Reliability and Queueing

by David G. Messerschmitt

Supplementary section for Understanding Networked Applications: A First Course, Morgan Kaufmann, 1999.

The reliability of a system of components is important to anyone designing high-availability systems. The congestion phenomenon affects the scalability and capacity of both servers and networks. What both these issues have in common is random "arrivals". In the case of reliability, these arrivals are component failures, in the case of servers they are tasks to compute, and in the communication links they are packets. These three problems share a common modeling framework based on probability theory.

This appendix will characterize the effectiveness of component redundancy, in which a component is replicated to mitigate the effect of component failure. It will also consider the statistics of waiting time in queues, applying to server task or network queues. A basic knowledge of probability theory is presumed, and further details can be found in [Rad89], [Wal91], and [Wal96].

The Exponential Distribution

A exponentially distributed random variable $X \geq 0$ has a probability density function $f(x) = \lambda e^{-\lambda x}$ with mean value $1/\lambda$ and probability distribution function $F(x) = 1 - e^{-\lambda x}$. This distribution plays a major role in both reliability analysis and queueing analysis.

The exponential distribution has one important property that helps understand the following: it is *memoryless*. If we let $0 < x_0 < x$, then the conditional probability that $X < x$ given that $X \geq x_0$ is

$$\Pr\{X < x | X \geq x_0\} = \frac{F(x) - F(x_0)}{1 - F(x_0)} = 1 - e^{-\lambda(x - x_0)} = F(x - x_0) \ . \qquad \text{EQ 1}$$

This is an exponential distribution referenced to $x = x_0$ rather than $x = 0$—the distribution is unaffected by the knowledge that $X \geq x_0$.

Component Failure

In modeling the failure of components in a system, statistical techniques should be used because the time of failure cannot be predicted with certainty (see "How Effective is Redundancy?" in Chapter 13). The simplest assumption, and a reasonable one for electronic components and subsystems is that a component fails at a constant rate λ. This means that for a large number of components, in an interval of length Δ a fraction $\Delta\lambda$ of the components that are still working at the beginning of that interval will fail on average, in the limit as $\Delta \to 0$.

Let X be a random variable denoting the time to failure for one component, and let its distribution function be $F(x)$; that is, $F(x)$ is the probability that $X \leq x$. Then at some time x, a fraction $1 - F(x)$ of these components will not have failed, and the probability of component failure in the

interval $[x, x+\Delta]$ is $F(x+\Delta) - F(x)$. Thus, the condition for constant failure rate is

$$F(x+\Delta) - F(x) = \Delta\lambda \cdot (1 - F(x)) \text{ as } \Delta \to 0 . \qquad \text{EQ 2}$$

This can be rewritten as

$$\dot{F}(x) = \lambda \cdot (1 - F(x)) . \qquad \text{EQ 3}$$

The solution to this differential equation (with the appropriate boundary condition $F(0) = 0$) is the exponential distribution, $F(x) = 1 - e^{-\lambda x}$. The average time to failure is the mean value of this distribution, $T = 1/\lambda$.

On reflection, the memoryless property of the exponential distribution is consistent with the constant failure rate assumption. Knowing that a component has lasted for some period x_0 says nothing (in the statistical sense) about remaining time to failure—the component has no mechanism for "wearing out", which would result in an increasing failure rate with time and make its previous lifetime relevant. This assumption is reasonably valid for electronic components, although it would be questionable for components with mechanical elements (like disk drives or keyboards).

Component Redundancy

Suppose n replicas of a component are provided, only one of which need work for system availability. Presumably this redundancy increases the time to system failure, but how much?

Example... If a computer must be functioning as part of a networked application, then the availability of the application can be increased by replicating the computer (and its software) n times. If one or more of these computers fail, one of the still-functioning computers is used. It is only necessary for any one of these computers to be functioning for the application to be available.

Of course, a different analysis is required in different circumstances. For example, in some systems there may be n different components, all of which have to be working in order for the system to be available.

Assume there are n components, all of which must fail for a system to fail. Further assume that each has a constant failure rate λ, and the components fail independently. Then at time x, the probability that one component has failed is $F(x)$ and the probability that all n components have failed is $F^n(x)$, since they fail independently. The probability density of the time to failure is thus the derivative of this distribution function, $n \cdot F^{n-1}(x) \cdot \dot{F}(x)$, and the average time to failure of all n components is

$$T_n = \int_0^\infty nxF^{n-1}(x) \cdot \dot{F}(x) \cdot dx . \qquad \text{EQ 4}$$

Substituting the exponential distribution,

$$T_n = \int_0^\infty nx(1 - e^{-\lambda x})^{n-1}(\lambda e^{-\lambda x}) \cdot dx \qquad \text{EQ 5}$$

which (after some tedious algebra and reference to a table of integrals) evaluates to

$$T_n = \frac{1}{\lambda}\sum_{i=1}^{n}(-1)^{i+1}\cdot\binom{n}{i}\cdot\frac{1}{i} = \frac{1}{\lambda}\sum_{i=1}^{n}\frac{1}{i}\ . \qquad \text{EQ 6}$$

This last identity can be established by defining the polynomials

$$g(x) = \sum_{i=1}^{n}(-1)^{i+1}\cdot\binom{n}{i}\cdot\frac{1}{i}\cdot x^i \text{ and } h(x) = \sum_{i=1}^{n}\frac{1}{i}\cdot x^i\ , \qquad \text{EQ 7}$$

where it suffices to show that $g(1) = h(1)$. The derivatives are easily determined,

$$\dot{g}(x) = \frac{1-(1-x)^n}{x} \text{ and } \dot{h}(x) = \frac{1-x^n}{1-x} = \dot{g}(1-x)\,. \qquad \text{EQ 8}$$

Integrating the right side of Equation 8, it is readily shown that $g(1)-g(0) = h(1)-h(0)$ and the expected result follows from $g(0) = h(0) = 0$.

Task and Packet Arrivals

Suppose tasks or packets arrive a random times $0 < T_1 < T_2 < T_3 < \ldots$. These arrival times form a *Poisson process* with rate λ if the interarrival times $T_1, T_2 - T_1, T_3 - T_2, \ldots$ are independent random variables with identical exponential distributions with mean $1/\lambda$. The arrival rate (expected number of arrivals in unit time) is λ. Note the parallel to reliability, where the concern is with a single “arrival”; namely, a failure. With queueing, there is a sequence of random arrivals, but by assumption the distribution of inter-arrival times is the same as the distribution of time to failure. From the memoryless property, knowing the elapsed time from the last arrival says nothing about the time to the next arrival—it has the same exponential distribution and takes $1/\lambda$ on average. Thus, each arrival has no awareness (in the statistical sense) of the last arrival, which is a reasonable assumption if the arrivals (tasks or packets) are from independent sources.

In order to characterize the waiting time in a queue (see "Modeling Congestion" in Chapter 17), assumptions must also be made as to the statistics of task processing time or packet transmission time (which is queueing theory is called the *service time*). The simplest assumption is that there is a single server, and the service times are random, independent, and distributed with an exponential distribution with mean $1/\mu$. This says that as long as there are tasks or packets waiting, the constant rate at which they are serviced (processed or transmitted) is μ. A queue with this exponential service time and Poisson arrivals is called an M/M/1 queue. It is the simplest case—many more complicated possibilities are possible.

The *utilization* of the server is defined as $\rho = \lambda/\mu$. For the server to be able to keep up with the arrivals, we must have an arrival rate less than a service rate; that is, $\lambda < \mu$ or $\rho < 1$. Further derivations of queueing results is beyond the scope of this book, other than to note two useful results. The average number of tasks or packets waiting in the queue is

$$L = \frac{\rho}{1-\rho} \qquad \text{EQ 9}$$

and the average completion time of a task (from arrival to completion of service) or latency of a packet (from arrival until the end of transmission time) is

$$T = \frac{1}{\mu - \lambda} \ . \qquad \text{EQ 10}$$

Statistical Multiplexing

Statistical multiplexing can now be modeled using the M/M/1 queueing model (see "Sharing Communication Links: Statistical Multiplexing" in Chapter 18). Assume that N different streams of packets, each with independent Poisson process arrival rate λ, are multiplexed together. The aggregate arrivals can then be shown to be a Poisson process with rate $N\lambda$. Packets are assumed to have random lengths independent of one another and with an exponential distribution with mean L bits. This implies that the transmission time for a communication link with bitrate R is exponential with average transmission time $1/\mu = L/R$ or rate $\mu = R/L$. The performance parameter of interest is the average latency, which is

$$T = \frac{1}{R/L - N\lambda} = \frac{L}{R - N\lambda L} \ . \qquad \text{EQ 11}$$

In this equation, $N\lambda L$ is the average aggregate incoming bitrate (packet arrival rate times average packet length), which must be less than the link bitrate R. The latency is proportional to the average packet length, and increases as the aggregate bit rate increases. It approaches a minimum of $T = L/R$, the average transmission time, for low arrival rates.

The statistical multiplexing “advantage” arises from comparing this latency to the case where the link bitrate R is statically divided into N lower-bitrate streams each with rate R/N, with one incoming packet stream applied to each (this is called *time-division multiplexing*). For each of these “subchannels”, the packet arrival rate is λ and the average service time is $1/\mu = NL/R$ or rate $\mu = R/NL$, and thus the latency is

$$\frac{1}{R/NL - \lambda} = \frac{NL}{R - N\lambda L} = NT \ . \qquad \text{EQ 12}$$

Thus, the average latency has been increased by a factor N over the statistical multiplexing case.

This quantifies one advantage of statistical multiplexing—the reduction in packet latency—but there are others. For example, time-division multiplexing limits the average bitrate of each and every packet stream to R/N, but statistical multiplexing limits only the *aggregate* average bitrate. Although not reflected in the assumptions and analysis above, this is of practical importance because the bitrate resources are allocated more dynamically and flexibly.

Exercises

E1. Suppose a simple system is decomposed into two component. If either component fails, the system fails. Assume the components fail independently at a constant rate λ_1 and λ_2 respectively, and show that the mean time to system failure is $1/(\lambda_1 + \lambda_2)$.

E2. Assume a system is composed of n components, all with the same constant failure rate λ. Failure of

any one of the components means system failure.

a. Find the distribution of time to system failure.

b. Find the mean time to system failure.