Queueing Theory Notes, BMGT 835

Dr. Michael C. Fu
Robert H. Smith School of Business,
University of Maryland, College Park, MD 20742-1815 USA

February 2000

Bibliography

- Introductory Operations Research textbooks with queueing theory chapters

  Winston, Wayne L., *Operations Research: Applications and Algorithms*, PWS-Kent.

  Hillier, and Lieberman, *Introduction to Operations Research*, McGraw-Hill.

  Ross, Sheldon M., *Introduction to Probability Models*, Academic Press.

- Recommended Queueing Theory Books

  Cooper, Robert B., *Introduction to Queueing Theory*, North Holland.

  Gross, Donald and Carl M. Harris, *Fundamentals of Queueing Theory*, Wiley.

  Kleinrock, Leonard, Queueing Systems, Volumes 1 and 2, Wiley.

  Walrand, Jean, *An Introduction to Queueing Networks*, Prentice Hall, 1988.

  Wolff, Ronald W., *Introduction to Stochastic Modeling and the Theory of Queues*, Prentice-Hall, 1989.

1 Brief Overview of Continuous-Time Markov Chains

We will use $\{X_t, t \geq 0\}$ to denote the CTMC, and, we wish to find the **p.m.f.** of X_t, which in the short-run will always be a function of the initial conditions.
Definition. For state space given by

$$\{0, 1, ..., s\},$$

we will represent the **p.m.f. of** X_t by the vector

$$\mathbf{p}(t) = [p_0(t)p_1(t)...p_S(t)]\,,$$
$$\text{where}\ \ p_i(t) = P\{X_t = i\},$$

i.e, $\mathbf{p}(t)$ gives the probability of being in any possible state at time t. Since $\mathbf{p}(t)$ is a p.m.f., we have

$$\sum_{i=0}^{S} p_i(t) = 1.$$

In words, the entries of $\mathbf{p}(t)$ must sum to 1, since the CTMC must be in one of the states.

The dynamics of a CTMC are characterized by its transition probabilities as a function of time. In general, we define the probability of making a transition to state j after an interval of length s, given that the chain is in state i at time t:

$$p_{ij}(s) = P(X_{t+s} = j|X_t = i).$$

We shall assume henceforth that all of the probabilities are independent of t. Since these are probabilities, they satisfy some obvious properties:

1. $p_{ij}(t) \geq 0$ for all i, j.
2. $\sum_{\text{all } j} p_{ij}(t) = 1$.

It will be convenient to represent the transition probabilities in the form of a matrix:
Definition. The **probability transition matrix** for a CTMC is defined by

$$\mathbf{P}(t) = [p_{ij}(t)].$$

The Chapman-Kolmogorov equations can be derived by using simple conditioning arguments. We do this by conditioning on an intermediate state k:

$$p_{ij}(t+s) = P(X_{t+s} = j|X_0 = i) = \sum_{\text{all} k} P(X_{t+s} = j|X_t = k)P(X_t = k|X_0 = i), = \sum_{\text{all} k} p_{ik}(t)p_{kj}(s).$$

Since the transition probabilities require specification for every possible time interval, they are quite cumbersome to work with. Luckily, the Markovian property of the process allows us to work with something similar, a single transition **rate** matrix:
Definition. The **transition rate matrix** for a CTMC is defined by

$$\mathbf{Q} = [q_{ij}].$$

This matrix is also called the **infinitesimal generator** matrix, and the relationship to $p_{ij}(s)$ is given by

$$q_{ij} = \lim_{s \to 0} \frac{p_{ij}(s)}{s}, \quad j \neq i,$$

$$\Lambda_i = \sum_{j \neq i} q_{ij} = -q_{ii}.$$

The last equation defines the diagonal entries, and is in fact arbitrary, but it makes for convenient mathematical properties; in particular, it means that the rows of $\mathbf{Q}$ must sum to 0. The rate Λ_i is the rate of the exponential distribution that defines the time to leave state i. The corresponding probabilities are given by

$$p_{ij} = \frac{q_{ij}}{\Lambda_i}, j \neq i \quad (p_{ii} = 0).$$

These one-step transition probabilities define what is called the **embedded** (discrete-time) Markov chain for the CTMC.
Property. For a transition rate matrix,

1. $q_{ij} \geq 0$ for all $i \neq j$.

2. $\sum_{\text{all } j} q_{ij} = 0$.

The memoryless property of the exponential distribution ensures that the rates are in fact a sufficient amount of information to summarize these transition probabilities.

Example:

$$\begin{bmatrix} -0.1 & 0.1 \\ 0.2 & -0.2 \end{bmatrix}.$$

This matrix will correspond to the Wandering Hippie example. As we shall see, in the long run, she will spend approximately 2/3 of her time in city 0 and 1/3 of her time in city 1. This type of chain is called an ergodic chain, which means that in the long-run it doesn't matter where she is now.

A homogeneous CTMC is completely characterized by its transition rate matrix. Given an initial p.m.f. vector $\mathbf{p}(0)$, any $\mathbf{p}(t)$ can be determined through the transition rate matrix. In general, we have the following linear (matrix) differential equation to solve to get the short-run p.m.f. vector $\mathbf{p}(t)$:

$$\mathbf{p}(t) = \mathbf{p}(0)e^{\mathbf{Q}t},$$

where the exponential of a matrix is defined by

$$e^{\mathbf{P}} = \mathbf{I} + \mathbf{P} + \frac{1}{2}\mathbf{P}^2 + \frac{1}{3!}\mathbf{P}^3 + \ldots$$

In general, carrying out the matrix multiplication directly is not a very efficient method; instead spectral decomposition is used, but this is beyond the scope of these notes.

1.0.1 Birth-Death Processes

A birth-death process is a special type of stochastic process. Let $N(t)$ be the state at time t, which we can think of as the population. Then, a birth-death process has the following characteristics:

1. Given $N(t) = n$, the time until the next birth is exponentially distributed with rate $\lambda_n (n \geq 0)$.

2. Given $N(t) = n$, the time until the next death is exponentially distributed with rate $\mu_n (n \geq 1)$.

3. Births and deaths can occur only one at a time, and are independent of each other.

Because the times between births and the times between deaths are exponentially distributed, a birth-death process is a special type of CTMC, in which transitions can only take place to "neighboring" states. In steady-state, we use flow balance and normalization to solve.

Clearly, a birth process is a special case of a birth-death process when the death rate is zero in all states. Thus, the Poisson process is also a special case of a birth-death process.

We summarize the analogy between the discrete-time and the continuous-time cases, as follows:
In using CTMC steady-state models, we have the following steps:

1. Define the **state** of the system.

2. Draw the transition **rate** diagram (or write down the transition rate matrix).

3. Write down the flow balance equations.

4. With the normalizing equation, solve for the steady-state probabilities.

5. Find the quantities of interest by **expressing them in terms of the probabilities**, e.g., the average is $\sum np_n$.

We illustrate this for three examples.
Example 1: Wandering Hippie A hippie wanders between two cities: 0 and 1. The state is the city in which the hippie is found. Let us assume that the average time between a movement (of the wandering type) from city 0 to 1 is 10 days, and that the average time between a movement from city 1 to city 0 is 5 days. We must also assume that these times are exponentially distributed in order to model the wanderings as a CTMC.

Comparison of DTMC and CTMC

	DTMC	CTMC
Markov property	$P\{X_{n+1}=j\|X_n=i, X_m=i_m, m=0,...,n-1\}$ $=P\{X_{n+1}=j\|X_n=i\}=p_{ij}$	$P\{X_{t+u}=j\|X_t=i, X_s=i_s, 0\leq s<t\}$ $=P\{X_{t+u}=j\|X_t=i\}=p_{ij}(u)$
pmf	$\boldsymbol{\pi}(n)=[\pi_0(n)\ \ \pi_1(n)\ldots]$ $\pi_i(n)=P\{X_n=i\}$	$p(t)=[p_0(t)\ \ p_1(t)\ldots]$ $p_i(t)=P\{X_t=i\}$
transition	one-step transition matrix $\mathbf{P}=[p_{ij}]$	transition *rate* matrix $\mathbf{Q}=[q_{ij}]$, $q_{ij}=\lim_{u\to 0} p_{ij}(u)/u, i\neq j$, $q_{ii}=\lim_{u\to 0}(p_{ii}-1)/u$
dynamics	$\boldsymbol{\pi}(n+1)=\boldsymbol{\pi}(n)\mathbf{P}$	$d\mathbf{p}(t)/dt=\mathbf{p}(t)\mathbf{Q}$
short-run (transient)	$\boldsymbol{\pi}(n)=\boldsymbol{\pi}(0)\mathbf{P}^n$	$\mathbf{p}(t)=\mathbf{p}(0)e^{\mathbf{Q}t}$
long-run ergodic (steady-state)	$\boldsymbol{\pi}=\boldsymbol{\pi}\mathbf{P},\ \ \boldsymbol{\pi}(\mathbf{P}-\mathbf{I})=0$ $\sum\pi_i=1$	$\mathbf{pQ}=0$ $\sum p_i=1$

The flow balance equations are given by:

$$\lambda p_0=\mu p_1,$$

where $\lambda=0.1$/day and $\mu=0.2$/day. The normalizing equation here is

$$p_0+p_1=1.$$

Solving, we get

$$p_0=1/3, \quad p_1=2/3.$$

Example 2: Two Light Bulbs On average a light bulb lasts for 22 days, with its lifetime being exponentially distributed. When a light bulb fails, the hippie's brother places an order for a replacement. On average, the replacement will arrive in 2 days, where the replacement time is exponentially distributed. Now, it is important to define the state and make sure we remember our definition. We will define the state as the number of *working* light bulbs, for which we get the following transition rate diagram. Note that we could just as well as defined the state as the number of *nonworking* bulbs, and the diagram would have been reversed.

The flow balance equations are given by:

$$2\mu p_0=\lambda p_1, \mu p_1=2\lambda p_2,$$

where $\lambda=1/22$/day and $\mu=0.5$/day. The normalizing equation here is

$$p_0+p_1+p_2=1.$$

Solving, we get

$$p_0=1/144, \quad p_1=22/144, \quad p_2=121/144.$$

Example 3: A Gas Station with Limited Space Potential customers arrive at a rate of 10 per hour according to a Poisson process, and service times are exponentially distributed with a mean of 3 minutes. The state of the system will be the number of cars at the station, so the state space is $\{0,1,2\}$, due to the limited capacity. Here, we assume that there is negligible time elapsed in going from the waiting space to the gas pump when a customer completes service. If this is not true, and the time can be modeled by an exponential distribution, then the state space must be expanded, with the single state "1" now divided into two states "(0,1)" and "(1,0)".

The flow balance equations are given by:

$$\lambda p_0=\mu p_1, \lambda p_1=\mu p_2,$$

where $\lambda=10$/hr and $\mu=20$/hr. The normalizing equation here is the same as in the previous example:

$$p_0+p_1+p_2=1.$$

Solving, we get

$$p_0=4/7, \quad p_1=2/7, \quad p_2=1/7.$$

Now consider adding a pump next to the former waiting space, so that there is no longer any waiting room for any cars, but there are two pumps for serving them. Assume that both spaces are accessible from the street. Verify that we have the following flow balance equations

$$\lambda p_0 = \mu p_1, \lambda p_1 = 2\mu p_2,$$

with the following steady-state probabilities:

$$p_0 = 8/13, \quad p_1 = 4/13, \quad p_2 = 1/13.$$

Thus, the gas station is empty 61.5% of the time.

Now, one can calculate the average number for the last two cases (for the wandering hippie, the average doesn't make any physical sense, unless we add costs as we did before), and for the last case, find the average waiting times by applying an important and useful result called Little's Law, which will be used extensively in the following section on queueing theory.

Example 2: Two Light Bulbs

$$E[N] = \sum_{n=0}^{2} np_n = (1)22/144 + (2)121/144 = 264/144 \approx 1.82.$$

Example 3: A Gas Station with Limited Space

$$E[N] = \sum_{n=0}^{2} np_n = (1)2/7 + (2)1/7 = 4/7 \approx 0.57.$$

In the next chapter, we will be able to show that the average time spent in the gas station by a customer that is served is given by

$$W = \frac{E[N]}{\lambda(1-p_2)} = \frac{4/7}{10/\text{hr}(6/7)} = 1/15\text{hr} = 4\text{min}.$$

Hence, the average waiting time W_q is 1 minute.

For the two-pump case, we have

$$E[N] = \sum_{n=0}^{2} np_n = (1)4/13 + (2)1/13 = 6/13 \approx 0.46.$$

$$W = \frac{E[N]}{\lambda(1-p_2)} = \frac{6/13}{10/\text{hr}(12/13)} = 0.05\text{hr} = 3\text{min}.$$

The last result is obvious, since if there are two pumps and no waiting space, a customer will either be served immediately or leave.

2 Queueing Theory

In this chapter, we will analyze queues, those irritating lines that we have all had to wait in. However, people are not the only entities on the earth that have to wait, or queue. Whenever there is a scarcity of resources that have to be allocated, queues form. Thus, queueing models are used to study such systems as computer/communications networks, manufacturing systems, airports, and many other man-made systems.

Example 1: Stamp Union Automatic Teller Machine (ATM).

Here, you wait in a first-come, first-served queue in order to use the ATM for services such as making deposits, checking balances, and withdrawing cash (most likely the latter, right?).

Example 2: Computer Room printer.

Here, each job you sent to the printer for printing is put into a printer queue, where it usually waits first-come, first-served, although special persons such as superusers can probably override this queue discipline to give priority to certain jobs. A generalization of this is the local area network (LAN) that connects client computers, servers, and peripherals in a building or on a campus.

Example 3: Airport.

At the airport, there are many queueing systems. There are queues at the ticket counter and queues at the gate

check-in. Even more important perhaps are the queues to use the runways (on the ground for take off, and in the air, albeit not a physical queue, for landing) and sometimes the unseen queues to use the gates. Other queueing systems include the curbside drop-off/pick-up lanes for passenger vehicle traffic and the baggage handling system.
Example 4: Manufacturing System.
Each part in the system has to undergo a series of manufacturing operations at various stations in the system. Queues are often physical buffers where parts are held in temporary storage while awaiting availability of the particular machine required to perform the operation at the station.
Example 5: Telecommunications Network.
Message packets — containing data, voice, and/or video information — are sent along the wide area network (WAN) via switches, at which there may be buffers for storing packets while awaiting sending. (In many parts of the systems, there are no buffers, in which case packets may be lost.)

2.1 Elements of a Queueing System

The three main elements of a queueing system are the following:

- customers;
- servers;
- queues.

In an ATM queueing "system," we have customers, ATM(s), and waiting room. In a manufacturing system, we have parts, machines, and buffers. In an airport runway, we have planes, runways, and waiting areas (both in the sky and on the ground). In a machine repair shop, we have machines (as customers this time!), repair persons, and waiting/storage room. In a communication network (either WAN or LAN), we have packets, switches, channels, and buffers.

2.2 Specification of a Queueing System

The main elements that specify a queueing system are the following:

- input sources (calling population): interarrival times, arrival process;
- queues: finite or infinite space;
  queue discipline, e.g., FCFS (FIFO), LCFS (LIFO), shortest processing time (SPT), random, priority, preemption (resume vs. non-resume); other customer behavior, such as jockeying, balking, reneging (impatient);
- service process: number of servers, service time characteristics.
- routing/topology, for queueing networks.

Some factors that must be considered in applying queueing theory, because they affect the complexity of the analysis:

- transient vs. steady state: we will consider only steady-state results, just as for CTMCs.
- entire distribution vs. means: for the number in system, we can derive the entire distribution $\{p_n\}$ for Markovian queues, but for waiting time, we usually only derive the mean via Little's Law (exception: $M/M/1$ queue).

Notation

In this section, we summarize the notation used throughout this chapter for a single queueing system. The first three quantities are **input parameters**, whereas the others are output **performance measures**.

$$\begin{aligned}
s &= \text{number of servers,}\\
\lambda_n &= \text{arrival rate when } N(t)=n,\\
\mu_n &= \text{service rate when } N(t)=n,\\
N(t) &= \text{number of customers in the system at time } t,\\
p_n(t) &= P(N(t)=n), \text{given } N(0)=0,
\end{aligned}$$

W_i = time in system of ith customer,
$(W_q)_i$ = time in queue of ith customer,
N = number of customers in the system in steady state,
p_n = $P(N = n)$,
L = $E[N]$ = steady-state mean number of customers in the system ,
L_q = steady-state mean number of customers in queue ,
q_n = number of customers in the system found by an *arriving* customer in steady state,
$\mathcal{W}$ = r.v. steady-state time in system,
W = $E[\mathcal{W}]$ = mean steady-state time in system,
$\mathcal{W}_q$ = r.v. steady-state time in queue,
W_q = $E[\mathcal{W}_q]$ = mean steady-state time in queue,
ρ = system utilization.

When the input parameters λ_n and μ_n have no subscript, they usually denotes a specific rate, oftentimes (but not always) when they are independent of n. In the case of queueing networks, the subscripts will usually index a station in the network. Performance measure with the argument t or n removed indicates the corresponding steady-state (or long-run) quantity. For Markovian systems, we will use CTMC models to find p_n, which represents the probability that there are n customers in the system, or equivalently the (long-run) proportion of time there are n customers in the system. When we come to queueing networks, we will have to refer to vector states, i.e., instead of n, a state would be represented by $\mathbf{n} = (n_1, ..., n_M)$, where M is the number of stations in the network. This is similar to the state $\mathbf{X}$ in the reliability network, where each X_i represented the state of the corresponding component.
Caution: the term "waiting time" is often used in the research literature to denote what we call the system time; similarly, the "queue length" is often used to denote what we call the number in the system and not just the number in queue.

2.3 Kendall Notation

We will be using the following notation to represent single-station queues:

$$\cdot\ /\ \cdot\ /\ \cdot\ /\ \cdot\ /\ \cdot$$

1. The first position represents the arrival process (interarrival time distribution).
2. The second position represents the service time distribution.
3. The third position represents the number of servers.
4. The fourth position represents the space in the system, which includes queue plus spaces at the servers; if this is omitted, then the space is assumed unlimited (∞).
5. The fifth position represents the population of the system, i.e., will the system "run out" after a certain number of customers have been served; if this is omitted, then the population is assumed unlimited (∞).

Thus, there will always be at least the first three indicators used in the notation. The last three indicators are positive integers, whereas the first two are letters representing distributions. The distributions we will use will be the following:

M Exponential (Markovian) distribution,

U Uniform distribution,

D Deterministic, which means that the time is not a random variable, but a known number,

E_k k-Erlang distribution, where recall that the Erlang distribution has the following p.d.f.:

$$f(t) = \frac{(\mu k)^k}{(k-1)!} t^{k-1} e^{-k\mu t},$$

with mean $1/\mu$ and variance $1/(k\mu^2)$.

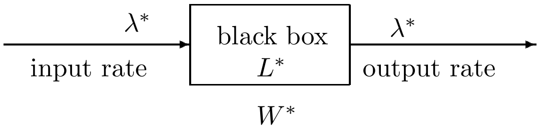

Figure 1: Little's Law: $L^* = \lambda^* W^*$.

G General distribution, which means that we have not specified any particular distribution.

Examples.

$M/M/1$ single-server queue with exponential interarrival times (Poisson arrival process) and exponential service times, no limit on queue length or customer population.

$G/G/1$ single-server queue with general interarrival times and general service times, no limit on queue length or customer population.

$M/G/1$ single-server queue with Poisson arrival process and general service times, no limit on queue length or customer population.

$D/U/2$ two-server queue with deterministic arrival process and uniformly distributed service times, no limit on queue length or customer population.

$M/G/s/s$ s-server queue with Poisson arrival process and general service times, no queueing (all customers in service, else turned away) and no limit on customer population.

$M/M/s/c/c$ ($c \geq s$) s-server queue with Poisson arrival process and exponential service times, and system size and customer population size of c.

2.4 Little's Law

Little's Law is a very simple concept, but also very powerful. Draw a "black box" around the portion of the system in which interest lies (which could be the entire system), and let λ^* be the input (arrival) rate into the black box, L^* be the average number in the black box, and W^* be the average time spent in the black box. Little's Law states the following:

If the system is stable, then

- The output (departure) rate out of the black box is equal to the input (arrival) rate λ^*.
- $L^* = \lambda^* W^*$.

Important Note: Little's Law requires neither assumptions on Poisson processes nor on exponentially distributed service times. Thus, it is a very general result that is independent of distributions!

The key to applying Little's Law is in making sure one defines the quantities λ^*, L^*, W^* correctly, especially the arrival rate that actually *enters* the black box, not the nominal arrival rate (which might include customers routed elsewhere).

Example: For an entire system, we have the black box being the system, and λ being the total rate into the system, so

$$L = \lambda W,$$

which is the way Little's Law is usually summarized. Applied to just the queue portion, we have

$$L_q = \lambda W_q.$$

Applied to just the service portion, we have

$$L_s = \lambda E[X],$$

where L_s represents the average number in service. (Clearly, by definition, $L = L_s + L_q$.)

The Arrival Rate

The application of Little's Law requires careful consideration of the arrival rate, which may have a few complications such as the following:

- the "nominal" arrival rate is not the rate that actually enters the system, because there are rejections, balkings, etc.;
- the arrival rate is state dependent, e.g., in a finite customer population, it is proportional to the number of outstanding customers.

In general, the average arrival rate is given by

$$\overline{\lambda} = \sum_{\text{all} i} \lambda_i p_i.$$

Caution: As discussed earlier, the λ_i here is the rate in state i; in networks, we use the same notation to indicate the total arrival rate into station i. The meaning should be clear from the context, but do not be confused by the dual usage of the notation.

From a sample path perspective, the arrival rate can be found by

$$\lambda = \lim_{t\to\infty} \frac{N_a(t)}{t},$$

where $N_a(t)$ is the number of arrivals by time t.

Example: ATM machine.

Student arriving at rate of 5 per minute (not necessarily Poisson process!), and average number there (waiting for or using machine) is 6 persons. What is the average total time spent there (waiting and transactions)?

$$\lambda = 5/\text{min}, \quad L = 6 \Longrightarrow W = \frac{L}{\lambda} = 1.2\text{min}.$$

Example: G/G/1 Queue.

Let λ be the arrival rate and μ the service rate (not necessarily of exponential distributions!), and define $\rho = \lambda/\mu$, which is the utilization and offered load. Then, we have the usual relationships: $L = \lambda W$ and $L_q = \lambda W_q$. Since $W = W_q + 1/\mu$, we can also apply Little's Law to obtain $L = \lambda(W_q + 1/\mu) = L_q + \rho$, i.e., $L_s = \rho$. Note in particular that $L \neq L_q + 1$.

Example: Motor Vehicle Administration.

In this case, there is a network of stations, and Little's Law can be applied to the entire system or just portions of it, as desired. A simple example is a sequence of just two stations that every customer must go through. (This is known as a tandem queueing network.) If the arrival rate is $\lambda = 1/\text{min}$ (and there are no discouraged customers who leave) and one observes 10 customers at the first station (total in service and in queue) and 20 customers at the second station, the estimated time an entering customer would expect to spend at each station and in the MVA total can be easily found:

$$W_1 = L_1/\lambda = 10\text{min}, \quad W_2 = L_2/\lambda = 20\text{min}.$$

Since everyone goes through both station 1 and station 2, then we can simply sum the individual station system times to get the total system time:

$$W = W_1 + W_2 = 30\text{min}.$$

This is not the case for general queueing networks, where system times do NOT necessarily sum, but the number in system is summable, i.e.,

$$W \neq W_1 + W_2 + ... + W_M, \quad L = L_1 + L_2 + ... + L_M,$$

where there are M stations in the system. In this case, one applies Little's Law to L:

$$W = L/\lambda = (L_1 + L_2)/\lambda.$$

For example, let's modify the case to where there is also an arrival stream that just goes to station 2 and leaves, with the same arrival rate of 1/min. Then, defining λ_i as the total arrival rate to station i, we have

$$\lambda_1 = 1/\text{min}, \ \lambda_2 = 2/\text{min}, \quad W_1 = 10\text{min}, \ W_2 = 10\text{min}, \quad W = 30/(2/\text{min}) = 15\text{min} \neq W_1 + W_2,$$

where the total arrival rate to the system is 2/min. Again, note that there are *no distributional assumptions!*

Example: G/G/s Queue with discouraged arrivals.

If q denotes the probability that an arriving customer departs before reaching service, then the arrival rate into service is reduced to $\lambda(1 - q)$. How Little's Law is applied then depends on whether the "zero service time" discouraged customers are to be counted in the overall statistics.

Service Completion Rate

The output (departure) rate is usually the sum of the various service completion rates at the stations in a system. Viewing each station separately, stability ensures that the arrival rate to a station is equal to the service completion rate. If p_n is the probability that there are n at the station, and μ_n is the service rate when there are n at the station, then the service completion rate is given by

$$\sum_{n \geq 1} \mu_n p_n.$$

If the service rate is independent of the number of customers at the station, as in a single-server queue, then we have a simplification:

$$\sum_{n \geq 1} \mu p_n = \mu(1 - p_0).$$

Since for a stable system, the arrival rate must equal the service completion rate, we thus have the following general result for a single-server queue:

$$p_0 = 1 - \lambda/\mu = 1 - \rho,$$

without need for exponential assumptions on either the arrivals or departures.

Sketch of Proof for Little's Law

A rough justification of Little's Law can be derived by viewing two different sample path (stochastic) processes: the system time and the number in system. The former is a discrete-time, continuous state space stochastic process, whereas the latter is a continuous-time, discrete state space process. We consider a single station of servers (e.g., a single-server queue). The key result to note is that at the end of a busy period — defined as a point when the system empties, the number of departures $N_d(t)$ is equal to the number of arrivals $N_a(t)$, and we have the following relationship:

$$\sum_{i=1}^{N_a(t)} W_i = \sum_{i=1}^{N_d(t)} W_i = \int_0^t N(u)du,$$

i.e., the area under the curve of $N(t)$ is exactly equal to the sum of the system times of the customers served. Thus, in general, we have the relationship:

$$\sum_{i=1}^{N_d(t)} W_i \leq \int_0^t N(u)du \leq \sum_{i=1}^{N_a(t)} W_i,$$

which we can rewrite as

$$\frac{1}{t}\int_0^t N(u)du = \frac{N_a(t)}{t}\frac{1}{N_a(t)}\sum_{i=1}^{N_a(t)} W_i + \frac{e(t)}{t},$$

where the "error term" $e(t)$ represents the difference between the sum of the system times and the area under the curve, and is hence equal to 0 whenever the system is empty. Now, if we take the limit as $t \to \infty$, then assuming that the "error term" is well behaved (numerator bounded and returns to zero often enough), we get Little's Law:

$$L = \lambda W.$$

2.5 PASTA

PASTA stands for Poisson Arrivals See Time Averages, and it was coined not by someone with Italian origin, but by someone with a German name (Wolff). The idea is very simple, once we understand the difference between a time average and a customer average. PASTA just says that if the customers follow a Poisson arrival process, then these two quantities are the same. The idea is that Poisson arrivals take a "random" look at the system.

First, we begin with a very simple example where the two quantities are not the same, and are in fact very different. This is because everything will be deterministic, so intuitively arrivals are not at all taking a random look at the system. This suggests that practically speaking, you can often improve the performance of your system by reducing randomness.

Example: D/D/1 Queue Consider a single-server FCFS queue where the interarrival times are 10 and the service times are 9. Then, in the long run, the system has one in the system 90% of the time, and is empty the other 10% of the time. Thus, by our definition, we have

$$p_0 = 0.10, p_1 = 0.90, p_n = 0, n > 1.$$

However, it is also clear that every customer arrives to an empty system, so that we have

$$q_0 = 1, q_n = 0, n > 0.$$

Since $p_0 \neq q_0$ and $p_1 \neq q_1$, PASTA does not hold for this system.
Example: M/M/1 Queue Soon, we will solve the $M/M/1$ queue by using a CTMC model. We will find all the steady-state probabilities for number in system, from which we can determine other performance measures of interest. But first, we derive the average time in queue directly by applying three important concepts: conditional expectation, PASTA, and Little's Law. In addition, we apply the memoryless property of the exponential distribution. Let us define λ as the arrival rate, μ as the service rate (hence mean service time $1/\mu$), $\mathcal{L}_q^*$ as the number found in the system by an arriving customer, and S_r^* as the remaining service time for the customer in service found by an arriving customer. Then by conditioning on the number found in the system by an arriving customer, $\mathcal{L}_q^*$, we have

$$W_q = E[S_r^*] + E[\mathcal{L}_q^*]/\mu,$$

i.e., a customer's wait consists of the wait for the customer in service to finish (if any) plus all the customers in the queue. First, we compute the expected remaining service time of a customer by conditioning on the status of the server. Since service times are exponential, by the memoryless property, the remaining service time found by an arriving customer is a full service time if the server is busy and 0 otherwise. The probability that the server is busy is by PASTA just equal to $1 - p_0 = \rho$, so we have

$$E[S_r^*] = \rho/\mu.$$

Applying PASTA and then Little's Law, we have

$$E[\mathcal{L}_q^*] = L_q = \lambda W_q.$$

Substituting, we have

$$W_q = \rho/\mu + L_q/\mu = \rho/\mu + \lambda W_q/\mu,$$

$$\text{hence} \quad W_q = \frac{\rho/\mu}{1-\rho}.$$

This formula will be derived again later in the chapter using a CTMC model, but this particular derivation is interesting in its own right, because it brought together three very important results/techniques that we have learned:

- conditional expectation;
- PASTA;
- Little's Law.

More on Departures and Arrivals

When we discussed Little's Law, we said that if the system is stable, then the departure rate is equal to the arrival rate. In fact, we have an even stronger result. Let r_n denote the (steady-state) probability that a departure leaves n customers in the system. Then, the entire departure *distribution* is equal to the arrival distribution.
Caution: This result does *not* mean that the departure *process* is the same as the arrival *process*.

Proposition. In a stable birth-death queue,

$$q_n = r_n \text{ for all } n.$$

A birth-death queue is a queueing system that can be modeled by a birth-death process (a special type of CTMC). We will discuss this further shortly.

The justification for this result is somewhat similar to the argument used to establish Little's Law. Basically the number of transitions from n to $n+1$ gives the number of times that an arrival sees n in the system, whereas the number of transitions from $n+1$ to n gives the number of times that a departure leaves n in the system. If we define these respective quantities by $N_n^a(t)$ and $N_n^d(t)$, where

$$q_n = \lim_{t\to\infty} \frac{N_n^a(t)}{t}, \quad r_n = \lim_{t\to\infty} \frac{N_n^d(t)}{t},$$

then since these quantities can never differ by more than 1:

$$|N_n^a(t) - N_n^d(t)| \leq 1.$$

Dividing by t and taking the limit establishes the proposition, since the righthand side goes to zero in the limit.

2.6 A Little Bit of Renewal Theory

An important branch of applied probability that has practical applications is renewal theory. We will just summarize one result that is extremely useful, called the regenerative theorem. First, we define informally the idea of regeneration. A process is said to be regenerative with regenerative points if at these regenerative points the process looks probabilistically the same. A regenerative cycle is the period of time between regenerative points. An example we will see is when a single-server queue empties.

Regenerative Theorem. Let $\{X(t)\}$ be a regenerative process with regenerative (i.i.d.) cycle lengths $\{T_i\}$, and f a performance measure defined on the state space of $\{X(t)\}$. Then the long-run average can be expressed as the ratio:

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T f(X(t))dt = \frac{E\left[\int_0^{T_1} f(X(t))dt\right]}{E[T_1]},$$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} f(X_i) = \frac{E\left[\sum_{i=1}^{T_1} f(X_i)\right]}{E[T_1]},$$

corresponding to continuous time or discrete time $\{X_i\}$, respectively.

Example: $G/G/s$ Queues

Let B_i be the length of the ith busy period and η_i the corresponding number of customers served in the busy period. Then, the following holds quite generally for a stable system:

$$L = \frac{E\left[\int_0^{B_1} N(t)dt\right]}{E[B_1]},$$

$$W = \frac{E\left[\sum_{i=1}^{\eta_1} W_i\right]}{E[\eta_1]}.$$

Note that aside from being distribution independent, this results holds for most service disciplines, not just FCFS.

Example: Alternating Renewal Process

In this type of process, there are two types (or sets) of states: say A and B. Then, let p_A be the long-run probability of being in state A, and let p_B be the long-run probability of being in state B.

For example, in a single-server queue, set A could correspond to all the busy states, and set B could correspond to the single idle state. Then, we have for an $M/G/1$ queue:

$$p_0 = \frac{E[T_A]}{E[T_A]+E[T_B]} = \frac{1/\lambda}{1/\lambda + E[B]},$$

where $E[T_A] = 1/\lambda$ by the memoryless property of the exponential distribution.

2.7 Birth-Death Queues

A birth-death queue is simply a queueing system that can be represented by a birth-death process, where births correspond to arrivals, deaths correspond to departures, and $N(t)$ is the state at time t, representing the number of customers in the system:

- population ↔ customers in system
- birth ↔ arrival
- death ↔ departure
- birth rates ↔ arrival rate
  (possibly dependent on customer population)
- death rates ↔ service rates
  (dependent on number of servers)

Since a birth-death process is a special type of CTMC, in which transitions only to neighboring states, we can use the usual CTMC technique of flow balance and normalization to solve for the steady-state probabilities. Thus, the process for analyzing a birth-death queue is as follows:

1. Define the state of the system.
2. Determine the state-transition rate diagram or matrix.
3. Use flow balance or $\mathbf{pQ} = 0$ PLUS normalization (probabilities must sum to 1) to solve for all the probabilities $p_0, p_1, ...$
4. Express the performance measures that you can in terms of $p_0, p_1, ...$
5. Use Little's Law and/or PASTA to determine other performance measures

Example: $M/M/1$ **queue**.

1. The state of the system is the number in the system.
2. λ from state i to $i+1$ and μ from state i to $i-1$, where λ is the arrival rate and μ is the service rate.
3. Flow balance or $\mathbf{pQ} = 0$ gives $p_n = \rho^n p_0, \quad \rho = \lambda/\mu$;
   Normalization gives $p_0 = 1 - \rho$ if $\rho < 1$.
4. $L = \sum_{i=0}^{\infty} n p_n$.
5. Little's Law and other relationships can be used to find W, W_q, L_q.

$$L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}, \quad W = \frac{E[X]}{1-\rho} = \frac{1}{\mu-\lambda}, \quad W_q = W - E[X] = \frac{\rho E[X]}{1-\rho} = \frac{\rho}{\mu-\lambda}, \quad L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda\rho}{\mu-\lambda} = \rho L.$$

Note the stability requirement:

$$\rho < 1 \text{ or } \lambda < \mu.$$

Numerical Example At the single ATM, arrivals follow a Poisson process at rate 10 per hour, service times exponentially distributed with mean of 4 minutes.

$$\lambda = 10/\text{hr}, \mu = 15/\text{hr}.$$

Thus, $\rho = 2/3$, and so we have

$$L = 2, L_q = 4/3, W_q = 2/15\text{hr} = 8\text{min}, W = 12\text{min}.$$

If the arrival rate increase by 20%, rework the problem.

$$\lambda = 12/\text{hr}, \mu = 15/\text{hr}.$$

Thus, $\rho = 4/5$, and so we have

$$L = 4, L_q = 16/5 = 3.2, W_q = 4/15\text{hr} = 16\text{min}, W = 1/3\text{hr} = 20\text{min}.$$

The average time in system and number in system have doubled!
In numerical examples, be sure to keep your units straight by carrying them along for the ride! For example, be consistent in using hours or minutes throughout (until the very end, if you like to convert back), in order to get ρ correct. Also, interpret rate versus mean correctly.

Distribution of Waiting Time in $M/M/1$ Queue

By PASTA, we know that

$$P(\mathcal{W}_q = 0) = p_0 = 1 - \rho$$

Assuming $t > 0$, we note that

$$P(\mathcal{W}_q > t) = P(\mathcal{W}_q > t | \mathcal{W}_q > 0)P(\mathcal{W}_q > 0) + P(\mathcal{W}_q > t | \mathcal{W}_q = 0)P(\mathcal{W}_q = 0).$$

The distribution of system time can be derived by conditioning on the number of customers found upon arrival by a customer, q_n, and applying PASTA to use p_n instead. Then, the probability reduces to the probability of the sum of the exponential service times:

$$P(\mathcal{W} > t) = \sum_{n=0}^{\infty} p_n P(\sum_{i=1}^{n+1} X_i > t).$$

The algebra is rather messy, and so we skip directly to the results:

$$\begin{aligned} P(\mathcal{W} > t) &= e^{-\mu(1-\rho)t}, \\ P(\mathcal{W}_q > t) &= \rho e^{-\mu(1-\rho)t}. \end{aligned}$$

Thus, the system time random variable also has an exponential distribution (with mean $\mu - \lambda$)! This is quite a remarkable and unexpected result.

Numerical Example At the single ATM, arrivals follow a Poisson process at rate 10 per hour, service times exponentially distributed with mean of 4 minutes. Find the following quantities:

1. percentage of time the ATM idle;
2. average number of persons at the ATM;
3. average time spent waiting in line;
4. average number of persons served per hour;
5. probability you will spent over 10 minutes at the ATM (waiting and transactions; as you are in a rush to get back to class!).

The first, and in many ways most important, step is to be able to "translate" the above requirements into the respective mathematical quantities to be found: $p_0; L; W_q; \lambda; P(\mathcal{W} > 1/6)$. Then, as before, we have $\lambda = 10\text{hr}, \mu = 15/\text{hr}, \rho = 2/3$, and

1. $p_0 = 1 - \rho = 1/3$;
2. $L = 2$;
3. $W_q = 8\text{min}$;
4. $\lambda = 10\text{hr}$;
5. $P(\mathcal{W} > 1/6\text{hr}) = e^{(15-10)/6} \approx 0.435$;

where the last result follows from the fact that the system time is exponentially distributed with rate $(\mu - \lambda)$.

2.7.1 $M/M/1/c$ Queue

Now consider a single-server queue with a limited capacity of c spaces in the system, where the arrival process is Poisson with rate λ and the service times are i.i.d. exponentially distributed with rate μ. As a birth-death queue, the analysis is identical to the $M/M/1$ queue, except that the state space is now finite instead of infinite. The flow balance equations are unchanged, but the normalization is over a finite sum instead of an infinite one.

$$p_n = (\lambda/\mu)p_0, \sum_{n=0}^{c} p_n = 1.$$

Solving, yields

$$p_0 = \frac{1-\rho}{1-\rho^{c+1}}, \text{where}\rho = \lambda/\mu, \quad p_n = \rho^n p_0.$$

$$L = \sum_{n=0}^{c} np_n = \frac{\rho[1-(c+1)\rho^c + c\rho^{c+1}]}{(1-\rho^{c+1})(1-\rho)}, \quad L_s = 1 - p_0, \quad L_q = L - L_s.$$

Note that since the system has limited capacity, there is no problem with stability (since customers that find the system full are assumed to leave). In fact, a special case is the one in which $\lambda = \mu$, in which we have that each state is equally likely, so that the average number in system is half full:

$$p_n = \frac{1}{c+1}, \quad L = \frac{c}{2}.$$

Also, assuming that W and W_q refer only to customers that actually enter the system, we do not have the usual version of Little's Law, i.e.,

$$L \neq \lambda W,$$

since λ is the *nominal* rate of customers, as some customers do not actually enter the system if the system is full. By PASTA, we know that the rate of customers not entering the system is given by $\lambda q_c = \lambda p_c$, and so we can apply Little's Law for the rate given by $\lambda(1-p_c)$ to get

$$W = \frac{L}{\lambda(1-p_c)}, W_q = \frac{L_q}{\lambda(1-p_c)}.$$

Example Consider a one-man barber shop with 10 seats (including the cutting seat). Assume potential customers arrive according to a Poisson process on the average every 3 minutes, and the barber's haircut times are exponentially distributed with an average time of 12 minutes. Find the following (long-run average):

1. the number of haircuts given by the barber per hour;

2. the time spent in shop by a customer;

3. percentage of time the barber is busy;

4. the probability that an arriving customer will have to wait;

5. the probability that an arriving customer will leave without receiving a haircut.

We have

$$\lambda = 20\text{hr}, \mu = 5/\text{hr}, \rho = 4.$$

Notice that ρ is much greater than 1, which indicates that the barber is very busy indeed, and we would expect a utilization of near 100%. We have

$$p_0 = \frac{1-4}{1-4^{11}}, \quad p_{10} = 4^{10} p_0 = 0.75,$$

and thus

1. $\lambda^* = \lambda(1-p_c) = (20/\text{hr})(0.25) = 5/\text{hr}$;

2. $L = 92/3; W = \frac{L}{\lambda^*} = 1.93\text{hrs}$;

3. $1 - p_0 = 99.99992\%$;

4. same as previous question: $1 - p_0 = 0.9999992$;

5. $p_c = 0.75$.

2.7.2 $M/M/s$ Queue

Now consider a single queue with s servers (like in a bank and most airline counters these days), where the arrival process is Poisson with rate λ and the service times are i.i.d. exponentially distributed with rate μ, for any of the servers. As a birth-death queue, we have the following:

$$\lambda_n = \lambda, \mu_n = n\mu, n \leq s; \mu_n = s\mu, n \geq s.$$

Flow balance and normalization lead to the following result:

$$p_0 = \left(\sum_{n=0}^{s-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!(1-\lambda/\mu)}, \right)^{-1}$$

$$p_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} p_0 & \text{if } 0 \leq n \leq s; \\ \frac{(\lambda/\mu)^n}{s! s^{n-s}} p_0 & \text{if } n \geq s. \end{cases}$$

$$L_q = \frac{p_0 (s\rho)^2 \rho}{s!(1-\rho)^2}, \quad \rho = \frac{\lambda}{s\mu}.$$

Note the stability requirement:

$$\rho < 1 \text{ or } \lambda < s\mu.$$

Specific Case: $M/M/2$ queue, where $\rho = \frac{\lambda}{2\mu}$.

$$p_0 = \frac{1-\rho}{1+\rho}, \quad p_n = 2\rho^n p_0, \quad L = \frac{2\rho}{1-\rho^2}, \quad W = \frac{E[X]}{1-\rho^2}.$$

For $M/M/3$ queue,

$$p_0 = \frac{1-\rho}{1+2\rho+1.5\rho^2}, \quad \rho = \frac{\lambda}{3\mu}.$$

Example 2 ATMs, Poisson arrival process at rate 80 per hr, exponentially distributed service times with mean 1.2 minutes.

$$\lambda = 80/\text{hr}, \quad \mu = 50/\text{hr}, \quad \rho = 0.8.$$

1. expected number in system: $L \approx 4.44$.
2. expected time in system: $W \approx 3.33$ min.
3. % time at least one ATM idle: $p_0 + p_1 \approx 0.29$.

2.7.3 Machine Repair Model: $M/M/s/c/c$ Queue

This is a finite-source, finite capacity, multi-server system. The interpretation for this model is that there are a set of machines (c of them) that occasionally break down and a set of repair workers (s of them) that repair them. Usually, it is assumed that $s < c$; otherwise, there is never any queue, and the problem is almost trivial. Thus, in this model, the machines are the customers and the people are the servers (as opposed to the ATM model, where the reverse was true). To set this up as a birth-death queue, we define the state as the number of broken machines in the system. This then corresponds to the number of machines in repair. Note that one could also equivalently define the state of the system as the number of working machines.

$$\begin{aligned} c &= \text{\# machines}, \\ s &= \text{\# repair persons}, \\ \lambda &= \text{breakdown rate of } \textbf{single} \text{ machine}, \\ \mu &= \text{repair rate for each service person.} \end{aligned}$$

Note that λ is defined individually now, and not over the entire population, as before. This is usually the case when a finite customer population is considered, and it is analogous to the rate μ for the service rate.

In terms of the birth-death process model, we have

$$\mu_n = \begin{cases} n\mu & \text{if } n \leq s; \\ s\mu & \text{if } n \geq s. \end{cases} \quad \lambda_n = (c-n)\lambda.$$

Note that the "birth" rate is defined on the number of good machines $(c-n)$, i.e., there are customers generated when a breakdown occurs. Letting $\rho = \lambda/\mu$, we find

$$p_n = \begin{cases} \left(\begin{array}{c} c \\ n \end{array}\right) \rho^n p_0 & \text{if } n \le s; \\ \left(\begin{array}{c} c \\ n \end{array}\right) \frac{\rho^n n!}{s! s^{n-s}} p_0 & \text{if } n \ge s. \end{cases}$$

where normalization

$$\sum_{n=0}^{c} p_n = 1$$

must be used to solve for p_0 as usual. The interpretation of L, L_q, W, W_q are as the expected (or average) number of broken down machines, number of machines waiting for repair, time a machine spends broken, time a machine spends waiting for repair. The first two can be found in the usual way via

$$L = \sum_{n=0}^{c} n p_n, \quad L = \sum_{n=s}^{c} (n-s) p_n,$$

and the second two can be found by applying Little's Law, $W = L/\lambda^*$, $W_q = L_q/\lambda^*$, where the average breakdown rate is given by

$$\lambda^* = \sum_{n=0}^{c} p_n \lambda_n = \sum_{n=0}^{c} p_n (c-n)\lambda = \lambda(c-L).$$

This rate could also be "derived" by common sense, since $(c-L)$ is the average number of good machines. **Example**: Consider a system of 5 machines and 2 repair workers. A machine breaks down every 30 days on average, exponentially distributed, and repair times are exponentially distributed with mean 3 days.

$$c = 5, \ s = 2, \ \lambda = 1/30/\text{day}, \ \mu = 1/3/\text{day}.$$

Find the following (long-run average):

1. average number of working machines;
2. average downtime for a broken machine;
3. % time one of the repair persons is idle.

In terms of our defined notation, we seek the quantities $c - L; W = L/\lambda^*; p_0 + p_1$. First we write down flow balance as usual:

$$p_1 = 0.5p_0, \quad p_2 = 0.1p_0, \quad p_3 = 0.015p_0, \quad p_4 = 0.0015p_0, \quad p_5 = 0.000075p_0.$$

Normalization is $p_0 + p_1 + p_2 + p_3 + p_4 + p_5 = 1$, and solving gives

$$p_0 \approx 0.619 \Longrightarrow L \approx 0.465.$$

1. $c - L \approx 4.535$;
2. $W = L/\lambda^* = \frac{L}{\lambda(c-L)} \approx 3.08$;
3. $p_0 + p_1 \approx 0.929$.

2.7.4 $M/G/\infty$ Queue

This is a system where there are an infinite number of servers, so that there is never a wait for service. In practical applications, this model is used for systems where there are ample number of servers, so that the likelihood of waiting is negligible. For the special case where the service times are exponentially distributed, we have a birth-death queue. If we denote λ as the arrival rate and μ as the service rate for each server, then the birth-death parameters are given by

$$\lambda_n = \lambda, \quad \mu_n = n\mu,$$

i.e., it is the same as the multi-server case, except that there is no upper limit on the service rate. Because of the latter, this system is always stable. We define

$$\rho = \lambda/\mu,$$

as in the single-server queue (not the multi-server queue!), and flow balance gives

$$p_n = \frac{\rho^n}{n!}p_0,$$

which with normalization applied (an infinite sum), we find

$$p_n = \frac{\rho^n}{n!}e^{-\rho},$$

i.e., the number in system has a Poisson distribution!

What is just as amazing is that the same distribution is true even when the service time distribution is not exponential, i.e., the equation above holds even for an $M/G/\infty$ queue. What this means is that for an $M/G/\infty$ queue, the distribution of the number in system depends only on the mean of the service time distribution and not on any other higher moments! This is truly an incredible result at first glance. With some advanced probability theory (beyond the scope of this text), however, it turns out that it is not that difficult to show this result. This is called an **insensitivity** result, for obvious reasons. Insensitivity also holds for $M/G/s/s$ queues, i.e., one can solve for the distribution of number in system in this queue by simply solving the Markov chain associated with the $M/M/s/s$ queue having the same service time mean. The discovery of this result, by Erlang near the beginning of the last century, was a seminal contribution, because telephone trunks could be modeled using this queue to determine the appropriate number of lines needed to obtain a desired service level, usually called the quality of service (QoS). As a result, the formulas associated with this system are usually referred to as Erlang's loss formula (loss, because it is a loss sytem, whereby customers are lost if they arrive to the system filled to capacity). This system will be discussed shortly.

2.7.5 $G/G/\infty$ Queue

Denoting λ as the arrival rate and μ as the service rate for each server, we see that as before there is no wait, and by common sense, we know that

$$\mathcal{W}_{\rm II} = 0, \quad \mathcal{W} = X \Longrightarrow W_q = 0, \quad W = E[X] = 1/\mu,$$

and by Little's Law, we have

$$L_q = 0, \quad L = \lambda E[X] = \lambda/\mu = \rho.$$

Note that here we only have the means and not the entire distribution for the number in system, as we had in the previous case. Of course, the waiting time distribution is just the service time distribution.

Numerical Example: If about two babies are born every second, and their average lifetime is 70 years, find the expected (or average) population in steady state?

Modeling the world as a $G/G/\infty$ queue, we find

$$L = \lambda/\mu = 2/\text{sec}(70\text{yrs})(365\text{days/yrs})(24\text{hrs/days})(3600\text{sec/days}) \approx 4.4\text{billion}.$$

2.7.6 Erlang's Loss Formula: $M/G/c/c$ Queue

Erlang's loss formula is one of the earliest results in queueing theory, and stemmed from the Danish mathematician Erlang's work on analyzing telephone congestion at the beginning of this century. This work formed the basis and impetus for all the queueing theory research that followed, and this line of research continues today in the research of ATM (asynchronous transfer mode) telecommunication networks. Of course, now other types of queueing systems (airports, computers, manufacturing systems, and service operations such as post offices, banks, etc.) also motivate research in queueing theory, as well. Note that these systems are all man-made, and not a product of mother nature.

The $M/G/c/c$ queue is a system in which the arrivals follow a Poisson process, but the service times can follow a general distribution. There are c servers and no space for queueing. Think of a set of telephone (called trunk) lines. When you pick up a phone, you can make a call if a line is available; otherwise, you must hang up and try again later. Contrast this with many customer service "support help" lines (call centers) these days, which is after your call gets through, and there is no service person available to help you. In this case, you are placed on hold, in a queue, so the system is more like a $M/G/c$ queue.

If the service times are in fact exponentially distributed, then we can solve this system as a birth-death queue, with

$$\lambda_n = \lambda, \quad \mu_n = n\mu, \quad \rho = \lambda/\mu,$$

i.e., the same parameters as in the $M/G/\infty$ queue, the only difference being that the state space is finite, truncated at c (note that ρ is defined differently from the $M/M/s$ queue). With flow balance and normalization (over a finite number of states), we find

$$p_n = \frac{\rho^n/n!}{\sum_{i=0}^{c} \rho^i/i!}.$$

Again, what is remarkable about this formula is that the distribution derived using CTMC analysis for the $M/M/c/c$ queue is valid for any service time distribution, where in general, $\rho = \lambda E[X]$. In other words, the system distribution depends only on the mean of the service time distribution, and is insensitive to anything else about the service time distribution. This "insensitivity" result is the same as that observed for the $M/G/\infty$ queue. Lest the reader generalize unwisely, we should note that this insensitivity result does not hold for the $M/G/1$ queue (nor for the $M/G/s$ queue when $s > 1$).

2.8 Decision Making

Now that we can handle multiple servers in a queueing system, we consider the application of some queueing theory results to decision making. Again, the idea is simply to apply the Law of the Unconscious Statistician to calculate certain quantities of interest which can then be used to aid decision-making. In this section, we will chiefly be applying our queueing results thus far to "inventory cost" comparisons. In a communications networks or a manufacturing system, there may be some cost to either excess delay or inventory in the system. If we can calculate these quantities, we can decide whether we need to add new equipment or reconfigure the system or perform some other type of change on the system.

The most common forms of decisions involve the following types of choices:

- which type of server to use;
- how many servers to use;
- which configuration to use.

For the last item, an example is whether to use a single line or multiple parallel lines. Sometimes the decision is dictated by physical considerations. For example, in a highway toll booth it would be difficult to implement a single parallel line, even though as we said earlier that is the fairest system, in the sense of minimizing the variance of a customer's waiting time.

In terms of our models, the above choices translate into decisions on the values of the following parameters: μ, s, λ, p_{ij}. In order to make these decisions, costs must be attached to the servers and to the customers waiting time or number in system (usually called as inventory when the "customers" are nonhuman).

Example: How many servers?

- cost per server;
- cost for waiting time or inventory;

Computer repair:
$70 per day for each repairer;
$100 per day for each computer out of service.

Analysis:

- system is repair shop;
- servers are repairers;
- "customers" are computers;
- cost on number of computers in repair (out of service).

Cost function:

$$G(s) = 70s + 100L(s).$$

Find s to minimize the cost function.
Extension to state-dependent costs:

- no cost for one computer out of service;
- \$50 per day for two computers out of service;
- \$60 per day for *each additional* computer out of service (after two).

Modified cost function:

$$G(s) = 70s + \sum_{i\geq 2}(50 + (i-2)60)p_i.$$

Example: Which computer?

- cost per computer;
- cost for waiting time or inventory;

Computer choices:
\$70 per hour for TYD, which has speed of 10 jobs per hour;
\$100 per hour for VMH, which has speed of 15 jobs per hour.
Waiting cost:
\$5 per hour in system for a job.
Analysis:

- system is computer system;
- servers are computers;
- "customers" are jobs;
- cost on system time of jobs in system.

Cost functions:

$$G_1 = 70 + 5L_1,$$

$$G_2 = 100 + 5L_2.$$

Choose the machine that yields the lowest average cost.

General form of cost function:

$$\frac{\text{expected cost}}{\text{time unit}} = \frac{\text{service cost}}{\text{time unit}} + \frac{\text{expected “inventory” cost}}{\text{time unit}},$$

where the latter usually is the product of L and a per time unit cost. Even when the costs are given in terms of waiting, we end up with the same form, because in order to convert into the correct cost units, we apply Little's Law by multiplying by the arrival rate:

$$\frac{\text{expected waiting cost}}{\text{time unit}} = \frac{\text{waiting cost per time unit}}{\text{customer}} \times \frac{\text{\# customers}}{\text{time unit}}\text{average \# customers},$$

where the latter is just L.
Caution: sometimes there is a distinction between in system costs and in queue costs; in the latter case, you will be using L_q or W_q instead.
Example: ATM machine
Assume Poisson arrival process and exponential service times, sufficient waiting room.
ATM1 costs \$6 an hour to operate, with rate of 12 per hour.
ATM1 costs \$10 an hour to operate, with rate of 15 per hour.
Waiting costs are \$10 an hour for a customer.

The arrival rate is 10 per hour.
ATM1:

$$\rho = 5/6 \Longrightarrow L = 5 \Longrightarrow \text{cost per hour} = \$6 + (\$10)(5) = \$56.$$

ATM2:

$$\rho = 2/3 \Longrightarrow L = 5 \Longrightarrow \text{cost per hour} = \$10 + (\$10)(2) = \$30.$$

Choose ATM2.

Should we add another one?
2 ATM2s:

$$\rho = 1/3 \Longrightarrow L = 3/4 \Longrightarrow \text{cost per hour} = (\$10)(2) + (\$10)(3/4) = \$27.50.$$

Yes!

Should we add yet another one?
No, because the operating cost itself would be \$30 per hour, exceeding the total cost of two ATM2 machines. However, we note that two ATM1 machines actually do better than two ATM2 machines, even though just one of them did worse!

$$\rho = 5/12 \Longrightarrow L = 120/119 \approx 1.0084 \Longrightarrow \text{cost per hour} = (\$6)(2) + (\$10)(120/119) = \$22.08.$$

And three ATM1 machines may do even better!

2.9 The M/G/1 Queue

This is our first example (not counting the $G/G/\infty$ queue, which was kind of trivial) of a queueing system where we allow one of the distributions to be completely general. Thus, since we don't restrict random variables to exponential distributions, we cannot model the system as a Markov chain (although there is a way to model it as an embedded Markov chain, but that is beyond the scope of this course).

The main result is the Pollacek-Khinchin formula, which henceforth will be called the P-K formula for short. Actually the P-K formula gives the capability to compute all the steady-state probabilities p_n, by providing the Laplace transform of the p.d.f. in terms of the p.d.f. of the service time distribution, but in order to do so one has to understand Laplace transforms. In this course, we will deal only with the so-called mean-value version of the P-K formula. We will derive this important formula using some common sense conditioning, PASTA, and a small result from renewal theory, which we can establish.

Recall that λ is the Poisson arrival rate and let X represent the service time random variable. The following is the P-K mean-value formula for the $M/G/1$ queue:

$$W_q = \frac{\lambda E[X^2]}{2(1-\rho)}, \quad \rho = \lambda E[X], \quad p_0 = 1 - \rho, \quad \rho = \lambda E[X]. \tag{1}$$

From this, one can calculate W, L, L_q by the appropriate application of Little's Law and common sense. The key thing to note is that this quantity only depends on the first two moments of the service time, $E[X]$ and $E[X^2]$, which says that any two distributions with the same mean and variance will yield the same mean waiting time.

One simple derivation of this result is similar to the example used in the PASTA section to derive W_q. Essentially, the same arguments hold, the only difference being in the calculation of $E[S_r]$, which is no longer simply $\rho E[X]$ in general, since the distribution need not be memoryless. Simple renewal theory results give

$$E[S_r] = \rho \frac{E[X^2]}{2E[X]} = \frac{\lambda E[X^2]}{2},$$

and going through the algebra yields the P-K formula.
Example ATM with Poisson arrivals and service time distribution (a) exponential; (b) deterministic; (c) Erlang 2-stage; (d) uniform. All with $\lambda = 5$/hr and $E[X] = 1/8$hr, so $\rho = 5/8$.

(a) $E[X^2] = 2/\mu^2$.

$$W_q = 5/24\text{hr}, \quad L_1 = 25/24.$$

(b) $E[X^2] = 1/\mu^2 = X^2$.

$$W_q = 5/48\text{hr}, \quad L_1 = 25/48.$$

(c) $E[X] = 1/8 \Longrightarrow \alpha = 16/\text{hr} \Longrightarrow Var[X] = 1/128\text{hr}^2$.

$$W_q = 5/32\text{hr}, \quad L_1 = 25/32.$$

(b) $X \sim U(0, 1/4)\text{hr} \Longrightarrow E[X^2] = 1/48\text{hr}^2$.

$$W_q = 5/36\text{hr}, \quad L_1 = 25/36.$$

2.9.1 $G/M/1$ Queue

This "dual" to the $M/G/1$ queue can be solved by a technique called the embedded (discrete-time) Markov chain. The idea is that we look at the system only at certain discrete points in time (called epochs) that we select, and then analyze the system as a DTMC. Assuming that we have selected our discrete time points cleverly, there is then a way to go back from the DTMC analysis to the original system.

For the $G/M/1$ queue, we look at the customer *arrival* epochs. We let F denote the c.d.f. of the service time distribution. We define our embedded DTMC by

$$X_n = \text{number in system that } n\text{th arrival sees.}$$

Next we note that since service times are exponentially distributed, the number of service completions in a time interval of length t is distributed Poisson(μt). Conditioning on the service time distribution, we can thus write

$$p_{i,j} = \int_0^\infty e^{-\mu t} \frac{(\mu t)^{i+1-j}}{(i+1-j)!} dF(t), \; j = 1, ..., i+1,$$

where $dF(t) = f(t)dt$ if the service times are continuous with p.d.f. f; otherwise, if the service times are discrete with p.m.f. p, $dF(t) = p(t)$ and the integral becomes a sum. For the case of $j = 0$, it is easiest to use

$$p_{i,0} = 1 - \sum_{j=1}^{i+1} p_{i,j}.$$

Then, one solves for π_n as usual for DTMC to find (non-trivially)

$$\pi_n = (1-\eta)\eta^n, \quad \eta = \int_0^\infty e^{-\mu t(1-\eta)} dF(t), \quad 0 < \eta < 1.$$

Recall that p_n is the probability that there are n in the system. If PASTA held, then we would have $p_n = \pi_n$ here, but PASTA does not hold here. In fact, it can be shown that (beyond the scope of this text)

$$\lambda \pi_{n-1} = \mu p_n.$$

Note that we could also solve the $M/G/1$ queue by this technique, by considering customer departure epochs.

2.10 Queueing Networks

Queueing networks involve "hooking together" individual stations of queues. Most systems of practical interest involve the interactions between stations. We first distinguish between two main classes of queueing networks.

Example 1: Student Union Consider the simplified version of the Student Union made up of only two components: UBC, the university bookstore and Roy's, a fast-food eatery. Customers arriving at the Union can go to either UBC or Roy's, and upon completion, may go to the other or leave the Union.

Example 2: Hippie Trapped in Student Union At some point the doors of the Union are closed, and the hippie and others are trapped forever in the Union (thank God one of the places is an eatery, albeit a fast-food one!).

Example 1 is an example of an open queueing network. Customers enter and eventually leave. Example 2 is an example of a closed queueing network. The number in the system is fixed. In our case, customers neither enter nor leave. Lest this seem unrealistic a system to model anything, we note that the actual requirement for a closed system is that the number of customers in the system be kept as a fixed number. This can also be accomplished by letting a customer into the system only when another customer leaves the system. Popular bars on weekend nights often operate this way, and as in our case, the motivation is to keep the system "stable" in some sense of the word. Another example would be the number of jobs allowed into a network at any time, or in the case of mainframe computers, a limit to the number of batch jobs that can be processed simultaneously. A pre-determined limit could be set, corresponding to our fixed number of jobs. Of course, if in the actual network the number of jobs is also allowed to fall below the limit, then the fixed job case would usually give bounds on performances in the maximum loaded situations which would probably be of most interest to the network manager.

More practical examples include communication and computer networks, airports, and manufacturing systems. Software implementing queuing models have become a very important part of design and control of these systems and many other discrete-event systems.

2.10.1 State of a Network

In the continuous-time Markov chain examples, we had a sneak preview of what is needed to define the "state" of a network. Since the network is made up of a set of individual stations, it is no longer sufficient in most cases to specify simply the total number of customers in the system (in fact, in the closed network case, this number never changes!). The state is in general defined by the number of customers at each station, and is represented by a vector:

$$(n_1,\ n_2, ..., n_M),$$

where M is the number of stations in the network, and n_i represents the number of customers at station i. The corresponding probability vector will be given by

$$p(n_1,\ n_2, ..., n_M).$$

Normalization means that the sum of the probabilities must be 1. In the case of an open queueing network (where customers enter and leave), this usually results in infinite sums, whereas in closed queueing networks (where the number of customers in the system is fixed), this results in a finite sum, albeit of combinatorially increasing size. For example, if there are two stations and three customers in the system that can be at either station, then the state space of the system is given by

$$(3,0),\ (2,1),\ (1,2),\ (0,3),$$

and normalization means that

$$p(3,0) + (2,1) + (1,2) + (0,3) = 1.$$

You should become comfortable working with these probabilities, and translating between them and descriptive quantities. For example, "the proportion of time that both stations are busy" is given by

$$p(2,1) + p(1,2).$$

Also, expectations are calculated by carefully applying the definition or the law of the unconscious statistician. For example, the expected number at station 1 is given by

$$L_1 = \sum_{i=0}^{3} np_n = 0p(0,3) + 1p(1,2) + 2p(2,1) + 3p(3,0).$$

Similarly, if there were three stations and three customers, you should verify that the state space is given by

$$(3,0,0),\ (2,1,0),\ (1,2,0),\ (0,3,0),\ (2,0,1),\ (1,1,1),\ (0,2,1),\ (1,0,2),\ (0,1,2),\ (0,0,3).$$

In general, the number of possible states is given by

$$\left(\begin{array}{c} N+M-1 \\ M-1 \end{array} \right).$$

The proportion of time that all stations are busy is given by

$$p(1,1,1),$$

and the expected number at station 1 is given by

$$L_1 = \sum_{i=0}^{3} np_n = 1(p(1,2,0) + p(1,1,1) + p(1,0,2)) + 2(p(2,1,0) + p(2,0,1)) + 3p(3,0,0).$$

Note that there are three states in which there is one customer at station 1, two states in which there are two, and only one in which there are (all) three.

The concept of service completion rate will become crucial for the analysis of closed queueing networks, so we will revisit it here briefly. Considering station 1, and assuming that the service rate at station 1 is given by μ_1, the service completion rate is given by the service rate times the probability that the server is busy. For the first example, it is thus given by

$$\lambda_1 = \mu_1(1 - p(0,3)),$$

and for the second example, it is given by

$$\lambda_1 = \mu_1(1 - p(0,3,0) - p(0,0,3) - p(0,1,2) - p(0,2,1)).$$

2.10.2 Burke's Theorem

We begin with a very simple yet powerful result. Basically, Burke's theorem says that the output, or departure, process of an $M/M/s$ queue is Poisson. The implication of this is that if you hook another station of exponential servers to the output of an $M/M/s$ queue, you will get another $M/M/s$ queue (probably with a different s). Also the Poisson process has the property that if you split it up probabilistically, you still end up with Poisson processes. Thus, if you feed half the output of the original $M/M/s$ queue to one station of exponential servers and the other half to another station of exponential servers, you can still analyze each of the two stations as before, with the arrival rate halved. Conversely, the Poisson process also has the property that if you combine two (or more) Poisson processes, you end up with a Poisson process again. These two properties allow us to split up and combine Poisson processes.

Thus, for queueing networks of exponential service times and Poisson arrivals **with no feedback**, Burke's Theorem allows us to use common sense and our results from before to analyze the system.

Example MVA: go to apply for title (2 servers), then pay cashier and pick up (1 server).
Assume Poisson arrivals at rate 5/hr; service times all exponentially distributed, with title service rate 5/hr (each) and cashier service rate 8/hr.
Model as a tandem queueing network, with an $M/M/2$ queue followed by $M/M/1$ queue. Check stability first!

$$\rho_1 = 1/2, \quad \rho_2 = 5/8, \quad W_1 = 4/15\text{hr} = 16\text{min}, \quad W_2 = 1/3\text{hr} = 20\text{min}, \quad W = W_1 + W_2 = 36\text{min}.$$

$$L_1 = 4/3, \quad L_2 = 5/3, \quad L = L_1 + L_2 = 3 = \lambda W.$$

Though we have not done so here, note that we can also calculate joint probabilities! In other words, Burke's Theorem also states that the joint probabilities can be calculated by treating the individual queues as independent, i.e., it is simply the product of the marginal probabilities for each of the $M/M/s$ queues. Again, as noted earlier, we can add system times for tandem networks. In general networks, we cannot do this, so be careful!!! But one can always add number in system.

In summary, Burke's Theorem allows us to analyze all queueing networks that have no "feedback" in the system, i.e., no customers ever revisit a station more than once. In graph-theoretic terms, the network structure corresponds to a tree. The main result can be summarized as follows:

- Each station can be treated as an independent $M/M/s_i$ queue.

Note that in reality the stations are *not* independent, since a departure at one station is an arrival at another. However, in steady state, the probability distributions act as though the stations were independent.

Example (previous MVA) We can find the probability that station 1 is empty and that station 2 is empty. The probability that the entire system is empty is then obtained by the product. For an $M/M/1$, $p_0 = 1 - \rho$, whereas for an $M/M/2$ queue, $p_0 = (1-\rho)/(1+\rho)$, so we have

$$p_{0,0} = (1-\rho_2)(1-\rho_1)/(1+\rho_1) = 1/8 = 12.5\%.$$

2.10.3 Open Jackson Networks

The words "node" and "station" will be used interchangeably in the discussion of queueing networks. The open Jackson queueing network model has the following assumptions:

1. Outside arrival process are Poisson.
2. Service times at servers are exponentially distributed.
3. Infinite queue space at each station.

4. Queue discipline is FCFS.

For both open and closed queueing networks, we will define the following parameters:

M = number of stations,
s_j = number of servers at station $j, j = 1, ..., M$,
r_j = rate of external Poisson arrival process to station $j, j = 1, ..., M$,
μ_j = service rate at station $j, j = 1, ..., M$,
λ_j = arrival rate to station i = service completion rate at station $j, j = 1, ..., M$,
p_{ij} = proportion of customers completing service at station i that go to station $j, i = 1, ..., M, j = 0, 1, ..., M$,

where "0" represents leaving the system, i.e., p_{i0} is the proportion of customers completing service at station i that leave the system. Also, $r_j = 0$ for all stations in a closed network.

From the model assumptions, it should be clear that we could in principle model the system as a CTMC, where the state is the vector $(n_1, n_2, ..., n_M)$, with n_i giving the number at station i. However, this will turn out to be a somewhat messy exercise for anything but very small values of M. It turns out that the solution can be found by a decomposition of the network into individual nodes. The main idea is to solve the traffic equations and then to treat each station separately as an appropriate $M/M/s_i$ node.

Solution (by **decomposition**):

1. Solve the **traffic equations**:
$$\lambda_j = r_j + \sum_{i=1}^{M} \lambda_i p_{ij}, \quad j = 1, ..., M. \tag{2}$$
2. Check that the system is stable by checking stability individually at each node.
3. Solve each node individually as an $M/M/s_i$ node to get the probabilities at each station (and hence be able to compute mean number at station, mean time at station, etc.).

Note that the traffic equations by themselves are simply a set of linear equations expressing conservation of flow, and are thus independent of the distributions of the interarrival times (i.e., they would hold for non-Poisson arrival processes, as well).

Although we have assumed exponential service times, we can get similar results for systems with general service times, including the following (see, e.g., the classic paper BCMP):

- infinite server nodes;
- single-server nodes with processor sharing;
- single-server nodes with LCFS.

The three main issues of interest for the **entire system** are the following:

- Stability – whether or not the system will reach steady state in the long run.
- Throughput – the rate at which the system outputs customers.
- Average time or number at a station.

For an open network:

- Stability: check $\rho < 1$ at each station.
- Throughput: if stable, then rate out is simply rate in, so sum all the external arrival rates.
- Average time or number at a station: use the queueing results to find L_i at each station; sum to get L; use Little's Law to get W.

For a closed network:

- Stability: never a problem, since the number of customers in the system is fixed.
- Throughput: usually need to calculate the **service completion rate** at a *specified* node (or set of nodes) corresponding to the input or output station(s).

- Average time or number at a Station: find L_i by using the probabilities appropriately; use the service completion rates and Little's Law to get W_i.

Caution: As noted earlier, $L_1 + L_2 + ...L_M = L$, but $W_1 + W_2 + ...W_M \neq W$, in general, unless it is a simple set of queues in series. Also, throughput is often defined in the queueing literature as the service completion rate throughout the entire network, meaning that a completion at any station contributes to the network throughput.

2.10.4 Closed Jackson Networks

The closed Jackson queueing network model has the following assumptions:

1. Service times at servers are exponential.
2. Sufficient queue space at each station.
3. Queue discipline is FCFS.
4. Single server at each station.

The last assumption can actually be relaxed, but the equations get messier (see, e.g., Wolff). "Sufficient" queue space would be no more thatn the fixed network capacity at each station, since that is the maximum number of customers that could be at any station at one time (less might be needed, depending on the routing). Since the number of customers in the network is fixed, we need that as an additional parameter:

$$N \quad = \quad \text{number of customers in the system (fixed).}$$

Solution (by finding the **joint p.m.f.**):

1. Solve the **visit ratio equations**:.
$$\pi_j = \sum_{i=1}^{M} \pi_i p_{ij}, \quad j = 1, ..., M, \tag{3}$$
$$\sum_{j=1}^{M} \pi_j = 1,$$
where
$$\pi_i = \text{proportion of station } i \text{ visits vs. visits to all stations.}$$
2. Write down the joint probability.
3. Normalize.
4. Find the average number at each station by taking the expected value appropriately.
5. Find the service completion rates at each station by conditioning and taking expected values appropriately.
6. Find the average time at each station by applying Little's Law.

Note that the visit ratio equations are simply a set of equations expressing the visits of a customer through the system, just like our hippie governed by a DTMC, i.e., we are just solving $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ again, where $\mathbf{P}$ this times represents the routing matrix in the network. As we said, one can think of the hippie as being trapped in the closed queueing network.

In order to compute throughput, we have to be able to compute the **service completion rates**, which are the arrival rates to the stations. In an open network, we are given the arrival rates, so the service completion rates are then known, assuming the system is stable. For closed networks, however, we have to compute the service completion rates by **conditioning** on the state of the server at the station. In our simplified case, we will always have just two states: busy or idle. In the idle state, the service completion rate is obviously zero, since the server is not working. On the other hand, in the busy state, the server works at its service rate. Thus, putting this together, at station j, we have

$$\lambda_j = \mu_j P\{\text{station } j \text{ is busy}\} = \mu_j(1 - P\{\text{station } j \text{ is idle}\}). \tag{4}$$

The calculations will be illustrated later on in examples. First, we start with an open system just to show that the concept is consistent.

Example: M/M/1 Queue: Assume parameters λ and μ. Then, the service completion rate is given by

$$\mu(1 - p_0) = \mu\rho = \lambda,$$

the arrival rate.

Comparison of Open and Closed (Jackson) Queueing Networks

	OPEN	**CLOSED**
stability	check each station	guaranteed
TP	easy: $\sum r_i$	function of some λ_i
solve	traffic equations for λ_i	visit ratio equations for π_i
$p(n_1, ..., n_M)$	decomposes: $p^{(1)}_{n_1}...p^{(M)}_{n_M}$	need to normalize: $C\left(\frac{\pi_1}{\mu_1}\right)...\left(\frac{\pi_M}{\mu_M}\right)$
L_i	evaluate at each station	$\sum np_n$ **properly** applied
W_i	evaluate at each station, or L_i/λ_i	L_i/λ_i
L	$\sum L_i$	N (given)
W	L/TP	N/TP

NOTE:

$$(W_q)_i = W_i - 1/\mu_i, \quad (L_q)_i = \lambda_i (W_q)_i.$$

2.10.5 Extensions

Some standard extensions that can be easily handled include the following:

- multiple servers: this can be handled by making the service rate at a station dependent on the state (number at the station, for example), such as $\mu, 2\mu, 3\mu, 3\mu, 3\mu$ for the case of three servers at a station;
- non-exponential servers: similar to the open network case, need to either change the queue discipline, or to have an infinite number of servers.

Example. Consider a network of 3 stations of single-server queues. All customers go to station 1 first, then half proceed to station 2 and the other half to station 3, after which they leave the system. Service times are all exponentially distributed with means 0.5 min, 2 min, and 1 min, respectively, at stations 1, 2, and 3. Assume first that the number of customers in the network is fixed at 3 at all times. Find the following:

1. probability that all stations are busy;
2. probability that no stations are busy;
3. probability that stations 1 and 2 are both busy (station 3 may or may not be);
4. probability that exactly 2 stations are busy;
5. mean number in system at each station;
6. service completion rate at each station;
7. mean (system) time spent at each station;
8. mean queue time spent at each station;
9. system throughput;
10. mean total system time in network.

First note that the set of possible states is as follows:
(3,0,0),(2,1,0),(2,0,1),(1,2,0),(1,0,2),(1,1,1),(0,3,0),(0,2,1),(0,1,2),(0,0,3).

Denoting $\mu_1 = 2/\text{min}, \mu_2 = 0.5/\text{min}, \mu_3 = 1/\text{min}$, the visit ratio equations are given as follows:

$$\begin{aligned} \pi_1 &= \pi_2 + \pi_3; \\ \pi_2 &= 0.5\pi_1; \\ \pi_3 &= 0.5\pi_1; \end{aligned}$$
$$\pi_1 + \pi_2 + \pi_3 = 1.$$

Solving yields $\pi_1 = 0.5$, $\pi_2 = \pi_3 = 0.25$, so the probability distribution is given by

$$p(n_1, n_2, n_3) = C(0.25)^{n_1+n_3}(0.5)^{n_2},$$

$$C = \frac{1}{1/8 + 1/16 + 1/16 + 1/32 + 1/32 + 1/32 + 1/64 + 1/64 + 1/64 + 1/64} = \frac{32}{13}.$$

Specifically, $p(0,3,0) = 4/13$, $p(1,2,0) = p(0,2,1) = 2/13$,
$p(2,1,0) = p(0,1,2) = p(1,1,1) = 1/13$,
$p(3,0,0) = p(0,0,3) = p(2,,0,1) = p(1,0,2) = 1/26$.

1. probability that all stations are busy: $p(1,1,1) = 1/13$;
2. probability that no stations are busy: $p(0,0,0) = 0$;
3. probability that stations 1 and 2 are both busy: $p(1,2,0) + p(2,1,0) + p(1,1,1) = 4/13$;
4. probability that exactly 2 stations are busy: $p(1,2,0)+p(0,2,1)+p(2,1,0)+p(0,1,2)+p(2,0,1)+p(1,0,2) = 7/13$;
5. $L_1 = 1[p(1,2,0) + p(1,1,1) + p(1,0,2)] + 2[p(2,1,0) + p(2,0,1)] + 3p(3,0,0) = 8/13$,
   $L_2 = 1[p(2,1,0) + p(0,1,2) + p(1,1,1)] + 2[p(1,2,0) + p(0,2,1)] + 3p(0,3,0) = 23/13$,
   $L_3 = 1[p(0,2,1) + p(1,1,1) + p(2,0,1)] + 2[p(0,1,2) + p(1,0,2)] + 3p(0,0,3) = 8/13$;
6. $\lambda_1 = \mu_1[p(1,2,0) + p(1,1,1) + p(1,0,2) + p(2,1,0) + p(2,0,1) + p(3,0,0)] = 11/13$ per min,
   $\lambda_2 = \mu_2[p(2,1,0) + p(0,1,2) + p(1,1,1) + p(1,2,0) + p(0,2,1) + p(0,3,0)] = 11/26$ per min,
   $\lambda_3 = \mu_3[p(0,2,1) + p(1,1,1) + p(2,0,1) + p(0,1,2) + p(1,0,2) + p(0,0,3)] = 11/26$ per min;
7. $W_1 = L_1/\lambda_1 = 8/11$ min,
   $W_2 = L_2/\lambda_2 = 46/11$ min,
   $W_3 = L_3/\lambda_3 = 16/11$ min;
8. $(W_q)_1 = 5/22$ min,
   $(W_q)_2 = 24/11$ min,
   $(W_q)_3 = 5/11$ min;
9. $TP = \lambda_1 = \lambda_2 + \lambda_3 = 11/13$ per min;
10. $W = N/TP = 39/11$ min $= W_1 + 0.5W_2 + 0.5W_3$.

Example. Consider the previous network with external arrivals to station 1 following a Poisson process at rate 30/hour. Then, the number of possible states is infinite, of course. We have $r_1 = 0.5$/min, given the following simple traffic equations:

$$\begin{aligned} \lambda_1 &= r_1 = 0.5/\text{min}, \\ \lambda_2 &= 0.5\lambda_1 = 0.25/\text{min}, \\ \lambda_3 &= 0.5\lambda_1 = 0.25/\text{min}, \end{aligned}$$

$$\rho_1 = 0.25, \quad \rho_2 = 0.5, \quad \rho_3 = 0.25.$$

$$L_i = \frac{\rho_i}{1 - \rho_i}.$$

1. probability that all stations are busy: $(1 - p_0^{(1)})(1 - p_0^{(2)})(1 - p_0^{(2)}) = \rho_1\rho_2\rho_3 = 1/32$;
2. probability that no stations are busy: $p(0,0,0) = (0.75)(0.5)(0.75)$;
3. probability that stations 1 and 2 are busy: $1 - p(0,\cdot,\cdot) - (\cdot,0,\cdot) + p(0,0,\cdot) = 1 - p_0^{(1)} - p_0^{(2)} + p_0^{(1)}p_0^{(2)} = 0.125$;
4. probability that exactly 2 stations are busy: $\sum_{n_1,n_2>0} p(n_1,n_2,0) + \sum_{n_1,n_2>0} p(n_1,0,n_2) + \sum_{n_1,n_2>0} p(0,n_1,n_2) = (1 - p_0^{(1)})(1 - p_0^{(2)})p_0^{(3)} + (1 - p_0^{(1)})p_0^{(2)}(1 - p_0^{(3)}) + p_0^{(1)}(1 - p_0^{(2)})(1 - p_0^{(3)}) = 7/32$;
5. $L_1 = 1/3$, $L_2 = 1$, $L_3 = 1/3$, $L = 5/3$;
6. found already by the traffic equations;
7. $W_1 = 2/3$ min, $W_2 = 4$ min, $W_3 = 4/3$ min;
8. $(W_q)_1 = 1/6$ min, $(W_q)_2 = 2$ min, $(W_q)_3 = 1/3$ min;
9. $TP = r_1 = \lambda_2 + \lambda_3 = 30$ per hour;
10. $W = L/TP = (5/3)/0.5$ min $= 10/3$ min $= W_1 + 0.5W_2 + 0.5W_3$.

Exercises for Chapter 2

1. Consider a bank with two tellers. Assume that the arrival process is Poisson at a rate of 2 per minute and that service times are exponential with a mean of 40 seconds. It is currently 10 AM.

   (a) Find the probability that the next arrival will arrive
      (i) before 10:01 AM.
      (ii) between 10:01 AM and 10:02 AM.
      (iii) after 10:02 AM.

   (b) Find the probability that between now and 10:02 AM there will be
      (i) no arrivals.
      (ii) exactly one arrival.
      (iii) at least one arrival.

   (c) Given that there has been no arrivals by 10:30 AM, find the probability that the next arrival will arrive
      (i) before 10:31 AM.
      (ii) between 10:31 AM and 10:32 AM.
      (iii) after 10:32 AM.

   (d) If both tellers are currently busy serving customers, find the probability that
      (i) neither customer will be finished before 10:01 AM.
      (ii) both customers will be finished between 10:01 AM and 10:02 AM.
      (iii) at least one of the customers will be finished by 10:01 AM.

   (e) Assume that there is sufficient capacity for any number of customers in queue. In steady-state, find
      (i) the probability that the system is empty.
      (ii) the probability that both servers will be busy.
      (iii) the probability that an arriving customer will have to wait.
      (iv) the average number of customers in the system.
      (v) the average number of customers in queue.
      (vi) the average time a customer spends in the system.
      (vi) the average time a customer spends waiting.

      The bank is considering reducing the number of tellers. Would this be a good idea? Why or why not?

   (f) Assume that there is no room for any waiting. In steady-state, find
      (i) the probability that the system is empty.
      (ii) the probability that both servers will be busy.
      (iii) the probability that an arriving customer will have to wait.
      (iv) the average number of customers in the system.
      (v) the average number of customers in queue.
      (vi) the average time a customer spends in the system.

2. Suppose that babies are born in the U.S. according to a Poisson process at a rate of one a second. (a) What is the expected time between now and five more births? (b) What is the probability that the time between the tenth and eleventh birth exceeds two seconds? (c) What is the expected number of births for an entire (seven-day) week? (d) What is the probability that there will be more than 3 births in the next 5 seconds?

3. Suppose that students arrive to the ATM machine at a rate of 5 per minute, that the average number waiting for or using the machine is 6 persons, and that the average time it takes for a transaction at the machine is 30 seconds. Find the average time a person spends waiting in line. NOTE that no assumptions have been made on distributions! (Hint: use Little's Law.)

4. The Motor Vehicle Administration is considering the following two options for processing driver's license renewals at their facility:

   (a) There is a single station at which the entire renewal process is completed. The station is served by two clerks, each of which can serve customers at a rate of 40 per hour.

(b) There are two stations, and the customer must go through both. At the first station, the customer fills out the appropriate forms, and at the second station, the customer pays the cashier and receives the license. Each station has one clerk each. The average time at the first station is 60 seconds, whereas the average time at the second station is 30 seconds.

Assuming that all service times are exponential and that the arrival process is Poisson with rate 40 per hour, which option will give the customer a smaller average total time at the facility? Would the answer change if in the second option, the service times were distributed equally as 45 seconds (average) at each station?

5. During the peak period each semester, the UM registration office serves approximately 120 students as hour (assume according to a Poisson process). Each student must go through three stations, each of which has exponentially distributed service times with means 20 seconds, 12 seconds, and 15 seconds, corresponding to stations 1, 2, and 3, respectively. Find the average time a student spends in the office and the expected number of students in the office.

6. A small store has a single checkout with a cashier. Customer arrive according to a Poisson process with rate 30/hr; service times exponentially distributed with mean 1.5 min. Hiring a bagger to help the cashier would reduce the mean service time to 1 min. The bagger would cost $8/hr, the cashier costs $16/hr, and customer waiting time costs are estimated at $0.08/min. Based on expected costs, should the bagger be hired?

7. Consider a queueing network of three single-server stations of sufficient buffer capacity, exponentially distributed service times with rates per hr 40, 50, 30, respectively, at stations 1, 2, and 3, and external Poisson arrival processes with rates 10/hr and 20/hr, respectively, to stations 1 and 2. 60% of the jobs finishing at station 1 go to station 2, whereas the rest leave; half of the jobs finishing at station 2 go back to station 1 and the other half go to station 3; 10% of the jobs finishing at station 3 go back to station 2 and the rest leave. Find the following quantities:

   (a) proportion of time that there is exactly 1 customer in entire network;

   (b) expected system time, waiting time, and number in queue at each station;

   (e) probability that the system is empty;

   (d) expected total number in system and expected total system time for a customer;

   (e) expected number of jobs served in half an hour.

8. An assembly line has two operations on a part: painting and buffing. Service times are exponentially distributed with a mean of 2.4 minutes for painting and 3.75 minutes for buffing. Arrivals to the line follow a Poisson process at a rate of 112 every 5 hours. Find (in steady state)

   (a) the average number of painted but not yet fully buffed parts in the system.

   (b) the average time in queue at each operation.

   (c) the average number of parts in the assembly line.

   (d) the average time spent by a part in the assembly system.

   (e) the throughput of the line.

   Assuming that all service times are exponential and that the arrival process is Poisson with rate 5 per hour, which option will give the customer a smaller average total time at the facility?

9. The plant manager wants to decide whether to buy printer A or printer B. Machine A prints at a rate of 20 jobs/hour at a cost of $3/hour. Machine B prints at a rate of 30 jobs/hour at a cost of $3.30/hour. Print job times are exponentially distributed. The manager estimates that on the average the “inventory” cost per job is $1/hour. If print jobs arrive according to a Poisson process at a rate of 10 per hour, then (in steady state)

   (a) what is the average cost per hour using Machine A?

   (b) what is the average cost per hour using Machine B?

   (c) if the manager chooses according to average cost per hour, which should he buy?

10. In which case would the average steady-state system time be shorter?
Case 1 - M/M/1 with arrivals at rate 1/hour and service rate 2/hour
Case 2 - M/M/2 with arrivals at rate 1/hour and service rate 1/hour
Prove your result, and give an intuitive explanation as to why it is true.

11. Repeat the previous problem for expected steady-state waiting time.

12. Suppose it costs \$1/hour per person in heating costs at the Stamp Union near the ATM machine and \$5/hour to run an ATM machine. If arrivals to ATM are Poisson with rate 5 per minute, and the service time of an ATM machine is exponential with a mean of 10 seconds,
    (a) what is the average cost per hour using one machine?
    (b) what is the average cost per hour using two machines?
    (c) if the number of machines chosen is based solely on cost, how many should be chosen?
    (d) why might the number not be chosen solely on cost to the bank?
    (Note that we have NOT included the *customer's* cost of waiting!)

13. Gory Auto Repairs does brake work and muffler work. Cars arrive at a rate of 4 per hour to the brake shop and at a rate of 2 per hour to the muffler shop. However, half of the cars finishing brake work also end up needing muffler work, while the rest are done. All cars finishing muffler work need no more work. Average time for a brake job is ten minutes, while the average time for a muffler job is twenty minutes. There is one worker in the brake shop and two workers in the muffler shop. Assuming exponential interarrival times and service times, find (in steady state)
    (a) average number of cars in the brake shop;
    (b) average time a car spends in the brake shop;
    (c) average number of cars in the muffler shop;
    (d) average time a car spends in the muffler shop;
    (e) average number of cars in Gory Auto Repairs;
    (f) average time a car spends in Gory Auto Repairs.
    (g) Why doesn't the sum of (b) and (d) equal (f)? Can you find a way of appropriately combining (b) and (d) to get (f)?
    (h) Find the throughput of the system.
    (i) Find the probability that the system is empty.

14. Rodham's Car Wash has a secret two-step process. The car first goes through a wash cycle and then through a rinse cycle. However, half of the cars that go through the wash cycle have to go through again, due to problems with the process, and one-quarter of the cars that go through the rinse cycle have to be rinsed again. If arrivals to the system are Poisson with rate 1/minute, service times are exponential with rate 2/minute at the wash cycle and 1.25/minute at the rinse cycle:
    (a) Show that the system is unstable in steady-state (by solving the traffic equations).
    (b) However, if we fix the number of parts allowed in the system (CLOSED network), it will be stable. Fix the number of parts in the system at 2 and find the steady-state
        (i) the average number at each station;
        (ii) the service completion rate at each station;
        (iii) the average time spent at each station;
        (iv) the average time spent in the system.
        Why don't the two values in the previous part add up to this value?
        (v) the throughput of the system;
        (vi) the probability that the system is empty.

15. Consider the queueing network example at the end of the chapter. Assume there are external arrivals to station 1 following a Poisson process at rate 48/hour. Find all the corresponding quantities as found in the example.