Elements of Queuing Theory

Antonio Carzaniga

March 5, 2025

A computer network can be seen as one big queuing system. Packets travel through a network effectively switching from one queue to another. A router is itself a queuing system that contains other queuing systems within it, such as its network interfaces. So yes, queuing theory is fundamental for computer networking.

However, queuing theory has a plethora of other applications beyond computer networking. Many of those affect your life quite directly on a daily basis. When you are in line to get a coffee, or when you are having dinner at a restaurant; when you are going through the check-out process at the grocery store, or when you are stuck in traffic; when you are taking an elevator, or when you sent in your job application. In all these circumstances *you* are in a queuing system.

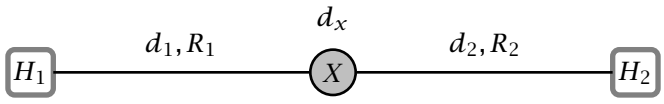
A queuing system consists of a set of processes that produce, consume, and exchange "customers". As I was saying, sometimes the customer is a person, and more specifically you. However, in computer networking and other similar computer systems, the customer is some discrete data unit, such as a network packet or a service request.

When a server process receives a customer that requires some processing but the server process is already busy processing another customer, the server process might delay the processing of the new customer by holding it in a temporary *queue*.

1 Basic Queuing Model in Networking

Consider a much simplified abstraction of a router in a very simple two-hop network.

Delay in Store-and-Forward Networks



$$d_{end\text{-}end} = d_1 + \frac{\ell}{R_1} + d_x + \frac{\ell}{R_2} + d_2$$

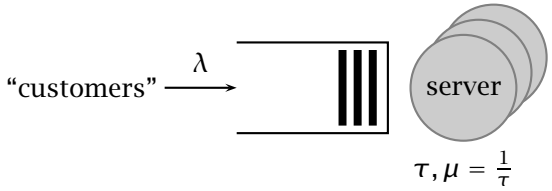
$$d_x = d_{cpu} + d_{queue}$$

In this case, the end-to-end delay consists of the sum of the

- propagation delays: $d_1 + d_2$

- transmission delays: $\ell/R_1 + \ell/R_2$
- processing delay: $d_x = d_{cpu} + d_{queue}$

A packet arriving at a router must be processed for forwarding—to determine which output port it should be forwarded through; then the packet must be switched from the input port over to the chosen output port; and then the packet must be transmitted onto the output medium. In essence, each stage requires some processing, which takes some time τ, performed by one of potentially multiple *servers.* However, for each processing stage, the packet might also have to wait in a queue. This model of a processing stage with incoming customers, a queue, and a set of server processes the most basic model of a queuing system.



Main Model Parameters

λ arrival rate

τ service time

Therefore, the service rate is $\mu = \frac{1}{\tau}$

s number of servers

M total capacity of the system, including server slots

Therefore, the queue capacity is $M - s \geq 0$

A primary parameter for the queuing system is the arrival rate λ, which measures the number of "customers" arriving at the system per unit time. In the case of a network processor, the arrival rate is typically measured in packets per second. The capacity M of the queuing system is measured in packets (although sometime it also makes sense to talk about the size in *bytes*) and includes the s servers.

The processing of each customer/packet (by a server) requires τ time units. Thus we refer to τ as the *service time,* and based on that we can also define a service *rate* $\mu = 1/\tau$, which in our case is once again measured in packets per second.

We assume that the queuing system is *stable,* meaning that the queue occupancy remains bounded over time. Since the system consists of s servers, the queuing system is stable when

$$\lambda < s\mu$$

meaning that the arrival rate λ, which is the rate at which customers enter the system, is strictly less than the aggregate service rate $s\mu$, which can in fact be thought of as the maximal rate at which customers can *leave* the system.

Performance Indicators (Service Perspective)

a offered load (or demand): $a = \lambda\tau$

ρ utilization ratio: percentage of time that a server is serving customers (as opposed to being idle)

$$\rho = \frac{\lambda}{s\mu}$$

We can then define some performance indicators. The utilization ratio ρ is the ratio between the service time—which is the time it takes a server to process one customer—and the inter-arrival time—that is, the time between arrivals, which is $1/\lambda$. So, for one server ($s = 1$) we have $\rho = \frac{\lambda}{\mu}$. However, for s servers capable of working in parallel, the service time is effectively divided by s, therefore $\rho = \frac{\lambda}{s\mu}$. And since we assume the queuing system to be stable, we have $\rho < 1$.

Example: Post Office Staff

Let's review these concepts with an example. On average, 40 persons go to the post office every hour. Also on average, the service time for each person is four minutes. How many clerks should the post office employ (to serve clients) so that the queuing delay remains limited? In that case, what is the probability that a clerk is not serving a client at any given time?

2 Little's Theorem

One of the most basic results in queuing theory is Little's theorem. Let's state the theorem right away, initially a bit informally. Consider a stable system over a long-enough period of time. Let N be the average number of customers in the system. Let T be the average time that a customer spends in the system. Let λ be the arrival rate. Then

$$N = \lambda T$$

This result is quite simple and intuitive. One can immediately derive this relation between N, T, and λ by considering the physical dimensions of those variables. N is measured in customers, λ is customers per unit time, and T is time. Also, again intuitively, the number of customers in the system should be proportional to the arrival time, and to the time each

The theorem is also very general. The results applies to all queuing subsystems, regardless of the arrival process or the service-time distribution.

Example: Restaurant Business

You are the manager of a successful restaurant. On average, each customer spends c (money) for a meal, and for that price and quality, the restaurant is always full. How can you increase revenues?

The revenue flow is $r = c\lambda$ (money per time unit) which allows you to compute the daily or monthly revenues. So, by Little's theorem, we have

$$r = c\frac{N}{T}$$

where N is effectively the number of seats (or the number of tables) in the restaurant, and T is the average time that each customer spends in the restaurant. So, you can either increase prices, and therefore increase c, which is obvious but also risky, or you can increase the number of seats N, or reduce the service time T—or both.

Example: Car Traffic on a Rainy Day

Considering car traffic in the city of Lugano, what is the relation between the number of cars and their average speed? Consider commuter traffic, that is, people traveling from their homes to their place of work, and back. What happens when it rains?

Considering the whole system of roads as the queuing system, you can assume that the average arrival rate λ is always the same, since that is simply the number of commuters over a day—that is, all the people who work in Lugano but live outside the city commute in (and out) every day. You may also assume that each commuter travels the same distance to and from their place of work. So, this means that the average distance ℓ is also a constant.

The two variables to characterize traffic are therefore the average commuting time T, and the average number of cars on the road, N.

By Little's theorem, $\lambda T = N$, which means that the commuting time is proportional to the number of cars on the road. This is very intuitive. High traffic, meaning a large N, also implies long commuting times, meaning a large T.

So, what happens when it rains? Well, one thing that happens is that the average velocity is reduced due to the reduced visibility and for safety reasons in general. Since the average velocity is simply ℓ/T, and since ℓ is constant, that means that T is higher, and therefore N is higher. In other words, the rain causes traffic to slow down and, by Little's theorem, N to go up and therefore congestion to also increase.

Example: Transit Time

Consider again cars on a road, and in particular on a single stretch of highway without exits, such as a tunnel. How can you measure the transit time, meaning the time it takes for cars to go through the tunnel?

These days one can set up cameras and computers that read plate numbers and then measure the transit time for each car, and then compute the average. However, that is unnecessarily complicated, also from a computational point of view, since it requires to maintain state, specifically entry and exit times, proportional to the number of cars.

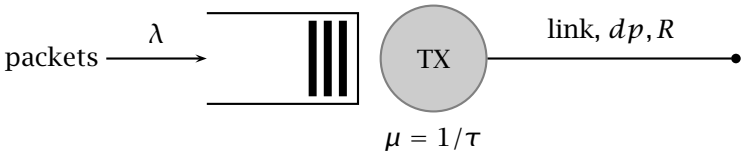
We can do better with Little's theorem. The average transit time is $T = N/\lambda$. So, we can compute T by measuring N and λ with two, much simpler sensors at the entrance and exit of the tunnel. The sensors record the passage of a car. So, λ is the number of cars entering the tunnel per unit time, and N is the difference between the count at the entrance and the count at the exit. We have to assume that the system starts from an empty highway or in any case from a known value N_0. However, once we have N and λ, we can compute the transit time using Little's theorem:

$$T = \frac{N}{\lambda}$$

.

2.1 Examples in Networking: Transmission Link

Consider a transmission link, consisting of as a buffer (queue), a transmitter with transmission rate R (server), and a link with propagation delay d. In this case, we consider fixed-size, and more specifically and for simplicity *unit*-size packets. The beauty of Little's theorem is that it is applicable to every subsystems, as well as the system as a whole.



Consider first the transmitter alone, without queue and without link. Little's theorem ($\lambda T = N$) can be written as $R = 1/d_{TX}$. Here λ is effectively the transmission rate R expressed in *packets* per unit time, and the service time is the transmission delay $\tau = d_{TX}$. And $N = 1$, since there is only one packet (at most) in the system (transmitter) at any given time. Therefore we have the simple and perhaps obvious relation

$$\mu = R = \frac{1}{d_{TX}}$$

Consider now the transmission line alone. Here Little's theorem gives us the link capacity, which means the number of packets that are "in flight":

$$\lambda d_p = N$$

Consider now the buffer/queue alone. Little's theorem gives us the queuing delay

$$d_q = N/\lambda$$

where N is the average number of packets in the queue.

Consider now the whole link or in fact a whole network carrying multiple flows with intensities λ_i. For each flow, we have $\lambda_i T_i = N_i$. We can then reason about the global flow. The global intensity is $\lambda = \sum_i \lambda_i$, and similarly the total queue occupancy is $N = \sum_i N_i$. So, by Little's theorem, we have the average time

$$T = \frac{N}{\lambda} = \frac{\sum_i N_i}{\sum_i \lambda_i} = \frac{\sum_i \lambda_i T_i}{\sum_i \lambda_i}$$

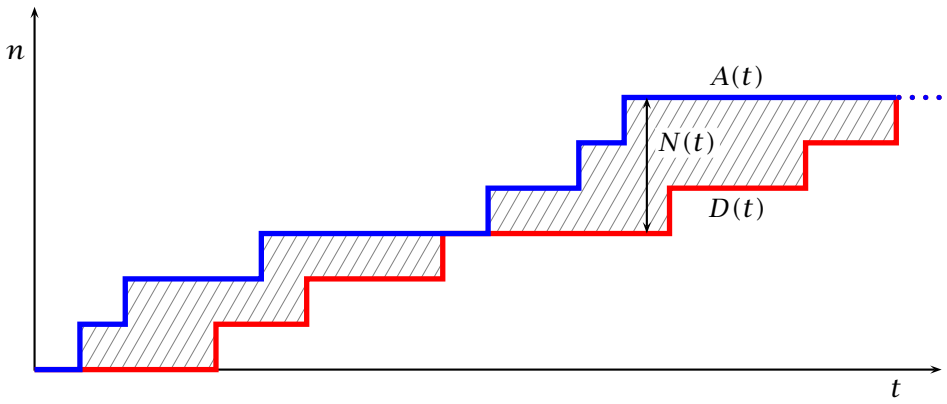
which effectively says that the average transit time is the weighted average of the individual per-flow transit time, where the weights are the flow intensities.

2.2 Proof of Little's Theorem

Let $A(t)$ and $D(t)$ be the total number of arrivals and departures until time t, respectively. Then

$$N(t) = A(t) - D(t)$$

is the number of customers in the system at time t.



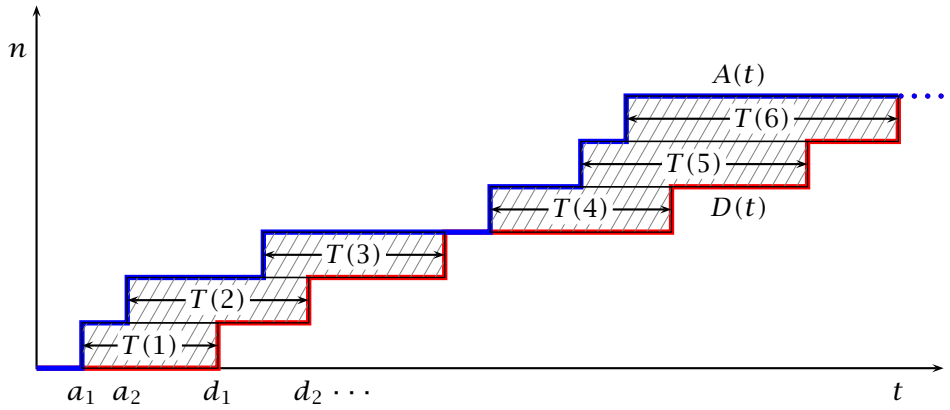
Little's theorem says that $N = \lambda T$, where N is the average number of customers in the system, where the average is taken over a long period of time. This notion of a long-term average, which also applies to λ and T, allows us to ignore the initial and final conditions. More specifically, the effect of the customers that are already in the system at the beginning of the period, and those that remain in the system at the end, vanishes over a long-enough period of time. In fact, the theorem is *exact* for any time interval (t_0, t^*) provided that $A(t_0) = D(t_0)$ and $A(t^*) = D(t^*)$, meaning the system is empty at the beginning and at the end time t^*. So, that is what we assume to prove the theorem.

Formally, given the instantaneous number of customers $N(t)$, we can consider the average of $N(t)$ over an interval $(0, t^*)$:

$$N = \frac{1}{t^*} \int_0^{t^*} N(t)dt$$

The integral above represents the area between the two curves $A(t)$ and $D(t)$. Now, we can look at the same area in a different way. Suppose for now that the system is FIFO, meaning that each departure corresponds to the first customer that is still in the system. Graphically, this means that considering the same level $n = 1, 2, \ldots$ on the vertical axis, the two vertical segments of the arrival and departure curves, $A(t)$ depicted in blue and $D(t)$ depicted in red, going from $n-1$ to n on the vertical axis, represent the arrival and departure of the n-th customer, respectively.

Therefore, the area between the two curves, measured from $t = 0$ to $t = t^*$ can also be seen as the sum of the areas of $A(t^*)$ rectangles, each with height 1 and with width $T(n)$, for $n = 1, 2, \ldots$, where $T(n)$ is the time that customer n spends in the system. This is depicted in the graph below:



Since the area between the departures and arrivals curves is equal to the sum of the areas of the rectangles, we can rewrite

$$N = \frac{1}{t^*} \sum_{i=1}^{A(t^*)} T(i)$$

Now, if we define T as the *average* time that a customer spends in the system

$$T = \frac{1}{A(t^*)} \sum_{i=1}^{A(t^*)} T(i)$$

then we have

$$N = \frac{A(t^*)}{t^*} T \qquad \text{and therefore} \quad N = \lambda T$$

We assumed that arrivals and departures are FIFO. That is, we assume that the first departure time d_1 is the departure time of the first customer arrived at time a_1, and that the second departure time d_2

is the departure time of the second customer arrived at time a_2, and so on. We used this assumption to compute the time $T(i) = d_i - a_i$ that a customer spends on the system. However, the result is the same for any departure order. In fact, the average time is given by the total time $\sum_{i=1}^{A(t^*)} T(i)$, which can be rewritten as the sum of all the departure times minus the sum of all the arrival times:

$$T = \frac{1}{A(t^*)} \sum_{i=1}^{A(t^*)} T(i) = \frac{1}{A(t^*)} \left(\sum_{i=1}^{A(t^*)} d_i - \sum_{i=1}^{A(t^*)} a_i \right)$$

The difference between the totals—departures minus arrivals—is the same regardless of the departure order.

3 Poisson Processes

In studying queuing systems, we must model the arrival process. The term *process* here refers to the distribution of the arrivals over time. And of course the same can be applied to other processes.

The simplest example is a process whereby customers arrive at times $0, c, 2c, 3c$, etc., meaning at regular intervals, where the inter-arrival time is always $c = 1/\lambda$. We call this a *deterministic* process.

A more interesting case is one where arrivals are somewhat random. In general, we call this a *stochastic* process. More specifically, we are interested in an arrival process where each arrival is random but the process as a whole maintains a constant long-term arrival rate λ. What would such a process look like? What does it even mean, exactly, that each arrival is random but the average rate is constant?

Imagine a large number n of independent sources, where each individual source produces events with a very small probability, but such that *in aggregate* there are λ events per unit time. This is what we call a Poisson process, and as it turns out, this process models very well a wide range of phenomena that are very relevant to many applications of queuing theory, including the arrival of service requests to a server or the arrival of packets to a network interface.[1]

As a concrete example, let's consider the radioactive decay with the emission of alpha or beta particles by some materials. If we use a Geiger counter to measure the radioactive emissions of, say, a banana, which contains potassium, including a particular isotope of Potassium that is radioactive, we might detect a few emissions per second. These emissions do not come at regular intervals and instead seem to be distributed randomly, but still with a stable long-term average of a handful of events per second. What is the statistical nature of these emissions? How can we figure out their distribution over time?

At its core, the process of radioactive decay is probabilistic, due to the quantum nature of the nuclear and electromagnetic forces that are responsible for it. With a good level of approximation, we can say that an atom has some probability of decaying and therefore emitting a particle within a fixed period of, say, one second. This probability is typically extremely low for each individual atom. For example, Potassium-40, the radioactive isotope of Potassium found in trace quantities in bananas, has a half-life of about 1.25 billion years. This means that each atom has a 50% chance of decaying (one in two) in a period of a billion years, and since there are about 31.5 million seconds in a year, that means that a Potassium-40 nucleus has one in 79 million billion chances of decaying within a period of 1 second (one in $2 \times 1.25\text{billion} \times 31.5\text{million} \approx 79$ million-billion).

So, for each atom of Potassium-40 in our banana, the probability that that atom emits a particle is indeed tiny ($p = 1.3 \times 10^{-17}$). However, there are a huge number of such atoms in a banana, roughly $n \approx 10^{18}$, and their decay processes are all independent of each other. As a result, from a banana, we observe an aggregate rate of about $np \approx 13$ emissions per second.

[1] Experimental observations of packet arrivals show that the

The same idea and model applies to many other processes, including, at least to a first-level approximation, the arrivals of packets at a router. So, let's analyze this process. In particular, we want to analyze the arrival function $A(t)$ that counts arrivals over time.

3.1 Poisson Distribution

Let λ be the total number of events per unit time from a large population of independent sources. Consider now the time interval $(0, t)$. During this interval of duration t, on average, we observe λt events as a sum of n potential events each occurring with probability p. That is, in expectation, $np = \lambda t$ and therefore the probability of each event is

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

Now, depending on the model of the arrival process, a source might generate arrival events "with replacement," meaning that the same source could generate multiple events. (This is not the case for some radioactive nuclei.) However, the probability of even a single event is so minuscule that we can safely ignore the probability that *the same source* would fire twice. In other words, in an interval t, each of the n sources generates *one* event with probability $\frac{\lambda t}{n}$, or *zero* events with probability $1 - \frac{\lambda t}{n}$.

In summary, we model $A(t)$ as the sum of n independent indicator random variables X_i with probability $p = P(X_i = 1) = \lambda t/n$ (binomial distribution) taken to the limit for $n \to \infty$.

$$P(A(t) = k) = \lim_{n\to\infty} \binom{n}{k} p^k (1-p)^{n-k}$$

And, since $p = \frac{\lambda t}{n}$, we have

$$\begin{aligned}
P(A(t) = k) &= \lim_{n\to\infty} \binom{n}{k} \left(\frac{\lambda t}{n}\right)^k \left(1 - \frac{\lambda t}{n}\right)^{n-k} \\
&= \lim_{n\to\infty} \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \left(\frac{\lambda t}{n}\right)^k \left(1 - \frac{\lambda t}{n}\right)^{n-k} \\
&= \lim_{n\to\infty} \frac{n^k + O(n^{k-1})}{n^k} \frac{(\lambda t)^k}{k!} \left(1 - \frac{\lambda t}{n}\right)^{n-k} \\
&= \lim_{n\to\infty} \frac{(\lambda t)^k}{k!} \left(1 - \frac{\lambda t}{n}\right)^{n-k} \\
&= \frac{(\lambda t)^k}{k!} \lim_{n\to\infty} \left(1 - \frac{\lambda t}{n}\right)^{n-k} \\
&= \frac{(\lambda t)^k}{k!} e^{-\lambda t}
\end{aligned}$$

Thus we derived the probability mass function of the Poisson distribution. This is the probability that the system gets k arrivals over a period t. Notice again that λ is the total number of events *per unit time.* Therefore, we derive a Poisson distribution with parameter λt, for a time interval of duration t.

We derived this distribution by considering one random variable X_i to count the events occurring for each source i out of a very large number n of independent sources. However, the calculation is exactly the same for a different model in which we always consider the *aggregate* of all sources, but we split the interval t in a number n of sub-intervals of duration $\delta = t/n$, and for each interval $i = 1, 2, \ldots, n$

we let the random variable X_i be the number of events occurring during the i-th sub-interval. Now, for a large-enough n and therefore for a small-enough δ, we can consider the probability of two or more events occurring in the same δ interval to be negligible. Therefore, we can consider the X_i's to be indicator variables, $X_i \in \{0, 1\}$, indicating whether or not an event occurs in the i-th sub-interval—which leads to the same binomial distribution taken to the limit for $n \to \infty$.

Example: To make all of this a bit more concrete, imagine you are setting out to admire shooting starts in the clear night sky. Your astrophysicist friend told you that tonight there will be one visible shooting star per minute on average. So you're gazing at the vastness of the sky when—*wow!*—you see a shooting star. Wonderful! Now, what is the probability that you will see another shooting star within the next minute? Or what would be the probability of seeing two of them in the next 20 seconds?

The Poisson distribution gives you the answers right away. The long-term average is $\lambda = 1$, one star per minute. The probability of seeing one shooting in a period of $t = 1$ minute ($\lambda t = 1$) is therefore:

$$P(A(t) = 1) = \frac{(\lambda t)^1}{1!} e^{-\lambda t} = e^{-1} \approx 0.37$$

What about two stars in 20 seconds? Just plug-in the numbers. The period is $t = 20$ seconds, so $\lambda t = 1/3$, and we're looking for $k = 2$ arrivals: $P(A(t) = 2) = \frac{(\lambda t)^2}{2!} e^{-\lambda t} = \left(\frac{1}{3}\right)^2 \frac{1}{2} e^{-1/3} \approx 0.02$.

3.2 Inter-Arrival Times

The Poisson distribution is a *discrete* distribution. It characterizes the probability of k events (e.g., packet arrivals) occurring in a given time period t. We are now interested in characterizing the time between arrivals.

Let $T_1, T_2, \ldots$ be the times between arrivals, such that T_i is the time between arrival $(i-1)$ and arrival i. We want to characterize the distribution of the random variables T_i. Now, let t_0 be the arrival time of event $i-1$, then the probability that T_i is greater than a certain time t is the probability that there are $k = 0$ arrivals in the t interval starting at t_0.

$$P(T_i > t) = P(A(t) = 0) = e^{-\lambda t}$$

Example: Let's go back to the spectacle of shooting stars. After you see one, you wonder how long you'll have to wait to see another one. What is the probability that you will have to wait at least 30 seconds?

Assuming the long-term average is still $\lambda = 1$ star per minute, then the probability that the inter-arrival time and therefore your waiting time is at least 30 seconds is $P(T_i > 0.5) = P(A(0.5) = 0) = e^{-0.5} \approx 0.61$.

What about two minutes? $P(T_i > 2) = P(A(2) = 0) = e^{-2} \approx 0.14$.

3.2.1 Memoryless Property

We characterized the inter-arrival time with an exponential distribution. In particular, $P(T_i > t) = e^{-\lambda t}$. This distribution has an important property: we say it is *memoryless.* Let's see what that means with an example.

Example: Three minutes passed since you saw a shooting star. What is the probability that you will have to wait another 30 seconds?

This example might seem counter-intuitive. We already computed the probability of waiting for at least 30 seconds right after the event. That is about 61%. However, you already waited a while, so you might think that you'd have a better chance to see a shooting star before the 30 second period, and

therefore that the probability that you'd have to wait at least 30 seconds would be lower. But that is not the case. The probability is the same. Let s be the time already passed since the previous event. Then the probability of waiting another t time is $P(T_i > s + t | T_i > s)$, which is

$$P(T_i > s + t | T_i > s) = \frac{P(T_i > s + t, T_i > s)}{P(T_i > s)} = \frac{P(T_i > s + t)}{P(T_i > s)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} = e^{-\lambda t}$$

In essence, if the last event occurred s time units ago, the probability of a Poisson event occurring at least t time units from now is still $e^{-\lambda t}$, independent of s. That is, the waiting time for Poisson processes is "memoryless", in the sense that it is totally independent of what happened in the past.

3.3 Combining and Splitting Poisson Processes

Poisson processes have nice and intuitive properties when multiple flows are combined into one, and vice-versa when a single flow is split into multiple ones. In essence, the sum of two Poisson processes is also a Poisson process, and more specifically, if the two separate processes have rates λ_1 and λ_2, respectively, then the combined process is Poisson with rate $\lambda = \lambda_1 + \lambda_2$. Similarly, if you randomly split a Poisson process with rate λ into two, meaning that each event is considered an event of the first or second process with probabilities p and $1 - p$, respectively, then you obtain two Poisson processes with rates $p\lambda$ and $(1 - p)\lambda$, respectively. The same applies to the combination of, or the splitting into more than two processes.

We only prove the combination property, namely that the combination of two Poisson processes with rates λ_1 and λ_2 is a Poisson process with rate $\lambda_1 + \lambda_2$. One way to prove that is to focus on inter-arrival times. This proof is particularly interesting because it is very straightforward and at the same time shows that the combination property can be seen as a consequence of the memoryless nature of Poisson processes.

So, let's see. Since the two component processes are Poisson with rates λ_1 and λ_2, respectively, the inter-arrivals of the two processes, $T_{1,i}$ and $T_{2,i}$, are distributed exponentially, and in particular for the two initial inter-arrivals, we have $P(T_{1,1} > t) = e^{-\lambda_1 t}$ and $P(T_{2,1} > t) = e^{-\lambda_2 t}$. Now, if we combine the two processes, then that means that the first arrival of the combined process is the earliest arrival between the two component processes, which in turn means that the inter-arrival time is the minimum of the two. Yet in other words, the probability of the combined process *not* having an event before time t is the probability that neither of the two component processes has an event before t, and since the two processes are independent, we can write

$$P(T_{(1+2),1} > t) = P(T_{1,1} > t)P(T_{2,1} > t) = e^{-\lambda_1 t} e^{-\lambda_2 t} = e^{-(\lambda_1+\lambda_2)t}.$$

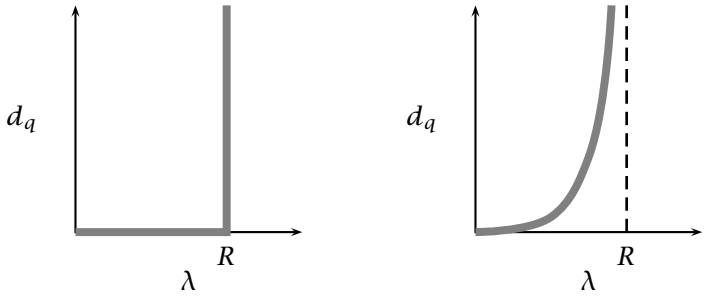
And the same calculation applies to any subsequent inter-arrival time of the combined process, thanks to the memoryless property of its components. We conclude that the inter-arrival times of the combined processes are distributed exponentially—which means that the process is Poisson—with rate $\lambda_1 + \lambda_2$.

4 Analytical Characterization of Basic Queuing Systems

We are now equipped with the necessary models and basic mathematical tools to analyze queuing delays in transmission systems , which is the primary purpose in the application of queuing theory to computer networking. In reviewing the basics of congestion control, we discussed queuing delays in a transmission system with the following steady-state equations:

$$d_q = \begin{cases} 0 & \lambda < R \\ \frac{\lambda - R}{R} t & \lambda > R \end{cases}$$

These equations give a very simplistic characterization of the queuing delay as a function of the input flow λ and the transmission rate R. This simplistic view corresponds to the picture on the left-hand side below, which simply says that the queue is empty when the input rate is lower than the output rate, and instead goes to infinity as soon as the input rate saturates the output rate.



ideal input flow
λ constant

realistic input flow
λ variable

Then, without further analysis, we said that reality is closer to the picture on the right-hand side. Now we provide that more realistic analysis.

4.1 Deterministic Arrivals, Deterministic Service Times: D/D/1

Let's first consider a perfectly deterministic arrival process with rate λ and a perfectly deterministic service process with service time $\tau = 1/\mu$. Deterministic here means that (1) arrivals are equally spaced over time, meaning that inter-arrival times are constant (time-invariant), and (2) service times are also constant.

Such queuing system is denoted D/D/1, where the first letter "D" indicates the type of arrival process ("deterministic"), the second letter "D" indicates the type of service process (also deterministic), and the number 1 indicates the number of servers. In this case, we imagine that the queue is infinite. With a finite queue of size q, we would then denote the system as D/D/1/q.

What is the queuing delay in this case? The system receives a customer/packet every $c = 1/\lambda$ time units, and each customer is served in exactly τ time unit. Assuming that the system is stable, meaning that $\lambda < 1/\tau$ and therefore $\tau < c$, then that means that the system contains exactly one customer for a duration of τ time units, and zero customers for a duration of $c - \tau$ time units. Thus we have the probabilities

$$P(N = 0) = 1 - \lambda\tau; \quad P(N = 1) = \lambda\tau$$

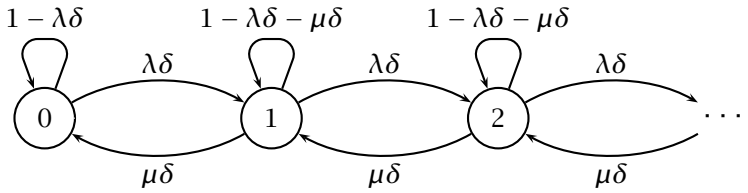
4.2 Poisson Arrivals, Exponential Service Times: M/M/1

The deterministic case is simply unrealistic, and in any case not very interesting. A more realistic case is when we consider a stochastic arrival process and a stochastic service process. In particular, we consider Poisson arrivals with rate λ and exponentially distributed service times with average τ and therefore rate $\mu = 1/\tau$. We denote such a queuing system M/M/1, where the letter "M" that describes the arrival and service processes means *Markovian* or *memoryless*, which are properties of Poisson processes and exponentially-distributed times, and the number 1 indicates the number of servers. We

further assume that the queue has infinite capacity, and that the system is stable, meaning that $\lambda < \mu$. As usual, we also indicate the utilization ratio as

$$\rho = \frac{\lambda}{\mu} < 1$$

The analysis of an M/M/1 queue relies on the Markovian and memoryless properties of arrivals and service times, respectively. Arrivals and departures are "Markovian" in the sense that the evolution of the system from a state S depend only on S, regardless of any previous state. Thus we model the queue as a Markov chain as follows:



The states of the chain, labeled $0, 1, 2, \ldots$, represent the state when there are $n = 0, 1, 2, \ldots$ customers in the system, respectively. A transition of the Markov chain (edge) represents an arrival or a service termination, and is in general a continuous process, meaning that the chain can in principle switch state at any time. However, we can also discretize this Markov chain by considering small-enough time intervals of duration δ. In such small intervals, we consider the probability of having two arrival or service termination events to be negligible, and instead the chain can switch only from state n to state $n + 1$ or vice-versa.

The probability of having one arrival, and therefore the probability of transitioning from state n to state $n + 1$ is $\lambda\delta$ for every state n. Similarly, in every state $n > 0$, the probability of transitioning to state $n - 1$ is the probability $\mu\delta$ of seeing a service termination. For the remaining probability $1 - \lambda\delta - \mu\delta$, the state does not change.

Thus we have a Markov chain, and our goal is to find the stationary distribution of the chain, and therefore the expected queue size.

As usual, in the analysis of the stationary distribution of a Markov chain, we can write a balance equation between any two connected states. In particular, let $P_0, P_1, P_2, \ldots$ be the stationary probabilities for each state $0, 1, 2, \ldots$, respectively. Consider states 0 and 1. Since P_0 is stationary, it must be that $\lambda\delta P_0 = \mu\delta P_1$, which then implies that $\lambda\delta P_1 = \mu\delta P_2$, and in general then implies that

$$\lambda\delta P_n = \mu\delta P_{n+1} \qquad P_{n+1} = \rho P_n \qquad P_n = \rho^n P_0$$

Combining this last equation with the total probability $\sum_{n=0}^{\infty} P_n = 1$, we have $\sum_{n=0}^{\infty} \rho^n P_0 = 1$, which allows us to compute P_0 as the inverse of the geometric series $\sum_{n=0}^{\infty} \rho^n = 1/(1 - \rho)$, which means that $P_0 = 1 - \rho$ and therefore

$$P_n = \rho^n (1 - \rho)$$

With the probabilities for every queue occupancy n, we can then find the expected queue occupancy $E(n) = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n\rho^n(1 - \rho)$, which means

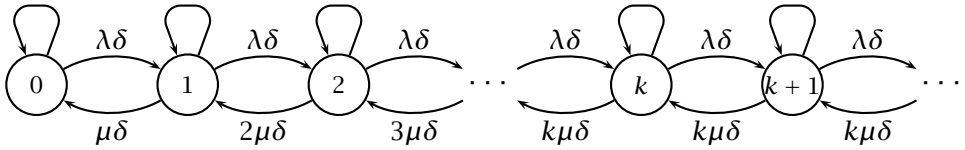
$$E(n) = \frac{\rho}{1 - \rho}$$

4.3 Multiple Servers: M/M/k

Consider now the case in which we have k servers. As in the previous analysis, we assume the queue has an infinite capacity. We denote this queuing system as M/M/k. The two letters "M" describe the arrival and service processes, respectively, and the letter "k" indicates that we have k servers. Thus the arrival is Poisson with rate λ, and each of the k servers serves customers with exponentially-distributed service times, where the parameter of the exponential distribution is a rate μ. Here, the utilization ratio is

$$\rho = \frac{\lambda}{k\mu} < 1$$

The analysis of an M/M/k queue uses the same analytical tool, namely a Markov chain. The chain is depicted below. We omit the probabilities of staying in a given state, since those probabilities are implied by the other transition probabilities, and in any case are irrelevant for the analysis.



Notice the difference with an M/M/1 queue. The transition probabilities going up in the number of customers in the system is $\lambda\delta$ that corresponds to a Poisson arrival with rate λ. However, the transition probabilities going down are proportional to the number of busy servers. Thus the probability of switching from state 2 to state 1 is $2\mu\delta$, and in general for every state $1 \le n \le k$, the probability to switch from state n to state $n-1$ is $n\mu\delta$. This is because any one of the n busy servers can finish serving a customer in a δ time slot. For states $n > k$, the transition probabilities remain $k\mu\delta$.

In summary,

$$P_n = \frac{1}{n}\frac{\lambda}{\mu}P_{n-1} \quad \text{and therefore} \quad P_n = \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n P_0 \quad \text{if } n \le k$$

and

$$P_n = \frac{1}{k}\frac{\lambda}{\mu}P_n \quad \text{and therefore} \quad P_n = \frac{1}{k!}\left(\frac{1}{k}\right)^{n-k}\left(\frac{\lambda}{\mu}\right)^n P_0 \quad \text{if } n > k.$$

and, using the utilization ratio $\rho = \frac{\lambda}{k\mu}$:

$$P_n = \begin{cases} \frac{(k\rho)^n}{n!}P_0 & \text{if } n \le k \\ \frac{k^k\rho^n}{k!}P_0 & \text{if } n > k \end{cases}$$

computing P_0 using $\sum_0^\infty P_n = 1$, we obtain:

$$\left[1 + \sum_0^{k-1}\frac{(k\rho)^n}{n!} + \sum_k^\infty \frac{k^k\rho^n}{k!}\right]P_0 = 1$$

and therefore:

$$P_0 = \left[\sum_0^{k-1}\frac{(k\rho)^n}{n!} + \frac{(k\rho)^k}{k!(1-\rho)}\right]^{-1}$$

Now that we have all the P_n probabilities, we can compute the overall performance probabilities. For starters, we can compute the queuing probability P_Q, meaning the probability that an arrival finds all k servers busy. That is simply the sum of all the probabilities P_n for $n \geq k$, which is

$$P_Q = \frac{(k\rho)^k}{k!(1-\rho)} P_0$$

With that, we can then compute the expected length of the queue $E(N_Q)$. This one is easy to figure out just by conditioning on the system having a queue. In essence, there are two cases: the system has a queue, in which case, $N_Q = 0$, or the system does have a queue. The question, then, is what is the expected queue size *given that the system has a queue*, which we know happens with probability P_Q. And this is easy to figure out. When an M/M/k system does have a queue, the system behaves exactly like an M/M/1 queue, which we know has an expected queue size of $\frac{\rho}{1-\rho}$. Therefore, the expected queue size for M/M/k is simply

$$E(N_Q) = \frac{\rho}{1-\rho} P_Q$$

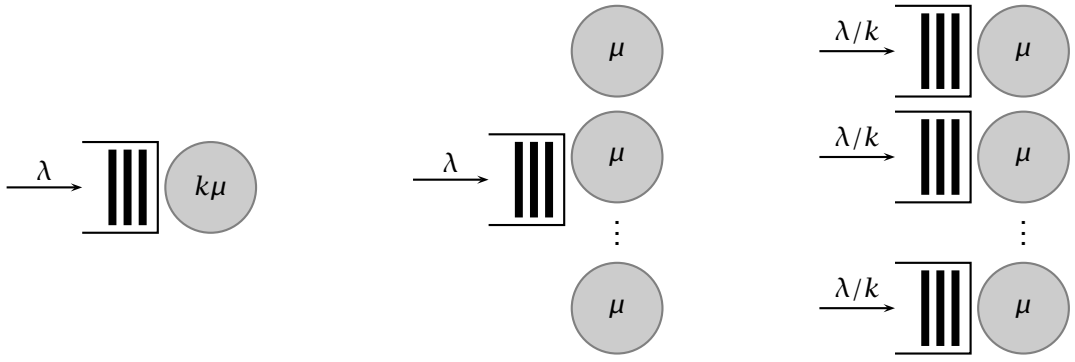
Now, using Little's theorem, we can compute the average queuing time $E(T_Q)$, and the average response time $E(T)$:

$$E(T_Q) = \frac{1}{\lambda} \frac{\rho}{1-\rho} P_Q$$

$$E(T) = \frac{1}{\lambda} \frac{\rho}{1-\rho} P_Q + \frac{1}{\mu}$$

4.4 Statistical Multiplexing

It is interesting to consider different ways to serve customers. The arrival rate is λ, and the total service rate is $k\mu$. However, this same $k\mu$ service can be used in different ways, as illustrated in the three diagrams below:



(a) one queue, one server with service rate $k\mu$

(b) one queue, k servers, each with service rate μ

(c) k queues, each with one server with service rate μ

In the first configuration (left), all customers arrive at a single queue served by one server with service rate $k\mu$. In the second configuration (middle), all customers arrive at a single queue served by k servers, each with service rate μ. In the third configuration (right), the customers are initially uniformly divided into k groups, each arriving with rate λ/k at one of k queues served by one server with service rate μ.

This latter configuration corresponds to a multiplexing scheme where each group has a reserved channel. For example, in telecommunication systems, this kind of multiplexing can be achieved by using different and independent frequencies, one for each group. We therefore refer to this configuration as frequency-division multiplexing (FDM). The first two schemes are instead based on a single communication channel, and instead differ in the way that they allocate computational resources. The first scheme is an M/M/1 queue. The second scheme is an M/M/k queue. In any case, the overall utilization ratio $\rho = \frac{\lambda}{k\mu}$ is the same in all three configurations.

The question we examine is how well each scheme performs in term of response time. We therefore measure the expected total time T that a customer spends in the system.

For the M/M/1 system, from Little's theorem, we have $T = N/\lambda$, where N is the expected state of the M/M/1 system:

$$E(T)_{\text{M/M/1}} = \frac{1}{\lambda}E(N) = \frac{1}{\lambda}\frac{\rho}{1-\rho} = \frac{1}{k\mu - \lambda}$$

For the M/M/k system, we consider the total time $T = T_Q + \tau$ as the sum of the queuing time plus the processing time. Again using little's theorem, the queuing time is $T_Q = N_Q/\lambda$, while the processing time is simply $\tau = 1/\mu$. Altogether:

$$E(T)_{\text{M/M/k}} = \frac{1}{\lambda}P_Q\frac{\rho}{1-\rho} + \frac{1}{\mu}$$

For the FDM system, the analysis is the same as an M/M/1 queue in which the arrival rate is λ/k and the service rate is μ. So,

$$E(T)_{\text{FDM}} = \frac{k}{\lambda}E(N) = \frac{k}{\lambda}\frac{\rho}{1-\rho} = \frac{k}{k\mu - \lambda}$$

Comparing the response times of the FDM and M/M/1 system, we can immediately see that M/M/1 is always better by a factor k. This result is very important. We'll come back to it at some point.

Let's now compare M/M/1 and M/M/k. Consider the ratio

$$\frac{E(T)_{\text{M/M/k}}}{E(T)_{\text{M/M/1}}} = \frac{\frac{1}{\lambda}P_Q\frac{\rho}{1-\rho} + \frac{1}{\mu}}{\frac{1}{\lambda}\frac{\rho}{1-\rho}} = P_Q + k(1-\rho)$$

We can analyze this ratio by considering to extreme cases.

Case 1: $\rho \approx 0$. In this case, the queuing probability is almost zero, $P_Q \approx 0$, so the ratio is close to k. This means that, when the load is low, the $M/M/1$ system is k times faster than the corresponding $M/M/k$. This makes sense also intuitively: when the load is low, there is little or no queuing, so the only time that matters is the processing time, and the $M/M/1$ server is k times faster.

Case 2: $\rho \approx 1$. In this case, there is most certainly a queue, $P_Q \approx 1$, and the ratio is close to 1. This means that, when the load is high, the $M/M/1$ system is about as fast as the corresponding $M/M/k$. This also makes sense intuitively: when the load is high, all k servers are likely to be busy (there is queuing) so the $M/M/k$ system behaves like an $M/M/1$ with arrival rate λ and service rate $k\mu$.