsf{E}\{L_Q\} &= \lambda\,\mathsf{E}\{W_Q\} = \frac{\rho^2}{1-\rho}.\end{aligned}$$

Finally, the expected number of jobs in service $\mathsf{E}\{L_s\}$, which is equal to the expected number of busy servers, must be

$$\mathsf{E}\{L_s\} = \mathsf{E}\{L\} - \mathsf{E}\{L_Q\} = \frac{\rho}{1-\rho} - \frac{\rho^2}{1-\rho} = \rho,$$

again in accordance with our earlier result.

Exercise 2.14.1. It is an easy mistake to think that $\mathsf{E}\{S_r\} = \mathsf{E}\{S\}$ when service times are exponential. Why is this wrong?

Exercise 2.14.2. Try to derive relation (2.62) with sample path arguments.

Exercise 2.14.3.

1. Why is this true for the $M/M/1$ queue

$$\mathsf{E}\{L_Q\} = \sum_{n=1}^{\infty}(n-1)\pi(n)?$$

2. Derive an expression for $\mathsf{E}\{L_Q\}$ based on this observation.

Exercise 2.14.4. Why is Eq. (2.65) not true in general for the $M/G/1$ queue?

Exercise 2.14.5. 1. Use the PASTA property to see that the expected waiting time in the system must also be equal to

$$\mathsf{E}\{W\} = \sum_{n=0}^{\infty} \mathsf{E}\{W_Q \,|\, N = n\} \pi(n) + \mathsf{E}\{S\},$$

where $\mathsf{E}\{W_Q \,|\, N = n\}$ is the waiting time in queue given that an arrival sees $N = n$ customers upon arrival.

2. Motivate that $\mathsf{E}\{W_Q \,|\, N = 1\} = \mathsf{E}\{S\}$.

3. Combine the above to see that

$$\mathsf{E}\{W\} = \mathsf{E}\{S\} \sum_{n=0}^{\infty} n\pi(n) + \mathsf{E}\{S\} = \mathsf{E}\{S\}\,\mathsf{E}\{L\} + \mathsf{E}\{S\}.$$

Exercise 2.14.6. Compute the variance of the number of jobs in the system L for the $M/G/1$ queue.

Exercise 2.14.7. Use the PASTA property and the ideas of Section 2.9 to derive for the $M/M/1$ queue that

$$(\lambda + \mu)\pi(n) = \lambda\pi(n-1) + \mu\pi(n+1)$$

Exercise 2.14.8. What would you guess for $\mathsf{E}\{S_r \,|\, S_r > 0\}$ for the $M/D/1$ queue?

Exercise 2.14.9. (Hall 5.2) After observing a single server queue for several days, the following steady-state probabilities have been determined: $p(0) = 0.4$, $p(1) = 0.3$, $p(2) = 0.2$, $p(3) = 0.05$ and $p(4) = 0.05$. The arrival rate is 10 customers per hour.

1. Determine $\mathsf{E}\{L\}$ and $\mathsf{E}\{L_Q\}$.
2. Using Little's formula, determine $\mathsf{E}\{W\}$ and $\mathsf{E}\{W_Q\}$.
3. Determine $\mathsf{V}\{L\}$ and $\mathsf{V}\{L_Q\}$.
4. Determine the service time and the utilization.

Exercise 2.14.10. (Hall 5.5) An $M/M/1$ queue has an arrival rate of 100 per hour and a service rate of 140 per hour.

1. What is $p(n)$?
2. What are $\mathsf{E}\{L_Q\}$, $\mathsf{E}\{L\}$

Exercise 2.14.11. (Hall, 5.6) An $M/M/1$ queue has been found to have an average waiting time in queue of 1 minute. The arrival rate is known to be 5 customers per minute.

1. What are the service rate and utilization?
2. Calculate $\mathsf{E}\{L_Q\}$, $\mathsf{E}\{L\}$ and $\mathsf{E}\{W\}$.
3. The queue operator would like to provide chairs for waiting customers. He would like to have a sufficient number so that all customers can sit down at least 90 percent of the time. How many chairs should he provide?

Exercise 2.14.12. (Hall 5.7). A single server queueing system is known to have Poisson arrivals and exponential service times. However, the arrival rate and service time are state dependent. As the queue becomes longer, servers work faster, and the arrival rate declines, yielding the following functions (all in units of number per hour): $\lambda(0) = 5$, $\lambda(1) = 3$, $\lambda(2) = 2$, $\lambda(n) = 0, n \geq 3$, $\mu(0) = 0$, $\mu(1) = 2$, $\mu(2) = 3$, $\mu(n) = 4, n \geq 3$. Calculate the state probabilities, i.e., $p(n)$ for $n = 0, \ldots$.

Exercise 2.14.13. (Hall 5.14) An airline phone reservation line has one server and a buffer for two customers. The arrival rate 6 customers per hour, and a service rate of just 5 customers per hour. Arrivals are Poisson and service times are exponential.

1. Estimate $\mathsf{E}\{L_Q\}$ and the average number of customers served per hour.
2. Estimate $\mathsf{E}\{L_Q\}$ for a buffer of size 5. What is the impact of the increased buffer size on the number of customers served per hour?

Exercise 2.14.14. (Hall 5.3) After observing a queue with two servers for several days, the following steady-state probabilities have been determined: $p(0) = 0.4$, $p(1) = 0.3$, $p(2) = 0.2$, $p(3) = 0.05$ and $p(4) = 0.05$. The arrival rate is 10 customers per hour.

1. Determine $\mathsf{E}\{L\}$ and $\mathsf{E}\{L_Q\}$.
2. Using Little's formula, determine $\mathsf{E}\{W\}$ and $\mathsf{E}\{W_Q\}$.
3. Determine $\mathsf{V}\{L\}$ and $\mathsf{V}\{L_Q\}$.
4. Determine the service time and the utilization.

Exercise 2.14.15. (Hall 5.8) The queueing system at a fast-food stand behaves in a peculiar fashion. When there is no one in the queue, people are reluctant to use the stand, fearing that the food is unsavory. People are also reluctant to use the stand when the queue is long. This yields the following arrival rates (in numbers per hour): $\lambda(0) = 10$, $\lambda(1) = 15$, $\lambda(2) = 15$, $\lambda(3) = 10$, $\lambda(4) = 5$, $\lambda(n) = 0, n \geq 5$. The stand has two servers, each of which can operate at 5 per hour. Service times are exponential, and the arrival process is Poisson.

1. Calculate the steady state probabilities.
2. What is the average arrival rate?
3. Determine $\mathsf{E}\{L\}$, $\mathsf{E}\{L_Q\}$, $\mathsf{E}\{W\}$ and $\mathsf{E}\{W_Q\}$.

Exercise 2.14.16. (Hall 5.10) A repair/maintenance facility would like to determine how many employees should be working in its tool crib. The service time is exponential, with mean 4 minutes, and customers arrive by a Poisson process with rate 28 per hour. The customers are actually maintenance workers at the facility, and are compensated at the same rate as the tool crib employees.

1. What is $\mathsf{E}\{W\}$ for $c = 1, 2, 3$, or 4 servers?
2. How many employees should work in the tool crib?

Exercise 2.14.17. An $M/M/2$ queueing system is very heavily utilized, with an arrival rate of 11 customers per hour, and a service rate of 6 customers per hour per server.

1. Determine $\mathsf{E}\{L_Q\}$ and $\mathsf{E}\{W_Q\}$.
2. Compare your solution to an $M/M/1$ queue with $\rho = 11/12$. Why are your answers similar or different?

Exercise 2.14.18. Generalize the single-server relation $\mathsf{E}\{W_Q\} = \rho\,\mathsf{E}\{S_r \,|\, S_r > 0\} + \mathsf{E}\{L_Q\}\,\mathsf{E}\{S\}$ to the $M/M/c$ multi-server queue.

We will use this result later to derive algorithms to compute the performance measures for the analysis of closed-queueing network.

Hints

Hint 2.14.1: Realize again that $\mathsf{E}\{S_r\}$ includes the jobs that arrive at an empty system.

Hint 2.14.2: This requires some extra definitions, similar to $A(n,t)$ as defined in (2.37a).

Hint 2.14.4: Think about the consequences of memoryless service times.

Hint 2.14.12: Use the level-crossing equations of the $M(n)/M(n)/1$ queue.

Hint 2.14.13: This is a queueing system with loss, in particular the $M/M/1/1+2$ queue.

Hint 2.14.16: Realize that we have to control the number of servers. Hence, we are dealing with a multi-server queue, i.e., the $M/M/c$ queue. Use the formulas of Eq. (2.49).

The remark that maintenance workers are compensated at the same rate as the tool crib workers confused me a bit at first. Some thought revealed that the consequence of this remark is that is it just as expensive to let the tool crib workers wait (to help maintenance workers) as to let the maintenance workers wait for tools. (Recall, in queueing systems always somebody has to wait, either the customer in queue or the server being idle. If it is very expensive to let customers wait, the number of servers must be high, whereas if servers are relatively expensive, customers have to do the waiting.)

Hint 2.14.18: Interpret each component at the right hand side of the equation and generalize it to a multi-server queueing system.

Solutions

Solution 2.14.1: $\mathsf{E}\{S_r \,|\, S_r > 0\} = \mathsf{E}\{S\}$ for the $M/M/1$ queue, and $\mathsf{E}\{S_r\} = \rho\,\mathsf{E}\{S_r \,|\, S_r > 0\}$ for the $M/G/1$ queue, it follows that

$$\mathsf{E}\{S_r\} = \rho\,\mathsf{E}\{S_r \,|\, S_r > 0\} = \rho\,\mathsf{E}\{S\}.$$

Solution 2.14.2: This is, admittedly, not simple, so let us work in stages. Define $A(+,t)$ as the number of arrivals that see a job in service. As the server is only busy at time t when $L(t) > 0$, we define

$$A(+,t) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k \leq t}\, \mathbb{1}_{L(A_k)>0}.$$

Next, define $\tilde{D}_k = \min\{D_i : D_i \geq A_k\}$ as the first departure after the arrival time A_k of job k. We need to consider two cases. If $\tilde{D}_k = D_k$ then, obviously, the first departure after A_k coincides with the departure time of job k. This is only possible if $L(A_k) = 0$. But then, it must be that the remaining service time of the job in service at time A_k is 0, as there is no job in service. Otherwise, if $L(A_k) > 0$ it must be that $\tilde{D}_k < D_k$, and, as a consequence, the remaining service time of the job in service at time A_k is equal to $\tilde{D}_k - A_k$. All in all, the total amount of remaining service times added up for all arrivals up to time t can be written as

$$S_r(t) = \sum_{k=1}^{\infty} (\tilde{D}_k - A_k) \mathbb{1}_{A_k \le t, L(A_k)>0}.$$

The remaining service time averaged over all arrivals up to time t is therefore $S_r(t)/A(t)$. We can now rewrite this to

$$\frac{S_r(t)}{A(t)} = \frac{A(+,t)}{A(t)} \frac{S_r(t)}{A(+,t)},$$

and interpret these fractions. First, consider

$$\frac{A(+,t)}{A(t)} = \frac{\sum_{k=1}^{\infty} \mathbb{1}_{A_k \le t, L(A_k)>0}}{\sum_{k=1}^{\infty} \mathbb{1}_{A_k \le t}}.$$

This is the number of jobs up to time t that see at least one job in the system divided by the total number of arrivals up to time t. Thus, as $t \to \infty$,

$$\frac{A(+,t)}{A(t)} \to \sum_{n=1}^{\infty} \pi(n) = 1 - \pi(0).$$

With PASTA we can conclude that $1 - \pi(0) = 1 - p(0) = \rho$. Second,

$$\frac{S_r(t)}{A(+,t)} = \frac{\sum_{k=1}^{\infty} (\tilde{D}_k - A_k) \mathbb{1}_{A_k \le t, L(A_k)>0}}{\sum_{k=1}^{\infty} \mathbb{1}_{A_k \le t, L(A_k)>0}},$$

is the total amount of remaining service time up to time t divided by the number of arrivals up to time t that see at least one job in the system. Thus, if the limit exists, we can *define*

$$\mathsf{E}\{S_r \,|\, S_r > 0\} = \lim_{t\to\infty} \frac{S_r(t)}{A(+,t)}.$$

Solution 2.14.3: 1. The fraction of time the system contains n jobs is $\pi(n)$ (by PASTA). When the system contains $n > 0$ jobs, the number in queue is one less, i.e., $n - 1$.

2.

$$\begin{aligned}
\mathsf{E}\{L_Q\} &= \sum_{n=1}^{\infty} (n-1)\pi(n) = (1-\rho) \sum_{n=1}^{\infty} (n-1)\rho^n \\
&= \rho(1-\rho) \sum_{n=1}^{\infty} (n-1)\rho^{n-1} = \rho \sum_{n=1}^{\infty} (n-1)\pi(n-1) \\
&= \rho \sum_{n=0}^{\infty} n\pi(n) = \rho \frac{\rho}{1-\rho}.
\end{aligned}$$

Another way to get the same result is by splitting:

$$\begin{aligned}
\mathsf{E}\{L_Q\} &= \sum_{n=1}^{\infty} (n-1)\pi(n) = \sum_{n=1}^{\infty} n\pi(n) - \sum_{n=1}^{\infty} \pi(n) \\
&= \mathsf{E}\{L\} - (1 - \pi(0)) = \mathsf{E}\{L\} - \rho.
\end{aligned}$$

Solution 2.14.4: Because the remaining service time of the job in service, provided there is a job in the system upon arrival, is not exponentially distributed in general. Only for the $M/M/1$ queue the service times are exponentially distributed, hence memoryless. And only when the service time is memoryless, the service time after an interruption is still exponential.

Solution 2.14.5:

1. The time in the system $\mathsf{E}\{W\}$ is the sum of the time in queue plus the service time of the job itself. Conditioning on the number of jobs N in the system at arrival moments, i.e., conditioning on $\pi(n)$, gives the result.

2. $\mathsf{E}\{W_Q \,|\, N = 0\} = 0$, of course. If $N = 1$, there must be a job in service. Because of the memoryless property of the service times, the remaining service time of the customer in service is still exponentially distributed with mean $\mathsf{E}\{S\}$. Thus, $\mathsf{E}\{W_Q \,|\, N = 1\} = \mathsf{E}\{S\}$.

3. The expected service time for a customer still in queue is $\mathsf{E}\{S\}$. Therefore, for each n we have that, when service times are exponential, $\mathsf{E}\{W_Q \,|\, N = n\} = n\,\mathsf{E}\{S\}$. The probability to see n jobs in the system by an arrival is $\pi(n)$ which, by PASTA, is equal to $p(n)$. Since $\mathsf{E}\{L\} = \sum_n np(n)$, the result follows.

Solution 2.14.6: $\mathsf{V}\{L\} = \mathsf{E}\{L^2\} - (\mathsf{E}\{L\})^2$. We already know $\mathsf{E}\{L\}$ so it remains to compute $\mathsf{E}\{L^2\}$. With wolfram alpha we get

$$\begin{aligned}\mathsf{E}\{L^2\} &= \sum_{n=0}^{\infty} n^2\pi(n) \\ &= \rho\frac{1+\rho}{(1-\rho)^2}.\end{aligned}$$

Thus,

$$\mathsf{V}\{L\} = \frac{\rho(1+\rho)}{(1-\rho)^2} - \frac{\rho^2}{(1-\rho)^2} = \frac{\rho}{(1-\rho)^2}.$$

To see how large this variance is, relative to the mean number of jobs in the system, we typically consider the square coefficient of variation (SCV). As $\mathsf{E}\{L\} = \rho/(1-\rho)$,

$$\frac{\mathsf{V}\{L\}}{(\mathsf{E}\{L\})^2} = \frac{1}{\rho}.$$

Thus, the SCV becomes smaller as ρ increases, but does not become lower than 1. So, realizing that the SCV of the exponential distribution is 1, the distribution of the number of jobs in the system has larger relative variability than the exponential distribution.

Lets see whether I can get the result for $\mathsf{E}\{L^2\}$ by myself. One way is to use the standard formula for a geometric series with $\rho < 1$:

$$\frac{1}{1-\rho} = \sum_{n=0}^{\infty} \rho^n.$$

If we differentiate the left and right hand side with respect to ρ and then multiply with ρ we obtain

$$\frac{\rho}{(1-\rho)^2} = \sum_{n=0}^{\infty} n\rho^n.$$

Again, differentiating and multiplying with ρ yields,

$$\begin{aligned}
\rho\frac{(1-\rho)^2+\rho 2(1-\rho)}{(1-\rho)^4} &= \rho\frac{1-2\rho+\rho^2+2\rho-2\rho^2}{(1-\rho)^4} \\
&= \rho\frac{(1-\rho)^2}{(1-\rho)^4} \\
&= \rho\frac{1+\rho}{(1-\rho)^3} \\
&= \sum_{n=0}^{\infty} n^2\rho^n
\end{aligned}$$

and hence

$$(1-\rho)\sum_{n=0}^{\infty} n^2\rho^n = \rho\frac{1+\rho}{(1-\rho)^2}$$

Observe that here we just compute the second moment of a geometric random variable.

Another way to derive the result is by noting that $\sum_{i=1}^n i = n(n+1)/2$ from which we get that $n^2 = -n + 2\sum_{i=1}^n i$. Substituting this relation into $\sum_n n^2\rho^n$ leads to

$$\begin{aligned}
\sum_{n=0}^{\infty} n^2\rho^n &= \sum_{n=0}^{\infty}\left(\sum_{i=1}^{\infty} 2i\,\mathbb{1}_{i\leq n} - n\right)\rho^n = \sum_{n=0}^{\infty}\sum_{i=0}^{\infty} 2i\,\mathbb{1}_{i\leq n}\rho^n - \sum_{n=0}^{\infty} n\rho^n \\
&= \sum_{i=0}^{\infty} 2i\sum_{n=i}^{\infty}\rho^n - \frac{\mathsf{E}\{L\}}{1-\rho} = \sum_{i=0}^{\infty} 2i\rho^i\sum_{n=0}^{\infty}\rho^n - \frac{\mathsf{E}\{L\}}{1-\rho} \\
&= \frac{2}{1-\rho}\sum_{i=0}^{\infty} i\rho^i - \frac{\mathsf{E}\{L\}}{1-\rho} = \frac{2}{(1-\rho)^2}\mathsf{E}\{L\} - \frac{\mathsf{E}\{L\}}{1-\rho} \\
&= \frac{\mathsf{E}\{L\}}{1-\rho}\left(\frac{2}{1-\rho}-1\right) = \frac{\mathsf{E}\{L\}}{1-\rho}\frac{1+\rho}{1-\rho} \\
&= \frac{\rho}{1-\rho}\frac{1+\rho}{(1-\rho)^2}.
\end{aligned}$$

A last method is based on z-transforms:

$$\phi(z) = \mathsf{E}\left\{z^L\right\} = \sum_{n=0}^{\infty} z^n p(n) = (1-\rho)\sum_{n=0}^{\infty}(\rho z)^n = \frac{1-\rho}{1-\rho z}.$$

Then

$$\mathsf{E}\{L\} = \left.\frac{d}{dz}\phi(z)\right|_{z=1} = \frac{\rho}{1-\rho},$$

and

$$\mathsf{E}\{L(L-1)\} = \phi''(z)\big|_{z=1} = \frac{2\rho^2}{(1-\rho)^2}.$$

Thus,

$$\mathsf{E}\{L\}^2 = \mathsf{E}\{L(L-1)\} + \mathsf{E}\{L\}.$$

Hence,

$$\begin{aligned}
\mathsf{E}\{L\}^2 &= \frac{2\rho^2}{(1-\rho)^2} + \frac{\rho}{1-\rho} \\
&= \frac{2\rho^2}{(1-\rho)^2} + \frac{\rho}{(}1-\rho)(1-\rho)2 \\
&= \frac{\rho^2+\rho}{(1-\rho)^2} = \rho\frac{1+\rho}{(1-\rho)^2}.
\end{aligned}$$

A bit of algebra gives the previous results.

Solution 2.14.7: Consider some state n and count all transitions that 'go in and out of' this state. Specifically, $A(n,t)+D(n-1,t)$ counts all transitions out of state n: $A(n,t)$ counts the number of arrivals that see n in the system upon arrival, hence immediately after such arrivals the system contains $n+1$ jobs; likewise, $D(n-1,t)$ counts all jobs that leave $n-1$ jobs behind, hence immediately before such jobs depart the system contains n jobs. In a similar way, $A(n-1,t)+D(n,t)$ counts all transitions into state n (Recall once again, $D(n,t)$ counts the jobs that leave n behind. Hence, when such departures occur, state n is entered). Therefore the difference between the 'in' transitions and the 'out' transitions is at most 1 for all t, so that we can write

$$\begin{aligned}
A(n,t)+D(n-1,t) \approx A(n-1,t)+D(n,t) &\iff \\
\frac{A(n,t)+D(n-1,t)}{t} \approx \frac{A(n-1,t)+D(n,t)}{t} &\iff \\
\frac{A(n,t)}{t}+\frac{D(n-1,t)}{t} \approx \frac{A(n-1,t)}{t}+\frac{D(n,t)}{t}&.
\end{aligned}$$

Using the ideas of Section 2.9 this becomes for $t\to\infty$,

$$(\lambda(n)+\mu(n))p(n) = \lambda(n-1)p(n-1)+\mu(n+1)p(n+1).$$

Since we are concerned here with the $M/M/1$ queue we have that $\lambda(n)=\lambda$ and $\mu(n)=\mu$, and using PASTA we have that $p(n)=\pi(n)$. We are done.

Solution 2.14.8: Since the service times are deterministic (and constant), I would guess that on average half of the service time remains at the moment a job arrives. If the service time is D always, then $\mathsf{E}\{S_r \,|\, S_r>0\} = \lambda D/2$.

Solution 2.14.9:

1. When a problem is mainly of a computational type, I coded the solutions and show you all the numerical answers. Thus, please read the code too since this shows the precise steps by which I obtain the answer. (The code is typically nearly identical to the mathematical formulas; you should not have any difficulty understanding the code.) I also show many intermediate numerical results so that you can check each step in your computations. (In the computations below I typically use the simplest, but often not the most efficient, code.)

```
P = [0.4, 0.3, 0.2, 0.05, 0.05]
L = sum(n*P[n] for n in range(len(P)))
L
```

```
1.05
```

There can only be a queue when a job is in service. Since there is $m=1$ server, we subtract m from the amount of jobs in the system. Before we do this, we need to ensure that $n-m$ does not become negative. Thus, $\mathsf{E}\{L_Q\} = \sum_n \max\{n-m,0\}p(n)$.

```
m = 1
Lq = sum(max(n-m,0)*P[n] for n in range(len(P)))
Lq
```

```
0.45000000000000007
```

2.
```
labda = 10./60
Wq = Lq/labda # in minutes
Wq
```

```
2.7000000000000006
```

```
Wq/60 # in hours
```

```
0.04500000000000001
```

```
W = L/labda # in minutes
W
```

```
6.300000000000001
```

```
W/60 # in hours
```

```
0.10500000000000001
```

3.
```
from math import sqrt
var_L = sum((n-L)**2*P[n] for n in range(len(P)))
var_L
```

```
1.2475
```

```
sqrt(var_L)
```

```
1.116915395184434
```

```
var_Lq = sum((max(n-m,0)-Lq)**2*P[n] for n in range(len(P)))
var_Lq
```

```
0.6475
```

```
sqrt(var_Lq)
```

```
0.8046738469715541
```

4.
```
mu = 1./(W-Wq)
1./mu # in minutes
```

```
3.5999999999999996
```

```
rho = labda/mu
rho
```

```
0.6
```

```
rho = L-Lq
rho
```

```
0.6
```

This checks the previous line.

The utilization must also by equal to the fraction of time the server is busy.

```
u = 1 - P[0]
u
```

```
0.6
```

Yet another way: Suppose we have m servers. If the system is empty, all m servers are idle. If the system contains one customer, $m-1$ servers are idle. Therefore, in general, the average fraction of time the server is idle is

$$1-u=\sum_{n=0}^{\infty}\max\{n-m,0\}p_n,$$

as in case there are more than m customers in the system, the number of idle servers is 0.

```
idle = sum( max(m-n,0)*P[n] for n in range(len(P)))
idle
```

```
0.4
```

Solution 2.14.10: 1. $p(n)=(1-\rho)\rho^n$

2.
```
labda = 100. # per hour
mu = 140. # per hour
ES = 1./mu
rho = labda/mu
rho
```

```
0.7142857142857143
```

```
1-rho
```

```
0.2857142857142857
```

```
L = rho/(1.-rho)
L
```

```
2.5
```

```
Lq = rho**2/(1.-rho)
Lq
```

```
1.7857142857142858
```

```
W = 1./(1.-rho) * ES
W
```

```
0.024999999999999998
```

```
Wq = rho/(1.-rho) * ES
Wq
```

```
0.017857142857142856
```

Solution 2.14.11: 1. $\mathsf{E}\{W_Q\} = \frac{1}{\lambda}\frac{\rho^2}{1-\rho}$. Since $\mathsf{E}\{W_Q\}$ and λ is given we can use this formula to solve for ρ with the abc-formula (and using that $\rho > 0$):

```
labda = 5. # per minute
Wq = 1.
a = 1.
b = labda*Wq
c = -labda*Wq
rho = (-b + sqrt(b*b-4*a*c))/(2*a)
rho
```

```
0.8541019662496847
```

```
ES = rho/labda
ES
```

```
0.17082039324993695
```

2.

```
Lq = labda*Wq
Lq
```

```
5.0
```

```
W = Wq + ES
W
```

```
1.170820393249937
```

```
L = labda*W
L
```

```
5.854101966249685
```

3. The problem is to find n such that $\sum_{j=0}^{n} p_j > 0.9$.

```
total = 0.
j = 0
while total <= 0.9:
    total += (1-rho)*rho**j
    j += 1

total
```

```
0.9061042738330157
```

```
j
```

```
15
```

```
n = j- 1 # the number of chairs
```

Observe that j is one too high once the condition is satisfied, thus subtract one.

As a check, I use that $(1-\rho)\sum_{j=0}^{n}\rho^j = 1-\rho^{n+1}$.

```
1-rho**(n) #  this must be too small.
```

```
0.890064968964683
```

```
1-rho**(n+1) # this must be OK.
```

```
0.9061042738330156
```

And indeed, we found the right n.

Solution 2.14.12: Follows right away from the hint.

Solution 2.14.13: 1.
```
labda = 6.
mu = 5.
rho = labda/mu
c = 1
b = 2
```

Set $p(n) = \rho^n$ initially, and normalize later. Use the expressions for the $M(n)/M(n)/1$ queue. Observe that $\rho > 1$. Since the size of the system is $c + b + 1$ is finite, all formulas work for this case too.

There are 4 states in total: $0, 1, 2, 3$.

```
P = [rho**n for n in range(c+b+1)]
P
```

```
[1.0, 1.2, 1.44, 1.7279999999999998]
```

```
G = sum(P)
G
```

```
5.368
```

```
P = [p/G for p in P] # normalize
P
```

```
[0.18628912071535023,
 0.22354694485842025,
 0.2682563338301043,
 0.3219076005961251]
```

```
L = sum(n*P[n] for n in range(len(P)))
L
```

```
1.725782414307004
```

```
Lq = sum((n-c)*P[n] for n in range(c,len(P)))
Lq
```

```
0.9120715350223545
```

The number of jobs served per hour must be equal to the number of jobs accepted, i.e., not lost. The fraction of customers lost is equal to the fraction of customers that sees a full system.

```
lost = labda*P[-1] # the last element of P
lost
```

```
1.9314456035767507

accepted = labda*(1.-P[-1]) # rate at which jobs are accepted
accepted

4.06855439642325
```

2. Increase b to 5

```
b = 5
P = [rho**n for n in range(c+b+1)]
P

[1.0,
 1.2,
 1.44,
 1.7279999999999998,
 2.0736,
 2.4883199999999994,
 2.9859839999999993]

G = sum(P)
G

12.915903999999998

P = [p/G for p in P] # normalize
P

[0.07742392634692857,
 0.09290871161631428,
 0.11149045393957714,
 0.13378854472749255,
 0.16054625367299108,
 0.19265550440758927,
 0.23118660528910712]

L = sum(n*P[n] for n in range(len(P)))
L

3.7098374221424995

accepted = labda*(1.-P[-1])
accepted

4.6128803682653565
```

Solution 2.14.14: 1. `P = [0.4, 0.3, 0.2, 0.05, 0.05]`

```
c = 2
Lq = sum((n-c)*P[n] for n in range(c,len(P)))
Lq

0.15000000000000002

L= sum(n*P[n] for n in range(len(P)))
L
```

```
1.05
```

2.
```python
labda = 10./60
Wq = Lq/labda # in minutes
Wq
```

```
0.9000000000000001
```

```python
Wq/60 # in hours
```

```
0.015000000000000003
```

```python
W = L/labda
W
```

```
6.300000000000001
```

3.
```python
from math import sqrt
var_L = sum((n-L)**2*P[n] for n in range(len(P)))
var_L
```

```
1.2475
```

```python
sqrt(var_L)
```

```
1.116915395184434
```

4.
```python
var_q = sum((max(n-c,0)-Lq)**2*P[n] for n in range(len(P)))
var_q
```

```
0.22750000000000004
```

```python
sqrt(var_q)
```

```
0.4769696007084729
```

Solution 2.14.15: 1.
```python
import numpy as np
labda = [10., 15., 15., 10., 5.]
c = 2
mn = 2*np.ones(len(labda)+1, dtype=int)  # number of active servers
mn[0] = 0  # no service if system is empty
mn[1] = 1  # one busy server if just one job present
mu = 5*mn # service rate is 5 times no of active servers
mu
```

```
array([ 0,  5, 10, 10, 10, 10])
```

Since there can be arrivals in states $0,\dots,4$, the system can contain 0 to 5 customers, i.e., $p(0),\dots,p(5)$.

Use the level crossing result for the $M(n)/M(n)/1$ queue:

```python
P = [1]*(len(labda)+1)
for i in range(1,len(P)):
    P[i] = labda[i-1]/mu[i]*P[i-1]
print(P) # unnormalized probabilities
```

```
[1, 2.0, 3.0, 4.5, 4.5, 2.25]
```

```
G = sum(P) # normalization constant
G
```

```
17.25
```

```
P = [p/G for p in P]
P # normed
```

```
[0.057971014492753624,
 0.11594202898550725,
 0.17391304347826086,
 0.2608695652173913,
 0.2608695652173913,
 0.13043478260869565]
```

2. $\lambda = \sum_n \lambda(n)p(n)$.

```
labdaBar = sum(labda[n]*P[n] for n in range(len(labda)))
labdaBar
```

```
8.8405797101449277
```

The average number in the system is:

3.
```
Ls = sum(n*P[n] for n in range(len(P)))
Ls
```

```
2.9420289855072461
```

The average number in queue

```
c = 2
Lq = sum((n-c)*P[n] for n in range(c,len(P)))
Lq
```

```
1.1739130434782608
```

And now the waiting times:

```
Ws = Ls/labdaBar
Ws # time in the system
```

```
0.33278688524590161
```

```
Wq = Lq/labdaBar
Wq # time in queue
```

```
0.13278688524590163
```

Solution 2.14.16: 1. Would one server/person do?

```
labda = 28./60 # arrivals per minute
ES = 4.
labda*ES
```

```
1.8666666666666667
```

If $c = 1$, the load $\rho = \lambda \mathsf{E}\{S\}/c > 1$ is clearly undesirable for one server. We need at least two servers.

It is not relevant to focus on the time in the system , as time in service needs to be spent anyway. Hence, we focus on the waiting time in queue.

I just convert the formulas of (2.49) to python code. This saves me time during the computations.

```python
from math import factorial

def WQ(c, labda, ES):
    rho = labda*ES/c
    G = sum([(c*rho)**n/factorial(n) for n in range(c)])
    G += (c*rho)**c/(1.-rho)/factorial(c)
    Lq = (c*rho)**c/(factorial(c)*G) * rho/(1.-rho)**2
    return Lq/labda # Wq, Little's law
```

Considering the scenario with one server is superfluous as $\rho > 1$ in that case.

What is the waiting time for $c = 2$ servers?

```python
WQ(2, 28./60, 4) # in minutes
```

```
27.034482758620694
```

```python
WQ(2, 28./60, 4)/60. # in hours
```

```
0.4505747126436782
```

What is the waiting time for $c = 3$ servers?

```python
WQ(3, 28./60, 4)    # in minutes
```

```
1.3542675591474136
```

```python
WQ(3, 28./60, 4)/60. # in hours
```

```
0.022571125985790228
```

What is the waiting time for $c = 4$ servers?

```python
WQ(4, 28./60, 4) # in minutes
```

```
0.26778942672317635
```

```python
WQ(4, 28./60, 4)/60. # in hours
```

```
0.004463157112052939 6
```

In the next part of the question we will interpret these numbers.

2. Since both types of workers cost the same amount of money per unit time, it is best to divide the amount of waiting/idleness equally over both types of workers. I am inclined to reason as follows. The average amount of waiting time done by the maintenance workers per hour is $\lambda \mathsf{E}\{W_Q\}$. To see this, note that maintenance workers arrive at rate λ, and

each worker waits on average $\mathsf{E}\{W_Q\}$ minutes. Thus, worker time is wasted at rate $\lambda\,\mathsf{E}\{W_Q\}$. Intestingly, with Little's lawy, $\mathsf{E}\{L_Q\} = \lambda\,\mathsf{E}\{W_Q\}$, i.e., the rate at which workers waste capacity (i.e. waiting in queue) is $\mathsf{E}\{L_Q\}$. On the other hand, the rate of work capacity wasted by the tool crib employees being idle is $c - \lambda\,\mathsf{E}\{S\}$, as $\lambda\,\mathsf{E}\{S\}$ is the average number of servers busy, while c crib servers are available.

As both types of employees are equally expensive, we need to choose c such that the number of maintenance workers waiting (i.e., being idle because they are waiting in queue), is equal to the number of crib workers being idle. In other words, we search for a c such that $\mathsf{E}\{L_Q\} \approx c - \lambda\,\mathsf{E}\{S\}$ (where, of course, $\mathsf{E}\{L_Q\}$ depends on c.)

```
labda = 28./60
ES = 4.
c = 2
ELQ = labda*WQ(c, labda, ES)
ELQ

12.616091954022991

c-labda*ES

0.1333333333333333
```

Now the maintenance employees wait more than the tool crib employees.

```
c = 3
ELQ = labda*WQ(c, labda, ES)
ELQ

0.6319915276021264

c-labda*ES

1.1333333333333333

c = 4
ELQ = labda*WQ(c, labda, ES)
ELQ

0.1249683991374823

c-labda*ES

2.1333333333333333
```

Clearly, $c = 3$ should do.

Solution 2.14.17: 1.

```
labda = 11.
mu = 6
c = 2

rho = labda/mu
rho < c  # is the system stable?

True
```

Compute Ls and so on

```
G = sum(rho**n/factorial(n) for n in range(c))
G += rho**c/factorial(c)/(1.-rho/c)
P0 = 1./G
P0
```

```
0.043478260869565244
```

```
Lq = rho**(c+1)/c/(factorial(c)*(1.-rho/c)**2)*P0
Lq
```

```
9.64492753623188
```

```
Lq/labda # Wq
```

```
0.8768115942028981
```

```
Ls = Lq + rho
Ls
```

```
11.478260869565213
```

```
Ls/labda # Ws
```

```
1.0434782608695647
```

```
ES = 1./mu # expected service time
ES
```

```
0.16666666666666666
```

2.
```
mu = c*6
```

```
rho = labda/mu
```

```
P0 = 1-rho
P0
```

```
0.08333333333333337
```

```
Lq = rho**2/(1.-rho)
Lq
```

```
10.083333333333327
```

```
Lq/labda # Wq
```

```
0.9166666666666661
```

```
Ls = rho/(1.-rho)
Ls
```

```
10.999999999999995
```

```
Ls/labda # Ws
```

```
0.9999999999999996
```

```
ES = 1./mu # expected service time
ES
```

```
0.08333333333333333
```

It is of interest to compare the M/M/2 system to the M/M/1 system with twice the capacity.

Solution 2.14.18: Recall that for the $M/M/1$ queue, $\mathsf{E}\{S_r \mid S_r > 0\} = 1/\mu$. Applying Little's law to the queue, we have that $\lambda \mathsf{E}\{W_Q\} = EL_Q$, so that it follows that

$$\mathsf{E}\{W_Q\} = \rho \mathsf{E}\{S_r \mid S_r > 0\} + \mathsf{E}\{L_Q\} \mathsf{E}\{S\} = \frac{\rho}{\mu} + \rho \mathsf{E}\{W_Q\},$$

where we use that $\rho = \lambda \mathsf{E}\{S\}$. In words, the first term (at the right-hand side) is the probability to find a job in service, which is ρ times the expected remaining service time if the server is busy, which is $1/\mu$ for exponentially distributed processing times. The second term is the probability to find the server busy, so that the arriving job has to wait, times the expected waiting time.

For the $M/M/c$ queue we now derive the corresponding expressions for this first and second term.

For the first, note that the probability that the job has to wait is the same as the probability that all servers are occupied at arrival. This is, clearly, $\sum_{k=c}^{\infty} p(k)$. If all servers are busy, the expected time until a service completion is $1/c\mu$, since there are c servers and the expected time to completion is $1/\mu$ for each individual server. Thus, the time until the first departure is the minimum of the remaining services at each of the servers. This has expectation $1/c\mu$. Thus, for the $M/M/c$ queue

$$\rho \mathsf{E}\{S_r \mid S_r > 0\} = \sum_{k=c}^{\infty} p(k) \frac{1}{c\mu}.$$

For the second term, since there are c servers, jobs depart from the queue (hence enter service) at rate $c\mu$. The expected time in queue when there are n jobs in front, is therefore $n/c\mu$. Thus, on average, since the expected queue length is $\mathsf{E}\{L_Q\} = \sum_{k=c}^{\infty}(k-c)p(k)$, we get

$$\mathsf{E}\{L_Q\} \frac{\mathsf{E}\{S\}}{c} = \sum_{k=c}^{\infty} \frac{k-c}{c\mu} p(k).$$

All in all,

$$\begin{aligned}\mathsf{E}\{W_Q\} &= \rho \mathsf{E}\{S_r \mid S_r > 0\} + \mathsf{E}\{L_Q\} \frac{\mathsf{E}\{S\}}{c} \\ &= \sum_{k=c}^{\infty} p(k) \frac{1+k-c}{c\mu}.\end{aligned}$$

We can also give this a natural interpretation. How many jobs have to leave before the service of the arriving job can start? Well, the entire queue must be cleared plus one job in service. Thus, if there are $k-c$ job in queue, we have to wait for these jobs to finish, plus one of the jobs in service.

Observe also that, by PASTA, we have $\pi(k) = p(k)$, so that the arriving job also sees $p(k)$.

2.15 M^X/M/1 Queue: Expected Waiting Time

Theory and Exercises

It is not always the case that jobs arrive in single units, they can also arrive as batches. For instance, cars and busses arrive at a fast food restaurant so that the batch size consists of the number of people in the vehicle. In this section we derive a formula to compute the expected queue length for this queueing process.

Assume that jobs arrive as a Poisson process with rate λ and each *job* contains multiple *items*. Let A_k be the arrival time of job k and $A(t)$ the number of job arrivals up to time t. Denote by B_k the number of items that job k brings into the system. We assume that $\{B_k\}$ is an i.i.d. sequence of discrete random variables distributed as a generic random variable B such that $\mathsf{P}\{B = k\} = f(k)$, where $f(k)$ is a given set of probabilities. We write

$$G(k) = \mathsf{P}\{B > k\} = \sum_{m=k+1}^{\infty} f(m),$$

for the *survivor function* of B. We also assume that the service time of each item is exponentially distributed with average $1/\mu$. Thus, the average time to serve a job is $\mathsf{E}\{B\}\,\mathsf{E}\{S\} = \mathsf{E}\{B\}/\mu$. This queueing situation denoted by the shorthand $M^X/M/1$.

The first criterion we must check for the $M^X/M/1$ queue is the stability: the service rate must be larger than the arrival rate of work. To determine the latter, the total number of items $N(t)$ arrived up to time t must be equal to the number of arrivals $A(t)$ up to time t times the batch size of each arrival, i.e.,

$$N(t) = \sum_{k=1}^{\infty} B_k \mathbb{1}_{A_k \leq t} = \sum_{k=1}^{A(t)} B_k.$$

The (stochastic) process $\{N(t)\}$ is known as the *compound Poisson process*. Clearly, the rate at which items arrive is approximately

$$\frac{N(t)}{t} = \frac{A(t)}{t} \frac{1}{A(t)} \sum_{i=k}^{A(t)} B_k.$$

As in the limit $A(t)/t \to \lambda$ and $(A(t))^{-1} \sum_{k=1}^{A(t)} B_k \to \mathsf{E}\{B\}$, we see that

$$\frac{N(t)}{t} \to \lambda\,\mathsf{E}\{B\}.$$

Thus, for stability, it is necessary that the service rate $\mu = 1/\mathsf{E}\{S\}$ for items is larger than the rate at which items arrive. Hence we require that the load is such that

$$\rho = \lambda\,\mathsf{E}\{B\}\,\mathsf{E}\{S\} = \frac{\lambda\,\mathsf{E}\{B\}}{\mu} < 1.$$

For the average waiting time, we can use the derivation of $\mathsf{E}\{W_Q\} = \mathsf{E}\{S_r\}/(1-\rho)$ of Section 2.14 for inspiration. Suppose a batch finds $\mathsf{E}\{L\}$ items in the system upon arrival. By the memoryless property of the service distribution, the expected remaining service time of the item in service (if any) is $\mathsf{E}\{S\}$. Therefore the expected time the entire batch, i.e., the job, spends in queue is

$$\mathsf{E}\{W_{Q,b}\} = \mathsf{E}\{L\}\,\mathsf{E}\{S\};$$

compare Eq. 2.66. Note that this is not the same as $\mathsf{E}\{W_Q\}$, which is the expected time an *item* spends in queue. If B_r is the number of items of the batch currently in service ($B_r = 0$ if the server is idle), and $L_{Q,b}$ the number of batches in queue, then

$$\mathsf{E}\{L\} = \mathsf{E}\{L_{Q,b}\}\,\mathsf{E}\{B\} + \mathsf{E}\{B_r\}.$$

With Little's law, $\mathsf{E}\{L_{Q,b}\} = \lambda\,\mathsf{E}\{W_{Q,b}\}$, hence

$$\mathsf{E}\{W_{Q,b}\} = \lambda\,\mathsf{E}\{S\}\,\mathsf{E}\{B\}\,\mathsf{E}\{W_{Q,b}\} + \mathsf{E}\{S\}\,\mathsf{E}\{B_r\} = \rho\,\mathsf{E}\{W_{Q,b}\} + \mathsf{E}\{S\}\,\mathsf{E}\{B_r\},$$

hence,

$$\mathsf{E}\{W_{Q,b}\} = \frac{\mathsf{E}\{S\}}{1-\rho}\,\mathsf{E}\{B_r\}.$$

And then, from the first equation,

$$\mathsf{E}\{L\} = \frac{\mathsf{E}\{W_{Q,b}\}}{\mathsf{E}\{S\}} = \frac{\mathsf{E}\{B_r\}}{1-\rho}.$$

Below we prove that

$$\mathsf{E}\{B_r\} = \rho\frac{\mathsf{E}\{B^2\}}{2\,\mathsf{E}\{B\}} + \frac{\rho}{2}. \tag{2.67}$$

In an exercise below we rewrite this to

$$\mathsf{E}\{L\} = \frac{\mathsf{E}\{B_r\}}{1-\rho} = \frac{1+C_s^2}{2}\frac{\rho}{1-\rho}\,\mathsf{E}\{B\} + \frac{1}{2}\frac{\rho}{1-\rho}, \tag{2.68}$$

where

$$C_s^2 = \frac{\mathsf{V}\{B\}}{(\mathsf{E}\{B\})^2},$$

is the square coefficient of variation of the batch sizes. Observe that this is nearly the same as $M/G/1$ waiting time formula 2.72, a result we derive in Section 2.16.

Thus, to compute the average number of items in the system, we only need to know the first and second moment (or the variance) of the batch size B. Thus, no matter how 'complicated' the distribution of B, when its second moment exists, the average queue length and waiting time can be computed with the above result.

We now turn to proving (2.67) with sample-path arguments and counting; it is an elegant line of reasoning. First concentrate on the time the server is busy, i.e., just remove all idle times, and plot the remaining job size during the service of the job, c.f, Figure 2.20. Second, define R as the remaining number of items to be served of the job in service at some arbitrary point in time. Let us show that

$$\mathsf{P}\{R = i\} = \frac{\mathsf{P}\{B \geq i\}}{\mathsf{E}\{B\}} = \frac{G(i-1)}{\mathsf{E}\{B\}}.$$

In Figure 2.20 concentrate on level i. Only jobs whose initial batch size is larger or equal to i will produce, during its service, a remaining number of items equal to i. Thus, by counting, we see in Figure 2.20 that $\sum_{k=1}^n \mathbb{1}_{B_k \geq i}$ is the number of times there are precisely i remaining items. We also see that $\sum_{k=1}^n B_k$ is the total time the server is busy. Thus, the fraction of time there are i remaining items is

$$\frac{\sum_{k=1}^n \mathbb{1}_{B_k\geq i}}{\sum_{k=1}^n B_k}.$$

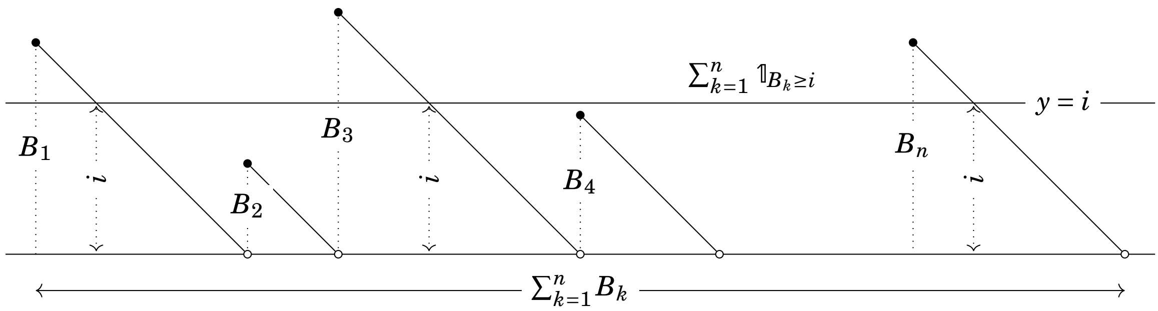


Figure 2.20: The remaining job size as a function of time. A batch crosses the line $y = i$ iff it contains an ith item. Thus, the number of ith items in n batches is $\sum_{k=1}^n \mathbb{1}_{B_k \geq i}$. The total number of items arrived is $\sum_k^n B_k$. The fraction of time an ith item is in service is therefore $\sum_k^n \mathbb{1}_{B_k \geq i} / \sum_k^n B_k$.

By PASTA this is also the fraction of jobs that see i items in service. Finally, assuming that the limits exist,

$$\mathsf{P}\{R = i\} = \lim_{n\to\infty} \frac{\sum_{k=1}^n \mathbb{1}_{B_k \geq i}}{\sum_{k=1}^n B_k} = \lim_{n\to\infty} \frac{n^{-1}\sum_{k=1}^n \mathbb{1}_{B_k \geq i}}{n^{-1}\sum_{k=1}^n B_k} = \frac{G(i-1)}{\mathsf{E}\{B\}}.$$

With this expression for $\mathsf{P}\{R = i\}$, the expected remaining time becomes

$$\begin{aligned}
\mathsf{E}\{R\} &= \sum_{i=1}^\infty i \mathsf{P}\{R = i\} = \sum_{i=1}^\infty i \frac{G(i-1)}{\mathsf{E}\{B\}} \\
&= \sum_{i=0}^\infty (i+1) \frac{G(i)}{\mathsf{E}\{B\}} = \sum_{i=0}^\infty i \frac{G(i)}{\mathsf{E}\{B\}} + \sum_{i=0}^\infty \frac{G(i)}{\mathsf{E}\{B\}} \\
&= \frac{\mathsf{E}\{B^2\}}{2\mathsf{E}\{B\}} - \frac{\mathsf{E}\{B\}}{2\mathsf{E}\{B\}} + \frac{\mathsf{E}\{B\}}{\mathsf{E}\{B\}} = \frac{\mathsf{E}\{B^2\}}{2\mathsf{E}\{B\}} + \frac{1}{2},
\end{aligned}$$

where we use the results of the exercises below.

Finally, recalling that in the above we conditioned on the server being busy, we get that $\mathsf{E}\{B_r\} = \rho \mathsf{E}\{R\}$. Combining this with the above expression we obtain (2.67).

Remark 2.15.1. The above derivation for $\mathsf{P}\{R = i\}$ is important for its own sake. $\mathsf{P}\{R = i\}$ is known as the *remaining lifetime distribution* or the *residual life* and used in, for instance, maintenance planning. The derivation is worth remembering, in particular the sample path derivation.

Exercise 2.15.1. Why is $\rho = \lambda \mathsf{E}\{B\} \mathsf{E}\{S\}$ the appropriate definition of load for the $M^X/M/1$ queue?

Exercise 2.15.2. Relate $f(k)$, the distribution $F(k)$ of the batch size B and the survivor function G. Which of the three alternatives is true: $G(k) = 1 - F(k)$, $G(k) = 1 - F(k-1)$, $G(k) = 1 - F(k+1)$?

Exercise 2.15.3. Show that the expression $\mathsf{E}\{L(M^X/M/1)\}$ reduces to $\mathsf{E}\{L(M/M/1)\}$ when the batch size is 1.

Exercise 2.15.4. What is $\mathsf{E}\{L\}$ in case $B_k = 3$ always? Suppose $\lambda = 1$ and $\mu = 6$ so that we can get a numerical answer.

Exercise 2.15.5. Show that

$$\frac{\lambda}{\mu}\mathsf{E}\left\{B^2\right\} = \rho(1+C_s^2)\mathsf{E}\{B\},$$

Exercise 2.15.6. If the batch size is geometrically distributed with parameter p, what is $\mathsf{E}\{L\}$?

Exercise 2.15.7. A common operational problem is a machine that receives batches of various sizes. Management likes to know how a reduction of the variability of the batch sizes affects the average queueing time. Suppose, for the sake of an example, that the batch size

$$\mathsf{P}\{B=1\} = \mathsf{P}\{B=2\} = \mathsf{P}\{B=3\} = \frac{1}{3}.$$

Batches arrive at rate 1 per hour. The average processing for an item is 25 minutes. How much does the waiting time decrease if batch sizes are 2 always?

Exercise 2.15.8. In a production enviroment, a machine replenishes an inventory of items (e.g., hamburgers) at a fixed rate of 1 per 3 minutes. If the inventory reaches some level S, the machine stops. Customers arrive at rate of 6 per hour. A customer can buy items in different quanties, $B = 1,2,3,4$, all with equal probability. What is a sensible value for the stopping level S?

Exercise 2.15.9. Use indicator functions to prove that

$$\sum_{n=0}^{\infty} G(n) = \mathsf{E}\{B\}.$$

Exercise 2.15.10. Use indicator functions to prove that

$$\sum_{i=0}^{\infty} iG(i) = \frac{\mathsf{E}\{B\}^2}{2} - \frac{\mathsf{E}\{B\}}{2}.$$

Exercise 2.15.11. Use indicator functions to prove that for a continuous random variable S with distribution function F,

$$\mathsf{E}\{S\} = \int_0^\infty x\,\mathrm{d}F = \int_0^\infty G(y)\,\mathrm{d}y,$$

where $G(x) = 1-F(x)$.

Exercise 2.15.12. Use indicator functions to prove that for a continuous random variable S with distribution function F,

$$\mathsf{E}\left\{S^2\right\} = \int_0^\infty x^2\,\mathrm{d}F = 2\int_0^\infty yG(y)\,\mathrm{d}y,$$

where $G(x) = 1-F(x)$.

Exercise 2.15.13. Use integration by parts to show that for a continuous random variable S with distribution function F and survivor function $G = 1-F$,

$$\int_0^\infty yG(y)\,\mathrm{d}y = \frac{\mathsf{E}\left\{S^2\right\}}{2}.$$

Exercise 2.15.14. Use that $\mathsf{E}\{S\} = \int_0^\infty x\,\mathrm{d}F = \int_0^\infty G(y)\,\mathrm{d}y$ to check that $\mathsf{E}\{S\} = \mu^{-1}$ if $F(x) = 1-e^{-\mu x}$.

Exercise 2.15.15. Compare $\mathsf{E}\left\{L(M^X/M/1)\right\}$ to $\mathsf{E}\{L(M/M/1)\}$ when the loads are the same. What do you conclude?

Hints

Hint 2.15.2: This exercise is just meant to become familiar with the notation.

Hint 2.15.3: What is the distribution of the batch size B for the $M/M/1$ queue?

Hint 2.15.4: Use Eq. 2.68. What are $\mathsf{E}\{B^2\}$, $\mathsf{E}\{B\}$ and $\mathsf{V}\{B\}$ for this case?

Hint 2.15.6: $f_k = q^{k-1}p$ with $q = 1-p$. Use generating functions to compute $\mathsf{E}\{B\}$ and $\mathsf{E}\{B^2\}$

Hint 2.15.8: Realize that the inventory process $I(t)$ behaves as $I(t) = S - L(t)$ where $L(t)$ is a suitable queueing process.

Hint 2.15.9: Write $\sum_{n=0}^\infty G(n) = \sum_{n=0}^\infty \sum_{i=n+1}^\infty \mathsf{P}\{B = i\}$ and then do the algebra.

Hint 2.15.10: $\sum_{i=0}^\infty iG(i) = \sum_{n=0}^\infty \mathsf{P}\{B = n\} \sum_{i=0}^\infty i\,1\{n \geq i+1\}$.

Hint 2.15.11: $\mathsf{E}\{S\} = \int_0^\infty x \,\mathrm{d}F = \int_0^\infty \int_0^\infty 1_{y\leq x} \,\mathrm{d}y \,\mathrm{d}F(x)$.

Hint 2.15.12: $\int_0^\infty yG(y)\,\mathrm{d}y = \int_0^\infty y \int_0^\infty 1\{y \leq x\} f(x)\,\mathrm{d}x\,\mathrm{d}y$.

Solutions

Solution 2.15.1: Jobs arrive at rate λ. The average number of items per job is $\mathsf{E}\{B\}$. Hence, work arrives as rate $\lambda \mathsf{E}\{B\}\mathsf{E}\{S\}$.This rate must be less than 1 for stability.

Solution 2.15.2:

$$
\begin{aligned}
f(k) &= \mathsf{P}\{B = k\} = \mathsf{P}\{B \leq k\} - \mathsf{P}\{B \leq k-1\} = F(k) - F(k-1), \\
G(k) &= \mathsf{P}\{B > k\} = 1 - \mathsf{P}\{B \leq k\} = 1 - F(k).
\end{aligned}
$$

It is all too easy to make, so called, off-by-one errors, such as in the three alternatives above. I nearly always check simple cases to prevent such simple mistakes. I advise you to acquire the same habit.

Solution 2.15.3: For the $M/M/1$ queue, each job contains just one item. Thus, $B \equiv 1$, hence $\mathsf{P}\{B = 1\} = 1$, $\mathsf{E}\{B^2\} = \mathsf{E}\{B\} = 1$. Therefore, $\mathsf{E}\{B_r(M/M/1)\} = \rho$, and $\mathsf{E}\{L(M/M/1)\} = \rho/(1-\rho)$.

Such checks are always important to do. Does the new result reduce to the results of the models you know?

Solution 2.15.4: As B is constant and equal to 3, $\mathsf{E}\{B^2\} = 9$, hence $\mathsf{V}\{B\} = 0$, hence $C_s^2 = 0$. Also, $\rho = \lambda \mathsf{E}\{B\}/\mu = 1 \cdot 3/6 = 1/2$. Hence

$$
\mathsf{E}\{L\} = \frac{1}{2}\frac{1/2}{1-1/2} \cdot 3 + \frac{1}{2}\frac{1/2}{1-1/2}.
$$

Solution 2.15.5: We have

$$
\begin{aligned}
\frac{\lambda}{\mu}\mathsf{E}\{B^2\} &= \frac{\lambda \mathsf{E}\{B\}}{\mu}\frac{\mathsf{E}\{B^2\}}{(\mathsf{E}\{B\})^2}\mathsf{E}\{B\} = \rho\frac{\mathsf{E}\{B^2\}}{(\mathsf{E}\{B\})^2}\mathsf{E}\{B\} \\
&= \rho\frac{(\mathsf{E}\{B\})^2 + \mathsf{V}\{B\}}{(\mathsf{E}\{B\})^2}\mathsf{E}\{B\} = \rho(1 + C_s^2)\mathsf{E}\{B\}.
\end{aligned}
$$

Solution 2.15.6: We need $\mathsf{V}\{B\}$ and $\mathsf{E}\{B\}$. Consider

$$\begin{aligned}\phi(z) = \mathsf{E}\left\{z^B\right\} &= \sum_{k=0}^{\infty} z^k \mathsf{P}\{B=k\} = \sum_{k=0}^{\infty} z^k f_k \\ &= \sum_{k=0}^{\infty} z^k p q^{k-1} = \frac{p}{q}\sum_{k=0}^{\infty}(qz)^k = \frac{p}{q}\frac{1}{1-qz}.\end{aligned}$$

Then

$$\mathsf{E}\{B\} = \phi'(1) = \frac{p}{q}\frac{q}{(1-qz)^2}\bigg|_{z=1} = \frac{p}{(1-q)^2} = \frac{1}{p},$$

and

$$\mathsf{E}\{B(B-1)\} = \phi''(1) = \frac{p}{q}\frac{2q^2}{(1-qz)^3}\bigg|_{z=1} = 2\frac{q}{p^2} = \frac{2}{p^2} - \frac{2}{p}.$$

Hence,

$$\mathsf{E}\left\{B^2\right\} = \frac{2}{p^2} - \frac{2}{p} + \mathsf{E}\{B\} = \frac{2}{p^2} - \frac{1}{p},$$

and

$$\mathsf{V}\{B\} = \mathsf{E}\left\{B^2\right\} - (\mathsf{E}\{B\})^2 = \frac{2}{p^2} - \frac{1}{p} - \frac{1}{p^2} = \frac{1}{p^2} - \frac{1}{p},$$

and

$$C_s^2 = \frac{\mathsf{V}\{B\}}{(\mathsf{E}\{B\})^2} = p^2\left(\frac{1}{p^2} - \frac{1}{p}\right) = 1-p$$

Then,

$$(1+C_s^2)/2 = 1 - p/2.$$

and

$$\mathsf{E}\{L\} = \left(1-\frac{p}{2}\right)\frac{\rho}{1-\rho}\frac{1}{p} + \frac{1}{2}\frac{\rho}{1-\rho} = \frac{\rho}{1-\rho}\frac{1}{p}.$$

Can we check this in a simple way? If $\mathsf{P}\{B=1\} = f_1 = p = 1$, then $\mathsf{E}\{L\} = \rho/(1-\rho)$. Thus, we get the result for the $M/M/1$ queue. The result is at least consistent with earlier work.

Solution 2.15.7: Start with the simple case, $B \equiv 2$. Then $\mathsf{V}\{B\} = 0$ and $\mathsf{E}\{B\} = 2$. The load is $\rho = \lambda\,\mathsf{E}\{B\}\,\mathsf{E}\{S\} = 1 \cdot 2 \cdot 25/60 = 5/6$. hence,

$$\mathsf{E}\{L\} = \frac{1}{2}\frac{5/6}{1/6}2 + \frac{1}{2}\frac{5/6}{1/6} = 5 + \frac{5}{2}.$$

Now the other case. $\mathsf{E}\left\{B^2\right\} = (1+4+9)/3 = 14/3$. Hence, $\mathsf{V}\{B\} = 14/3 - 4 = 2/3$. Hence,

$$C_s^2 = \frac{\mathsf{V}\{B\}}{(\mathsf{E}\{B\})^2} = \frac{2/3}{4} = \frac{1}{6}.$$

And thus,

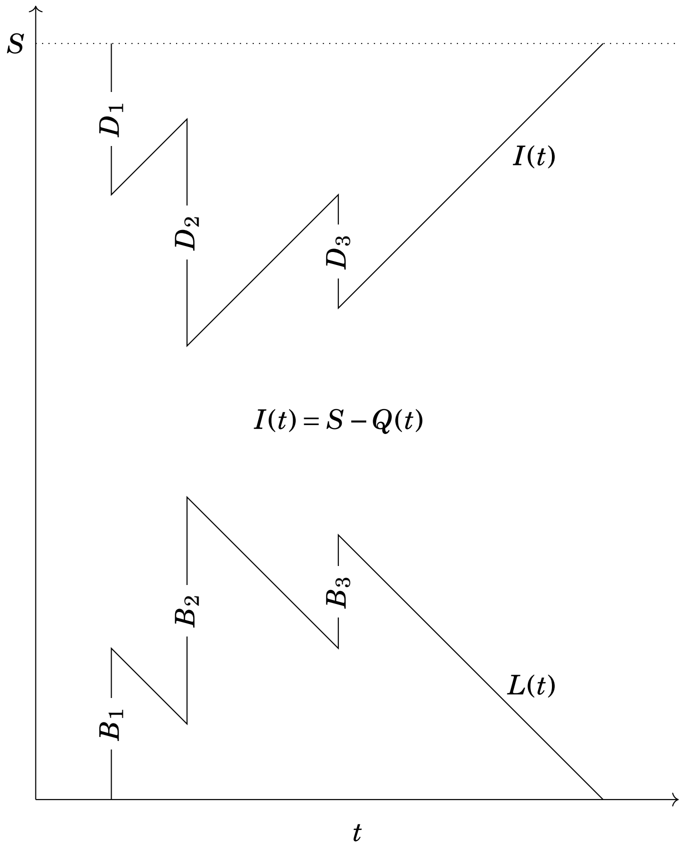
$$\mathsf{E}\{L\} = \frac{1+1/6}{2}\frac{5/6}{1/6}2 + \frac{1}{2}\frac{5/6}{1/6} = \frac{7}{6}5 + \frac{5}{2}.$$

If we divide these two answers, we see that the ratio between $\mathsf{E}\{L\}$ for both answers is 10/9. In other words, we can reduce about 10% of the number of items in the system by working in fixed batch sizes.

Observe how easy it is with these models to get insight into the order of magnitude that can be achieved with changing work habits, such as working in contant batch sizes.

Observe also that it is up to management to decide whether this reduction outweighs any efforts to reduce the variation in batch sizes. (In our model we assume that $B = 2$, i.e., we removed all variability. This is the best possible. Real life reductions will typically achieve less than the 10% reduction.)

Solution 2.15.8: First, to see why inventory systems and queueing systems are two sides of one coin, consider the figure below. Here we write B_k for the size of the kth batch that arrives at the queue and D_k for the size of the k-th demand at the inventory system. Note that when the machine is on, it replenishes the inventory. This is similar to the server in the queueing system that is processing jobs.



Now, continuing with the problem, consider a queueing system with job arrival rate $\lambda = 6$ per hour and the jobs have batch sizes as indicated in the problem. The average number of items in the system follows like this

$$
\begin{aligned}
\mathsf{E}\{B\} &= \frac{1+2+3+4}{4} = \frac{5}{2} \\
\mathsf{E}\{B^2\} &= \frac{1+4+9+16}{4} = \frac{30}{4} \\
\mathsf{V}\{B^2\} &= \frac{30}{4} - \frac{25}{4} = \frac{5}{4} \\
C_s^2 &= \frac{5/4}{25/4} = \frac{1}{5} \\
\rho &= \lambda\,\mathsf{E}\{B\}\,\mathsf{E}\{S\} = 6\frac{5}{2}\frac{1}{20} = \frac{3}{4}.
\end{aligned}
$$

Hence,

$$\mathsf{E}\{L\} = \frac{1+1/5}{2}\frac{3/4}{1/4}\frac{5}{2} + \frac{1}{2}\frac{3/4}{1/4} = 6.$$

Thus, if the level is set to $S = 4$, then on average there will be two items short. Clearly, then, S should be at least 6. However, $\mathsf{E}\{L\}$ is just the average. Roughly speaking, in this case half of the demand will then be lost. So, if we take variability into account, S should be quite a bit bigger than 6. A more detailed analysis is necessary to determine the right value of S such that not more than a certain fraction of demand is lost. We will address this issue in Section 2.17.

Solution 2.15.9: Realize that this sort of problem is just a regular probability theory problem, nothing fancy. We use/adapt the tools you learned in calculus to carry out 2D integrals (or in this case 2D summations.)

$$\begin{aligned}
\sum_{n=0}^{\infty} G(n) &= \sum_{n=0}^{\infty} \mathsf{P}\{B > n\} = \sum_{n=0}^{\infty}\sum_{i=n+1}^{\infty} \mathsf{P}\{B = i\} \\
&= \sum_{n=0}^{\infty}\sum_{i=0}^{\infty} 1\{n < i\}\mathsf{P}\{B = i\} = \sum_{i=0}^{\infty}\sum_{n=0}^{\infty} 1\{n < i\}\mathsf{P}\{B = i\} \\
&= \sum_{i=0}^{\infty} i\mathsf{P}\{B = i\} = \mathsf{E}\{B\}.
\end{aligned}$$

In case you are interested in mathematical justifications: the interchange of the two summations is allowed because the summands are all positive. (Interchanging the order of summations or integration is not allways allowed because the results can be different when part of the integrand is negative. Check Fubini's theorem for more on this if you are interested.)

Solution 2.15.10:

$$\begin{aligned}
\sum_{i=0}^{\infty} iG(i) &= \sum_{i=0}^{\infty} i \sum_{n=i+1}^{\infty} \mathsf{P}\{B = n\} = \sum_{n=0}^{\infty} \mathsf{P}\{B = n\} \sum_{i=0}^{\infty} i1\{n \geq i+1\} \\
&= \sum_{n=0}^{\infty} \mathsf{P}\{B = n\} \sum_{i=0}^{n-1} i = \sum_{n=0}^{\infty} \mathsf{P}\{B = n\}\frac{(n-1)n}{2} \\
&= \sum_{n=0}^{\infty} \frac{n^2}{2}\mathsf{P}\{B = n\} - \frac{\mathsf{E}\{B\}}{2} = \frac{\mathsf{E}\{B\}^2}{2} - \frac{\mathsf{E}\{B\}}{2}.
\end{aligned}$$

Solution 2.15.11:

$$\begin{aligned}
\mathsf{E}\{S\} &= \int_0^{\infty} x\,\mathrm{d}F = \int_0^{\infty}\int_0^x \mathrm{d}y\,\mathrm{d}F(x) \\
&= \int_0^{\infty}\int_0^{\infty} 1_{y\leq x}\,\mathrm{d}y\,\mathrm{d}F(x) = \int_0^{\infty}\int_0^{\infty} 1_{y\leq x}\,\mathrm{d}F(x)\,\mathrm{d}y \\
&= \int_0^{\infty}\int_y^{\infty} \mathrm{d}F(x)\,\mathrm{d}y = \int_0^{\infty} G(y)\,\mathrm{d}y
\end{aligned}$$

Solution 2.15.12:

$$\begin{aligned}
\int_0^{\infty} yG(y)\,\mathrm{d}y &= \int_0^{\infty} y\int_y^{\infty} f(x)\,\mathrm{d}x\,\mathrm{d}y = \int_0^{\infty} y\int_0^{\infty} 1\{y \leq x\}f(x)\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_0^{\infty} f(x)\int_0^{\infty} y1\{y \leq x\}\,\mathrm{d}x\,\mathrm{d}y = \int_0^{\infty} f(x)\int_0^{x} y\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_0^{\infty} f(x)\frac{x^2}{2}\,\mathrm{d}x = \frac{\mathsf{E}\{S\}^2}{2}.
\end{aligned}$$

Solution 2.15.13:

$$\int_0^\infty yG(y)\,\mathrm{d}y = \frac{y^2}{2}G(y)\bigg|_0^\infty - \int_0^\infty \frac{y^2}{2}g(y)\,\mathrm{d}y = \int_0^\infty \frac{y^2}{2}f(y)\,\mathrm{d}y = \frac{\mathsf{E}\{S^2\}}{2}, \tag{2.69}$$

since $g(y) = G'(y) = -F'(y) = f(y)$.

Solution 2.15.14: If $F(x) = 1 - e^{-\mu x}$, we obtain that

$$\mathsf{E}\{S\} = \int_0^\infty e^{-\mu x}\,\mathrm{d}x = \mu^{-1}\int_0^\infty e^{-x}\,\mathrm{d}x = \mu^{-1}.$$

Solution 2.15.15:

$$\frac{\mathsf{E}\{L(M^X/M/1)\}}{\mathsf{E}\{L(M/M/1)\}} = \frac{\mathsf{E}\{B_r\}}{\rho} = \frac{\mathsf{E}\{B^2\}}{2\,\mathsf{E}\{B\}} + \frac{1}{2}.$$

With this we can check whether this condition

$$1 \leq \frac{\mathsf{E}\{L(M^X/M/1)\}}{\mathsf{E}\{L(M/M/1)\}} = \frac{\mathsf{E}\{B^2\}}{2\,\mathsf{E}\{B\}} + \frac{1}{2}$$

is always true. Clearly, it reduces to

$$\mathsf{E}\{B\} \leq \mathsf{E}\{B^2\}.$$

Multiply this by $\mathsf{E}\{B\}$ for reasons to become clear presently to get

$$(\mathsf{E}\{B\})^2 \leq \mathsf{E}\{B^2\}\,\mathsf{E}\{B\}.$$

So, the initial inequality is converted to this, and we like to know whether this always true.

To see this, we can use Jensen's inequality $\phi(\mathsf{E}\{X\}) \leq \mathsf{E}\{\phi(X)\}$ when ϕ is convex. In this case take $\phi(x) = x^2$, so that Jensen's inequality states that $(\mathsf{E}\{B\})^2 \leq \mathsf{E}\{B^2\}$. (BTW, note that Jensen's inequality implies that $\mathsf{V}\{X\} = \mathsf{E}\{X^2\} - (\mathsf{E}\{X\})^2 \geq 0$.) Now noting that $B \geq 1$, as a job minimally contains one item, we get

$$\begin{aligned}(\mathsf{E}\{B\})^2 &\leq \mathsf{E}\{B^2\}, \quad \text{by Jensen's inequality}\\ &\leq \mathsf{E}\{B^2\}\,\mathsf{E}\{B\}, \quad \text{as } B \geq 1.\end{aligned}$$

Clearly, this is the inequality we tried to show. As a result,

$$1 \leq \frac{\mathsf{E}\{L(M^X/M/1)\}}{\mathsf{E}\{L(M/M/1)\}}$$

for all B.

In conclusion, if work arrives in batches, the average number of jobs in the system, increases, hence the average waiting time increases.

2.16 M/G/1 Queue: Expected Waiting Time

Theory and Exercises

Let's focus on one queue in a supermarket, served by one cashier, and assume that customers do not jockey, i.e., change queue. What can we say about the average waiting time in queue if service times are not exponential, like in the $M/M/1$ queue, but have a more general distribution? One of the celebrated results of queueing theory is the Pollackzek-Khintchine formula by which we can compute the average waiting time formula for the $M/G/1$ queue. In this section we derive this result by means of sample path arguments.

Recall that Eq. (2.64) states that

$$\mathsf{E}\{W_Q\} = \frac{\mathsf{E}\{S_r\}}{1-\rho}$$

This equation is valid for the $M/G/1$ queue, as in the derivation we used the PASTA property but did not make any assumption about the service time distribution, except that it has a finite mean. To use it, we need to compute the average remaining service time for generally distributed service times. This is what we will do here.

Consider the kth job of some sample path the $M/G/1$ queueing process. This job requires S_k units of service, let its service time start at time A_k so that it departs the server at time $D_k = A_k + S_k$, c.f. Figure 2.21. Define

$$R_k(s) = (D_k - s)\mathbb{1}_{A_k \leq s < D_k}$$

as the remaining service time at time s of job k. So see this, observe that when the time $s \in [A_k, D_k)$, the remaining service time until job k departs is $D_k - s$, while if $s \not\in [A_k, D_k)$, job k is not in service so it cannot have any remaining service as well.

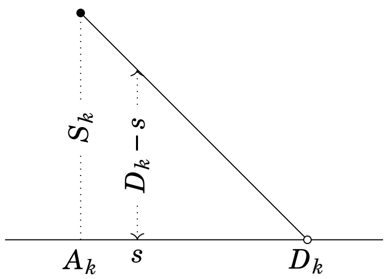


Figure 2.21: Remaining service time.

For the average remaining service time, we can add all remaining service times up to some time t, and then divide by t, specifically,

$$R(t) = \frac{1}{t}\int_0^t \sum_{k=1}^{\infty} R_k(s)\,\mathrm{d}s,$$

so that we can define

$$\mathsf{E}\{S_r\} = \lim_{t\to\infty} R_t,$$

provided this limit exists along the sample path.

Now observe that as t can be somewhere in a service interval, it is easiest to consider $R(t)$ at departure times $\{D_n\}$, as at a time D_n precisely n jobs departed. Thus,

$$R(D_n) = \frac{1}{D_n}\int_0^{D_n}\sum_{k=1}^{\infty} R_k(s)\,\mathrm{d}s = \frac{1}{D_n}\sum_{k=1}^{n}\int_0^{D_n} R_k(s)\,\mathrm{d}s,$$

the interchange being allowed by the positivity of $R_k(s)$. But, if $k < n$ so that $D_k < D_n$,

$$\begin{aligned}
\int_0^{D_n} R_k(s)\,\mathrm{d}s &= \int_0^{D_n} (D_k - s)\mathbb{1}_{A_k \leq s < D_k}\,\mathrm{d}s = \int_{A_k}^{D_k} (D_k - s)\,\mathrm{d}s \\
&= \int_{A_k}^{A_k+S_k} (A_k + S_k - s)\,\mathrm{d}s = \int_0^{S_k} (S_k - s)\,\mathrm{d}s \\
&= \int_0^{S_k} s\,\mathrm{d}s = \frac{S_k^2}{2}.
\end{aligned}$$

With this,

$$R(D_n) = \frac{1}{D_n}\sum_{k=1}^{n}\frac{S_k^2}{2} = \frac{n}{D_n}\frac{1}{n}\sum_{k=1}^{n}\frac{S_k^2}{2}.$$

Finally, assuming that along this sample path, $D_n \to \infty$ as $t \to \infty$, $n/D_n \to \gamma$, $\gamma = \lambda$ by rate stability, and the limit $n^{-1}\sum_{k=1}^{n} S_k^2$ exists as $n \to \infty$, we get

$$\mathsf{E}\{S_r\} = \lim_{D_n\to\infty} R(D_n) = \frac{\lambda}{2}\mathsf{E}\{S^2\}. \tag{2.70}$$

In the exercises we ask you to prove that

$$\mathsf{E}\{S_r\} = \frac{1+C_s^2}{2}\rho\,\mathsf{E}\{S\}, \tag{2.71}$$

where

$$C_s^2 = \frac{\mathsf{V}\{S\}}{(\mathsf{E}\{S\})^2},$$

With the above we have obtained the *Pollackzek-Khintchine* (PK) formula for the average waiting time in queue:

$$\mathsf{E}\{W_Q\} = \frac{1}{1-\rho}\mathsf{E}\{S_r\} = \frac{1+C_s^2}{2}\frac{\rho}{1-\rho}\mathsf{E}\{S\}. \tag{2.72}$$

This proves to be a very useful result; a few of the problems below will clarify how to use it.

Exercise 2.16.1. Show from Eq. (2.70) that

$$\mathsf{E}\{S_r\,|\,S_r > 0\} = \frac{\mathsf{E}\{S^2\}}{2\,\mathsf{E}\{S\}}$$

Exercise 2.16.2. Show (2.71).

Exercise 2.16.3. Show that $\mathsf{E}\{S_r\,|\,S_r > 0\} = 1/2\mu$ when S is deterministic and equal to $1/\mu$.

Exercise 2.16.4. Show that $\mathsf{E}\{S_r\,|\,S_r > 0\} = 2\mu^{-1}/3$ when S is $U[0,2/\mu]$.

Exercise 2.16.5. Show that $\mathsf{E}\{S_r\,|\,S_r > 0\} = \mu^{-1}$ when $S \sim \exp(\mu)$.

Exercise 2.16.6. A queueing system receives Poisson arrivals at the rate of 5 per hour. The single server has a uniform service time distribution, with a range of 4 minutes to 6 minutes. Determine $\mathsf{E}\{L_Q\}$, $\mathsf{E}\{L\}$, $\mathsf{E}\{W_Q\}$, $\mathsf{E}\{W\}$.

Exercise 2.16.7. Consider a workstation with just one machine. Jobs arrive, roughly, in accordance to Poisson process, with rate $\lambda = 3$ per day. The average service time $\mathsf{E}\{S\} = 2$ hours, $C_s^2 = 1/2$, and the shop is open for 8 hours. What is $\mathsf{E}\{W_Q\}$?

Suppose the expected waiting time has to be reduced to 1h. How to achieve this?

Exercise 2.16.8. (Hall 5.16) The manager of a small firm would like to determine which of two people to hire. One employee is fast, on average, but is somewhat inconsistent. The other is a bit slower, but very consistent. The first has a mean service time of 2 minutes, with a standard deviation of 1 minute. The second has a mean service time of 2.1 minute, with a standard deviation of 0.1 minute. If the arrival rate is Poisson with rate 20 per hour, which employee would minimize $\mathsf{E}\{L_Q\}$? Which would minimize $\mathsf{E}\{L\}$?

Exercise 2.16.9. Show that when services are exponential, the expected waiting time $\mathsf{E}\{W_Q(M/G/1)\}$ reduces to $\mathsf{E}\{W_Q(M/M/1)\}$.

Exercise 2.16.10. Compute $\mathsf{E}\{W_Q\}$ and $\mathsf{E}\{L\}$ for the $M/D/1$ queue. Assume that the service time is always T. Compare $\mathsf{E}\{L(M/D/1)\}$ to $\mathsf{E}\{L(M/M/1)\}$ where the mean service time is the same in both cases.

Exercise 2.16.11. Compute $\mathsf{E}\{L\}$ for the $M/G/1$ queue with $G = U[0,A]$, i.e., the uniform distribution on $[0,A]$.

Exercise 2.16.12. Show that for the $M/G/1$ queue, the expected idle time is $\mathsf{E}\{I\} = 1/\lambda$.

Exercise 2.16.13. Use Exercise 2.7.6 to show that for the $M/G/1$ queue, the expected busy time is given by $\mathsf{E}\{B\} = \mathsf{E}\{S\}/(1-\rho)$.

Exercise 2.16.14. What is the utilization of the $M/G/1/1$ queue?

Exercise 2.16.15. For the $M/G/1/1$ queue what is the fraction of jobs rejected (hence, what is the fraction of accepted jobs)?

Exercise 2.16.16. Why is the fraction of lost jobs at the $M/G/1/1$ queue not necessarily the same as for a $G/G/1/1$ queue with the same load?

Hints

Hint 2.16.11: Integrate $A^{-1}\int x\,\mathrm{d}x$, and likewise for the second moment.

Hint 2.16.12: What is the average time between two arrivals? Observe that the interarrivals are memoryless, hence the average time until the next arrival after the server becomes idle is also $1/\lambda$.

Hint 2.16.14: The job in the system is also in service (it is an $M/G/1/1$ queue...). The average idle time is $1/\lambda$. The average busy time is $\mathsf{E}\{S\}$–as there are no jobs in queue. The load ρ must also be equal to the fraction of time the server is busy.

Hint 2.16.15: The rate of accepted jobs is $\lambda\pi(0)$. What is the load of these jobs? Equate this to $1-\pi(0$ as this must also be the load. Then solve for $\pi(0)$.

Hint 2.16.16: Provide an example.

Solutions

Solution 2.16.1: From Eq. (2.62) and the sentences above this equation, we have that $\mathsf{E}\{S_r\} = \rho\,\mathsf{E}\{S_r \,|\, S_r > 0\} = \lambda\,\mathsf{E}\{S\}\,\mathsf{E}\{S_r \,|\, S_r > 0\}$. Hence

$$\lambda\,\mathsf{E}\{S_r \,|\, S_r > 0\} = \frac{\mathsf{E}\{S_r\}}{\mathsf{E}\{S\}} = \lambda \frac{\mathsf{E}\{S^2\}}{2\,\mathsf{E}\{S\}}.$$

Solution 2.16.2: Use that

$$\frac{\mathsf{E}\{S^2\}}{(\mathsf{E}\{S\})^2} = \frac{(\mathsf{E}\{S^2\} - (\mathsf{E}\{S\})^2) + (\mathsf{E}\{S\})^2}{(\mathsf{E}\{S\})^2} = \frac{\mathsf{V}\{S\} + (\mathsf{E}\{S\})^2}{(\mathsf{E}\{S\})^2} = C_s^2 + 1.$$

Then

$$\mathsf{E}\{S_r\} = \frac{\lambda\,\mathsf{E}\{S^2\}}{2} = \frac{\mathsf{E}\{S^2\}\,\lambda(\mathsf{E}\{S\})^2}{2(\mathsf{E}\{S\})^2} = \frac{\mathsf{E}\{S^2\}}{2(\mathsf{E}\{S\})^2}\rho\,\mathsf{E}\{S\}.$$

Solution 2.16.3: When S is deterministic, $\mathsf{P}\{S = 1/\mu\} = 1$. Therefore, $\mathsf{E}\{S\} = 1/\mu$ and $\mathsf{E}\{S^2\} = 1/\mu^2$.

Solution 2.16.4: When S is uniform, $\mathsf{E}\{S\} = \alpha^{-1}\int_0^\alpha x\,\mathrm{d}x = \alpha^2/2\alpha = \alpha/2$, where we take $\alpha = 2/\mu$ for notational convenience. Likewise $\mathsf{E}\{S^2\} = \alpha^{-1}\int_0^\alpha x^2\,\mathrm{d}x = \alpha^2/3$. Thus, $\mathsf{E}\{S_r \,|\, S_r > 0\} = \alpha/3 = 2/3\mu$.

Solution 2.16.5:

$$\begin{aligned}
\mathsf{E}\{S\} &= \mu\int_0^\infty x e^{-\mu x}\,\mathrm{d}x = 1/\mu, \\
\mathsf{E}\{S^2\} &= \mu\int_0^\infty x^2 e^{-\mu x}\,\mathrm{d}x = x^2 e^{-\mu x}\big|_0^\infty + 2\int_0^\infty x e^{-\mu x}\,\mathrm{d}x \\
&= 2x/\mu e^{-\mu x}\big|_0^\infty + 2/\mu\int_0^\infty e^{-\mu x}\,\mathrm{d}x \\
&= \frac{2}{\mu^2}.
\end{aligned}$$

Solution 2.16.6:
```
labda = 5./60 # arrivals per minute
a = 4.
b = 6.
ES = (a+b)/2.  # service time in minutes
Var = (b-a)*(b-a)/12.
SCV = Var/(ES**2)

rho = labda*ES

Wq = (1+SCV)/2.*rho/(1.-rho)*ES
Wq # in minutes
```

```
1.8095238095238095
```

```
Wq/60. # in hour
```

```
0.030158730158730160
```

```
W = Wq + ES
W
```

```
6.809523809523809
```

```
Lq = labda*Wq
Lq
```

```
0.15079365079365079
```

```
L = labda*W
L
```

```
0.5674603174603174
```

Solution 2.16.7: $\mathsf{E}\{W_Q\} = 4.5$ h. $\rho = \lambda\,\mathsf{E}\{S\} = (3/8)\cdot 2 = 3/4$.

One way to increase the capacity/reduce the average service time is to choose $\mathsf{E}\{S\} = 1$ hour and reduce C_s^2 to 1/4. Of course, there are many more ways. Reducing C_S^2 to zero is (nearly) impossible or very costly. Hence, ρ must go down, i.e., capacity up or service time down. Another possibility is to plan the arrival of jobs, i.e., reduce the variability in the arrival process. However, typically this is not possible. For instance, would you accept this as a customer?

Solution 2.16.8: The arrival process is assumed to be Poisson. There is also just one server. Hence, we can use the PK formula to compute the average queue length.

```
labda = 20./60 # per hour
ES = 2. # hour
sigma = 1.
SCV = sigma*sigma/(ES*ES)
rho = labda*ES
rho
```

```
0.6666666666666666
```

```
Wq = (1+SCV)/2 * rho/(1-rho) * ES
Wq
```

```
2.4999999999999996
```

```
Lq = labda * Wq # Little's law
Lq
```

```
0.8333333333333331
```

```
W = Ws + ES
```

```
---------------------------------------------------------------------------
    NameError
Traceback (most recent call last)<ipython-input-1-4b0da811f906> in
<module>()
----> 1 W = Ws + ES
NameError: name 'Ws' is not defined
```

```
W
```

```
6.809523809523809
```

```
L = labda * W
L
```

```
2.2698412698412698
```

```
ES = 2.1
SD = 0.1
SCV = SD**2/ES**2
rho = labda*ES
rho
```

```
0.7
```

```
Lq = rho**2/(1.-rho)*(1.+SCV)/2.
Lq
```

```
0.8185185185185182
```

```
L = rho + Lq
L
```

```
1.5185185185185182
```

Solution 2.16.9: Since the SCV for the exponential distribution is 1, hence $C_s^2 = 1$, we get

$$\mathsf{E}\left\{W_Q\right\} = \frac{1+1}{2}\frac{\rho}{1-\rho}\mathsf{E}\{S\} = \frac{\rho}{1-\rho}\mathsf{E}\{S\},$$

which we also found in the previous section.

Solution 2.16.10: For the $M/D/1$ the service time is deterministic.

Thus, the service time $S = T$ always; recall that S is deterministic under the assumptions in the exercise. Therefore, we must have that $\mathsf{P}\{S \leq T\} = 1$. As an immediate consequence, $\mathsf{P}\{S > T\} = 1 - \mathsf{P}\{S \leq T\} = 0$. Also, $\mathsf{P}\{S \leq x\} = 0$ for all $x < T$. Thus, all probability mass is concentrated on the point T, thus, it is also impossible that the service time is smaller than T. (If you are into maths, then you should be aware of the fact that 'impossible' and 'almost surely' are not the same. Thus, my using of the word 'impossible' is not completely correct.)

To compute the SCV, I first compute $\mathsf{E}\{S\}$, then $\mathsf{E}\left\{S^2\right\}$ and then $\mathsf{V}\{S\} = \mathsf{E}\left\{S^2\right\} - (\mathsf{E}\{S\})^2$, and use the definition of coefficient of variation. I use these steps time and again.

First, $\mathsf{E}\{S\} = T$ since $\mathsf{P}\{S = T\} = 1$. We can also obtain this in a more formal way. The density $F(x)$ of S has a, so-called, atom at $x = T$, such $F(T) - F(T-) = 1$. Using this,

$$\mathsf{E}\{S\} = \int_0^\infty x\,\mathrm{d}F(x) = T\cdot(F(T) - F(T-)) = T.$$

Similarly

$$\mathsf{E}\left\{S^2\right\} = \int_0^\infty x^2\,\mathrm{d}F(x) = T^2\cdot(F(T) - F(T-)) = T^2.$$

Hence $\mathsf{V}\{S\} = \mathsf{E}\left\{S^2\right\} - (\mathsf{E}\{S\})^2 = 0$, hence $C_s^2 = 0$.

From the Pollackzek-Khintchine formula, the first term is $1+C_s^2 = 1$ for the $M/D/1$ queue, and $1+C_s^2 = 2$ for the $M/M/1$ queue. Hence,

$$\mathsf{E}\{W_Q(M/D/1)\} = \frac{\mathsf{E}\{W_Q(M/M/1)\}}{2}$$

thus,

$$\mathsf{E}\{L_Q(M/D/1)\} = \frac{\mathsf{E}\{L_Q(M/M/1)\}}{2}.$$

In both cases the expected service times are equal to T so that the loads are the same. Thus,

$$\begin{aligned}\mathsf{E}\{L(M/D/1\} = \mathsf{E}\{L_Q(M/D/1)\} + \rho &= \frac{\mathsf{E}\{L_Q(M/M/1)\}}{2} + \rho \\ &= \frac{\rho^2}{2(1-\rho)} + \rho = \frac{\rho(2-\rho)}{2(1-\rho)}.\end{aligned}$$

Solution 2.16.11:

$$\begin{aligned}\mathsf{E}\{S\} &= A/2, \\ \mathsf{E}\{S^2\} &= \int_0^A x^2 \, dx/A = A^2/3 \\ \mathsf{V}\{S\} &= A^2/3 - A^2/4 = A^2/12 \\ C_s^2 &= (A^2/12)/(A^2/4) = 1/3, \\ \rho &= \lambda A/2, \\ \mathsf{E}\{W_Q\} &= \frac{1+C_s^2}{2}\frac{\lambda A/2}{1-\lambda A/2}\frac{A}{2}, \\ \mathsf{E}\{W\} &= \mathsf{E}\{W_Q\} + \frac{A}{2} \\ \mathsf{E}\{L\} &= \lambda\,\mathsf{E}\{W\}.\end{aligned}$$

Solution 2.16.12: Since the interarrival times are memoryless, the expected time to the first arrival after a busy time stops is also $\mathsf{E}\{X\} = 1/\lambda$.

Solution 2.16.13: Since $\rho = \lambda\,\mathsf{E}\{S\}$, $\mathsf{E}\{I\} = 1/\lambda$ and from Exercise 2.7.6

$$\mathsf{E}\{B\} = \frac{\rho}{1-\rho}\mathsf{E}\{I\} = \frac{\lambda\,\mathsf{E}\{S\}}{1-\rho}\frac{1}{\lambda}.$$

The result follows.

Solution 2.16.14: The system can contain at most 1 job. Necessarily, if the system contains a job, this job must be in service. All jobs that arrive while the server is busy are blocked, in other words, rejected. Just after a departure, the average time until the next arrival is $1/\lambda$, then a service starts with an average duration of $\mathsf{E}\{S\}$. After this departure, a new cycle start. Thus, the utilization is $\mathsf{E}\{S\}/(1/\lambda + \mathsf{E}\{S\}) = \lambda\,\mathsf{E}\{S\}/(1+\lambda\,\mathsf{E}\{S\})$. Since not all jobs are accepted, the utilization ρ cannot be equal to $\lambda\,\mathsf{E}\{S\}$.

Solution 2.16.15: Let $p(0)$ be the fraction of time the server is idle. By PASTA, $\pi(0) = p(0)$. Thus, the rate of accepted jobs is $\lambda\pi(0)$. Therefore, the departure rate $\gamma = \lambda\pi(0)$. The loss rate is $\lambda - \gamma = \lambda(1-\pi(0))$.

Since $\lambda\pi(0)$ is the rate at which jobs enter the system, the load must be $\lambda\pi(0)\,\mathsf{E}\{S\}$. Since the load is also $1-\pi(0)$, it follows from equating that

$$\lambda\pi(0)\,\mathsf{E}\{S\} = 1-\pi(0) \iff \pi(0) = \frac{1}{1+\lambda\,\mathsf{E}\{S\}} \iff 1-\pi(0) = \frac{\lambda\,\mathsf{E}\{S\}}{1+\lambda\,\mathsf{E}\{S\}},$$

which is the same an in the previous problem.

Note that $\gamma < \lambda$, as it should be the case.

Solution 2.16.16: Typically, in the $G/G/1$ queue, the arrivals do not see time-averages. Consequently, the fraction of arrivals that are blocked is not necessarily equal to the utilization ρ.

Again, take jobs with duration 59 minutes and interarrival times of 1 hour. The load is 59/60, but no job is lost, also not in the $G/G/1/1$ case. Thus, $\gamma = \lambda$ in this case.

2.17 $M^X/M/1$ Queue Length Distribution

Theory and Exercises

In this section we extend the analysis of the $M^X/M/1$ batch queue that we started in Section 2.15. We present a numerical, recursive, scheme to compute the stationary distribution of the number of items in queue.

To find the distribution $p(n)$, which is equal to $\pi(n)$ by PASTA, we turn to level-crossing arguments. However, the arguments that lead to the level-crossing equation (2.38) need to generalized. To see this, we consider an example. If $L(t) = 3$, the system contains 3 items (but not necessarily 3 jobs.) Since the server serves single items, down-crossings of level $n = 3$ occur in single units. However, due to the batch arrivals, when a job arrives it typically brings multiple items to the queue. For instance, suppose that $L(A_k-) = 3$, i.e., job k sees 3 items in the system at its arrival epoch. If $B_k = 20$, then $L(A_k) = 3+20 = 23$, that is, right after the arrival of job k the system contains 23 items. In this case, at the arrival of job k, all levels between states 3 and 23 are crossed. The left panel in Figure 2.22 demonstrates this in more general terms. Level n can be up-crossed from below from many states, in fact from any level $m < n$. However, it can only be down-crossed from state $n+1$.

As always with level-crossing arguments, we turn to counting how often level n is upcrossed and downcrossed as a function of time. The downcrossing rate is easy; it is exactly the same as for the $M/M/1$ queue. Thus, from Eq. (2.39b), the downcrossing rate is

$$\frac{D(n,t)}{t} = \frac{D(n,t)}{Y(n+1,t)}\frac{Y(n+1,t)}{t} \to \mu(n+1)p(n+1) = \mu\pi(n+1), \tag{2.73}$$

where we use that $\mu(n+1) = \mu$ by the memoryless property of the service times and PASTA to see that $\pi(n+1) = p(n+1)$. Observe that this part was easy because there is just one way (i.e., there is just one arrow from right to left in Figure 2.22) to downcross level n, namely from $n+1$ to n. Counting upcrossings requires quite some more work.

Let us write $\mathbb{1}_{L(A_k-)\leq n}\mathbb{1}_{L(A_k)>n}$ to track whether the kth job sees n or less items in the system upon arrival, i.e. condition $\mathbb{1}_{L(A_k-)\leq n}$, and is such that after its arrival the system contains

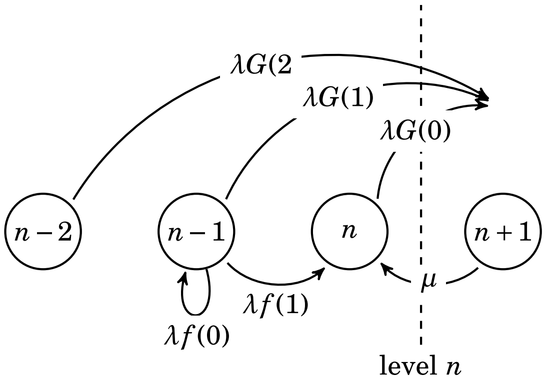


Figure 2.22: Level crossing of level n. Observe that when the system is in state $n-2$, the arrival of any batch larger than 2 ensures that level n is crossed from below. The rate at which such events happen is $\lambda G(2)$. Similarly, in state $n-1$ the arrival of any batch larger than one item ensures that level n is crossed, and this occurs with rate $\lambda G(1)$, and so on.

more than n items, i.e., condition $\mathbb{1}_{L(A_k)>n}$. Thus, if $\mathbb{1}_{L(A_k-)\leq n}\,\mathbb{1}_{L(A_k)>n} = 1$, job k crossed level n from below. Hence

$$\sum_{k=1}^{\infty} \mathbb{1}_{A_k\leq t}\,\mathbb{1}_{L(A_k-)\leq n}\,\mathbb{1}_{L(A_k)>n},$$

counts all arrivals up to time t that enforce a crossing of level n from below.

From Figure 2.22 we see that an upcrossing can be decomposed into:

$$\begin{aligned}
\mathbb{1}_{L(A_k-)\leq n}\,\mathbb{1}_{L(A_k)>n} &= \mathbb{1}_{L(A_k-)=n}\,\mathbb{1}_{B_k>0} + \mathbb{1}_{L(A_k-)=n-1}\,\mathbb{1}_{B_k>1} + \cdots + \mathbb{1}_{L(A_k-)=0}\,\mathbb{1}_{B_k>n} \\
&= \sum_{m=0}^{n} \mathbb{1}_{L(A_k-)=m}\,\mathbb{1}_{B_k>n-m}.
\end{aligned}$$

In other words, paths that upcross level n require that any job that sees m ($m \leq n$) in the system upon arrival must bring a batch larger than $n-m$ items.

Define

$$A(m,n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{A_k\leq t}\,\mathbb{1}_{L(A_k-)=m}\,\mathbb{1}_{B_k>n-m}.$$

as the number of jobs that, up to time t, see m in the system upon arrival and have batch size larger than $n-m$. Then the number of times level n is upcrossed up to time t can be written as $\sum_{m=0}^{n} A(m,n,t)$, hence the upcrossing rate up to time t is

$$\frac{1}{t}\sum_{m=0}^{n} A(m,n,t).$$

Simplifying this comes down to finding the right quantities by which to divide. Some experimentation will reveal that the following gives the most relevant insights:

$$\frac{A(m,n,t)}{t} = \frac{A(t)}{t}\frac{A(m,t)}{A(t)}\frac{A(m,n,t)}{A(m,t)}. \tag{2.74}$$

Here $A(t)/t$ and $A(m,t)/A(t)$ have the same meaning as in Section 2.12, hence the first of these fractions converges to λ and the second to $\pi(m)$ as $t \to \infty$. Now, provided the limit exists, we define

$$\lim_{t\to\infty}\frac{A(m,n,t)}{A(m,t)} = \lim_{t\to\infty}\frac{\sum_{k=1}^{\infty}\mathbb{1}_{A_k\leq t}\mathbb{1}_{L(A_k-)=m,B_k>n-m}}{\sum_{k=1}^{\infty}\mathbb{1}_{A_k\leq t}\mathbb{1}_{L(A_k-)=m}} = \mathsf{P}\{B>n-m\,|\,L(A-)=m\}, \tag{2.75}$$

where $\mathsf{P}\{L(A-)=m\}$ has the interpretation of the steady-state probability that a job sees m jobs in the system upon arrival. By *assumption*, B and $L(A-)$ are independent[8], and therefore,

$$\lim_{t\to\infty}\frac{A(m,n,t)}{A(n,t)} = \mathsf{P}\{B>n-m\} = G(n-m).$$

By combining the above and making the usual assumptions about the existence of all limits involved we find

$$\lim_{t\to\infty}\frac{A(m,n,t)}{t} = \lambda\pi(m)G(n-m).$$

Thus, Eq. (2.74) has the interpretation that the rate at which level n is crossed from below from state m is equal the rate at which jobs arrive times the fraction of jobs that see m jobs in the system times the fraction of jobs with batch size larger than $n-m$. Observe that the stream of jobs with batch size larger than i, say, is a Poisson process thinned at rate $G(i)$. Hence, this stream of jobs is a Poisson process in its own right with rate $\lambda G(i)$.

The last step is to relate the up- and downcrossing rates. Recall that for the $M/M/1$ the number of upcrossings and downcrossings of level n could not differ more than one, i.e., $|A(n,t)-D(n,t)|\leq 1$. Similarly, for the $M^X/M/1$ queue we have

$$\left|\sum_{m=0}^{n}A(m,n,t)-D(n,t)\right|\leq 1. \tag{2.76}$$

Thus, taking the limit $t\to\infty$ in this equation, we conclude that the upcrossing rate and the downcrossing rate of level n must be the same. Hence,

$$\lambda\sum_{m=0}^{n}G(n-m)\pi(m) = \mu\pi(n+1). \tag{2.77}$$

For the excess probabilities $\mathsf{P}\{L>n\}$ we need to compute the actual probabilities $\pi(n)$ from the recursion (2.77). To apply this, temporarily set $\pi(0)=1$. Then, for $n=1$, we have $\mu\pi(1) = \lambda\pi(0)=\lambda$, . Then, since $G(0)=1$, and writing $\alpha=\lambda/\mu$, the *unnormalized* state probabilities are

$$\begin{aligned}
&\pi(1)=\alpha\\
&\pi(2)=\alpha G(0)\pi(1)+\alpha G(1)\pi(0)=\alpha(\alpha+G(1))=\alpha^2+\alpha G(1),\\
&\pi(3)=\alpha[G(0)\pi(2)+G(1)\pi(1)+G(2)\pi(0)]=\alpha[\alpha^2+2\alpha G(1))+G(2)],
\end{aligned}$$

and so on. (Evidently, the computer is better suited to continue with this.)

It is left to find the normalization constant. As this recursion does not lead to a closed form expression for $p(n)$, such as Eq. (2.45), we need to use a criterion to stop this iterative procedure. Finding general conditions on when to stop is not directly easy, but a pragmatic approach is to simply stop at some (large) number N and take $G=\sum_{i=0}^{N}\pi(i)$ as the normalization constant. Then $\pi(0)=1/G$, $\pi(1)=\alpha/G$, and so on.

[8] Thus,

$$\mathsf{P}\{B>n-m\,|\,L(A-)=m\} = \frac{\mathsf{P}\{B>n-m, L(A-)=m\}}{\mathsf{P}\{L(A-)=m\}} = \frac{\mathsf{P}\{B>n-m\}\,\mathsf{P}\{L(A-)=m\}}{\mathsf{P}\{L(A-)=m\}} = \mathsf{P}\{B>n-m\}.$$

Remark 2.17.1. An interesting extension is to consider a queueing process with batch services, i.e., the $M/M^Y/1$ queue. Construction a recursion for the steady-state probabilities $p(n)$ for this case is not hard, in fact, mostly analogous to Eq. (2.77). However, solving the recursion is much less simple, hence we will not discuss this further.

Exercise 2.17.1. Show that Eq. (2.77) reduces to $\mu p(n+1) = \lambda p(n)$ for the $M/M/1$ case.

Exercise 2.17.2. Show that $A(n,n,t) = A(n,t)$.

Exercise 2.17.3. Why is Eq. (2.60), i.e., $\pi(n) = \delta(n)$, not true for the $M^X/M/1$ batch queue?

Exercise 2.17.4. Let us use the above recursion Eq. (2.77) for $\pi(n)$ to derive an expression for the expected number of units of work $\mathsf{E}\{L\}$ in the system. Substitute the recursion Eq. (2.77) in the definition of $\mathsf{E}\{L\}$ to show that

$$\mu \mathsf{E}\{L\} = \mu \sum_{n=0}^{\infty} n\pi(n) = \lambda \frac{\mathsf{E}\{B^2\}}{2} + \lambda \mathsf{E}\{B\} \mathsf{E}\{L\} + \lambda \frac{\mathsf{E}\{B\}}{2}, \tag{2.78}$$

Exercise 2.17.5. Use Eq. (2.78) and the definition for $\rho = \lambda \mathsf{E}\{B\}/\mu$ to show that

$$(1-\rho)\mathsf{E}\{L\} = \frac{\lambda}{\mu} \frac{\mathsf{E}\{B^2\}}{2} + \frac{\rho}{2}.$$

This is the same result as what we obtained in Section 2.15.

Exercise 2.17.6. (Finite queueing systems) We consider the $M^X/M/1/K$ queue, i.e., a batch queue in which at most K jobs fit into the system. When customers can be blocked, it is necessary to specify an acceptance, or equivalently a rejection, policy. Three common rules are

1. Complete rejection: if a batch does not fit entirely into the system, it will be rejected completely.
2. Partial acceptance: accept whatever fits of a batch, and reject the rest.
3. Complete acceptance: accept all batches that arrive when the system contains K or less jobs, and reject otherwise.

Derive a set of recursions, analogous to Eq. (2.77), to compute $\pi(n)$ for these three different acceptance rules.

Exercise 2.17.7. Implement the recursion (2.77) in a computer program for the case $f(1) = f(2) = f(3) = 1/3$ (recall $\mathsf{P}\{B = k\} = f_k$). Take $\lambda = 1$ and $\mu = 3$.

Hints

Hint 2.17.4: Substitute the recursion, and carry on with the algebra. We urge the reader to try it, its good (necessary) practice. Use also the results of the exercises of Section 2.15.

Hint 2.17.5: Divide both sides by μ.

Hint 2.17.6: This exercise tests your creativity and modeling skills, it is not analytically difficult. The best approach to problems like this is to try some simple cases first. For instance, consider the case with batch sizes of 1 first, then batches of sizes 1 or 2, and so on. If system contains K jobs, which batches can be accepted? If the system contains $K-3$, say, which batch sizes can be accepted, which will be refused, under which acceptance policy?

Hint 2.17.7: This is an important problem, it helps to check (numerically) the algebraic results we have been deriving up to now. Implementing the recursion is not hard, just try and see how far you get.

Solutions

Solution 2.17.1: The left hand side of Eq. (2.77) is identical, so we only have to concentrate on the right hand side. In the $M/M/1$ queue, all batches have size 1. Thus, $\mathsf{P}\{B=1\}=f(1)=1$ and $f(k)=0$ for $k\neq 1$. Thus, $G(0)=1$ and $G(1)=G(2)=\cdots=0$. Thus, $\sum_{m=0}^{n} G(n-m)p(m)=G(0)p(n)=p(n)$.

Solution 2.17.2: $A(n,n,t)$ counts all jobs up to time t that see n items and bring at least $n-n+1=1$ unit of work. As each job brings at least 1 item, $A(n,n,t)$ counts all jobs that see n items at arrival.

Solution 2.17.3: Because arrivals do not occur in single units, but in batches.

Solution 2.17.4: We use that $\mu\pi(n)=\lambda\sum_{i=0}^{n-1}\pi(i)G(n-1-i)$ and the results of the exercises of Section 2.15 to see that

$$
\begin{aligned}
\mu\,\mathsf{E}\{L\} &= \sum_{n=0}^{\infty} n\,\mu\pi(n), \quad \text{now substitute for } \mu\pi(n) \text{ the above recursion,}\\
&= \lambda\sum_{n=0}^{\infty} n\sum_{i=0}^{n-1}\pi(i)G(n-1-i) = \lambda\sum_{n=0}^{\infty} n\sum_{i=0}^{\infty} 1\{i<n\}\pi(i)G(n-1-i)\\
&= \lambda\sum_{i=0}^{\infty}\pi(i)\sum_{n=0}^{\infty} 1\{i<n\}nG(n-1-i) = \lambda\sum_{i=0}^{\infty}\pi(i)\sum_{n=i+1}^{\infty} nG(n-1-i)\\
&= \lambda\sum_{i=0}^{\infty}\pi(i)\sum_{n=0}^{\infty}(n+i+1)G(n) = \lambda\sum_{i=0}^{\infty}\pi(i)\left[\sum_{n=0}^{\infty} nG(n)+(i+1)\sum_{n=0}^{\infty}G(n)\right]\\
&= \lambda\sum_{i=0}^{\infty}\pi(i)\sum_{n=0}^{\infty} nG(n)+\lambda\,\mathsf{E}\{B\}\sum_{i=0}^{\infty}\pi(i)(i+1)\\
&= \lambda\sum_{i=0}^{\infty}\pi(i)\frac{\mathsf{E}\{B\}^2-\mathsf{E}\{B\}}{2}+\lambda\,\mathsf{E}\{B\}(\mathsf{E}\{L\}+1)\\
&= \lambda\frac{\mathsf{E}\{B\}^2-\mathsf{E}\{B\}}{2}+\lambda\,\mathsf{E}\{B\}\,\mathsf{E}\{L\}+\lambda\,\mathsf{E}\{B\}\\
&= \lambda\frac{\mathsf{E}\{B\}^2}{2}+\lambda\,\mathsf{E}\{B\}\,\mathsf{E}\{L\}+\lambda\frac{\mathsf{E}\{B\}}{2},
\end{aligned}
$$

Solution 2.17.5: Dividing both sides by μ and using that $\lambda\,\mathsf{E}\{B\}/\mu=\rho$,

$$
\mathsf{E}\{L\}=\frac{\lambda}{\mu}\frac{\mathsf{E}\{B\}^2}{2}+\rho\,\mathsf{E}\{L\}+\frac{\rho}{2}.
$$

Solution 2.17.6: Let's first deal with complete rejection. Suppose a batch of size k arrives when the system contains n jobs. When $k+n\leq K$, the batch can be accepted since the entire batch will fit into the queue. When, however, $k+n>K$, the batch has to be rejected.

Now consider an imaginary line between states with n and $n+1$ jobs in the system. This imaginary line separates the state space into two disjoint part: one part with states $0,1,\ldots,n$

and the other part with states $n+1, n+2, \ldots, K$. We call this imaginary line 'level n'. We call an 'upcrossing' a transition from some state $m \leq n$ to a state $l \geq n+1$. We also say in that case that level n is 'crossed from below', i.e., from some state $m \leq n$ to a state $l \geq n+1$. Likewise, a 'downcrossing' of level n happens when this level is crossed from 'above', i.e. from some state $l \geq n+1$ to some state $m \leq n$.

If the system contains n jobs, level n is crossed from below with rate $\lambda\pi(n)\mathsf{P}\{B \leq K-n\}$. More generally, when the system is in state $m \leq n$, we need a batch of at least $n+1-m$ to cross the n. Moreover, any batch larger than $K+1-m$ gets rejected. Thus, when the system contains $m \leq n$ jobs, the rate at which level n is crossed from below is

$$\lambda\pi(m)\mathsf{P}\{n+1-m \leq B \leq K+1-m\} = \lambda\pi(m)[G(n-m) - G(K-m)].$$

Since the server serves only single items, level n can only be crossed from above from state $n+1$. This happens at rate $\mu p(n+1)$.

Finally, since on the long run the number of up- and crossings must be the same, the up- and downcrossing rates must match. This implies that the balance equations becomes

$$\mu\pi(n+1) = \lambda \sum_{m=0}^{n} \pi(m)[G(n-m) - G(K-m)],$$

for $n = 0, \ldots, K-1$.

Before we continue with the other acceptance rules, it is important to check this result. In general, off-by-one errors are easily, and commonly, made, so we need to test the above on simple cases.

- If $K \to \infty$, then $G(K-m) = \mathsf{P}\{B > K-m\} \to 0$, so we get our earlier result.
- Take $n = K$. Then $G(n-m) - G(K-m) = 0$ for all m. Then the right hand side is 0, as it should.
- Taken $n = 0$ and $K = 1$, then $\mu\pi(1) = \lambda\pi(0)$. This also makes sense.
- Take n much smaller than K. If the batch size is maximally 2, then for small n the entire batch must fit. Let's see if this holds in the above formula. If n much smaller than K, then also m is much smaller than K (since in the right hand size, $m \leq n$. But then $G(K-m) \leq G(2) = 0$, as it should. (Observe that G is a decreasing function of it argument; its a survival function.)

The complete-acceptance policy is actually quite simple. As any batch will be accepted when $n \leq K$, the queue length is not bounded. Only when the number of jobs in the system is larger than K, we do not accept jobs.

$$\mu\pi(n+1) = \begin{cases} \lambda\sum_{m=0}^{n}\pi(m)G(n-m), & \text{for } n \leq K, \\ \lambda\sum_{m=0}^{K}\pi(m)G(n-m), & \text{for } n > K. \end{cases}$$

For the partial acceptance case, any job is accepted, but the system only admits whatever fits. Thus, the rate at which level $n < K$ is crossed is not affected by this acceptance policy, and therefore,

$$\mu\pi(n+1) = \lambda \sum_{m=0}^{n} \pi(m)G(n-m),$$

for $n = 0, 1, \ldots, K-1$.

Solution 2.17.7: The recursion for n becomes

$$\pi(n) = \frac{\lambda}{\mu} \sum_{i=0}^{n-1} \pi(n-1-i) G(i).$$

Since $G(i) = 0$ for $i \geq 3$ we rewrite this to

$$\pi(n) = \frac{\lambda}{\mu} \sum_{i=0}^{\min(\{n-1,l\}} \pi(n-1-i) G(i),$$

where $l = 3$.

In python this reads:

```
p[n] = labda / mu * sum(p[n - 1 - i] * G[i] for i in range(min(n, l)))
```

The following code carries out the recursion for the $M/M/1$ queue and compares the result to the *unnormalized* probabilities $(\lambda/\mu)^n$ (unnormalized means that I do not multiply with $1-\rho$). Thus, we can test the code right away.

```
import numpy as np

l = 1 # M/M/1 has batch size 1
f = np.ones(l + 1) / l # f[0] =1, f[1] = 1
f[0] = 0 # set f[0] = 0
F = np.cumsum(f) # distribution
G = np.ones_like(F) - F # survivor function

labda = 1
mu = 3
num = 30
p = np.ones(num)

for n in range(1, num):
    p[n] = labda / mu * sum(p[n - 1 - i] * G[i] for i in range(min(n, l)))
    print(n, p[n], (labda / mu)**n)
```

```
1 0.333333333333 0.3333333333333333
2 0.111111111111 0.1111111111111111
3 0.037037037037 0.037037037037037035
4 0.0123456790123 0.012345679012345677
5 0.00411522633745 0.004115226337448558
6 0.00137174211248 0.0013717421124828527
7 0.000457247370828 0.00045724737082761756
8 0.000152415790276 0.00015241579027587253
9 5.08052634253e-05 5.080526342529084e-05
10 1.69350878084e-05 1.6935087808430282e-05
11 5.64502926948e-06 5.645029269476759e-06
12 1.88167642316e-06 1.8816764231589195e-06
13 6.27225474386e-07 6.272254743863065e-07
14 2.09075158129e-07 2.090751581287688e-07
15 6.96917193763e-08 6.969171937625627e-08
16 2.32305731254e-08 2.3230573125418753e-08
17 7.74352437514e-09 7.743524375139585e-09
18 2.58117479171e-09 2.5811747917131946e-09
19 8.60391597238e-10 8.603915972377315e-10
20 2.86797199079e-10 2.867971990792438e-10
21 9.55990663597e-11 9.559906635974793e-11
```

```
22 3.18663554532e-11 3.186635545324931e-11
23 1.06221184844e-11 1.0622118484416435e-11
24 3.54070616147e-12 3.540706161472145e-12
25 1.18023538716e-12 1.1802353871573816e-12
26 3.93411795719e-13 3.934117957191272e-13
27 1.3113726524e-13 1.3113726523970905e-13
28 4.37124217466e-14 4.3712421746569684e-14
29 1.45708072489e-14 1.457080724885656e-14
```

The test is convincing.

```
l = 3
f = np.ones(l + 1) / l
f[0] = 0
F = np.cumsum(f)
G = np.ones_like(F) - F
num = 30
p = np.ones(num)
for n in range(1, num):
    p[n] = labda / mu * sum(p[n - 1 - i] * G[i] for i in range(min(n, l)))
    print(n, p[n])
```

```
1 0.333333333333
2 0.333333333333
3 0.296296296296
4 0.20987654321
5 0.172839506173
6 0.137174211248
7 0.107453132144
8 0.0855052583448
9 0.0676218056191
10 0.053481007299
11 0.0423546546089
12 0.0335164204493
13 0.0265266197627
14 0.0209963727551
15 0.0166176420267
16 0.013152476817
17 0.0104098985844
18 0.00823914349042
19 0.006521078273
20 0.00516126893158
21 0.00408501187013
22 0.00323318352739
23 0.00255898258378
24 0.0020253696307
25 0.00160302862078
26 0.00126875641195
27 0.00100418845646
28 0.000794789646005
29 0.000629055806986
```

Stopping at 30 seems ok. The last step is normalize.

```
p /= p.sum()  # normalize
for n in range(1, num):  # print normalized results
    print(n, p[n])
```

```
1 0.111199612316
```

```
2 0.111199612316
3 0.098844099836
4 0.0700145707172
5 0.0576590582377
6 0.0457611573315
7 0.0358462399097
8 0.0285244547366
9 0.0225585557067
10 0.0178412018337
11 0.0141294635168
12 0.0111810388805
13 0.00884924950098
14 0.00700436553119
15 0.00554362605289
16 0.00438765096912
17 0.00347273006048
18 0.00274856868584
19 0.00217542412751
20 0.00172179331274
21 0.00136275520879
22 0.00107858626437
23 0.000853673613716
24 0.000675660953187
25 0.000534768483485
26 0.000423255663394
27 0.000334996101149
28 0.000265140901524
29 0.000209852285585
```

Now that we have the probabilities we can do all experiments we like.

```python
EL = sum(n*p[n] for n in range(len(p)))
EL
```

```
3.309069824793065
```

It is of interest to compare this result to the equation above Eq. (2.68).

```python
EB = sum(k*fk for k,fk in enumerate(f))
EB2 = sum(k*k*fk for k,fk in enumerate(f))
EB2
```

```
4.6666666666666661
```

```python
rho=labda*EB/mu
RHS = labda/mu*EB2/2 + rho/2
EL=RHS/(1-rho)
EL
```

```
3.3333333333333326
```

So, after all, stopping at 30 seems not quite ok: there is a slight difference between the two expectations. Lets run the recursion up to 100, and see what we get then.

```python
num = 100
p = np.ones(num)
for n in range(1, num):
    p[n] = labda / mu * sum(p[n - 1 - i] * G[i] for i in range(min(n, 1)))
p /= p.sum()  # normalize
EL = sum(n*p[n] for n in range(len(p)))
print(EL)
```

```
3.33333332712
```

This does the job.

As a last step, I want to check Eq. (2.68).

```
VB = EB2 - EB*EB
C2 = VB/EB/EB
EL = (1+C2)/2*rho/(1-rho)*EB + rho/(1-rho)/2
EL
```

```
3.3333333333333326
```

The algebra is, most probably, correct. I spent at least one hour to get Eq. (2.68) correct. I am now, also numerically, convinced that all results are ok.

2.18 M/G/1 Queue Length Distribution

Theory and Exercises

In Section 2.16 we derived the Pollackzek-Khintchine Eq. (2.72) by which we can compute the average waiting time in queue for the $M/G/1$ queue. Then, with Little's law, we therefore also have the average queue length. However, if we need the loss probability $\mathsf{P}\{L > n\}$, we need expressions for the stationary distribution of the number of jobs in the system $p(n) = \mathsf{P}\{L = n\}$. Here we present a set of recursions by which we can compute $p(n)$ for a given arrival rate λ and service distribution F of the service times.

To derive the stationary distribution $p(n)$ we work in steps, and use similar level-crossing arguments as in Section 2.17. However, we cannot simply copy the derivation of the $M^X/M/1$ queue to the $M/G/1$ queue. To see this, note that in the $M^X/M/1$ queue, the service times are exponential, hence memoryless. For this reason, the service times 'restart from anew' at any moment in time, in particular at arrival moments. For the $M/G/1$ queue, service times are, in general, not exponential, hence not memoryless, hence do not restart at arrival times. For this reason we need to focus on moments in time in which the system 'restarts'. As we will see, the argumentation is quite subtle, as it uses an interplay of the PASTA property and the relation of Eq. (2.60) between $\pi(n)$, $p(n)$ and $\delta(n)$. The important idea is to focus on the *departure* moments, as these moments can be considered as restarts of the service times, and on the number of arrivals Y_k during the service time of the kth job. Assume that $f(j) = \mathsf{P}\{Y_k = j\}$ is given and write $G(j) = \mathsf{P}\{Y_k > j\}$.

First we concentrate on a downcrossing of level n, see Figure 2.23; recall that level n lies between states n and $n+1$. Suppose now that job '$k-1$' leaves $n+1$ jobs behind after its service completion. Precisely when job k leaves n jobs behind a downcrossing occurs, i.e., when $\mathbb{1}_{L(D_{k-1})=n+1}\mathbb{1}_{L(D_k)=n} = 1$. Let us write this in another way. Observe that if $L(D_{k-1}) = n+1$ and no other jobs arrive during the service time S_k of job k, i.e., when $Y_k = 0$, it must also be that job k leaves n jobs behind. If, however, $Y_k > 0$, then $L(D_k) \geq n+1$. Thus, we see that

$$\mathbb{1}_{L(D_{k-1})=n+1}\mathbb{1}_{L(D_k)=n} = \mathbb{1}_{L(D_{k-1})=n+1}\mathbb{1}_{Y_k=0}.$$

Consequently, the number of downcrossings up to time t is

$$D(n+1,0,t) = \sum_{k=1}^{\infty} \mathbb{1}_{D_k \leq t}\mathbb{1}_{L(D_{k-1})=n+1}\mathbb{1}_{Y_k=0}.$$

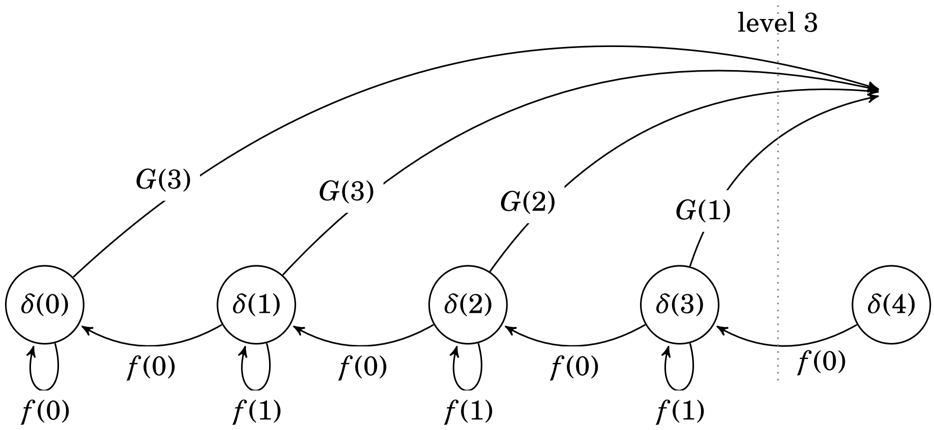


Figure 2.23: Level 3 is crossed from below with rate $\gamma\delta(0)G(3)+\gamma\delta(1)G(3)+\cdots\gamma\delta(3)G(1)$ and crossed from above with rate $\gamma\delta(4)f(0)$.

By using the definitions and limits developed in Sections 2.7 and 2.12,

$$\begin{aligned}\lim_{t\to\infty}\frac{D(n+1,0,t)}{t} &= \lim_{t\to\infty}\frac{D(t)}{t}\frac{D(n+1,t)}{D(t)}\frac{D(n+1,0,t)}{D(n+1,t)} \\ &= \gamma\delta(n+1)\lim_{t\to\infty}\frac{D(n+1,0,t)}{D(n+1,t)} \\ &= \gamma\delta(n+1)\mathsf{P}\{Y=0\} \\ &= \gamma\delta(n+1)f(0),\end{aligned}$$

where the last limit follows from the independence of Y_k and $L(D_{k-1})$, compare the derivation of (2.75).

For the upcrossings, suppose first that $L(D_{k-1})=n>0$ and observe that an upcrossing whenever $\mathbb{1}_{L(D_{k-1})=n}\mathbb{1}_{L(D_k)>n}=1$. Again, we can convert this into a statement with the number of arrivals during the service time S_k of job k. If $Y_k=0$, then job k must leave $n-1$ jobs behind, so no upcrossing can happen. Next, if $Y_k=1$, then job k leaves n jobs behind, so still no upcrossing occurs. In fact, level n is upcrossed when iff more than one job arrives during the service of job k, thus,

$$\mathbb{1}_{L(D_{k-1})=n}\mathbb{1}_{L(D_k)>n}=\mathbb{1}_{L(D_{k-1})=n}\mathbb{1}_{Y_k>1}.$$

More generally, level n is upcrossed for $0<m\leq n$ whenever

$$\mathbb{1}_{L(D_{k-1})=m}\mathbb{1}_{L(D_k)>n}=\mathbb{1}_{L(D_{k-1})=m}\mathbb{1}_{Y_k>n-m+1},$$

while if $m=0$ (think about this),

$$\mathbb{1}_{L(D_{k-1})=0}\mathbb{1}_{L(D_k)>n}=\mathbb{1}_{L(D_{k-1})=0}\mathbb{1}_{Y_k>n}.$$

Again we define proper counting functions (c.f., one of the exercises below), divide, and take suitable limits to find for upcrossing rate

$$\gamma\delta(0)G(n)+\gamma\sum_{m=1}^{n}\delta(m)G(n-m+1). \tag{2.79}$$

Finally, equating the downcrossing and upcrossing rates and dividing by γ gives

$$f(0)\delta(n+1) = \delta(0)G(n) + \sum_{m=1}^{n} \delta(m)G(n+1-m). \tag{2.80}$$

Noting that $\pi(n) = \delta(n)$, which follows from (2.60) and the fact that the $M/G/1$ queue length process has one-step transitions, we arrive at

$$f(0)\pi(n+1) = \pi(0)G(n) + \sum_{m=1}^{n} \pi(m)G(n+1-m).$$

Again, we obtain a recursion by which we can compute, iteratively, the state probabilities. Setting $\pi(0) = 1$ at first, $\pi(1)$ follows from the above recursion. Then, with $\pi(0) = 1$ and $\pi(1)$ we find $\pi(2)$, and so on, up to some limit N. Then, set $G = \sum_{n=0}^{N} \pi(n)$ as the normalization constant. As a practical approach to determine whether N is sufficiently large, compute $p(n)$ also for a couple for values above N. If these values decrease when n becomes larger and they are very small compared to the $p(n)$ with $n < N$, N is most surely sufficiently large. However, getting formal bounds on a proper size of N requires more work than we can do here.

Exercise 2.18.1. Provide the details behind the derivation of Eq. 2.79.

Exercise 2.18.2. Suppose that for the $M/G/1$ queue, $L(D_{k-1}) > 0$. Why is $S_k = D_k - D_{k-1}$? If $L(D_{k-1}) = 0$, the time between D_{k-1} and D_k is *not* equal to S_k. What is then the distribution of $D_k - D_{k-1}$?

Exercise 2.18.3. Explain that if the service time is s, then

$$\mathsf{P}\{Y_k = j \,|\, S = s\} = e^{-\lambda s}\frac{(\lambda s)^j}{j!}. \tag{2.81}$$

Exercise 2.18.4. Explain that

$$\mathsf{P}\{Y_k = j\} = \int_0^\infty e^{-\lambda x}\frac{(\lambda x)^j}{j!}\,\mathrm{d}F(x). \tag{2.82}$$

where F is the distribution of the service times.

Exercise 2.18.5. If S is deterministic and equal to s, show that Eq. (2.82) reduces to Eq. (2.81).

Exercise 2.18.6. If $S \sim \exp(\mu)$, show that

$$f(j) = \mathsf{P}\{Y_k = j\} = \frac{\mu}{\lambda+\mu}\left(\frac{\lambda}{\lambda+\mu}\right)^j. \tag{2.83}$$

Exercise 2.18.7. If $S \sim \exp(\mu)$, show that

$$G(j) = \sum_{k=j+1}^{\infty} f(k) = \left(\frac{\lambda}{\lambda+\mu}\right)^{j+1}. \tag{2.84}$$

Exercise 2.18.8. Clearly, the $M/M/1$ queue is a special case of the $M/G/1$ queue. For this reason, Eq. (2.83) must also apply to the $M/M/1$ queue. Thus, to check our work, show that this recursion, when applied to the $M/M/1$ queue, gives the queue length distribution for the $M/M/1$ queue, i.e., $p(n) = (1-\rho)\rho^n$.

Exercise 2.18.9. With the above results we can compute Eq. (2.82) for deterministic service times or exponential service times. When, however, the service distribution F is some general distribution function, we need numerical methods to evaluate Eq. (2.82). Can you design a suitable numerical method for this problem?

Hints

Hint 2.18.1: Define for $m = 1, \dots, n$

$$D(m,n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{D_k \leq t} \mathbb{1}_{L(D_{k-1})=m} \mathbb{1}_{Y_k > n-m+1},$$

and

$$D(0,n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{D_k \leq t} \mathbb{1}_{L(D_{k-1})=0} \mathbb{1}_{Y_k > n}.$$

Then, divide by $D(n,t)$ and $D(t)$ and take limits.

Hint 2.18.2: Realize that if $L(D_{k-1}) = 0$, job $k-1$ leaves behind an empty system. Thus, before job k can leave, it has to arrive. In other words, $D_{k-1} < A_k$. Since job k arrives to an empty system, his service starts right away, to that the time between A_k and D_k is equal to the service time of job k.

Hint 2.18.3: If s is deterministic, the number of arrival during a fixed period of time with length s must be Poisson distributed.

Hint 2.18.5: If $S = s$ then $\mathsf{P}\{S = s\} = 1$, i.e., all probability mass lies at s. Thus, all arrivals must occur during $[0,s]$.

Hint 2.18.8: Define shorthands such as $\alpha = \lambda/(\lambda+\mu)$, so that $1-\alpha = \mu/(\lambda+\mu)$, and $\alpha/(1-\alpha) = \lambda/\mu = \rho$. Then, with the previous exercise that $f(n) = \alpha^n(1-\alpha)$ and $G(n) = \alpha^{n+1}$. (I did not get this result for 'free'. Indeed, it's just algebra, but please try to derive this yourself. Its good to hone your computational skills.)

Solutions

Solution 2.18.1: Define, for the first term,

$$D(0,n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{D_k \leq t} \mathbb{1}_{L(D_{k-1})=0} \mathbb{1}_{Y_k > n}$$

Then

$$\begin{aligned}
\lim_{t\to\infty} \frac{D(0,n,t)}{t} &= \lim_{t\to\infty} \frac{D(t)}{t} \frac{D(0,t)}{D(t)} \frac{D(0,n,t)}{D(0,t)} \\
&= \gamma\delta(0) \lim_{t\to\infty} \frac{D(0,n,t)}{D(0,t)} \\
&= \gamma\delta(0)\mathsf{P}\{Y > n\} \\
&= \gamma\delta(0)G(n).
\end{aligned}$$

For the second term, define

$$D(m,n,t) = \sum_{k=1}^{\infty} \mathbb{1}_{D_k \leq t} \mathbb{1}_{L(D_{k-1})=m} \mathbb{1}_{Y_k > n-m+1}$$

then

$$\begin{aligned}
\lim_{t\to\infty}\frac{D(m,n,t)}{t} &= \lim_{t\to\infty}\frac{D(t)}{t}\frac{D(m,t)}{D(t)}\frac{D(m,n,t)}{D(m,t)} \\
&= \gamma\delta(m)\lim_{t\to\infty}\frac{D(m,n,t)}{D(m,t)} \\
&= \gamma\delta(m)\mathsf{P}\{Y > n-m+1\} \\
&= \gamma\delta(m)G(n-m+1).
\end{aligned}$$

Solution 2.18.2: When $L(D_{k-1}) > 0$, job k is already in the system when job $k-1$ finishes its service and leaves. Thus, at the departure time D_{k-1} of job $k-1$, the service of job k can start right away at D_{k-1}. Then, $D_k = D_{k-1} + S_k$.

When job $k-1$ leaves an empty system behind, $D_k = A_k + S_k$, since job k sees an empty system, hence its service can start right away after its arrival time at time A_k. Since the arrival process is Poisson by assumption, the time to the next arrival after D_{k-1} is exponentially distributed with rate λ. (Recall the memoryless property of the interarrival times.) Thus, $A_k - D_{k-1}$ has the same distribution as X_k, so that $\mathsf{P}\{D_k - D_{k-1} \leq x\} = \mathsf{P}\{X_k + S_k \leq x\}$. BTW: Can you simplify this expression by using conditioning on X_k?

Solution 2.18.3: If the service time is s units long, we need to know the number of arrivals during this interval. By assumption, the arrival process is Poisson.

Solution 2.18.4: We use a conditioning argument to arrive at this result. The probability that the service time is x units long is written in various ways in the literature: $F(\mathrm{d}x) = \mathrm{d}F(x) = \mathsf{P}\{S \in \mathrm{d}x\}$, but this all means the same thing, it is only the notation that differs. (As an aside, to properly define this we need measure theory...). When F has a density f, then $\mathrm{d}F(x) = f(x)\,\mathrm{d}x$. Note that when S is discrete, it does not have a density everywhere. With this,

$$\mathsf{P}\{Y_k = j\} = \int_0^\infty \mathsf{P}\{Y_k = j \mid S = x\}\,\mathsf{P}\{S \in \mathrm{d}x\} = \int_0^\infty \mathsf{P}\{Y_k = j \mid S = x\}\,\mathrm{d}F(x).$$

Using the answer of the previous problem we arrive at the result.

Solution 2.18.5: Let us first attack this problem from a general point of view. Suppose the service time S can take values $s_1 < s_2 < \cdots < s_n$, and $\mathsf{P}\{S = s_i\} = \alpha_i$. Then of course we want that $\mathsf{P}\{S \leq s_n\} = \sum_{i=1}^n \alpha_i = 1$. The distribution function F of S is in this case a step function, with steps at the points $s_1, s_2, \ldots$, and step sizes $\alpha_1, \alpha_2, \ldots$. Thus, $F(s_i) - F(s_i-) = \alpha_i$. We say that such a distribution function has *atoms* at the points $s_1, s_2, \ldots$. In this case we write

$$\int_0^\infty g(x)\,\mathrm{d}F(x) = \sum_{i=1}^n g(s_i)\alpha_i.$$

Thus, the integral of g with respect to F is the sum of g at the points at which F makes a jump times the weight of F at these points.

As an example, in case $S \equiv 10$ (the service time is always 10 time units long) the distribution function F makes just one jump at $s_1 = 10$ of size $\alpha_1 = F(10) - F(10-) = 1$, i.e., F has the form

$$F(x) = \begin{cases} 0, & x < 10, \\ 1, & x \geq 10. \end{cases}$$

With this,

$$\int_0^\infty g(x)\,\mathrm{d}F(x) = \sum_{i=1}^n g(s_i)\alpha_i = g(s_1)\alpha_1 = g(10)\cdot 1 = g(10).$$

Thus, the integral of g with respect to this distribution F is $g(10)$. More generally, when F puts all probability mass at the single point s (rather than at the point 10), then

$$\int_0^\infty g(x)\,\mathrm{d}F(x) = g(s).$$

Let us now copy the formula of the previous problem:

$$\mathsf{P}\{Y_k = j\} = \int_0^\infty \mathsf{P}\{Y_k = j \,|\, S = x\}\,\mathrm{d}F(x).$$

We see that here the integrand is the function $g(x) = \mathsf{P}\{Y_k = j \,|\, S = x\}$. It is given in the question that F puts all mass on the point s. Therefore,

$$\mathsf{P}\{Y_k = j\} = \int_0^\infty \mathsf{P}\{Y_k = j \,|\, S = x\}\,\mathrm{d}F(x) = \int_0^\infty g(x)\,\mathrm{d}F(x) = g(s) \qquad \star$$

Now we also know that if the service time takes precisely x time units, the number of arrivals is Poisson distributed. Therefore

$$\mathsf{P}\{Y_k = j \,|\, S = x\} = e^{-\lambda x}\frac{(\lambda x)^j}{j!}.$$

Hence,

$$g(x) = \mathsf{P}\{Y_k = j \,|\, S = x\} = e^{-\lambda x}\frac{(\lambda x)^j}{j!}.$$

Finally, using ($\star$), and taking $x = s$ in the above expression of g, we get

$$\mathsf{P}\{Y_k = j\} = \int_0^\infty \mathsf{P}\{Y_k = j \,|\, S = x\}\,\mathrm{d}F(x) = g(s) = e^{-\lambda s}\frac{(\lambda s)^j}{j!}.$$

Solution 2.18.6: Use conditional probability to see that

$$\begin{aligned}
\mathsf{P}\{Y_n = j\} &= \int_0^\infty e^{-\lambda x}\frac{(\lambda x)^j}{j!}\,\mathrm{d}F(x) = \int_0^\infty e^{-\lambda x}\frac{(\lambda x)^j}{j!}\mu e^{-\mu x}\,\mathrm{d}x \\
&= \frac{\mu}{j!}\lambda^j \int_0^\infty e^{-(\lambda+\mu)x}x^j\,\mathrm{d}x = \frac{\mu}{j!}\left(\frac{\lambda}{\lambda+\mu}\right)^j \int_0^\infty e^{-(\lambda+\mu)x}((\lambda+\mu)x)^j\,\mathrm{d}x \\
&= \frac{\mu}{j!}\left(\frac{\lambda}{\lambda+\mu}\right)^j \frac{j!}{\lambda+\mu}.
\end{aligned}$$

Here I also used the hints and results of Exercise 2.1.4.

In hindsight, this result could have been derived in another way, in fact by using the result of Exercise 2.1.13 in which we analyzed a merged Poisson process. Consider the Poisson process with rate $\lambda+\mu$ that arises when the arrival and service process are merged. The probability that an arrival corresponds to an epoch of the merged process is $\lambda/(\lambda+\mu$ and the probability that a departure corresponds to an epoch of the merged process is $\mu/(\lambda+\mu)$. The probability that j arrivals occur before a service occurs, is the same as the probability that a geometrically distributed random variable with success probability $\mu/(\lambda+\mu) = 1-p$ takes the value j.

Solution 2.18.7: Take $\alpha = \lambda/(\lambda+\mu)$ so that $f(j) = (1-\alpha)\alpha^j$.

$$\begin{aligned}
G(j) &= \sum_{k=j+1}^{\infty} f(k) = (1-\alpha)\sum_{k=j+1}^{\infty}\alpha^k \\
&= (1-\alpha)\sum_{k=0}^{\infty}\alpha^{k+j+1}, \text{ by change of variable} \\
&= (1-\alpha)\sum_{k=0}^{\infty}\alpha^k\alpha^{j+1} = (1-\alpha)\alpha^{j+1}\sum_{k=0}^{\infty}\alpha^k \\
&= (1-\alpha)\alpha^{j+1}\frac{1}{1-\alpha} = \alpha^{j+1}.
\end{aligned}$$

Solution 2.18.8: Take $n = 0$. Then $f(0)\pi(1) = (1-\alpha)\pi(1)$, and $\pi(0)G(0) = \pi(0)\alpha$. Thus, $\pi(1) = \pi(0)\alpha/(1-\alpha) = \rho\pi(0)$.

For $n = 1$,

$$\begin{aligned}
(1-\alpha)\pi(2) &= \pi(0)G(1) + \pi(1)G(1) = \pi(0)G(1)(1+\rho) \\
&= \pi(0)\alpha^2(1+\rho) = \pi(0)\alpha\alpha(1+\rho) = \pi(0)\alpha\rho.
\end{aligned}$$

Dividing by $1-\alpha$, we get

$$\pi(2) = \pi(0)\rho^2$$

Finally, now that we suspect that $\pi(n) = \rho^n\pi(0)$, let's fill it in, and see whether it checks. We divide both sides by $\pi(0)$ so that we are left with checking that

$$\begin{aligned}
(1-\alpha)\rho^{n+1} &= \alpha^{n+1} + \sum_{m=1}^{n}\rho^m\alpha^{n-m+2} \\
&= \alpha^{n+1} + \alpha^{n+2}\sum_{m=1}^{n}(\rho/\alpha)^m \\
&= \alpha^{n+1} + \alpha^{n+1}\rho\sum_{m=0}^{n-1}(\rho/\alpha)^m \\
&= \alpha^{n+1} + \alpha^{n+1}\rho\sum_{m=0}^{n-1}(\rho/\alpha)^m \\
&= \alpha^{n+1} + \alpha^{n+1}\rho\frac{1-(\rho/\alpha)^n}{1-\rho/\alpha)} \\
&= \alpha^{n+1} - \alpha^{n+1}(1-(\rho/\alpha)^n), \quad \text{as } \rho/\alpha = 1+\rho, \\
&= \alpha^{n+1}(\rho/\alpha)^n = \alpha\rho^n.
\end{aligned}$$

Since $\rho = \alpha/(1-\alpha)$ we see that the left and right hand sides are the same.

Thus we get that $\pi(n) = \delta(n) = (1-\rho)\pi(0)$; a result we obtained earlier for the $M/M/1$ queue.

Solution 2.18.9: A simple numerical method is as follows. Make a grid of size $\mathrm{d}x$, for some small number $\mathrm{d}x$, e.g. $\mathrm{d}x = 1/100$, and write $f_i = \mathsf{P}\{S \in (i\,\mathrm{d}x, (i+1)\,\mathrm{d}x]\} = F((i+1)\mathrm{d}x) - F(i\,\mathrm{d}x)$. Then

$$\mathsf{P}\{Y_k = j\} = \int_0^{\infty} e^{-\lambda s}\frac{(\lambda x)^j}{j!}\,\mathrm{d}F(x) \approx \sum_{i=1}^{\infty} e^{-\lambda i\,\mathrm{d}x}\frac{(\lambda i\,\mathrm{d}x)^j}{j!} f_i\,\mathrm{d}x$$

Lets try a numerical experiment. Topics like these are, of course, not necessary for the course, but I hope that you study it nonetheless. You will have to use it for other courses later anyway.

```python
import numpy as np

labda = 3
mu = 4
j = 5
dx = 1 / 100


def F(x):
    return 1 - np.exp(-mu * x)


def f(x):
    return F(x + dx) - F(x)


def term(i):
    res = np.exp(-labda * i * dx)
    res *= (labda * i * dx)**j / np.math.factorial(j)
    res *= f(i*dx) * dx
    return res

print(sum(term(i) for i in range(50)))
```

```
1.11588350072e-05
```

```python
print(sum(term(i) for i in range(500)))
```

```
8.09880771006e-05
```

```python
print(sum(term(i) for i in range(5000)))
```

```
8.09880771273e-05
```

Since I don't know when to stop the integral I just try a few values; of course stopping the integration at $x = 50$ is too small, since $50\,\mathrm{d}x = 50/100 = 1/2$, but I include it for illustrative purposes. Stopping at 500 seems ok, since the results of the last two integrals are nearly the same. This also suggestst that $\mathrm{d}x = 1/100$ is sufficiently small. In general, however, one must take care and try various values for $\mathrm{d}x$ and the integration limits.

In general, for more complicated situations is it best to use a numerical library to compute the above integral. The methods have been designed to produce good and reliable results, and, typically, it is very hard to improve these methods. Thus, let's try a real number cruncher.

```python
from scipy.integrate import quad

def g(x):
    return np.exp(-labda*x) * (labda*x)**j/np.math.factorial(j) * f(x)

print(quad(g, 0, np.inf))
```

```
(8.098807712760667e-05, 3.4086429822163874e-11)
```

Comparing these results to our simple estimates we see that they are the same.

2.19 A Relation Between the M^X/M/1 and M/G/1 Queue

Theory and Exercises

There is an interesting relation between the $M^X/M/1$ and the $M/G/1$ queueing systems. As an example, consider the $M^{10}/M/1$ queue in which each job corresponds to the arrival of ten items of work. Intuitively, it is reasonable that this queueing process is quite similar to an $M/G/1$ queue in which each job requires approximately 10 time units. Below we will make this relation precise. We will also use this to find another derivation of the Pollackzek-Khintchine equation, if we consider a certain limiting case.

The basic observation is that by discretizing the service time S of a job in an $M/G/1$ queue we can approximate S by the service time of a batch of small units of work. To make this more specific, assume that the distibution of the service time S is given. Use a grid of size $1/n$ and take

$$\mathsf{P}\left\{B^{(n)} = i\right\} = \mathsf{P}\left\{S \in \left(\frac{i}{n}, \frac{i+1}{n}\right]\right\} \tag{2.85}$$

as the distribution of the batch size $B^{(n)}$. Let the service time of one unit of the batch be exponentially distributed with mean $1/n$. Thus, if n becomes larger, the service time S is represented by ever larger batches, but such that the service time of each unit becomes ever smaller. These arguments can be made precise in a probabilistic sense, but here we rely on our intuition to conclude that, in some sense, $B^{(n)}/n \to S$[9].

Let us see how this limit can be used to show that the expected waiting in queue for the batch queue leads to the Pollackzek-Khintchine equation, i.e., the expected waiting time in queue for the $M/G/1$ queue. Writing $\mathsf{E}\{L^{(n)}\}$ for the batch queue with grid size $1/n$, we use (2.68) to see that

$$\frac{\mathsf{E}\{L^{(n)}\}}{n} = \frac{\rho}{1-\rho}\cdot\frac{1}{2}\left(1+\frac{\mathsf{V}\{B^{(n)}\}}{(\mathsf{E}\{B^{(n)}\})^2}\right)\frac{\mathsf{E}\{B^{(n)}\}}{n} + \frac{1}{2n}\frac{\rho}{1-\rho}. \tag{2.86}$$

By assumption[10]:

$$\frac{B^{(n)}}{n} \to S$$

so that must be that

$$\mathsf{E}\left\{\frac{B^{(n)}}{n}\right\} \to \mathsf{E}\{S\}$$

and

$$\mathsf{E}\left\{\frac{(B^{(n)})^2}{n^2}\right\} \to \mathsf{E}\{S^2\},$$

as $n\to\infty$. Hence,

$$\begin{aligned}\frac{\mathsf{V}\{B^{(n)}\}}{n^2} &= \frac{\mathsf{E}\{(B^{(n)})^2\} - (\mathsf{E}\{B^{(n)}\})^2}{n^2} = \mathsf{E}\left\{\frac{(B^{(n)})^2}{n^2}\right\} - \left(\mathsf{E}\left\{\frac{B^{(n)}}{n}\right\}\right)^2 \\ &\to \mathsf{E}\{S^2\} - (\mathsf{E}\{S\})^2 = \mathsf{V}\{S\}.\end{aligned}$$

[9] For the interested, we mean convergence in distribution here.

[10] Here we interchange limits and expectations. A formal approach should pay due attention to the validity of all these interchanges.

from which

$$\frac{\mathsf{V}\{B^{(n)}\}}{(\mathsf{E}\{B^{(n)}\})^2} = \frac{\mathsf{V}\{B^{(n)}\}}{n^2}\frac{n^2}{(\mathsf{E}\{B^{(n)}\})^2} \to \frac{\mathsf{V}\{S\}}{(\mathsf{E}\{S\})^2} = C_s^2,$$

as $n\to\infty$. Moreover, we *expect* that

$$\frac{\mathsf{E}\{L^{(n)}\}}{n} \to \mathsf{E}\{W_Q\},$$

since the number of items in the system, $\mathsf{E}\{L^{(n)}\}$, becomes ever larger as n increases, but the service time of one item is $1/n$, which becomes ever smaller. Now, taking the limit $n\to\infty$ in (2.86) we obtain

$$\frac{\mathsf{E}\{L^{(n)}\}}{n} \to \frac{\rho}{1-\rho}\frac{1+C_s^2}{2}\mathsf{E}\{S\}.$$

Thus, based on this limit we expect that

$$\mathsf{E}\{W_Q\} = \frac{\mathsf{E}\{L^{(n)}\}}{n} \to \frac{\rho}{1-\rho}\frac{1+C_s^2}{2}\mathsf{E}\{S\},$$

but this is the same as Eq. (2.72)!

Exercise 2.19.1. In Eq. (2.85) we define $\mathsf{P}\left\{B^{(n)}=i\right\} = \mathsf{P}\{S\in(i/n,(i+1)/n]\}$ and not

$$\mathsf{P}\left\{B^{(n)}=i\right\} = \mathsf{P}\{S\in[i/n,(i+1)/n]\},$$

i.e., we use a half open interval (,] rather than the closed interval [,]. Why is that?

Exercise 2.19.2. Why do we consider $\mathsf{E}\{L^{(n)}\}/n \to \mathsf{E}\{W_Q\}$, and not $\mathsf{E}\{L^{(n)}\}/n\to\mathsf{E}\{W\}$?

Exercise 2.19.3 (use=false)**.** Can you model the $M^X/M/1$ as an $M/G/1$ queue, thereby deriving Eq. (2.68) from Eq. (2.72)?

Hints

Hint 2.19.2: What is the expected service time of one unit of a job in the $M^X/M/1$ queue in the limiting case?

Solutions

Solution 2.19.1: Starting from the distribution function $\mathsf{P}\{S\le x\}$, we define

$$\mathsf{P}\left\{B^{(n)}=i\right\} = \mathsf{P}\{S\le(i+1)/n\} - \mathsf{P}\{S\le i/n\} = \mathsf{P}\{S\in(i/n,(i+1)/n]\}.$$

Solution 2.19.2: The time it takes to serve one unit is $1/n$. Thus, the fraction of time in service in the batch queue becomes negligible compared to the time in queue. Another way to see this, is to use

$$\frac{\mathsf{E}\{L^{(n)}\}}{n} = \frac{\mathsf{E}\left\{L_Q^{(n)}\right\}}{n} + \frac{\mathsf{E}\left\{L_s^{(n)}\right\}}{n}.$$

Since $L_s^{(n)}\le 1$, $\mathsf{E}\{L^{(n)}\}/n \approx \mathsf{E}\left\{L_Q^{(n)}\right\}/n$ for n large.

Solution 2.19.3: I guess this is a bit hard, so I include a partial answer for the interested.

Starting from the $M/G/1$ queue, suppose first that the service time S is an exponential random variable with mean $1/\mu$. Then the $M/G/1$ reduces to the $M/M/1$ queue, which is evidentally equal to the $M^1/M/1$ queue in which the jobs arrive as single units. Next, if S consists of the sum of k i.i.d. exponential random variables S_i with mean $1/\mu$, i.e., $S = \sum_{i=1}^k S_i$, then the $M^k/M/1$ queue results. Generalizing still further, we can take a random batch B of i.i.d. exponentials so that $S = \sum_{i=1}^B S_i$. Now

$$\mathsf{P}\{S \leq x\} = \sum_{k=1}^\infty \mathsf{P}\{S \leq x \,|\, B = k\}\,\mathsf{P}\{B = k\} = \sum_{k=1}^\infty f(k)\mathsf{P}\left\{\sum_{i=1}^B X_i \leq x\right\}$$

where $f(k) = \mathsf{P}\{B = k\}$, and we obtain the $M^X/M/1$ from the $M/G/1$ queue by taking this distribution function for the service times.

2.20 G/G/1 and G/G/c Queue: Approximations

Theory and Exercises

In manufacturing settings it is quite often the case that the arrival process at a station is not Poisson. For instance, if processing times at a station are nearly constant, and the jobs of this station are sent to a second station for further processing, the interarrival times at the second station must be more or less equal. Hence, in this case, the SCV of the arrivals at the second station $C_{a,2}^2$ is most probably smaller than 1. As a consequence the Pollackzek-Khintchine formula for the $M/G/1$-queue can no longer be reliably used to compute the average waiting times. As a second, trivial case, if the interarrival times of jobs are 1 hour always and service times 59 minutes always, there simply cannot be queue. Thus, the $M/G/1$ waiting time formula cannot, and should not, be applied to approximate the average waiting time of the $G/G/1$ queue.

There is no formula as yet by which the average waiting times for the $G/G/1$ queue can be computed; only approximations are available. One such simple and robust approximation is based on the following observation. Recall the waiting time in queue for the $M/M/1$ queue:

$$\mathsf{E}\left\{W_Q(M/M/1)\right\} = \frac{\rho}{1-\rho}\,\mathsf{E}\{S\} = \frac{1_a + 1_b}{2}\,\frac{\rho}{1-\rho}\,\mathsf{E}\{S\},$$

where we label the number 1 with a and b. When generalizing this result to the $M/G/1$ queue we get

$$\mathsf{E}\left\{W_Q(M/G/1)\right\} = \frac{1_a + C_s^2}{2}\,\frac{\rho}{1-\rho}\,\mathsf{E}\{S\},$$

Thus, 1_b in the expression for the $M/M/1$ queue is replaced by C_s^2 in the expression for the $M/G/1$ queue. As a second generalization, Kingman proposed to replace 1_a in this formula by the SCV of the interarrival times

$$C_a^2 = \frac{\mathsf{V}\{X\}}{(\mathsf{E}\{X\})^2}$$

resulting in

$$\mathsf{E}\left\{W_Q(G/G/1)\right\} \approx \frac{C_a^2 + C_s^2}{2}\,\frac{\rho}{1-\rho}\,\mathsf{E}\{S\}.$$

This formula is reasonably accurate; for related expressions we refer to Bolch et al. [2006] and Hall [1991]. With Little's law we can compute $\mathsf{E}\{L_Q\}$ from the above. Moreover, $\mathsf{E}\{W\} = \mathsf{E}\{W_Q\} + \mathsf{E}\{S\}$, and so on, c.f., Section 2.14

It is crucial to memorize the *scaling* relations that can be seen from the $G/G/1$ waiting time formula. Roughly:

1. $\mathsf{E}\{W_Q\} \sim (1-\rho)^{-1}$. The consequence is that the waiting time increases *very steeply* when ρ is large, hence is very sensitive to the actual value of ρ when ρ is large.
2. $\mathsf{E}\{W_Q\} \sim C_a^2$ and $\mathsf{E}\{W_Q\} \sim C_s^2$. Hence, reductions of the variation of the interarrival and service times do affect the waiting time, but only linearly.
3. $\mathsf{E}\{W_Q\} \sim \mathsf{E}\{S\}$. Thus, working in smaller job sizes reduces the waiting time also. The average queue length does not decrease by working with smaller batches, but jobs are more 'uniformly spread' over the queue. This effect lies behind the idea to do 'lot-splitting', i.e., rather than process large jobs, split jobs into multiple small jobs (assuming that setup times are negligible), so that the waiting time per job can be reduced.

These insights prove very useful when trying to reduce waiting times in any practical situation. First try to reduce the load (by blocking demand or increasing the capacity), then try to reduce the variability (e.g., by planning the arrival times of jobs), and finally, attempt to split jobs into multiple smaller jobs and use the resulting freedom to reschedule jobs in the queue.

For the $G/G/c$ queue we also only have an approximation:

$$\mathsf{E}\{W_Q\} = \frac{C_a^2 + C_e^2}{2} \frac{\rho^{\sqrt{2(c+1)}-1}}{c(1-\rho)} \mathsf{E}\{S\}. \tag{2.87}$$

Even though the above results are only approximate, they prove to be exceedingly useful when designing queueing systems and analyzing the effect of certain changes, in particular changes in capacity, variability and service times.

Clearly, Kingman's equation requires an estimate of the SCV C_a^2 of the interarrival times and the SCV C_s^2 of the service times. Now it is not always easy in practice to determine the actual service time distribution, one reason being that service times are often only estimated by a planner, but not actually measured. Similarly, the actual arrival moments of jobs are often not registered, mostly just the date or the hour, perhaps, that a customer arrived. Hence, it is often not possible to estimate C_a^2 and C_s^2 from information that is available. However, when for instance the number of arrivals per day have been logged for some time so that we know $\{a_n, n = 1,\dots,N\}$ for some N, we can use this information instead of the the interarrival times $\{X_k\}$ to obtain insight into C_a^2. The relation we present here to compute C_a^2 from $\{a_n\}$ can of course also be applied to estimate C_s^2.

Theorem 2.20.1. The SCV of the interarrival times can be estimated with the following formula

$$C_a^2 \approx \frac{\tilde{\sigma}^2}{\tilde{\lambda}},$$

where

$$\tilde{\lambda} = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n a_i, \quad \tilde{\sigma}^2 = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n a_i^2 - \tilde{\lambda}^2,$$

in words, $\tilde{\lambda}$ is the average number of arrivals per period, e.g., per day, and $\tilde{\sigma}^2$ is the variance of the number of arrivals per period.

Proof. The proof is based on an argument in Cox [1962]. We use quite a bit of the notation developed in Section 2.7. Let $\{A(t), t \geq 0\}$ be the number of arrivals that occur up to (and including) time t. We assume that $\{A(t)\}$ is a renewal process such that the interarrival times $\{X_k, k = 1, 2, \ldots\}$ with $X_k = A_k - A_{k-1}$, are i.i.d. with mean $1/\lambda$ and standard deviation σ. (Observe that σ is not the same as $\tilde{\sigma}$ above.) Note that C_a^2 is defined in terms of λ and σ as:

$$C_a^2 = \frac{\mathsf{V}\{X_i\}}{(\mathsf{E}\{X_i\})^2} = \frac{\sigma^2}{1/\lambda^2} = \lambda^2\sigma^2.$$

Next, let A_k be the arrival time of the kth arrival. The following useful relation between $A(t)$ and A_k enables us to prove our result (recall that we uses a similar relation in the derivation of the Poisson process):

$$\mathsf{P}\{A(t) < k\} = \mathsf{P}\{A_k > t\}$$

Since the interarrival times have finite mean and second moment by assumption, we can apply the central limit law to obtain that, as $k \to \infty$,

$$\frac{A_k - k/\lambda}{\sigma\sqrt{k}} \to N(0,1),$$

where $N(0,1)$ is a standard normal random variable with distribution $\Phi(\cdot)$. Similarly,

$$\frac{A(t) - \lambda t}{\alpha\sqrt{t}} \to N(0,1)$$

for an α that is yet to be determined. Thus, $\mathsf{E}\{A(t)\} = \lambda t$ and $\mathsf{V}\{A(t)\} = \alpha^2 t$.

Using that $\mathsf{P}\{N(0,1) \leq y\} = \mathsf{P}\{N(0,1) > -y\}$ (and $\mathsf{P}\{N(0,1) = y\} = 0$) we have that

$$\begin{aligned}\Phi(y) &\approx \mathsf{P}\left\{\frac{A_k - k/\lambda}{\sigma\sqrt{k}} \leq y\right\} \\ &= \mathsf{P}\left\{\frac{A_k - k/\lambda}{\sigma\sqrt{k}} > -y\right\} \\ &= \mathsf{P}\left\{A_k > \frac{k}{\lambda} - y\sigma\sqrt{k}\right\}.\end{aligned}$$

Define for ease

$$t_k = \frac{k}{\lambda} - y\sigma\sqrt{k}.$$

We can use the above relation between the distributions of $A(t)$ and A_k to see that $\mathsf{P}\{A_k > t_k\} = \mathsf{P}\{A(t_k) < k\}$. With this we get,

$$\begin{aligned}\Phi(y) &\approx \mathsf{P}\{A_k > t_k\} \\ &= \mathsf{P}\{A(t_k) < k\} \\ &= \mathsf{P}\left\{\frac{A(t_k) - \lambda t_k}{\alpha\sqrt{t_k}} < \frac{k - \lambda t_k}{\alpha\sqrt{t_k}}\right\}.\end{aligned}$$

Since, $(A(t_k) - \lambda t_k)/\alpha\sqrt{t_k} \to N(0,1)$ as $t_k \to \infty$, the above implies that

$$\frac{k - \lambda t_k}{\alpha\sqrt{t_k}} \to y,$$

as $t_k \to \infty$. Using the above definition of t_k, the left hand of this equation can be written as

$$\frac{k - \lambda t_k}{\alpha\sqrt{t_k}} = \frac{\lambda\sigma\sqrt{k}}{\alpha\sqrt{k/\lambda + \sigma\sqrt{k}}} y.$$

Since $t_k \to \infty$ is implied by (and implies) $k \to \infty$, we therefore want that α is such that

$$\frac{\lambda\sigma\sqrt{k}}{\alpha\sqrt{k/\lambda + \sigma\sqrt{k}}} y \to y,$$

as $k \to \infty$. This is precisely the case when

$$\alpha = \lambda^{3/2}\sigma.$$

Finally, for t large (or, by the same token k large),

$$\frac{\sigma_k^2}{\lambda_k} = \frac{\mathsf{V}\{A(t)\}}{\mathsf{E}\{A(t)\}} \approx \frac{\alpha^2 t}{\lambda t} = \frac{\alpha^2}{\lambda} = \frac{\lambda^3\sigma^2}{\lambda} = \lambda^2\sigma^2 = C_a^2,$$

where the last equation follows from the above definition of C_a^2. The proof is complete. $\square$

Exercise 2.20.1. Show that the approximation (2.87) for the $G/G/1$ queue reduces to the result known for the $M/M/1$ and $M/G/1$ queues.

Exercise 2.20.2. Is Eq. (2.63) also valid for the $G/G/1$ queue? Why (not)?

Exercise 2.20.3. Consider a queue n servers, with generally distributed interarrival times, generally distributed service times, and the system can contain at most K customers, i.e, the $G/G/n/K$ queue. Let λ be the arrival rate, μ the service rate, β the long-run fraction of customers lost, and ρ the average number of busy/occupied servers Show that

$$\beta = 1 - \rho\frac{\mu}{\lambda}.$$

Exercise 2.20.4.

1. Consider a single-server queue at which every minute a customer arrives, precisely at the first second. Each customer requires precisely 50 seconds of service. What are ρ, $\mathsf{E}\{L\}$, C_a^2, and C_S^2?
2. Consider the same single-server system, but now the customer service time is stochastic: with probability 1/2 a customer requires 1 minute and 20 seconds of service, and with probability 1/2 the customer requires only 20 seconds of service. What are ρ, C_a^2, and C_S^2?

Exercise 2.20.5. For the $G/G/1$ queue, prove that the fraction of jobs that see n jobs in the system is the same as the the fraction of departures that leave n jobs behind. What condition have you used to prove this?

Exercise 2.20.6. Suppose for the $G/G/1$ that a job sees n jobs in the system upon arrival. Can you use the central limit law to estimate the distribution of waiting time in queue for this job?

Exercise 2.20.7. (Hall 5.19) When a bus reaches the end of its line, it undergoes a series of inspections. The entire inspection takes 5 minutes on average, with a standard deviation of 2 minutes. Buses arrive with interarrival times uniformly distributed on [3,9] minutes.

1. Assuming a single server, estimate $\mathsf{E}\{W_Q\}$ with the $G/G/1$ waiting time formula.
2. Compare this result to an $M/G/1$ system with arrival rate 10 per hour and the same service time distribution. Explain why your previous answer is smaller.

Hints

Hint 2.20.1: Take $c = 1$.

Hint 2.20.6: Let $S_n = \sum_{k=1}^n X_k$ and $\sigma_n^2 = n \mathsf{V}\{X\}$. Then, under conditions you can find on the internet,

$$\frac{S_n - n\,\mathsf{E}\{X\}}{\sigma_n} \to \mathcal{N}(0,1), \quad \text{as } n \to \infty,$$

where $\mathcal{N}(0,1)$ is a normally distributed random variable with $\mu = 0$ and $\sigma^2 = 1$. But then

$$\begin{aligned}
\frac{S_n - n\,\mathsf{E}\{X\}}{\sigma_n} &\approx \mathcal{N}(0,1) \iff \\
S_n - n\,\mathsf{E}\{X\} &\approx \sigma_n \mathcal{N}(0,1) \iff \\
S_n - n\,\mathsf{E}\{X\} &\approx \mathcal{N}(0,\sigma_n^2) \iff \\
S_n &\approx n\,\mathsf{E}\{X\} + \mathcal{N}(0,\sigma_n^2) \iff \\
S_n &\approx \mathcal{N}(n\,\mathsf{E}\{X\},\sigma_n^2),
\end{aligned}$$

hence, S_n is approximately distributed as $\mathcal{N}(n\,\mathsf{E}\{X\},\sigma_n^2)$.

Hint 2.20.7: It took me some time to understand the details of part a. I then decided to read part b first, which clarified the intent of part a.

Solutions

Solution 2.20.1: If $c = 1$, as is the case for the $M/M/1$ queue, $\sqrt{2(c+1)} - 1 = 2 - 1 = 1$, so that (2.87) reduces to $\mathsf{E}\{W_Q\} = \rho/(1-\rho)\mathsf{E}\{S\}$. Recall that $C_a^2 = C_s^2 = 1$ for the $M/M/1$.

Solution 2.20.2: Not necessarily. Since jobs do not arrive in general as a Poisson process, we cannot use the PASTA property to conclude that the time average queue length $\mathsf{E}\{L_Q\}$ is the same as the average queue length as observed by customers.

(Can you provide an example that shows this difference? Hint, we did this earlier.)

Solution 2.20.3: This follows from the observation that $\lambda(1-\beta)$ is the net arrival rate, as jobs are lost at a rate $\lambda\beta$. Hence, the load must be $\lambda(1-\beta)/\mu$. As the load must be equal to ρ, it follows that $\rho = \lambda(1-\beta)/\mu$, from which the other result immediately follows.

Solution 2.20.4:

1. $\rho = \lambda\,\mathsf{E}\{S\} = 1/60 \cdot 50 = 5/6$. Since job arrivals do not overlap any job service, the number of jobs in the system is 1 for 50 seconds, then the server is idle for 10 seconds, and so on. Thus $\mathsf{E}\{L\} = 1 \cdot 5/6 = 5/6$. There is no variance in the interarrival times, and also not in the service times, thus $C_a^2 = C_s^2 = 0$. Also $\mathsf{E}\{W_Q\} = 0$ since $\mathsf{E}\{L_Q\} = 0$. Interestingly, we get the same from Kingman's formula.

2. Again $\mathsf{E}\{S\}$ is 50 seconds, so that $\rho = 5/6$. Also $C_a^2 = 0$. For the C_S^2 we have to do some work.

$$\begin{aligned}
\mathsf{E}\{S\} &= \frac{20}{2} + \frac{80}{2} = 50 \\
\mathsf{E}\{S^2\} &= \frac{400}{2} + \frac{6400}{2} = 3400 \\
\mathsf{V}\{S\} &= \mathsf{E}\{S\}^2 - (\mathsf{E}\{S\})^2 = 3400 - 2500 = 900 \\
C_S^2 &= \frac{\mathsf{V}\{S\}}{(\mathsf{E}\{S\})^2} = \frac{900}{2500} = \frac{9}{25}.
\end{aligned}$$

It is crucial to remember from this exercise that knowledge of the utilization is not sufficient to characterize the average queue length.

Solution 2.20.5: All follows straightaway from the definitions in the main text. In the $G/G/1$ queue jobs arrive and depart in single units. Hence, $|A(n,t)-D(n,t)| \leq 1$, c.f., Eq. (2.37a). Then,

$$\frac{A(t)}{t}\frac{A(n,t)}{A(t)} \approx \frac{D(t)}{t}\frac{D(n,t)}{D(t)}.$$

The left hand side goes to $\lambda\pi(n)$ as $t \to \infty$, and the right hand side to $\gamma\delta(n)$. Use the fact that we always assume, implicitly, that the system is stable, so that $\lambda = \gamma$. As a consequence $\delta(n) = \pi(n)$.

Solution 2.20.6: If the number in the system is n, then the total service time required to process these n jobs is the sum of the service times, i.e.,$W_{Q,k} = \sum_{k=1}^{n} S_k$ where k counts over the n jobs in the system. Then, with the hint,

$$W_{Q,k} \sim \mathcal{N}(n\,\mathsf{E}\{S\}, \sigma_n^2)$$

where $\sigma_n^2 = n\,\mathsf{V}\{S\}$. Thus, the waiting time in queue is approximately normally distributed with mean $n\,\mathsf{E}\{S\}$ and variance $n\,\mathsf{V}\{S\}$.

Solution 2.20.7: 1.
```
a = 3.
b = 9.
EA = (b+a)/2.
EA

6.0

labda = 1./EA # per minute
labda

0.16666666666666666

VA = (b-a)*(b-a)/12.
CA2 = VA/(EA*EA)
CA2

0.08333333333333333

ES = 5.
sigma = 2
VS = sigma*sigma
CS2 = VS/(ES*ES)
CS2

0.16

rho = labda*ES
rho

0.8333333333333333

Wq = (CA2+CS2)/2. * rho/(1.-rho) * ES
Wq
```

```
3.0416666666666665
```

2. ```
   Wq = (1.+CS2)/2.*rho/(1.-rho)*ES
   Wq
   ```

   ```
   14.499999999999993
   ```

   The arrival process with uniform interarrival times is much more regular than a Poisson process. In the first case, bus arrivals are spaced in time at least with 3 minutes.

2.21 Service Interruptions

Theory and Exercises

See the related section 1.11 of the book of Zijm.

No notes; only questions.

Exercise 2.21.1. Determine the expectation and variance of $S_N = \sum_{i=1}^{N} X_i$ where N is a random variable (with finite second moment) and $\{X_i\}$ are i.i.d. as the generic random variable X.

Exercise 2.21.2. A small company receives orders at a rate λ. Interarrival times between orders are exponentially distributed. Once B orders have been assembled, the company starts produces this batch of B jobs. The average service time of an order is $1/\mu$. Between two batches the company has to re-adjust some of the machines before it can start processing another batch. These setup times form an i.i.d. sequence of times with average $\mathsf{E}\{S\}$ time units.

1. Should the batch size be limited from above or below to ensure that the system remains stable?
2. How many orders arrive on average during a setup?
3. Use the above two points to determine a criterion on the minimal batch size B.

Exercise 2.21.3. Zijm.Ex.1.11.1

Exercise 2.21.4. Zijm.Ex.1.11.2

Exercise 2.21.5. Zijm.Ex.1.11.3

Exercise 2.21.6. Zijm.Ex.1.11.4 and 1.11.5

Exercise 2.21.7. Zijm.Ex.1.11.4

Exercise 2.21.8. Zijm.Ex.1.11.6

Exercise 2.21.9. Zijm.Ex.1.11.7

Hints

Hint 2.21.2: 1: If the batch sizes B are small, relatively much time is spent on setups. 2: If the setup time S is constant, then the number of orders that arrive is Poisson distributed. 3: Given a batch size B, find the time to produce such a batch, then include a setup time S to compute the length of one production cycle. The expected number of jobs served during such a cycle must not exceed the number of jobs served during the cycle.
   ```

Solutions

Solution 2.21.1: We first consider the expectation. If it is given that $N = n$, then

$$\mathsf{E}\{S_N \,|\, N = n\} = \mathsf{E}\left\{\sum_{i=1}^{n} X_i\right\} = \mathsf{E}\{X_1\} + \mathsf{E}\{X_2\} + \cdots \mathsf{E}\{X_n\} = n\,\mathsf{E}\{X\},$$

where the last equation follows from the fact that the X_i have the same distribution. With this,

$$\mathsf{E}\left\{\sum_{i=1}^{N} X_i\right\} = \mathsf{E}\left\{\mathsf{E}\left\{\sum_{i=1}^{N} X_i \,\middle|\, N\right\}\right\} = \mathsf{E}\{N\,\mathsf{E}\{X\}\} = \mathsf{E}\{X\}\,\mathsf{E}\{N\}.$$

This result in known as Wald's equation.

To compute the variance we use that $\mathsf{V}\{S_N\} = \mathsf{E}\{S_N^2\} - (\mathsf{E}\{S_N\})^2$. Thus, we first compute

$$\begin{aligned}
\mathsf{E}\{S_N^2 \,|\, N = n\} &= \mathsf{E}\left\{\left(\sum_{i=1}^{n} X_i\right)^2\right\} = \mathsf{E}\left\{\sum_{i=1}^{n} X_i^2 + \sum_{i\neq j}^{n} X_i X_j\right\} \\
&= \sum_{i=1}^{n} \mathsf{E}\{X_i^2\} + \sum_{i\neq j} \mathsf{E}\{X_i\}\,\mathsf{E}\{X_j\},
\end{aligned}$$

since $\mathsf{E}\{X_i X_j\} = \mathsf{E}\{X_i\}\,\mathsf{E}\{X_j\}$ by independence. Therefore, using that the $\{X_i\}$ are i.i.d. as X,

$$\mathsf{E}\{S_N^2 \,|\, N = n\} = n\,\mathsf{E}\{X^2\} + n(n-1)(\mathsf{E}\{X\})^2.$$

Thus, interpreting $\mathsf{E}\{S_N^2 \,|\, N\}$ as a function of N, we get

$$\mathsf{E}\{S_N^2 \,|\, N\} = N\,\mathsf{E}\{X^2\} + N(N-1)(\mathsf{E}\{X\})^2,$$

from which follows that

$$\begin{aligned}
\mathsf{E}\{S_N^2\} &= \mathsf{E}\{\mathsf{E}\{S_N^2 \,|\, N\}\} \\
&= \mathsf{E}\{N\,\mathsf{E}\{X^2\} + N(N-1)(\mathsf{E}\{X\})^2\} \\
&= \mathsf{E}\{N\}\,\mathsf{E}\{X^2\} + \mathsf{E}\{N^2\}(\mathsf{E}\{X\})^2 - \mathsf{E}\{N\}(\mathsf{E}\{X\})^2 \\
&= \mathsf{E}\{N\}(\mathsf{E}\{X^2\} - (\mathsf{E}\{X\})^2) + \mathsf{E}\{N^2\}(\mathsf{E}\{X\})^2 \\
&= \mathsf{E}\{N\}\,\mathsf{V}\{X\} + \mathsf{E}\{N^2\}(\mathsf{E}\{X\})^2.
\end{aligned}$$

And finally,

$$\begin{aligned}
\mathsf{V}\{S_N\} &= \mathsf{E}\{S_N^2\} - (\mathsf{E}\{S_N\})^2 \\
&= \mathsf{E}\{N\}\,\mathsf{V}\{X\} + \mathsf{E}\{N^2\}(\mathsf{E}\{X\})^2 - (\mathsf{E}\{N\})^2(\mathsf{E}\{X\})^2 \qquad (2.88) \\
&= \mathsf{E}\{N\}\,\mathsf{V}\{X\} + \mathsf{V}\{N\}(\mathsf{E}\{X\})^2.
\end{aligned}$$

Solution 2.21.2:

1. Only from below. If batches are very large, the fraction of time spent on setups becomes very small.
2. The expected number of arrivals is
$$\lambda\,\mathsf{E}\{S\}.$$
To see this, note that the number of arrivals N during a constant time is Poisson. Thus,
$$\mathsf{E}\{N\} = \mathsf{E}\{\mathsf{E}\{N \,|\, S\}\} = \mathsf{E}\{\mathsf{E}\{\lambda S \,|\, S\}\} = \lambda\,\mathsf{E}\{\mathsf{E}\{S \,|\, S\}\} = \lambda\,\mathsf{E}\{S\}.$$

3. The number of jobs that arrive, on average, during one production cycle must be smaller than the total amount of jobs that can be served, on average, during one cycle. The number of arrivals during the setup is $\lambda \mathsf{E}\{S\}$. The number of arrivals during serving the batch is $\lambda B/\mu$. Thus,
$$\lambda(B/\mu + \mathsf{E}\{S\}) \leq B.$$
When equality holds here, the system is critically loaded. As we discussed before, that it not a good idea.

Solution 2.21.3: Yes. The availability is 93%. Since $\mathsf{E}\{S_e\} = \mathsf{E}\{S_0\}/A$, and if $\mathsf{E}\{S_0\}$ is known, $\mathsf{E}\{S_e\}$ follows.

Solution 2.21.4: No, from Zijm.Eq.1.51, the average repair time has to be known. The repair time here is of course the time it takes for a mechanic to show up at work again.

Solution 2.21.5: It might, but perhaps a normal distribution would be better. It makes sense to make a histogram of the recover times to see whether some clear pattern is present.

Besides this, I don't know how sensitive the result for C_e^2 is on the distribution of the repair times. Perhaps it is not that sensitive, so in that case it would be ok to simply use the exponential distribution.

The sensitivity study would be an interesting topic for simulation.

Solution 2.21.6: We need to make an assumption about the distribution of the repair times. Inferring from the text, the repair time is always 2 days. Lets also assume that all jobs accumulate in front of the broken machine, in other words, the broken machine is part of the job routing of each job. Then

$$\mathsf{P}\{N(2) > 20\} = \sum_{n=21}^{\infty} e^{-5\cdot 2}\frac{10^n}{n!}.1 - \sum_{n=0}^{20} e^{-5\cdot 2}\frac{10^n}{n!} = 1 - 0.9984,$$

i.e., very small. This is the code I used:

```
from math import exp, factorial
exp(-10) * sum((10)**n / factorial(n) for n in range(21))
```

```
0.9984117393381421
```

Solution 2.21.7: If the shop already contains 10 jobs, somewhere upstream of the broken machine, then
$$\mathsf{P}\{N(2) > 10\} = 1 - 0.58.$$

where I used

```
exp(-10) * sum((10)**n / factorial(n) for n in range(11))
```

```
0.5830397501929855
```

Solution 2.21.8: Cleaning times will be pretty constant. Changing dies, or other machine parts, is also typically quite predictable, although it can take a lot of time, in particular in case a crane or other heavy machinery is needed to replace parts. If the machine require temporary adjustments, then the variation in setup times may be quite a bit higher.

Solution 2.21.9: Then the effective service times, and in particular, C_e^2 will be quite a bit bigger. It is preferable to avoid such a situation.

Mathematically, it is only given that N_s is a random variable. As, however, this does not state anything about its distribution, we cannot make any general claim. The intent of the problem is to have you check the relevant formulas and notice that the variance of N_s appears in the formulas.

2.22 Process Batching

Theory and Exercises

Exercise 2.22.1. Zijm.Ex.1.12.1

Exercise 2.22.2. Zijm.Ex.1.12.2

Exercise 2.22.3. Zijm.Ex.1.12.3

Hints

Solutions

Solution 2.22.1: Zijm, Eq. 1.66

```
labda = 10
mu = 30
ESs = 2 #days

labda*ESs/(1-labda/mu)

29.999999999999996
```

At least 30 jobs need to be served in a batch.

Solution 2.22.2:
```
N = 50
VarSs = 0 # fixed setup time
ES0 = 1/mu
VarS0 = 1/mu**2 # M/M/1, hence exponential service times

ESb = ESs + N*ES0 # Eq.1.64 Zijm

Csb2 = (VarSs + N*VarS0)/ESb**2
Csb2

0.004132231404958677

labdaB = labda/N
labdaB

0.2


rho = labdaB*ESb # Eq.1.65
rho

0.7333333333333334
```

When inspecting the SCV, I find it a bit small. To see whether I made a mistake, I'll try to make Figure 1.6 of Zijm.

```
VarSs = 0
ES0 = 1 / mu
VarS0 = 1 / mu**2  # M/M/1, hence exponential service times

for N in range(31, 120, 5):
    ESb = ESs + N * ES0   # Eq.1.64 Zijm
    Csb2 = (VarSs + N * VarS0) / ESb**2
    labdaB = labda / N
    rho = labdaB * ESb  # Eq.1.65
    Cab2 = 1 / N  # recall that Ca2=1 for M/M/1
    W = (Cab2 + Csb2) / 2 * rho / (1 - rho) * ESb + ESb
    print("{} {:f} {:f} {:f} {:f}".format(N, ESb, Csb2, rho, W))
```

```
31 3.033333 0.003744 0.978495 5.517742
36 3.200000 0.003906 0.888889 3.605556
41 3.366667 0.004019 0.821138 3.586216
46 3.533333 0.004094 0.768116 3.684511
51 3.700000 0.004139 0.725490 3.816106
56 3.866667 0.004162 0.690476 3.961630
61 4.033333 0.004166 0.661202 4.114252
66 4.200000 0.004157 0.636364 4.270960
71 4.366667 0.004137 0.615023 4.430224
76 4.533333 0.004109 0.596491 4.591190
81 4.700000 0.004074 0.580247 4.753341
86 4.866667 0.004035 0.565891 4.916348
91 5.033333 0.003991 0.553114 5.079995
96 5.200000 0.003945 0.541667 5.244129
101 5.366667 0.003896 0.531353 5.408644
106 5.533333 0.003847 0.522013 5.573461
111 5.700000 0.003796 0.513514 5.738522
116 5.866667 0.003745 0.505747 5.903782
```

This is not the same as the results of Figure 1.6. After closely inspecting my results above, I can't find a mistake... Perhaps you see what's wrong.

Solution 2.22.3: Definitely bigger. Observe that the waiting time is very sensitive to the batch size, hence to the estimates of all involved parameters such as the arrival rate, service times, setup time, and so on. Since, in general, it is very hard to obtain good estimates of these data (the order portfolio changes continuously, personnel does not always produce the same quality, rework may result, and so on and so forth), one must stay away from such situations. It is much better to be on the safe side of things.

Bibliography

F. Baccelli and W.A. Massey. A sample path analysis of the $M/M/1$ queue. *Journal of Applied Probability*, 26(2):418–422, 1988.

G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley & Sons, 2006.

M. Capiński and T. Zastawniak. *Probability through Problems*. Springer Verlag, 2nd edition, 2003.

D.R. Cox, editor. *Renewal Theory*. John Wiley & Sons Inc, New York, 1962.

M. El-Taha and S. Stidham Jr. *Sample-Path Analysis of Queueing Systems*. Kluwer Academic Publishers, 1998.

R.W. Hall. *Queueing Methods for Services and Manufacturing*. Prentice Hall, 1991.

H.C. Tijms. *Stochastic Models, An Algorithmic Approach*. J. Wiley & Sons, 1994.

H.C. Tijms. *A First Course in Stochastic Models*. John Wiley & Sons, Chichester, 2003.

A.A. Yushkevich and E.B. Dynkin. *Markov Processes: Theorems and Problems*. Plenum Press, 1969.