These slides cover a significant part of the book:

M. Zukerman, “Introduction to Queueing Theory and Stochastic Teletraffic Models”

The book is available on
http://www.ee.cityu.edu.hk/~zukerman/classnotes.pdf
and
https://arxiv.org/pdf/1307.2968.pdf

From the second set of slides onwards, the chapter number in the title of every set of slides corresponds to the relevant chapter or section in the book covered by the set of slides.

Basic Probability Topics

Moshe Zukerman

Electronic Engineering Department
City University of Hong Kong
Hong Kong SAR, PRC

Text/Reference Books

Moshe Zukerman, Introduction to Queueing Theory and Stochastic Teletraffic Models (Chapter 1)
http://www.ee.cityu.edu.hk/~zukerman/classnotes.pdf

D. Bertsekas and J. N. Tsitsiklis, Introduction to Probability, Athena Scientific, Belmont, Massachusetts 2002.

S. M. Ross, A first course in probability, Macmillan, New York, 1976.

Events, Sample Space, and Random Variables

- Consider an experiment (e.g. tossing a coin, or rolling a die).
- Sample space - set of all possible outcomes.
- Event - a subset of the sample space.
- example: experiment consisting of rolling a die once.

Sample space = {1, 2, 3, 4, 5, 6}

Possible events:

- {2, 3},
- {6},
- empty set {} (often denoted by Φ)
- the entire sample space {1, 2, 3, 4, 5, 6}

Question

Consider the experiment to be tossing a coin. What is the Sample Space? What are the events associated with this Sample Space?

Question

Consider the experiment to be tossing a coin. What is the Sample Space? What are the events associated with this Sample Space?

Answer

Notice that although the sample space includes only the outcome of the experiments which are Head (H) and Tail (T), the events associated with this samples space includes all subsets of the state space which include also the empty set which in this case is the event $\{H \cap T\}$ and the entire sample space which in this case is the event $\{H \cup T\}$.

Events are called mutually exclusive if their intersection is the empty set.

A set of events is exhaustive if its union is equal to the sample space.

Example 1: tossing a coin only once

The events {H} (Head) and {T} (Tail) are both mutually exclusive and exhaustive.
What is the sample space (the set) of all possible events in this case?

Example 2: rolling a die only once

The events {1}, {2}, {3}, {4}, {5}, and {6} are both mutually exclusive and exhaustive.
The events {4}, {5}, and {6} are mutually exclusive but are not exhaustive.

A random variable is a real valued function defined on the sample space.

This function $X = X(\omega)$ assigns a number to each outcome ω of the experiment.

Example: tossing a coin experiment
$X = 1$ for Head {H}
$X = 0$ for Tail {T}

Note that the function X is deterministic (not random), but the ω is unknown before the experiment is performed. Therefore $X(\omega)$ is called a random variable.

If X is a random variable then $Y = g(X)$ for some function g is also a random variable.

Examples:

$Y = cX$ for some scalar c is a random variable.

$Y = X^n$ for some integer n is a random variable.

If $X_1,\ X_2,\ X_3,\ \ldots,\ X_n$ is a sequence of random variables, then
$Y = \sum_{i=1}^{n} X_i$ is also a random variable.

Probability, Conditional Probability and Independence

Consider a sample space *S*. Let *A* be a subset of *S*. The probability of *A* is the function on *S* and all its subsets, denoted *P(A)* that satisfies the following three axioms:

1. $0 \leq P(A) \leq 1$
2. $P(S) = 1$
3. The probability of the union of mutually exclusive events is equal to the sum of the probabilities of these events.

Questions

Question 1

Consider again the experiment to be tossing a coin. Assume that $P(H) = P(T) = 0.5$. Illustrate each of the Probability Axioms for this case.

One intuitive interpretation of probability of an event is its **limiting relative frequency**

Let an outcome of an experiment be a person height.

Let p_i be the probability that a person's height is i cm.

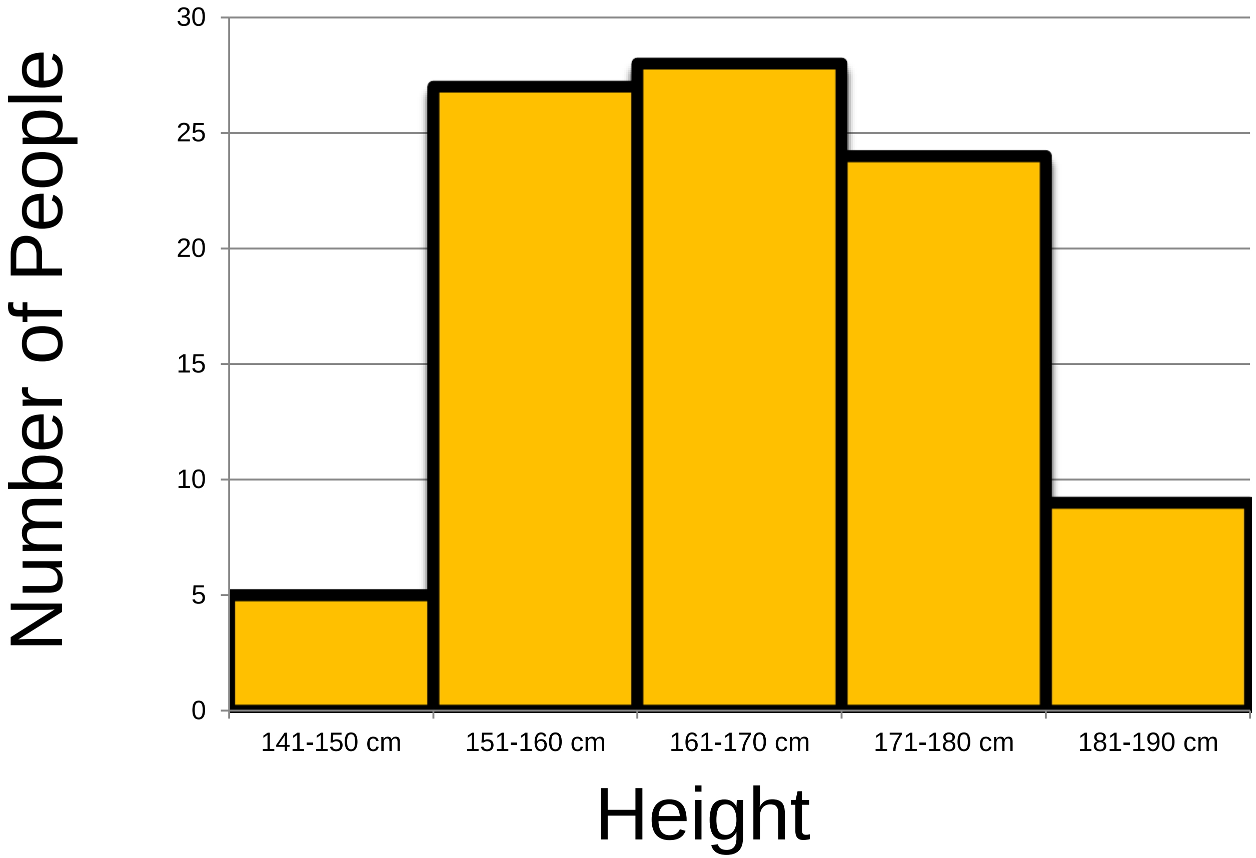
Consider a sample of N people.

Each one of them reports his/her height, rounded to the nearest cm.

Let n_i be the number of people reported a height of i cm.

These n_i values can be graphically presented as what is known as a **histogram**.

An example of a histogram is shown in the following graph.



An example of a Histogram with 5 ranges (bins) and each range is 10 cm.

In every range (bin), 10 n_i values are added up.

In this case, N = 93 people.

Limiting Relative Frequency (continued)

The relative frequency n_i/N approximates p_i.

This approximation becomes more and more accurate as N increases.

This approximation is consistent with the requirement

$$\sum_i p_i = 1.$$

If we set $p_i = n_i/N$, then since $\sum_i n_i = N$, we obtain

$$\sum_i p_i = 1.$$

The Average Height

$$\text{The average height} = \frac{\sum_i i n_i}{N}.$$

For large N, we set $p_i = n_i/N$, then,

$$\text{The average height} = \sum_i i p_i.$$

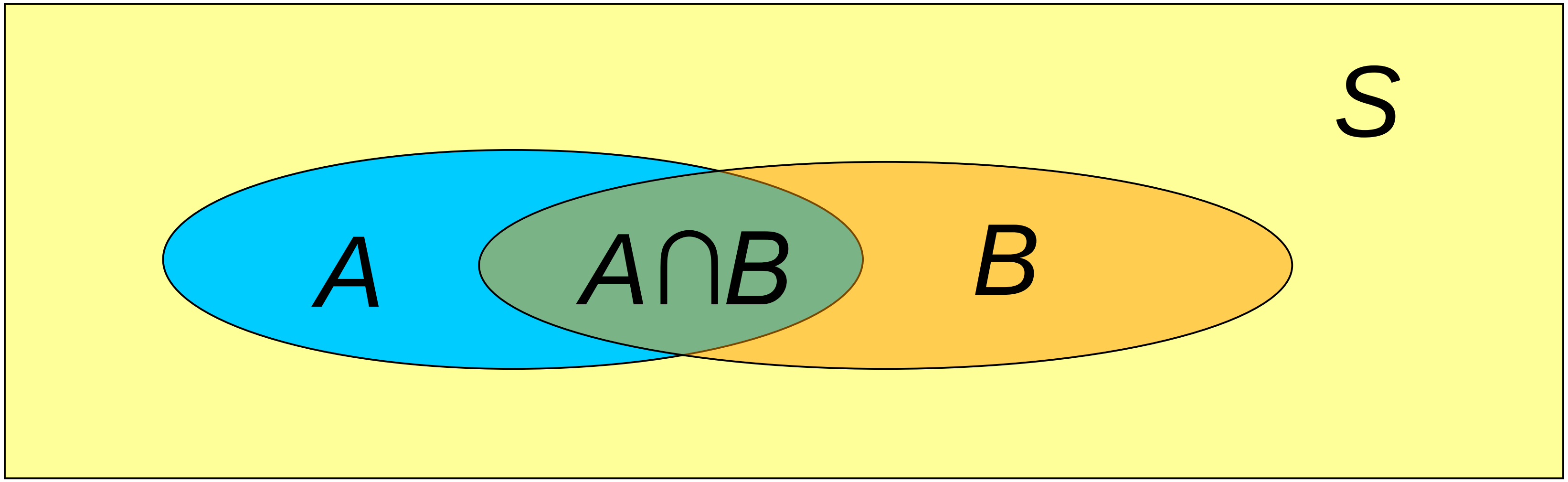
This is related to the well-known

Law of Large Numbers

Conditional Probability

The conditional probability of event A given event B is denoted by

$$P(A \mid B)$$



"Given event *B*" is equivalent to "*B* becomes the sample space".

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Example: consider rolling a die and *B*={1,2,3} (*B* = outcome is 1 or 2 or 3), and *A*={1}, then

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{1/6}{1/2} = \frac{1}{3}.$$

Now, since

$$P(A \cap B) = P(B \cap A) = P(B \mid A)P(A)$$

we obtain

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}.$$

Events *A* and *B* are said to be **independent** if and only if

$$P(A \mid B) = P(A).$$

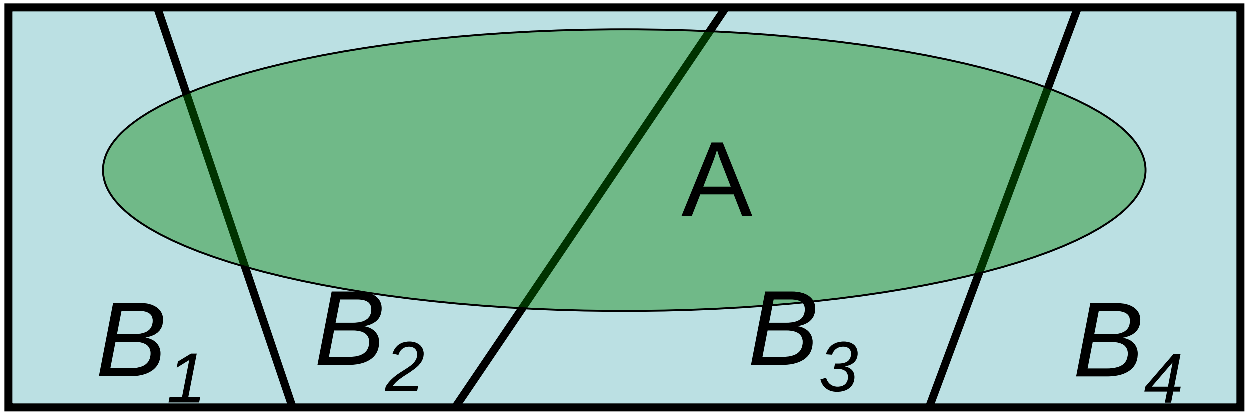
Equivalent definitions are:

$$P(A \cap B) = P(A)P(B)$$

$$P(B \mid A) = P(B).$$

Independence between two events means that if one of them occurs, the probability of the other to occur is not affected.

Homework: Show the equivalence between these three relationships.



$B_1,\ B_2,\ B_3,\ \ldots,\ B_n$ are mutually exclusive and exhaustive events in S. A is another event in S. Then, $A = \bigcup_{i=1}^{n}(A \cap B_i)$

Notice that $(A \cap B_1), (A \cap B_2), \cdots, (A \cap B_n)$ are also mutually exclusive (but not exhaustive).

Considering the 3rd probability axiom, we obtain:

$P(A) = \sum_{i=1}^{n} P(A \cap B_i)$,

Recall, $P(A \cap B) = P(A \mid B)P(B)$,

law of total probability:

$P(A) = \sum_{i=1}^{n} P(A \mid B_i) \times P(B_i)$

Bayes' theorem:

$$P(B_1 \mid A) = \frac{P(A|B_1)P(B_1)}{\sum_{i=1}^{n} P(A|B_i) \times P(B_i)}.$$

$P(B_1 \mid A)$ is the posterior probability of B_1.

$P(B_1)$ is the prior probability of B_1.

Other names for Bayes’ Theorem: Bayes' law and Bayes' rule

Homework: Make sure you know how to derive the law of total probability and Bayes’ theorem.

Question 2

Now consider an experiment involving three coin tosses. The outcome of the experiment is now a 3-long string of Heads and Tails. Assume that all coin tosses have probability 0.5, and that the coin tosses are independent events.

1. Write the sample space where each outcome of the experiment is an ordered 3-long string of Heads and Tails.

2. What is the probability of each outcome?

3. Consider the event
   $A = \{Exactly\ one\ head\ occurs\}$.
   Find $P(A)$ using the additivity axiom.

Partial Answer: $P(A) = 1/8 + 1/8 + 1/8 = 3/8$.

Question 3

Now consider again three coin tosses. Find the probability $P(A \mid B)$ where A and B are the events:

$A =$ more than one head came up

$B =$ 1st toss is a head.

Guide:

$P(B) = 4/8$; $P(A \cap B) = 3/8$;

$P(A \mid B) = (3/8)/(4/8) = 3/4$.

Question 4

Consider a medical test for a certain disease. The medical test detects the disease with probability 0.99 and fails to detect the disease with probability 0.01. If the disease is not present, the test indicates that it is present with probability 0.02 and that it is not present with probability 0.98. Consider two cases:

Case a: The test is done on a randomly chosen person from the population where the occurrence of the disease is 1/10000.

Case b: The test is done on patients that are referred by a doctor that have a prior probability (before they do the test) of 0.3 to have the disease.

Find the probability of a person to have the disease if the test shows positive outcome in each of these cases.

Guide:

A = person has the disease.
B = test is positive.
$\bar{A}$ = person does not have the disease.
$\bar{B}$ = test is negative.
We need to find $P(A \mid B)$.

Case a:

We know: $P(A) = 0.0001$.
$P(\bar{A}) = 0.9999$.
$P(B \mid A) = 0.99$.
$P(B \mid \bar{A}) = 0.02$. By the law of total probability:
$P(B) = P(B \mid A)P(A) + P(B \mid \bar{A})P(\bar{A})$.
$P(B) = 0.99 \times 0.0001 + 0.02 \times 0.9999 = 0.020097$.
Now put it all together and use Bayes' Theorem to obtain:
$P(A \mid B) = 0.004926108$.

Case b:

$P(A) = 0.3$.
Repeat the previous derivations to show that for this case $P(A \mid B) = 0.954983923$.

Question 5

In a multiple choice exam, there are 4 answers to a question. A student knows the right answer with probability 0.8 (Case 1), with probability 0.2 (Case 2), and with probability 0.5 (Case 3). If the student knows the answer s/he answers correctly with probability 1. However, if the student does not know the answer s/he always guesses with probability of success being 0.25. Given that the student marked the right answer, what is the probability he/she knows the answer.

Guide:

A = Student knows the answer.
B = Student marks correctly.
$\bar{A}$ = Student does not know the answer.
$\bar{B}$ = Student marks incorrectly.
We need to find $P(A \mid B)$.
Case 1: We know: $P(A) = 0.8$. $P(\bar{A}) = 0.2$. $P(B \mid A) = 1$.
$P(B \mid \bar{A}) = 0.25$. By the law of total probability:
$P(B) = P(B \mid A)P(A) + P(B \mid \bar{A})P(\bar{A})$.
$P(B) = 1 \times 0.8 + 0.25 \times 0.2 = 0.85$.
Now put it all together and by Bayes' Theorem obtain:
$P(A \mid B) = 0.941176471$.
Case 2: Repeat the previous derivations to obtain:
$P(A) = 0.2$. $P(B) = 0.4$. $P(A \mid B) = 0.5$.
Case 3: Repeat the previous derivations to obtain:
$P(A) = 0.5$. $P(B) = 0.625$. $P(A \mid B) = 0.8$.

Probability and Distribution Functions

X is a random variable (r.v.).

x is a number that represents an outcome of an experiment.

$\{X = x\}$ is an event.

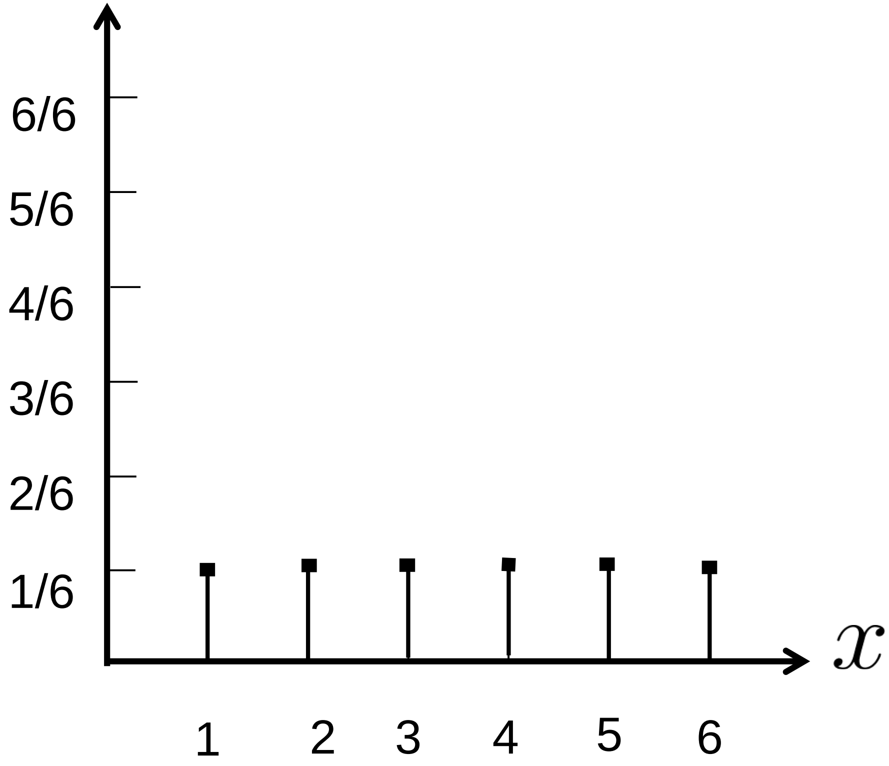
$P_X(x) = P(X = x)$ is a probability function.

Other names: probability distribution function, probability mass function.

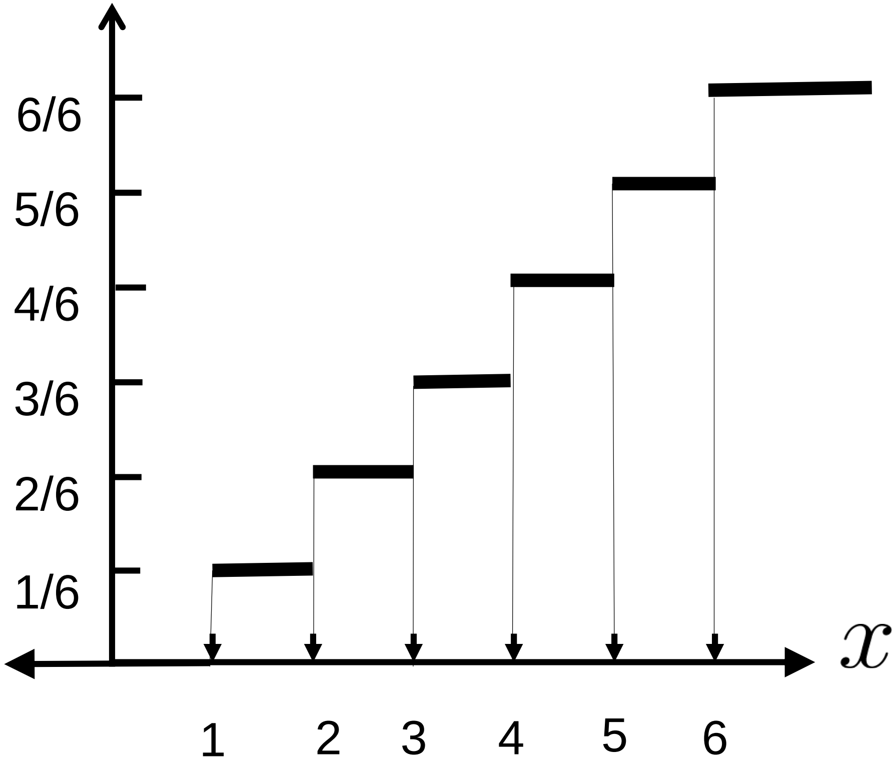
$F_X(x) = P(X \leq x)$ is the cumulative distribution function (CDF) of r.v. X.

Rolling a die

$$P_X(x) = P(X = x)$$



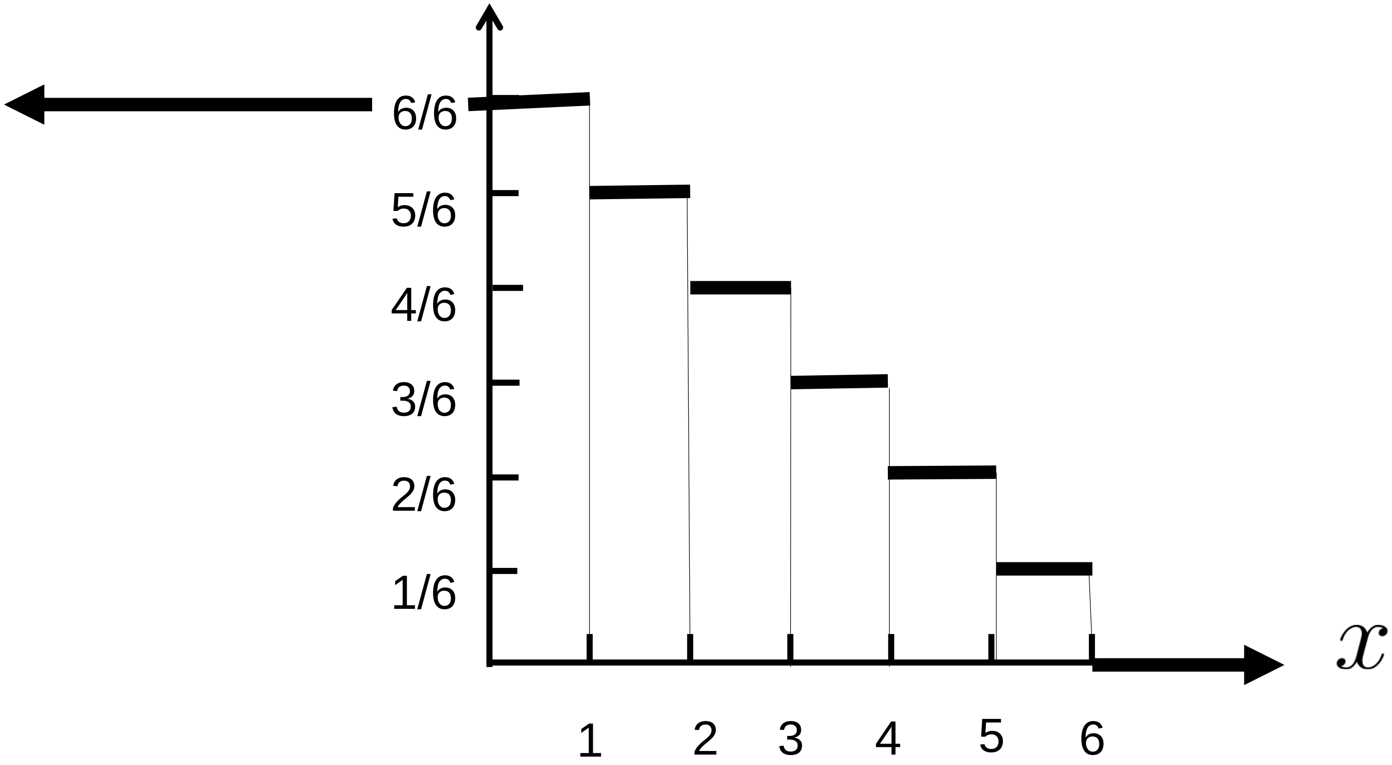
$$F_X(x) = P(X \leq x)$$



$\bar{F}_X(x) = P(X > x)$ is the <u>**complementary distribution function**</u> of random variable X

For the case of Rolling a die

$$\bar{F}_X(x) = P(X > x)$$



For every r.v. X and any $x \in R$,

$$F_X(x) + \bar{F}_X(x) = 1.$$

Notation: $A, B = A \cap B$

$$F_{X_1,\ ...,X_n}(x_1,\ ..., x_n) = P(X_1 \leq x_1,\ \ldots,\ X_n \leq x_n).$$

is the joint distribution function of the random variables $X_1,\ X_2,\ \ldots,\ X_n$.

Then

$$F_{X_1}(x_1) = F_{X_1,\ X_2,\ ...,\ X_n}(x_1,\ \infty,\ \ldots,\ \infty).$$

A random variable is called ***discrete*** if it takes at most a countable number of possible values.

A ***continuous*** random variable takes an uncountable number of possible values.

For discrete random variables $X_1,\ X_2,\ ...,X_n$ the joint probability function is:

$$P_{X_1,\ X_2,\ ...,\ X_n}(x_1,\ x_2,\ \ldots,\ x_n) = P(X_1 = x_1,\ X_2 = x_2,\ \ldots,\ X_n = x_n)$$

and the probability function of a single discrete random variable is:

$$P_{X_1}(x_1) = \sum\nolimits_{x_2} \cdots \sum\nolimits_{x_n} P_{X_1,\ X_2,\ ...,\ X_n}(x_1,\ x_2,\ \ldots,\ x_n).$$

Conditional Probability for Discrete Random Variables

$$P_{X|Y}(x \mid y) = P(X = x \mid Y = y) = \frac{P(X{=}x, Y{=}y)}{P(Y{=}y)} = \frac{P_{X,Y}(x,y)}{P_Y(y)}.$$

Because of the above and since $P_Y(y) = \sum_x P_{X,Y}(x,y)$,

we obtain $\sum_x P_{X|Y}(x \mid y) = 1.$

The implication is that the event {*Y*=*y*} is the new sample space and *X* has a legitimate distribution function in this new sample space.

$$P_Y(y) = \sum_x P_{X,Y}(x,y) = \sum_x P_{Y|X}(y \mid x) P_X(x)$$

is another version of the law of total probability.

Independence between Random Variables

"random variables U and V are independent" is equivalent to:
"the events $U = u$ and $V = v$ are independent for every u and v."
Accordingly, random variables U and V are independent if and only if:
$P_{U,V}(u, v) = P_U(u)P_V(v) \quad$ for all u, v.

An equivalent definition of independence between U and V is $P_{U|V}(u \mid v) = P_U(u)$ for all u, v.

Example

You roll a fair 6-side die twice. X is a result of the first roll and Y is the result of the second roll. Define U = max(X,Y) and V = min(X,Y).

Find: P(U=5|V=3)

Convolution

Consider independent random variables V_1 and V_2 with probability functions $P_{V_1}(v_1)$ and $P_{V_2}(v_2)$, respectively.

Let $V = V_1 + V_2$.

The **convolution** of $P_{V_1}(v_1)$ and $P_{V_2}(v_2)$ is

$$\begin{aligned} P_V(v) &= P(V_1 + V_2 = v) \\ &= \sum_{v_1} P(V_1 = v_1, V_2 = v - v_1) \\ &= \sum_{v_1} P_{V_1}(v_1) P_{V_2}(v - v_1). \end{aligned}$$

Question

Explain the last equation of convolution using the Law of Total Probability.

Guide

The factor $P_{V_2}(v - v_1)$ represents $P(V = v \mid V_1 = v_1)$ (conditioning) and the second is $P_{V_1}(v_1)$ (unconditioning).

Now consider *k* random variables $X_i,\ i = 1, 2, 3,\ \ldots,\ k.$

Let $P_{X_i}(x_i)$ be the probability function of X_i

$$Y = \sum_{i=1}^{k} X_i$$

The convolution of the *k* probability functions is:

$$P_Y(y) = \sum_{x_2,\ x_3,\ \ldots,\ x_k:\ x_2+x_3+\ \ldots\ +x_k \le y} \left(P_{X_1}(y - \Sigma_{i=2}^{k} x_i) \prod_{i=2}^{k} P_{X_i}(x_i) \right).$$

If all the X_i are independent and identically distributed (IID) random variables, with probability function $P_{X_1}(x)$, then $P_Y(y)$ is called the *k-fold convolution* of $P_{X_1}(x)$.

Some discrete random variables

1. Bernoulli (with parameter p)

$$\begin{aligned} P(X=1) &= p && \text{“success”} \\ P(X=0) &= 1-p && \text{“failure”} \end{aligned}$$

2. Geometric (with parameter p)

The number of independent Bernoulli trials until the first success

$$P(X=i)=(1-p)^{i-1}p \quad \text{for} \;\; i=1,2,3, \;\; \ldots$$

$P(X>i)=(1-p)^i$ for $i=0,1,2, \;\ldots\;,$ and $P(X>i)=1$ for $i<0$.

Geometric random variable is memoryless.

$$P(X>m+n \mid X>m)=P(X>n), \;\; m=0,1,2,\ldots, \;\; n=0,1,2,\ldots$$

It is the ONLY memoryless discrete random variable.

3. Binomial (with parameters p and n)

The number of successes in *n* independent Bernoulli trials

$$P(X = i) = \binom{n}{i} p^i (1-p)^{n-i} \qquad i = 0,\ 1,\ 2,\ \ldots,\ n.$$

Can be used to model users activity.

A user is active with probability *p* and non-active with probability 1-*p*.

X = *i* is the event where *i* users are active.

4. Poisson (with parameter λ)

$$P(X = i) = e^{-\lambda}\frac{\lambda^i}{i!} \qquad i = 0,\ 1,\ 2,\ 3,\ \ldots$$

How to compute these values?

Use Recursion and start from values around λ.

Set arbitrary initial value then normalize.

Poisson-Binomial Relationship

Consider a sequence of binomial random variables $X_n, \quad n = 1, 2, \ldots$ with parameters (n, p) where $\lambda = np$, or $p = \lambda/n$. Then the probability function

$$\lim_{n\to\infty} P(X_n = k)$$

is a Poisson probability function with parameter λ.

$\Rightarrow$ Poisson can be used to model traffic from a large number of sources.

To prove this we write:

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

Substituting $p = \lambda/n$, we obtain

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

or

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

Now notice that

$$\lim_{n\to\infty} \left(1 - \frac{\lambda}{n}\right)^n = e^{-\lambda},$$

$$\lim_{n\to\infty}\left(1-\frac{\lambda}{n}\right)^{-k}=1,$$

and

$$\lim_{n\to\infty}\frac{n!}{(n-k)!n^k}=1.$$

Therefore,

$$\lim_{n\to\infty}P(X_n=k)=\frac{\lambda^k e^{-\lambda}}{k!}.$$

QED.

Sum of two Poisson Random variables

Let $Y = X_1 + X_2$ where X_1 and X_2 are two independent Poisson random variables with parameters λ_1 and λ_2, respectively.

Use convolution to show that Y follows a Poisson distribution with parameter $\lambda_1 + \lambda_2$.

$$\begin{aligned}
P_Y(k) &= P(X_1 + X_2 = k) \\
&= \sum_{i=0}^{k} P(\{X_1 = i\} \cap \{X_2 = k - i\}) \\
&= \sum_{i=0}^{k} P_{X_1}(i) P_{X_2}(k - i) \\
&= \sum_{i=0}^{k} \frac{\lambda_1^i}{i!} e^{-\lambda_1} \frac{\lambda_2^{k-i}}{(k-i)!} e^{-\lambda_2} \\
&= e^{-(\lambda_1+\lambda_2)} \sum_{i=0}^{k} \frac{\lambda_1^i \lambda_2^{k-i}}{i!(k-i)!} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{k!} \sum_{i=0}^{k} \frac{k! \lambda_1^i \lambda_2^{k-i}}{i!(k-i)!} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{k!} \sum_{i=0}^{k} \binom{k}{i} \lambda_1^i \lambda_2^{k-i} \\
&= \frac{e^{-(\lambda_1+\lambda_2)} (\lambda_1 + \lambda_2)^k}{k!}. \qquad \textbf{QED}
\end{aligned}$$

5. Pascal (with parameters $k \geq 1, p \in (0,1]$)

The number of independent Bernoulli trials until the kth success, or equivalently, sum of k geometric random variables.

For Pascal random variable X, the event $\{X = i\}$ requires a "success" in the ith trial, and $k-1$ "successes" in $i-1$ trials. These two events are independent. The first is Bernoulli and the second is Binomial. Therefore,

$$P(X = i) = \binom{i-1}{k-1} p^k (1-p)^{i-k} \qquad i = k,\ k+1,\ k+2,\ \ldots .$$

6. Discrete Uniform

(with parameters a and b with $b > a$)

The discrete uniform probability function with integer parameters a and b has equal non-zero values for $x = a, a+1, a+2, \ldots, b$. It is given by

$$P_X(x) = \begin{cases} \frac{1}{b-a+1} & \text{if } x = a, a+1, a+2, \ldots, b \\ 0 & \text{otherwise.} \end{cases} .$$

Rolling a fair die is one example that govern by this probability function with $a = 1$ and $b = 6$.

Continuous Random Variables and Distributions

Now the set of possible outcomes is uncountable.

A continuous random variable X which assigns a real number to any outcome of an experiment, is characterized by the existence of a function called the *probability density function* (or simply the *density*) of X defined for all $x \in R$, which has the property that for any set $A \subset R$,

$$P(X \in A) = \int_A f(x)dx.$$

To guarantee that all the relevant probabilities are nonnegative (recall the first probability axiom), we only consider nonnegative density functions.

Let $A = [a, b]$, we obtain,

$$P(a \leq X \leq b) = \int_a^b f(x)dx.$$

Notice that the probability of a continuous random variable taking a particular value is equal to zero. If we set $a = b$ in the above, we obtain

$$P(X = a) = \int_a^a f(x)dx = 0.$$

Therefore, for continuous random variable X,

$F_X(x) = P(X \leq x) = P(X < x)$,

and $P(X \geq x) = P(X > x)$.

We obtain

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f(s)ds.$$

Hence, the probability density function $f(x)$ is the derivative of the distribution function $F_X(x)$.

Notation: $F_X(x) = F(x)$ and $f_X(x) = f(x)$

Let X and Y be two continuous random variables. The joint density of X and Y denoted $f_{X,Y}(x,y)$ is a nonnegative function that satisfies

$$P(\{X,Y\} \in A) = \iint_{\{X,Y\}\in A} f_{X,Y}(x,y)dxdy.$$

for any set $A \subset R^2$.

Equivalently to the discrete case:

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dx.$$

Let X and Y be two continuous random variables with joint density $f_{X,Y}(x,y)$. The conditional density of X given Y is defined as

$$f_{X|Y}(x \mid y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}.$$

For every given fixed y, it is a legitimate density because

$$\int_{-\infty}^{\infty} f_{X|Y}(x \mid y)dx = \int_{-\infty}^{\infty} \frac{f_{X,Y}(x,y)dx}{f_Y(y)} = \frac{f_Y(y)}{f_Y(y)} = 1.$$

By the definition of conditional density,

$$f_{X,Y}(x,y) = f_Y(y) f_{X|Y}(x \mid y)$$

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dy = \int_{-\infty}^{\infty} f_Y(y) f_{X|Y}(x \mid y)dy.$$

$$P(A) = P(X \in A) = \int_A f_X(x)dx = \int_A \int_{-\infty}^{\infty} f_Y(y) f_{X|Y}(x \mid y)dydx.$$

$$P(A) = \int_{-\infty}^{\infty} f_Y(y) \int_A f_{X|Y}(x \mid y)dxdy$$

and therefore

$$P(A) = \int_{-\infty}^{\infty} f_Y(y) P(A \mid Y = y)dy$$

which is the continuous equivalence of the Law of Total Probability.

Example

Consider the following joint density:

$$f_{X,Y}(x,y) = \begin{cases} 2 & 0 \leq x+y \leq 1, \quad x \geq 0, \quad y \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

1. Show that this is a legitimate density by showing first that all relevant probabilities are nonnegative and that the 2-dimensional integral of this joint density over the entire state space is equal to 1.

2. Derive the marginal density $f_Y(y)$.

3. Derive the conditional density $f_{X|Y}(x \mid y)$.

Please complete all steps in the following.

To show that this is a legitimate density observe that the joint density is nonnegative and also

$$\int_0^1 \int_0^{1-x} 2dydx = 1.$$

$$f_Y(y) = \begin{cases} \int_0^{1-y} f_{X,Y}(x,y)dx = 2 - 2y & 0 \leq y \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{X|Y}(x \mid y) = \begin{cases} \frac{2}{2-2y} = \frac{1}{1-y} & 0 \leq x \leq 1 - y \\ 0 & \text{otherwise.} \end{cases}$$

Convolution of continuous random variables

Consider independent random variables U and V. Let $X = U + V$.

$$f_X(x) = \int_u f_U(u) f_V(x-u).$$

The latter is the *convolution* of the densities $f_U(u)$ and $f_V(v)$.

Convolution of *k* continuous random variables

As in the discrete case the convolution $f_Y(y)$, of k densities $f_{X_i}(x_i)$, $i = 1, 2, 3, \ \ldots, \ k$, of random variables X_i, $i = 1, 2, 3, \ \ldots, \ k$, respectively, is given by

$$f_Y(y) = \iint_{x_2, \ ..., \ x_k: \ x_2 + \ ..., + x_k \leq y} \left(f_{X_1}(y - \Sigma_{i=2}^{k} x_i) \prod_{i=2}^{k} f_{X_i}(x_i) \right).$$

And again, in the special case where all the random variable X_i, $i = 1, 2, 3, \ \ldots, \ k$, are IID, the density f_Y is the k-fold convolution of f_{X_1}.

Equivalence between discrete and continuous random variables and their probability functions/densities and distributions

Discrete	**Continuous**
$P_Y(y) = \sum_x P_{X,Y}(x,y)$	$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dx$
$P_{X\|Y}(x \mid y) = \frac{P_{X,Y}(x,y)}{P_Y(y)}$	$f_{X\|Y}(x \mid y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$
$P(A) = \sum_{i=1}^{n} P(A \mid B_i) \times P(B_i)$	$P(A) = \int_{-\infty}^{\infty} f_Y(y)P(A \mid Y = y)dy$
If X and Y are independent $P_{X\|Y}(x \mid y) = P_X(x)$ $P_{X,Y}(x,y) = P_X(x)P_Y(y)$ $P(V_1 + V_2 = v) = \sum_{v_1} P_{V_1}(v_1)P_{V_2}(v - v_1)$	**If X and Y are independent** $f_{X\|Y}(x \mid y) = f_X(x)$ $f_{X,Y}(x,y) = f_X(x)f_Y(y)$ $P(U + V = x) = \int_u f_U(u)f_V(x - u)du$

Some Continuous Random Variables

1. Uniform (with parameters *a,b*)

Its probability density function is

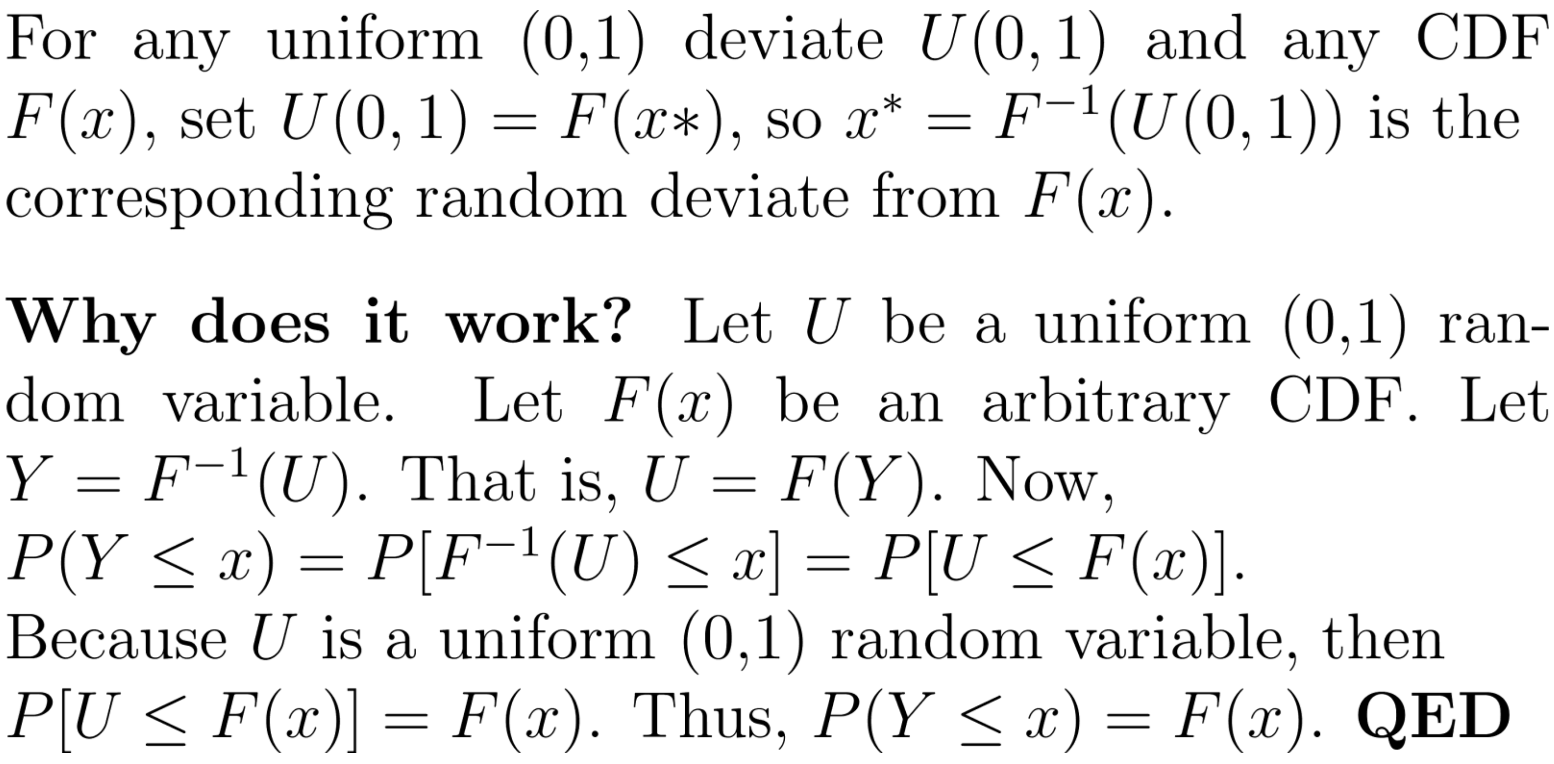
$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases}$$

A special case - uniform (0,1).
Its probability density function is

$$f(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Inverse transform sampling

Using uniform (0,1) deviates to generate sequence of random deviates of any distribution



For any uniform (0,1) deviate $U(0,1)$ and any CDF $F(x)$, set $U(0,1) = F(x*)$, so $x^* = F^{-1}(U(0,1))$ is the corresponding random deviate from $F(x)$.

Why does it work? Let U be a uniform (0,1) random variable. Let $F(x)$ be an arbitrary CDF. Let $Y = F^{-1}(U)$. That is, $U = F(Y)$. Now,
$P(Y \leq x) = P[F^{-1}(U) \leq x] = P[U \leq F(x)]$.
Because U is a uniform (0,1) random variable, then
$P[U \leq F(x)] = F(x)$. Thus, $P(Y \leq x) = F(x)$. **QED**

Derive the convolution of two independent uniform (0,1) random variables.

$$f_X(x) = \int_u f_U(u) f_V(x-u) du$$

Since U and V are uniform(0,1) random variables, for $f_U(u)f_V(x-u)$ to be non-zero, u and x must satisfy:

$0 \leq u \leq 1$ and $0 \leq x - u \leq 1$,

or

$\max(0, x-1) \leq u \leq \min(1, x)$

and

$0 \leq x \leq 2$.

Therefore,

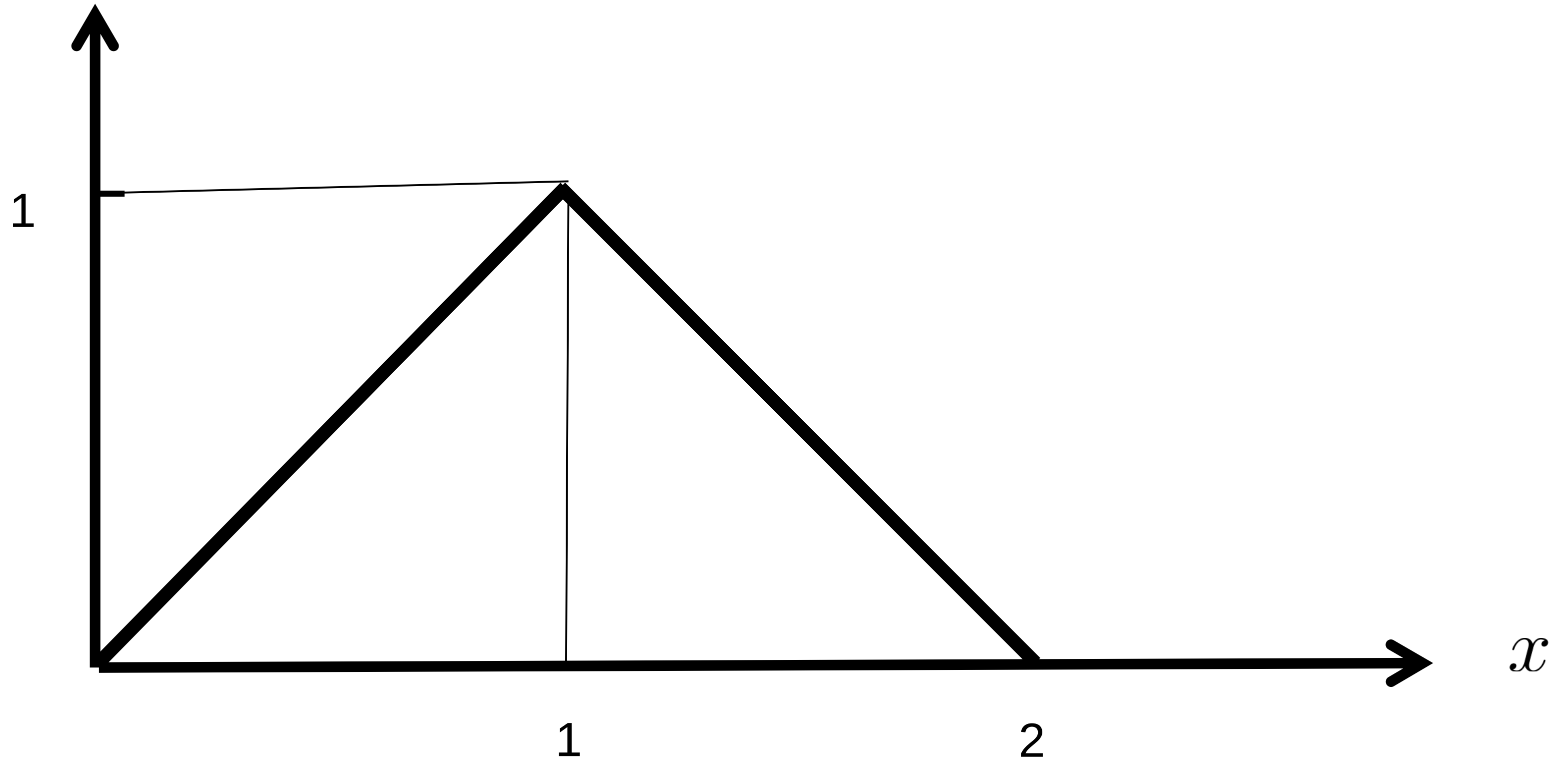
$$f_X(x) = \begin{cases} u\Big|_{\max(0,x-1)}^{\min(1,x)} & 0 \le x \le 2 \\ 0 & \text{otherwise.} \end{cases}$$

or

$$f_X(x) = \begin{cases} \min(1,x) - \max(0,x-1) & 0 \le x \le 2 \\ 0 & \text{otherwise.} \end{cases}$$

Convolution of two independent uniform (0,1) random variables

$f_X(x)$



2. Exponential (with parameter *μ*)

Density function:

$$f(x) = \begin{cases} \mu e^{-\mu x} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Cumulative Distribution Function (CDF):

$$F(x) = \begin{cases} \int_0^x \mu e^{-\mu s} ds = 1 - e^{-\mu x} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Complementary Distribution Function:

$$\bar{F}(x) = 1 - F(x) = \begin{cases} e^{-\mu x} & \text{if } x \geq 0 \\ 1 & \text{otherwise.} \end{cases}$$

Example

Show how to apply the Inverse transform sampling to generate exponential deviates.

Guide

Due to symmetry, we can use the complementary distribution function instead of the cumulative distribution function. Set $U(0,1) = \bar{F}(x*) = e^{-\lambda x^*}$, and obtain

$$\ln U(0,1) = -\lambda x^*$$

or

$$x^* = -\frac{\ln U(0,1)}{\lambda}.$$

Memorylessness

A continuous random variable is called memoryless if for any $t \geq 0$ and $s \geq 0$, $P(X > s + t \mid X > t) = P(X > s)$. The following proves that the exponential random variable is memoryless.

$$\begin{aligned} P(X > s + t \mid X > t) &= \frac{P(X > s + t \cap X > t)}{P(X > t)} \\ &= \frac{P(X > s + t)}{P(X > t)} \\ &= \frac{e^{-\mu(s+t)}}{e^{-\mu t}} \\ &= e^{-\mu s} = P(X > s). \end{aligned}$$

The exponential random variable is the ONLY memoryless continuous random variable. (Recall that the Geometric is the ONLY memoryless discrete random variable.)

The minimum of two independent expenential random variables

Let X_1 and X_2 be independent and exponentially distributed random variables with parameters λ_1 and λ_2. Let $X = \min[X_1, X_2]$. Then

$$\begin{aligned} P(X > t) &= P(\min[X_1, X_2] > t) \\ &= P(X_1 > t, X_2 > t) \\ &= e^{-\lambda_1 t} e^{-\lambda_2 t} \\ &= e^{-(\lambda_1 + \lambda_2)t}. \end{aligned}$$

Thus, the distribution of X is exponential with parameter $\lambda_1 + \lambda_2$.

Competition between two independent expenential random variables

Let X_1 and X_2 be independent and exponentially distributed random variables with parameters λ_1 and λ_2. What is the probability of $X_1 < X_2$?

By the continuous version of the law of total probability,

$$\begin{aligned} P(X_1 < X_2) &= \int_0^\infty (1 - e^{-\lambda_1 t})\lambda_2 e^{-\lambda_2 t} dt \\ &= \frac{\lambda_1}{\lambda_1 + \lambda_2}. \end{aligned}$$

To understand the latter, note that X_2 can obtain many values: $t_1, t_2, t_3, \ldots$, infinitely many values ...

All these values, that X_2 may take, lead to the events $X_2 = t_1, X_2 = t_2, X_2 = t_3, \ldots$ that are mutually exclusive and exhaustive.

Then, using the continuous version of the Law of Total Probability, namely, integration of the product $P(X_1 < t)$ times the density of X_2 at t, will give us the probability of $X_1 < X_2$.
By integrating over all t we "add up" the probabilities of infinitely many mutually exclusive and exhaustive events that make up the event $X_1 < X_2$. And this is exactly what the Law of Total Probability does!

Relationship between the memoryless random variables: Exponential and Geometric

Let X_{exp} and X_{geo} be exponential and geometric random variables with parameters λ and p, respectively. Let δ be an "interval" size used to discretize the continuous values that X_{exp} takes, and we are interested to find δ such that

$$F_{X_{exp}}(n\delta) = F_{X_{geo}}(n), \quad n = 1, 2, 3, \ldots.$$

To this end we consider the complementary distributions and aim to find δ that satisfies $P(X_{exp} > n\delta) = P(X_{geo} > n)$, or $e^{-\lambda n\delta} = (1-p)^n$, or $e^{-\lambda\delta} = 1 - p$, or $p = 1 - e^{-\lambda\delta}$.

We can observe that as the interval size δ approaches zero the probability of success p also approaches zero, and under these conditions the two distributions approach each other.

3. Hyper-Exponential

Let X_i for $i = 1, 2, 3, \ \ldots, \ k$ be k independent exponential random variables with parameters λ_i, $i = 1, 2, 3, \ \ldots, \ k$, respectively. Let p_i for $i = 1, 2, 3, \ \ldots, \ k$ be k nonnegative real numbers such that $\sum_{i=1}^{k} p_i = 1$. A random variable X that is equal to X_i with probability p_i is called Hyper-exponential. By the Law of total probability, its density is

$$f_X(x) = \sum_{i=1}^{k} p_i f_{X_i}(x).$$

4. Erlang (with parameters k and λ)

A random variable X has Erlang distribution with parameters λ (positive real) and k (positive integer) if its density is given by

$$f_X(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!}. \tag{1}$$

Let X_i, $i = 1, 2, \ldots, k$, be k independent exponentially distributed random variables each with parameter λ, prove by induction that the random variable X defined by the sum $X = \sum_{i=1}^{k} X_i$ has Erlang distribution with parameters k and λ.
Complete the other homework problems in the textbook on Erlang random variable.

5. Hypo-Exponential

Let X_i, $i = 1,\ 2,\ \ldots,\ k$ be k independent exponentially distributed random variables each with parameters λ_i, respectively.

The random variable X defined by the sum $X = \sum_{i=1}^{k} X_i$ is called hypo-exponential. In other words, the density of X is a convolution of the k densities $\lambda_i e^{-\lambda_i x}$, $i = 1,\ 2,\ \ldots,\ k$.

The Erlang distribution is a special case of hypo-exponential when all the k random variables are identically distributed.

6. Gaussian (with parameters m and σ^2)

The Gaussian random variable X with parameters m and σ^2 has the following density.

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-m)^2/2\sigma^2} \qquad -\infty < x < \infty.$$

This density is symmetric and bell shaped.

Very widely used due to the **The central limit theorem** which says that the sum of a large number of independent random variables (not necessarily of the same distribution, but each has a finite variance) has Gaussian (normal) distribution.

7. Pareto (with parameters γ and δ)

It is useful in modelling lengths of data bursts in data and multimedia networks.

Its complementary distribution function is defined by

$$P\left(X > x\right) = \begin{cases} \left(\frac{x}{\delta}\right)^{-\gamma}, & x \geq \delta \\ 1, & \text{otherwise.} \end{cases}$$

Here $\delta > 0$ is the scale parameter representing a minimum value for the random variable, and $\gamma > 0$ is the shape parameter of the Pareto distribution.

Mean

The **mean**, or the **expectation**, of a discrete random variable is defined by

$$E[X] = \sum_{\{n: P_X(n) > 0\}} n P_X(n).$$

This is related to the term **average** discussed before in the context of limiting relative frequency.

Equivalently, the mean of a continuous random variable is defined by

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx.$$

Example

Consider the following probability function:

$$P\left(X = x\right) = \begin{cases} 0.2 & x = 1 \\ 0.3 & x = 2 \\ 0.3 & x = 3 \\ 0.2 & x = 4 \end{cases}$$

Find the mean of X.

Solution:

$$E[X] = 1 \times 0.2 + 2 \times 0.3 + 3 \times 0.3 + 4 \times 0.2 = 2.5$$

As mentioned before, a function of a random variable is also a random variable.

Mean of a function of discrete random variables:

$$E[g(X)] = \sum_{\{k:P_X(k)>0\}} g(k)P_X(k)$$

Mean of a function of continuous random variables:

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f_X(x)dx$$

If a and b are constants then for a random variable X (either discrete or continuous) we have:

$$E[aX] = aE[X],$$

$$E[X - b] = E[X] - b,$$

and

$$E[aX - b] = aE[X] - b.$$

For random variables $X_1, X_2, \ldots X_n$ (not necessarily independent) we have:

$$E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i].$$

The n**th moment** of the random variable X is defined by $E[X^n]$. Substituting $g(X) = X^n$ in the definitions of $E[g(X)]$, the nth moment of X is given by:

$$E[X^n] = \sum_{\{k:P_X(k)>0\}} k^n P_X(k)$$

for a discrete random variable and

$$E[X^n] = \int_{-\infty}^{\infty} x^n f_X(x)dx$$

for a continuous random variable.

The n**th central moment** of random variable X is defined by $E[(X-E[X])^n]$. Substituting $g(X)=(X-E[X])^n$ in the definitions of $E[g(X)]$, the nth central moment of X is given by:

$$E[(X-E[X])^n] = \sum_{\{k:P(k)>0\}} (k-E[X])^n P_X(k)$$

for a discrete random variable and

$$E[(X-E[X])^n] = \int_{-\infty}^{\infty} (x-E[X])^n f_X(x)dx$$

for a continuous random variable.

The mean is the first moment.

The variance is the second central moment defined by:

$$Var[X] = E[(X - E[X])^2].$$

The variance of a discrete random variable X is

$$Var[X] = \sum_{\{k: P(k) > 0\}} (k - E[X])^2 P_X(k)$$

The variance of a continuous random variable X is

$$Var[X] = \int_{-\infty}^{\infty} (x - E[X])^2 f_X(x) dx.$$

Another equation for the variance can be obtained by

$$Var[X] = E[(X - E[X])^2] = E[X^2 - 2XE[X] + (E[X])^2] = E[X^2] - (E[X])^2.$$

Example

Consider again the following probability function:

$$P(X = x) = \begin{cases} 0.2 & x = 1 \\ 0.3 & x = 2 \\ 0.3 & x = 3 \\ 0.2 & x = 4 \end{cases}$$

Find the variance of X.

Solution:

We already know that $E[X] = 2.5$,
so $Var[X] = (1 - 2.5)^2 \times 0.2 + (2 - 2.5)^2 \times 0.3 +$
$(3 - 2.5)^2 \times 0.3 + (4 - 2.5)^2 \times 0.2 = 1.05$

An Alternative Solution:

Now we solve the same problem using the formula $Var[X] = E[X^2] - (E[X])^2$

We already know that
$(E[X])^2 = (2.5)^2 = 6.25$,
and
$E[X^2] = 1^2 \times 0.2 + 2^2 \times 0.3 + 3^2 \times 0.3 + 4^2 \times 0.2 = 7.3$.
Then,
$Var[X] = E[X^2] - (E[X]^2) = 7.3 - 6.25 = 1.05$.

If a is a constant then for a random variable X (either discrete or continuous) we have:

$$Var[aX] = a^2 Var[X],$$

If the random variables X_1, X_2, X_3, $\ldots$, X_n are **independent**, then

$$Var\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} Var[X_i].$$

The **standard deviation** of r.v. X is:

$$\sigma_X = \sqrt{Var[X]}.$$

When X is obvious, we use σ for the standard deviation. Hence the variance is often denoted by σ^2.

Covariance and Correlation

The **covariance** of two random variables X_1 and X_2, denoted by $cov(X_1, X_2)$, is defined by

$$cov(X_1, X_2) = E[(X_1 - E[X_1])(X_2 - E[X_2])].$$

The larger the dependency, the larger the covariance.

The variance of the sum of two random variables X_1 and X_2 is given by

$$Var[X_1 + X_2] = Var[X_1] + Var[X_2] + 2cov(X_1, X_2).$$

If X_1 and X_2 are independent, then $cov(X_1, X_2) = 0$, so

$$Var[X_1 + X_2] = Var[X_1] + Var[X_2].$$

Problem

Prove that $cov(X_1, X_2) = 0$ does not necessarily imply that X_1 and X_2 are independent.

Problem

Prove that $cov(X_1, X_2) = 0$ does not necessarily imply that X_1 and X_2 are independent.

Guide

The proof is by a counter example.

Consider two random variables X and Y and assume that both have Bernoulli distribution with parameter p.

Consider random variable X_1 defined by $X_1 = X + Y$ and another random variable X_2 defined by $X_2 = X - Y$. Show that $cov(X_1, X_2) = 0$ and that X_1 and X_2 are not independent.

The covariance can take any value between $-\infty$ and $+\infty$, and in some cases, it is convenient to have a normalized dependence measure - a measure that takes values between -1 and 1. Such measure is the **correlation**. Notice that the covariance is bounded by

$$cov(X_1, X_2) \leq \sqrt{Var[X_1]Var[X_2]},$$

the **correlation** of two random variables X and Y denoted by $corr(X,Y)$ is defined by

$$corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y},$$

assuming $Var[X] \neq 0$ and $Var[Y] \neq 0$.

Homework

Consider an experiment of tossing a die with 6 sides. Assume that the die is fair, i.e., each side has the same probability (1/6) to occur. Consider a random variable X that takes the value i if the outcome of the toss is i, for $i = 1, 2, 3, \cdots, 6$.

1. Find $E[X]$, $Var[X]$ and σ_X.
2. Plot the probability function, cumulative distribution function and the complementary distribution function of X.

Some answers

$E[X] = 3.5$; $E[X^2] = 15.16666667$; $Var[X] = 2.916666667$; $\sigma_X = 1.707825128$.

Homework

Consider an exponential random variable with parameter λ. Derive its mean and Variance.

Guide

Find the mean by

$$E[X] = \int_0^\infty x\lambda e^{-\lambda x} dx.$$

Use integration by parts to show that:

$$E[X] = -xe^{-\lambda x}\Big]_0^\infty + \int_0^\infty e^{-\lambda x} dx = \frac{1}{\lambda}.$$

Guide (continued)

Then use integration by parts to derive the second moment. Understand and verify the following derivations:

$$\begin{aligned} E[X^2] &= \int_0^\infty x^2 \lambda e^{-\lambda x} dx \\ &= -x^2 e^{-\lambda x}\Big]_0^\infty + 2\int_0^\infty x e^{-\lambda x} dx \\ &= \left(-x^2 e^{-\lambda x} - \frac{2}{\lambda} x e^{-\lambda x} - \frac{2}{\lambda^2} e^{-\lambda x}\right)\Bigg]_0^\infty \\ &= \frac{2}{\lambda^2}. \end{aligned}$$

$$Var[X] = E[X^2] - (E[X])^2 = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}$$

Sample Mean and Sample Variance

Consider a sample of n realizations of a random variable X, denoted $X(1), X(2), \ \ldots, X(n)$

The **Sample Mean**

$$S_m = \frac{\sum_{i=1}^{n} X(i)}{n}$$

is an estimator for the mean of X.

The **Sample Variance**

$$S_v = \frac{\sum_{i=1}^{n}[X(i) - S_m]^2}{n-1} \qquad n > 1$$

is an estimator for the variance of X.

The sample standard deviation is $\sqrt{S_v}$.

Homework

Consider the following data for students heights (in cm) taken from a sample of 10 students: 172, 178, 162, 167, 168, 175, 182, 161, 171, 170.

Compute the sample mean, the sample variance and the sample standard deviation.

Answers

Sample Mean = 170.6
Sample Variance = 43.6
Sample Standard Deviation = 6.603029608

Homework

Generate 10 deviates from an exponential distribution of a given mean and compute the Sample Mean and Sample Variance. Compare them with the real mean and variance. Then increase the sample to 100, 1000, . . . , 1,000,000. Observe the difference between the real mean and variance and the sample mean and variance. Repeat the experiment for a Pareto deviates of the same mean. Discuss differences.

Use your generated data of random deviates to compare between the exponential and Pareto densities and the equivalent histograms. Use small enough ranges to achieve good fit.

Conditional Expectation (or Mean)

$E[X \mid Y]$ denotes the conditional expectation of random variable X given the event $\{Y = y\}$ for each relevant value of y.

The conditional expectation of two discrete random variables is defined by

$$E[X \mid Y = j] = \sum_i iP(X = i \mid Y = j).$$

If X and Y are continuous, their conditional expectation is defined as

$$E[X \mid Y = y] = \int_{x=-\infty}^{\infty} x f_{X|Y}(x \mid y) dx.$$

$E[X \mid Y]$ is itself a random variable which is a function of the random variable Y. Therefore, for **discrete** random variables:

$$\begin{aligned} E_Y[E[X \mid Y]] &= \sum_j E[X \mid Y = j]P(Y = j) \\ &= \sum_j \sum_i iP(X = i \mid Y = j)P(Y = j) \\ &= \sum_i i \sum_j P(X = i \mid Y = j)P(Y = j) \\ &= \sum_i iP(X = i) = E[X]. \end{aligned}$$

Thus,

$$E[X] = E_Y[E[X \mid Y]].$$

For **continuous** random variables:

$$\begin{aligned}
E_Y[E[X \mid Y]] &= \int_{y=-\infty}^{\infty} E[X \mid Y = y] f_Y(y) dy \\
&= \int_{y=-\infty}^{\infty} \int_{x=-\infty}^{\infty} x f_{X|Y}(x \mid y) dx f_Y(y) dy \\
&= \int_{x=-\infty}^{\infty} x \int_{y=-\infty}^{\infty} f_{X|Y}(x \mid y) f_Y(y) dy dx \\
&= \int_{x=-\infty}^{\infty} x f_X(x) dx = E[X].
\end{aligned}$$

Thus,

$$E[X] = E_Y[E[X \mid Y]].$$

Homework: Show that the latter holds also for the case where X is discrete and Y is continuous and vice versa.

Question

Let X be a geometric random variable. Use the concept of conditional expectation to derive $E[X]$.

Guide

Condition on the result of the first Bernoulli trial and obtain $E[X] = p(1) + (1-p)(1+E[X])$ and then solve it to obtain $E[X] = 1/p$.

Note that $P(X = x \mid Y = y)$ is itself a random variable that is a function of the values y taken by random variable Y.

Therefore, by definition

$$E_Y[P(X = x \mid Y = y)] = \sum_y P(X = x \mid Y = y)P(Y = y)$$

which lead to another way to express the Law of Total Probability:

$$P_X(x) = E_Y[P(X = x \mid Y = y)].$$

Henceforth, we normally ommit the subsript X or Y for mean, variance and probability notations.

Define the **Conditional Variance** as

$$Var[X \mid Y] = E[(X - E[X \mid Y])^2 \mid Y].$$

This gives rise to the following useful formula for the variance of a random variable known as EVVE:

$$Var[X] = E[Var[X \mid Y]] + Var[E[X \mid Y]].$$

Homework: Try to prove it, and if you are not successful, see my book or other sources.

Example

The number of Internet flows that arrive at a router per second is ϕ which has mean ϕ_e and variance ϕ_v. The number of packets in each flow is ς which has mean ς_e and variance ς_v. Assume that ϕ and ς are independent. The total number of packets arriving at the router per second is W which has mean W_e and variance W_v. Assume $W = \varsigma\phi$. To meet certain quality of service requirements, it is required that the router has the capacity to serves the arriving packets at the rate of $s_r = W_e + 4\sqrt{W_v}$ per second. Find s_r.

Solution

To compute s_r one needs to have the values of W_e and W_v. Because ϕ and ς are independent $E[W|\phi] = \phi\varsigma_e$ and therefore

$$W_e = E[W] = E[E[W|\phi]] = E[\phi]E[\varsigma] = \phi_e\varsigma_e.$$

By EVVE,

$$Var[W] = E[Var[W|\phi]]+Var[E[W|\phi]] = \varsigma_v E[\phi^2]+(\varsigma_e)^2 Var[\phi].$$

Therefore

$$W_v = \phi_v\varsigma_v + \varsigma_v\phi_e^2 + \phi_v\varsigma_e^2.$$

The following table provides the mean and the variance of some of the above-mentioned random variables.

r.v.	parameters	mean	variance
Bernoulli	$0 \leq p \leq 1$	p	$p(1-p)$
geometric	$0 \leq p \leq 1$	$1/p$	$(1-p)/p^2$
binomial	n and $0 \leq p \leq 1$	np	$np(1-p)$
Poisson	$\lambda > 0$	λ	λ
discrete uniform	a and b	$(a+b)/2$	$[(b-a+1)^2-1]/12$
uniform	a and b	$(a+b)/2$	$(b-a)^2/12$
exponential	$\mu > 0$	$1/\mu$	$1/\mu^2$
Gaussian	m and σ	m	σ^2
Pareto	$\delta > 0$ and $1 < \gamma \leq 2$	$\delta\gamma/(\gamma-1)$	∞

The Central Limit Theorem

Let $X_1, X_2, X_3, \ \ldots, \ X_k$ be k independent and identically distributed (IID) random variables with common mean λ and variance σ^2. Define random variable Y_k as

$$Y_k = \frac{X_1 + X_2 + X_3 + \ \ \ldots \ \ + X_k - k\lambda}{\sigma\sqrt{k}}.$$

Then,

$$\lim_{k\to\infty} P(Y_k \leq y) = \Phi(y)$$

where $\Phi(\cdot)$ is the distribution function of a standard Gaussian random variable given by

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-t^2/2} dt.$$

The central limit theorem is considered the most important result in probability.

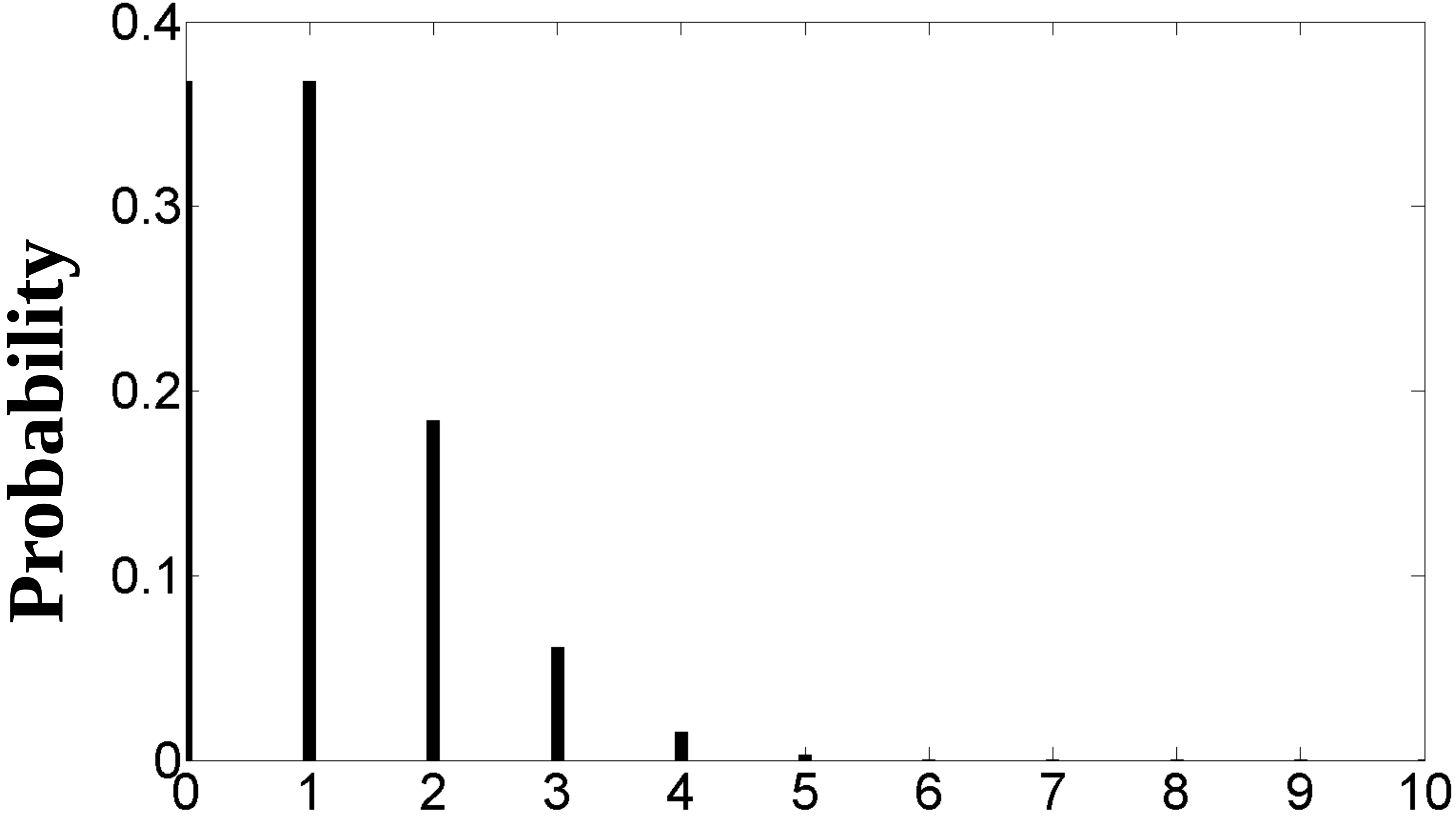
It implies that the sum of k IID random variable with common mean λ and variance σ^2 is approximately Guassian with mean $k\lambda$ and variance $k\sigma^2$ *regardless* of the distribution of these variables.

Furthermore, under certain conditions, the central limit theorem also applies in the case of sequences that are not identically distributed.

Homework:

Observe the following behavior of the Poisson probability function and provide explanation.

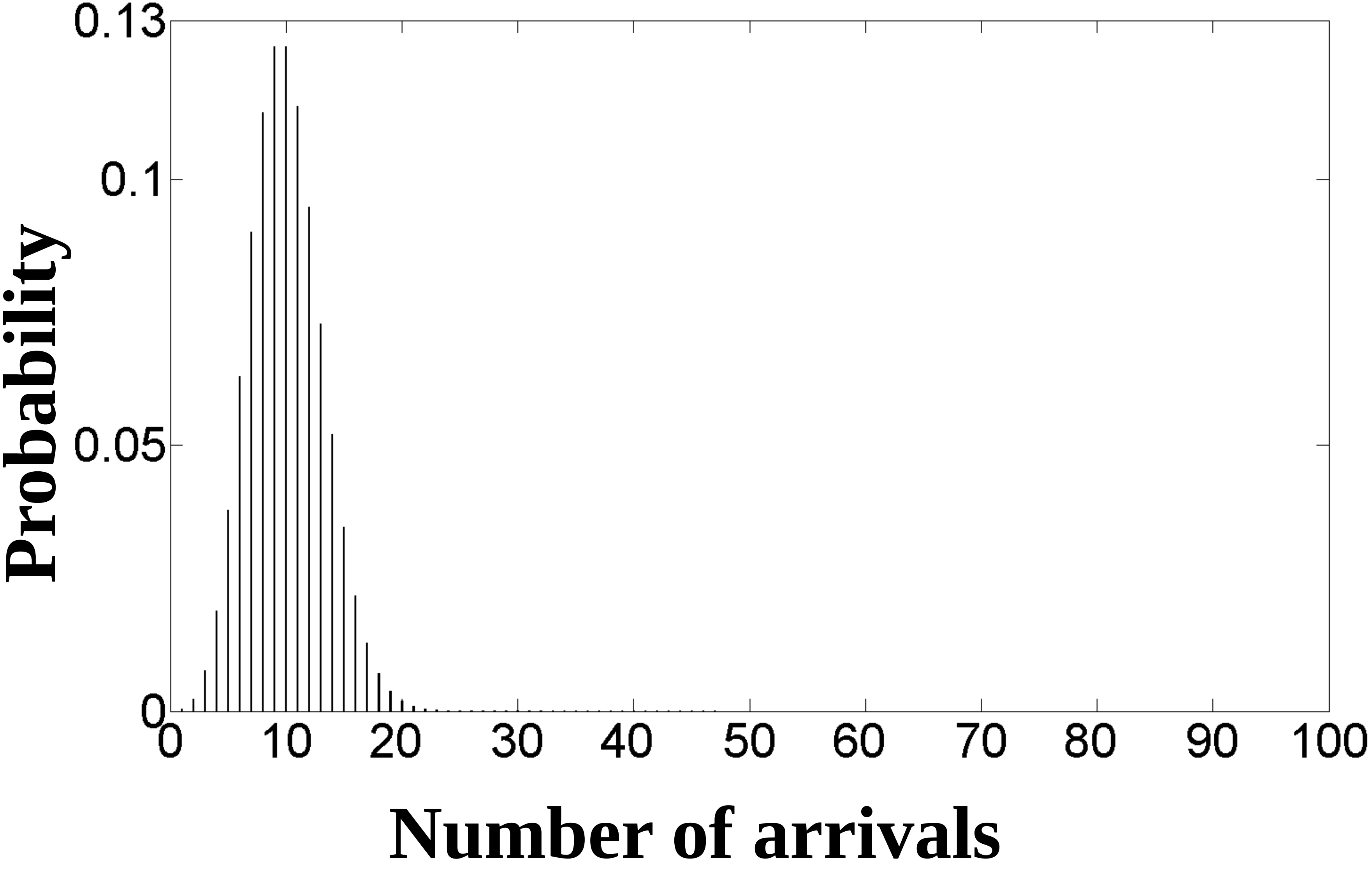
Poisson Probability function with λ= 1



Number of arrivals

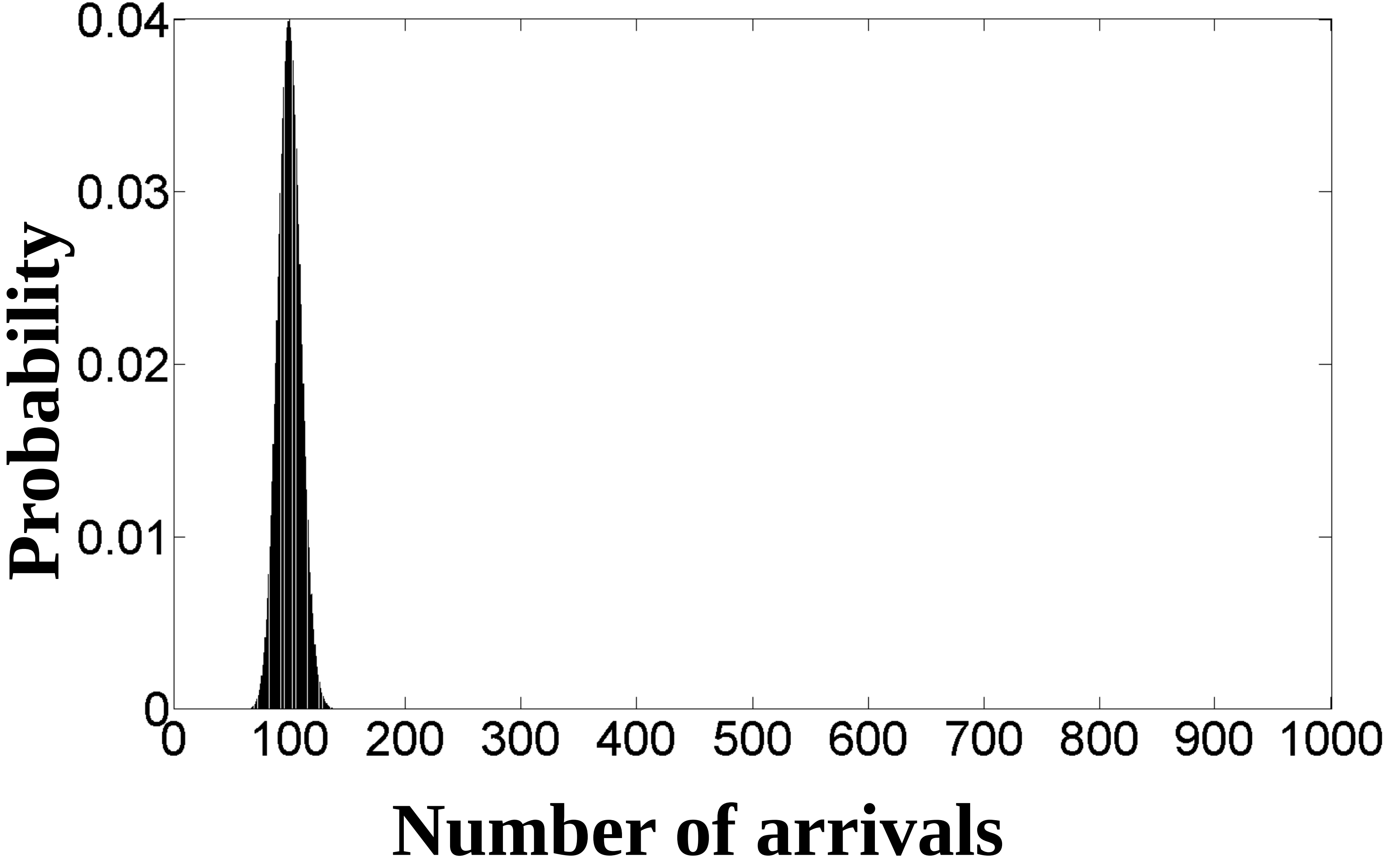
Credit: Li Fan

Poisson Probability function with λ= 10



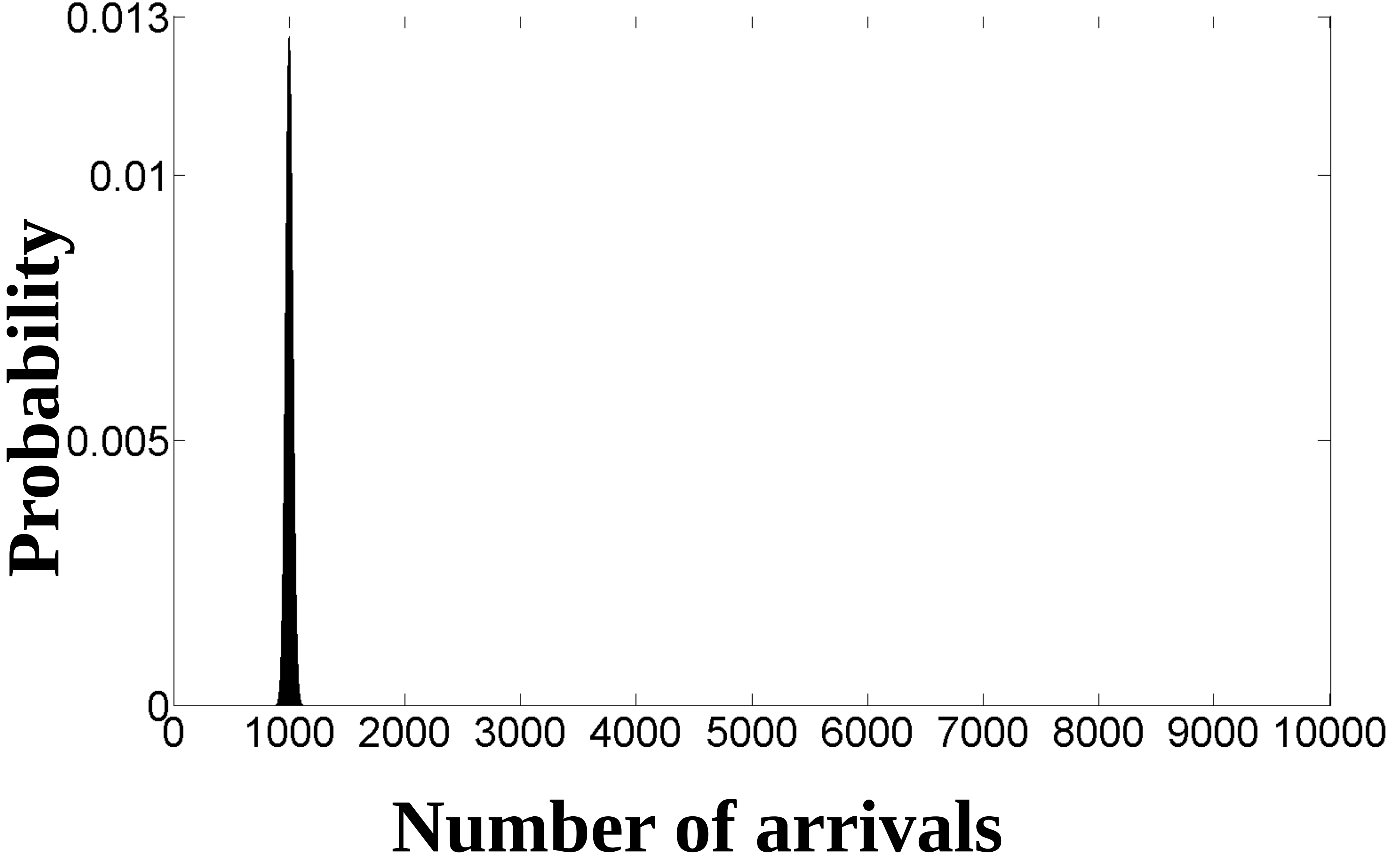
Credit: Li Fan

Poisson Probability function with λ= 100



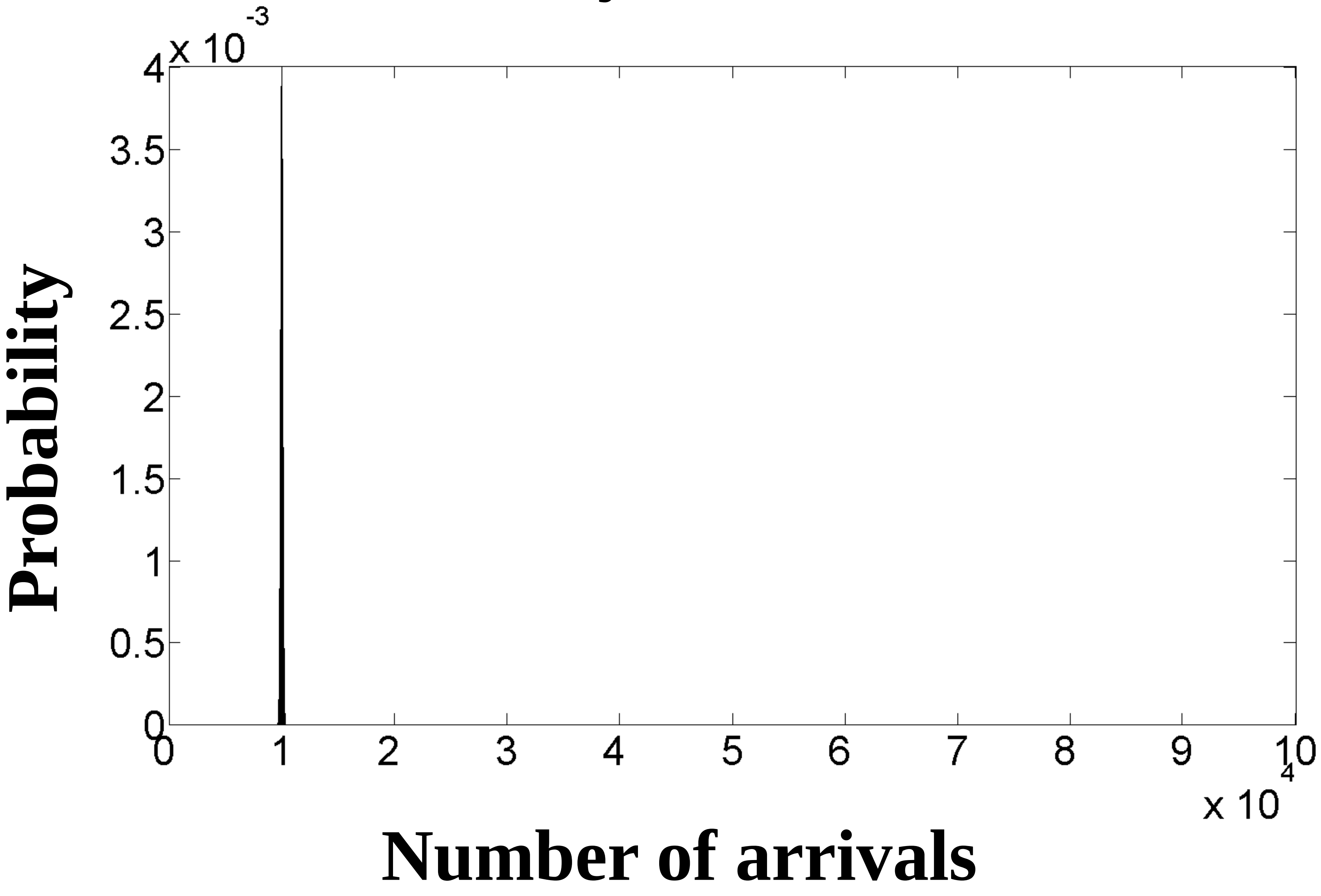
Credit: Li Fan

Poisson Probability function with λ= 1000



Credit: Li Fan

Poisson Probability function with λ= 10000



Credit: Li Fan

Link Dimensioning

We will consider several scenarios of sources (individuals or families) sharing a communication link. Each of the sources has certain requirements for capacity and the common link must be dimensioned in such a way that minimizes the cost for the telecommunications provider, but still meets the individual QoS requirements. The link dimensioning procedures that we consider apply to user requirements for capacity either upload or download.

Case 1: Homogeneous Individual Sources

Consider N independent sources (end-terminals), sharing a transmission link of capacity C [Mb/s]. Any of the sources transmits data in accordance with an on-off process. That is, a source alternates between two states: 1) the on state during which the source transmits at a rate R [Mb/s], and 2) the off state during which the source is idle. Assume that the proportion of time the source is in the on-state is p, so it is in the off-state $1 - p$ of the time. The question is how much capacity should the link have so it can serve all N sources such that the probability that the demand exceeds the total link capacity is no higher than α.

Without loss of generality, let us normalize the traffic generated by a source during on period by setting $R = 1$.

The demand generated by a single source is Bernoulli distributed with parameter p, so the demand generated by all N sources has Binomial distribution with parameters p and N.

Accordingly, finding the desired capacity is reduced to finding the smallest C such that

$$\sum_{i=C+1}^{N} \binom{N}{i} p^i (1-p)^{N-i} \leq \alpha.$$

If N is large we can use the central limit theorem and approximate the Binomial distribution by a Gaussian distribution.

Accordingly, the demand can be approximated by a Gaussian random variable with mean Np and variance $Np(1-p)$ and finding C_G such that the probability of our Gaussian random variable to exceed C_G is α.

It is well known that Gaussian random variables obey the so-called 68-95-99.7% Rule which means that the following apply to a random variable X with mean m and standard deviation σ.

$$\begin{aligned} P(m-\sigma \leq X \leq m+\sigma) &= 0.68 \\ P(m-2\sigma \leq X \leq m+2\sigma) &= 0.95 \\ P(m-3\sigma \leq X \leq m+3\sigma) &= 0.997. \end{aligned}$$

Therefore, if $\alpha = 0.0015$ then C_G should be three standard deviations above the mean, namely,

$$C_G = Np + 3\sqrt{Np(1-p)}.$$

Recall that for our original problem, before we introduced the Gaussian approximation, $C = N$ guarantees that there is sufficient capacity to serve all arriving traffic without losses. Therefore, we set our dimensioning rule for the optimal C value as follows:

$$C_{opt} = \min\left[N, Np + 3\sqrt{Np(1-p)}\right].$$

Case 2: Non-homogeneous Individual Sources

Now we generalize the above scenario to the case where the traffic and the peak rates of different sources can be different.

In this case where the sources are non-homogeneous, we must invoke a generalization of the central limit theorem that allows for non IID random variables (i.e., the so-called "Lyapunov's central limit theorem").

Consider N sources where the ith source transmits at rate R_i with probability p_i, and at rate 0 with probability $1 - p_i$.

Let $R_X(i)$ be a random variable representing the rate transmitted by source i. We obtain:

$$E[R_X(i)] = p_i R_i.$$

and

$$Var[R_X(i)] = R_i^2 p_i - (R_i p_i)^2 = R_i^2 p_i (1 - p_i).$$

The latter is consistent with the fact that $R_X(i)$ is equal to R_i times a Bernoulli random variable.

We now assume that the random variable

$$\Sigma_R = \sum_{i=1}^{N} R_X(i)$$

has a Gaussian distribution with mean

$$E[\Sigma_R] = \sum_{i=1}^{N} E[R_X(i)] = \sum_{i=1}^{N} p_i R_i$$

and variance

$$Var[\Sigma_R] = \sum_{i=1}^{N} Var[R_X(i)] = \sum_{i=1}^{N} R_i^2 p_i (1 - p_i).$$

Notice that the allocated capacity should not be more than the total sum of the peak rates of the individual sources. Therefore, in this more general case, for the QoS requirement $\alpha = 0.0015$, our optimal C value is set to:

$$C_{opt} = \min\left[\sum_{i=1}^{N} R_i, E[\Sigma_R] + 3\sqrt{Var[\Sigma_R]}\right].$$

For lower α value, mean + 4 or even 5 standard deviations may be required

Homework

There are 20 sources each transmits at a peak-rate of 10 Mb/s with probability 0.1 and is idle with probability 0.9, and there are other 80 sources each transmits at a peak-rate of 1 Mb/s with probability 0.05 and is idle with probability 0.95.

A service provider aims to allocate the minimal capacity C_{opt} such that no more than 0.0015 of the time, the demand of all these 100 sources exceeds the available capacity. Find an appropriate C_{opt}.

Answer: $C_{opt} = 64.67186$ Mb/s.

Notice the difference in contributions to the total variance of sources from the first group versus such contributions of sources from the second group.

Consider a range of examples where the variance is the dominant part of C_{opt} versus examples where the variance is not the dominant part of C_{opt}.

Chapter 2: Stochastic Processes

© Moshe Zukerman

February 12, 2023

Preliminaries

A stochastic process is a collections of random variables.
Stochastic processes are used to model real-life processes (natural and artificial) to understand their properties and their real-world effects.

The research in the field of stochastic processes has three facets:

Theory: mathematical explorations of stochastic processes models that aim to better understand their properties.

Measurements: taken on the real process in order to identify its statistical characteristics.

Modelling: fitting the measured statistical characteristics of the real process with those of a model and development of new models of stochastic processes that well match the real process.

Preliminaries (cont'd)

We will provide background on basic theoretical aspects of stochastic processes which form a basis for queueing theory and teletraffic models discussed later.

Throughout the course, we will use **mathematical/rigorous language** as well as **engineering/somewhat intuitive** language.

For a given **index set** T, a **stochastic process** $\{X_t,\ t \in T\}$ is an indexed collection of random variables.

They may or may not be identically distributed.

In many applications the index t is used to model time.

Accordingly, the random variable X_t for a given t can represent, for example, the number of telephone calls that have arrived at an exchange by time t.

Preliminaries (cont'd)

If the index set T is countable, the stochastic process is called a **discrete-time process**, or a **time series**.

Otherwise, the stochastic process is called a **continuous-time process**.

Considering our previous example, where the number of phone calls arriving at an exchange by time t is modelled as a continuous-time process $\{X_t, \; t \in T\}$, we can alternatively, use a discrete-time process to model, essentially, the same thing.

This can be done by defining the discrete-time process $\{X_n, \; n = 1, \; 2, \; 3, \; \ldots \; \}$, where X_n is a random variable representing, for example, the number of calls arriving within the nth minute.

Introductory Comments and Preliminaries (cont'd)

A stochastic process $\{X_t,\ t \in T\}$ is called **discrete space** stochastic process if the random variables X_t are discrete, and it is called **continuous space** stochastic process if it is continuous. We therefore have four types of stochastic processes:

1. Discrete Time Discrete Space
2. Discrete Time Continuous Space
3. Continuous Time Discrete Space
4. Continuous Time Continuous Space.

Strict Stationarity

A discrete-time stochastic process $\{X_n,\ n = 1,\ 2,\ 3,\ \ldots\ \}$ is **strictly stationary** if for any subset of $\{X_n\}$, say,
$\{X_{n(1)},\ X_{n(2)},\ X_{n(3)},\ \ldots,\ X_{n(k)}\}$,
for any integer m, the joint probability function
$P(X_{n(1)},\ X_{n(2)},\ X_{n(3)},\ \ldots,\ X_{n(k)})$,
is equal to the joint probability function
$P(X_{n(1)+m},\ X_{n(2)+m},\ X_{n(3)+m},\ \ldots,\ X_{n(k)+m})$.

In other words,
$P(X_{n(1)+m},\ X_{n(2)+m},\ X_{n(3)+m},\ \ldots,\ X_{n(k)+m})$
is independent of m.

In this case, the probability structure of the process does not change with time.

Strict Stationarity (cont'd)

An equivalent definition for strict stationarity is applied also for a continuous-time process.

Notice that for the process to be strictly stationary, the value of k is unlimited as the joint probability should be independent of m for any subset of $\{X_n,\ n = 1,\ 2,\ 3,\ \ldots\}$.

If k is limited to some value k^*, we say that the process is **stationary of order** k^*.

Strict Stationarity (cont'd)

An equivalent definition applies to a continuous-time stochastic process.

A continuous-time stochastic process X_t is said to be strictly stationary if its statistical properties do not change with a shift of the origin.

In other words, the process X_t statistically the same as the process X_{t-d} for any value of d.

Gaussian Process

An important stochastic process is the **Gaussian Process** defined as a process that has the property that the joint probability function (density) associated with any set of times is multivariate Gaussian.

For simplicity we only consider single dimensional Gaussian distribution and processes, but the definition of a Gaussian process applies to the more general case of multivariate Gaussian.

The importance of the Gaussian process lies in its property to be an accurate model for superposition of many independent processes.

Gaussian Process (cont'd)

This makes the Gaussian process a useful model for heavily multiplexed traffic which arrive at switches or routers deep in a major telecommunications network.

Fortunately, the Gaussian process is not only useful, but it is also relatively simple and amenable to analysis (but not as simple as the Poisson process that we will learn later).

Notice that for a Gaussian distribution, all the joint moments of the Gaussian random variables are fully determined by the joint first and second order moments of the variables.

Therefore, if the first and second order moments do not change with time, the Gaussian random variables themselves are stationary.

Weak Stationarity

This implies that for a Gaussian process, stationarity of order two (also called **weak stationarity**) implies strict stationarity.

For a time series $\{X_n, \ n = 1, \ 2, \ 3, \ \dots \}$, weak stationarity implies that, for all n, $E[X_n]$ is constant, denoted $E[X]$, independent of n. Namely, for all n,

$$E[X] = E[X_n].$$

Weak stationarity (because it is stationarity of order two) also implies that the covariance between X_n and X_{n+k}, for any k, is independent of n, and is only a function of k, denoted $U(k)$. Namely, for all n,

$$U(k) = Cov[X_n, X_{n+k}].$$

Autocovariance Function

Notice that, the case of $k = 0$, namely,

$$U(0) = Cov[X_n, X_n] = Var[X_n]$$

implies that the variance of X_n is also independent of n.

Also for all integer k,

$$U(-k) = U(k)$$

because $Cov[X_n, X_{n+k}] = Cov[X_{n+k}, X_n] = Cov[X_n, X_{n-k}]$.

The function $U(k)$, $k = 0, 1, 2, \ \ldots$, is called the **autocovariance function**.

The value of the autocovariance function at k, $U(k)$, is also called **the autocovariance of lag** k.

Autocorrelation Function

The **autocorrelation function** at lag k, denoted $C(k)$, is the normalized version of the autocovariance function, and since by weak stationarity, for all i and j, $Var[X_j] = Var[X_i]$, it is given by:

$$C(k) = \frac{U(k)}{Var[X_n]}.$$

Autocorrelation function (as well as correlation) is bounded between -1 and 1. In some areas of EE other definitions of autocorrelation functions (as well as of correlation) exist.

High values of autocorrelation function (close to 1) indicate strong dependencies. Equivalently, low (positive) values of autocovariance function indicate weak dependencies.

High negative values of autocovariance function and high negative values of autocorrelation function (close to -1) indicate strong negative dependencies.

Ergodicity

A stochastic process is called **ergodic** if every realization contains sufficient information on the probabilistic structure of the process.

For example, let us consider a process which can be in either one of two realization: either $X_n = 1$ for all n, or $X_n = 0$ for all n.

Assume that each one of these two realizations occur with probability 0.5.

If we observe any one of these realizations, regardless of the duration of the observations, we shall never conclude that $E[A] = 0.5$. We shall only have the estimations of either $E[A] = 0$ or $E[A] = 1$, depending on which realization we happen to observe. Such a process is not ergodic.

Statistics of a Stationary and Ergodic Process

Assuming $\{X_n,\ n = 1,\ 2,\ 3,\ \ldots\ \}$ is ergodic and stationary, and we observe m observations of this $\{X_n\}$ process, denoted by $\{\hat{A}_n,\ n = 1,\ 2,\ 3,\ \ldots,\ m\}$, then the mean of the process $E[A]$ can be estimated by

$$\hat{E}[A] = \frac{1}{m}\sum_{n=1}^{m}\hat{A}_n,$$

and the autocovariance function $U(k)$ of the process can be estimated by

$$\hat{U}(k) = \frac{1}{m-k}\sum_{n=k+1}^{m}(\hat{A}_{n-k} - E[A])(\hat{A}_n - E[A]).$$

Two Orderly and Memoryless Point Processes

We now consider a very special class of stochastic processes called **point** processes that also possess two properties: **orderliness** and **memorylessness**.

After providing, somewhat intuitive, definitions of these concepts, we will discuss two processes that belong to this special class: one is discrete-time - called the **Bernoulli process** and the other is continuous-time - called the **Poisson process**.

We consider here a physical interpretation, where a **point process** is a sequence of events which we call **arrivals** occurring at random in points of time t_i, $i = 1, 2, \ \ldots\ , t_{i+1} > t_i$, or $i = \ldots, \ -2, -1, 0, 1, 2, \ \ldots, t_{i+1} > t_i$.

The index set, namely, the time, can be continuous or discrete.

Two Orderly and Memoryless Point Processes (cont'd)

We call our events arrivals to relate is to the context of queueing theory, where a point process typically corresponds to points of arrivals, i.e., t_i is the time of the ith arrival that joints a queue.

A point process can be defined by its **counting process** $\{N(t),\ t \geq 0\}$, where $N(t)$ is the number of arrivals occurred within $[0, t)$.

A counting process $\{N(t)\}$ has the following properties:

1. $N(t) \geq 0$,
2. $N(t)$ is integer,
3. if $s > t$, then $N(s) \geq N(t)$ and $N(s) - N(t)$ is the number of occurrences within $(t, s]$.

Two Orderly and Memoryless Point Processes (cont'd)

Note that $N(t)$ is not an independent process because for example, if $t_2 > t_1$ then $N(t_2)$ is dependent on the number of arrivals in $[0, t_1)$, namely, $N(t_1)$.

Another way to define a point process is by the stochastic process of the inter-arrival times Δ_i where $\Delta_i = t_{i+1} - t_i$.

Orderliness for a point process means that the probability that two or more arrivals happen at once is negligible. Mathematically, for a continuous-time counting process to be **orderly**, it should satisfy:

$$\lim_{\Delta t \to 0} P(N(t + \Delta t) - N(t) > 1 \mid N(t + \Delta t) - N(t) \geq 1) = 0.$$

A stochastic process is **memoryless** if at any point in time, the future evolution of the process is statistically independent of its past.

Bernoulli Process

The Bernoulli process is a discrete-time stochastic process made up of a sequence of IID Bernoulli distributed random variables $\{X_i,\ i = 0, 1, 2, 3,\ \ldots\}$ where for all i, $P(X_i = 1) = p$ and $P(X_i = 0) = 1 - p$. In other words, we divide time into consecutive equal time slots.

Then, for each time-slot i, we conduct an independent Bernoulli experiment. If $X_i = 1$, we say that there was an **arrival** at time-slot i.

Otherwise, if $X_i = 0$, we say that there was no arrival at time-slot i.

The Bernoulli process is both orderly and memoryless. It is orderly because, by definition, no more than one arrival can occur at any time-slot as the Bernoulli random variable takes values of more than one with probability zero.

Bernoulli Process (cont'd)

It is also memoryless because the Bernoulli trials are independent, so at any discrete point in time n, the future evolution of the process is independent of its past.

The counting process for the Bernoulli process is another discrete-time stochastic process $\{N(n), n \geq 0\}$ which is a sequence of Binomial random variables $N(n)$ representing the total number of arrivals occurring within the first n time-slots.

Notice that since we start from slot 0, $N(n)$ does not include slot n in the counting.

That is, we have

$$P[N(n) = i] = \binom{n}{i} p^i (1-p)^{n-i} \qquad i = 0,\ 1,\ 2,\ \ldots,\ n.$$

Bernoulli Process (cont'd)

The inter-arrival times for Bernoulli process are memoryless, IID, and geometrically distributed, so we can drop the index i of Δ_i, designating the i inter-arrival time, and consider the probability function of the random variable Δ representing any inter-arrival time. Its probability function is given by

$$P(\Delta = i) = p(1-p)^{i-1} \qquad i = 1,\ 2,\ \ldots\ .$$

Another important question is what is the probability distribution of the time it takes until the ith arrival.

Bernoulli Process (cont'd)

This time is a sum of i inter-arrival times which is a sum of i geometric random variables which we already know has a Pascal distribution with parameters p and i, so we have

$$P[\text{the } i\text{th arrival occurs in time slot } n] =$$

$$= \binom{n-1}{i-1} p^i (1-p)^{n-i} \qquad i = i,\ i+1,\ i+2,\ \ldots\ .$$

Bernoulli Process (cont'd)

Notice that on-off sources could be modeled as Bernoulli processes where the on-periods are represented by consecutive successes of Bernoulli trials and the off-periods by failures.

In this case, for each on-off process, the length of the on- and the off-periods are both geometrically distributed.

Accordingly, the **superposition** of N Bernoulli processes with parameter p is another discrete-time stochastic process where the number of arrivals during the different slots are IID and binomial distributed with parameters N and p.

Homework: Prove the last statement.

Bernoulli Process (cont'd)

Another important concept is **merging** of processes which is different from superposition.

Unlike superposition in which we are interested in the total number of arrivals, in merging we are only interested to know if there was at least one arrival within a time-slot without any interest of how many arrivals there were in total.

This is of interest in sensor network application when we want to know if any of the sensors sounds an alarm and we are not interested in how many sensors sound an alarm.

Bernoulli Process (cont'd)

Consider an example of N sensors that are spread around a country to detect certain events. Assume that the ith sensor is active following a Bernoulli process with parameter p_i in any of the time slots. Assume independence between the sensors and between different time slots.

The probability that an alarm is sound in a given time-slot is the probability that at least one of the sensors is active which is given by

$$P_a = 1 - \prod_{i=1}^{N} (1 - p_i).$$

Now, considering the independence of the processes, we can realize that the alarms follow a Bernoulli process with parameter P_a.

This Bernoulli process with parameter P_a is the **merged** process.

Bernoulli Process (cont'd)

Another concept that applies to traffic in networks is **splitting**.

Consider a Bernoulli process with parameter p and then color each arrival, independently of all other arrivals, in red with probability q and in blue with probability $1 - q$.

Then, in each time-slot we have a red arrival with probability pq and a blue one with probability $p(1 - q)$.

Therefore, the red arrivals follow a Bernoulli process with parameter pq and the blue arrivals follow a Bernoulli process with parameter $p(1 - q)$.

Poisson process

The Poisson process is a continuous-time point process which is also memoryless and orderly.

It applies to many cases where a certain event occurs at different points in time.

Such occurrences of the events could be, for example, arrivals of phone call requests at a telephone exchange. We will use many terms to call such occurrences, including **“occurrences”**, **“events”**, **“arrivals”** **and** **“points”**.

A Poisson process can be described by its **counting process** $\{N(t),\ t \geq 0\}$ representing the total number of occurrences by time t.

Poisson process (cont'd)

A counting process $\{N(t)\}$ is defined as a Poisson process with rate $\lambda > 0$ if it satisfies the following three conditions.

1. $N(0) = 0$.
2. The number of occurrences in two non-overlapping intervals are independent. That is, for any $s > t > u > v > 0$, the random variable $N(s) - N(t)$, and the random variable $N(u) - N(v)$ are independent. This means that the Poisson process has what is called **independent increments**.
3. The number of occurrences in an interval of length t has a Poisson distribution with mean λt.

These three conditions will be henceforth called the **three Poisson process conditions**.

Poisson process (cont'd)

By definition, the Poisson process $N(t)$ has what is called **stationary increments**, that is, for any $t_2 > t_1$, the random variable $N(t_2) - N(t_1)$, and the random variable $N(t_2 + u) - N(t_1 + u)$ have the same distribution for any $u > 0$.

In both cases, the distribution is Poisson with parameter $\lambda(t_2 - t_1)$.

Intuitively, if we choose the time interval $\Delta = t_2 - t_1$ to be arbitrarily small (almost a "point" in time), then the probability of having an occurrence there is the same regardless of where the "point" is.

Loosely speaking, every point in time has the same chance of having an occurrence. Therefore, occurrences are equally likely to happen at all times. This property is also called **time-homogeneity**.

More Properties and Characteristic of the Poisson Process

- The Poisson process is a pure-chance point process on the real line.
- The points on the real line (that normally represents time) are events (e.g. call arrivals). Events are random and independent of each other.
- The parameter λ is the **rate** of the Poisson process.
- The time between consecutive events is exponentially distributed with parameter λ.
- The mean time between consecutive events is equal to $\frac{1}{\lambda}$.
- The number of events in an interval time T is Poisson distributed with parameter λT.
- The mean number of events in an interval time T is equal to λT.
- The variance of the number of events in an interval time T is also equal to λT.
- The number of events in disjoint intervals are independent random variables, so the number of future events is independent of the past.

Homework

Generate a Poisson process with rate $\lambda = 1$ for a period of time of length $T \geq 10,000$.

Pick a point in time from a uniform distribution within the interval [1,10000].

Record the length of the interval (between two consecutive Poisson occurrences) that includes the chosen point in time.

Repeat the experiment 1000 times.

Compute the average length of the intervals you recorded.

Explain your result.

Share your program, result and explanation on Group Learning.

Memorylessness Property of Poisson Process

By the memorylessness (memoryless) property of the exponential distribution, the time until the next occurrence is always exponentially distributed with the same parameter as that of the original distribution and therefore, at any point in time, not necessarily at points of occurrences, the future evolution of the Poisson process is independent of the past, and is always probabilistically the same.

The Poisson process is therefore memoryless.

Another explanation for the independence of the past can be explained by the Poisson process property of **independent increments**.

Superposition of Poisson Processes

Consider two Poisson processes: one with parameter λ_1 and the other with parameter λ_2.

The superposition of these two processes is a new point process that comprises all the points of both processes.

This superposition is a Poisson process with rate $\lambda = \lambda_1 + \lambda_2$.

Superposition of Poisson Processes (cont'd)

Notice that for any time interval T, the number of events in the process, which is the superposition of the two processes, will be the sum of the number of events occured in the two processes. Therefore, it is Poisson distributed with parameter $\lambda_1 T + \lambda_2 T = \lambda T$.

Notice also that the time between two consecutive events in the process, which is the superposition of the two processes, will be the minimum of two exponential random variables: one with parameter λ_1 and the other with parameter λ_2. We know that this minimum is exponentially distributed with parameter $\lambda = \lambda_1 + \lambda_2$.

In general: the superposition of k Poisson processes with parameters $\lambda_1, \lambda_2, \ldots, \lambda_k$, is a Poisson process with rate $\lambda = \lambda_1 + \lambda_2, \ldots, +\lambda_k$.

Again, show it by induction.

Homework

Consider superposition of two Poisson processes, namely, $X(t)$ is a superposition of $X_1(t)$ and $X_2(t)$ where $X_i(t)$ is a Poisson process with parameter λ_i, $i = 1, 2$.

What is the probability that the next event that occurs will be from $X_1(t)$ (or from $X_2(t)$).

Guide: This is equivalent to the question of having say two exponential random variables T_1 and T_2 with parameters λ_1 and λ_2, respectively, and we are interested in the probability of $T_1 < T_2$.

$$P(T_1 < T_2) = \frac{\lambda_1}{\lambda_1 + \lambda_2}.$$

Splitting a Poisson Process

In many networking applications, it is of interest to study the effect of **splitting** of traffic streams.

We will consider two types of splitting: **random splitting** and **regular splitting**.

Consider an arrival (counting) process $X(t), t \geq 0$, of packets to a certain switch called Switch X.

This packet arrival process is assumed to follow a Poisson process with parameter λ.

Some of these packets are then forwarded to Switch A and the others to Switch B.

The counting processes of packets forwarded to A and B are designated $X_A(t)$ and $X_B(t)$, $t \geq 0$, respectively.

Splitting a Poisson Process (cont.)

Under **random splitting**, every packet that arrives at Switch X is forwarded to A with probability p and to B with probability $1-p$ independently of any other event associated with other packets.

In this case, the process $X_A(t), t \geq 0$ follows a Poisson process with parameter $p\lambda$ and the process $X_B(t), t \geq 0$ follows a Poisson process with parameter $(1-p)\lambda$.

We assume that the forwarding is done instantly upon arrival at Switch X, so

$$X(t) = X_A(t) + X_B(t).$$

One simple way to show it is to show that the inter-arrival times of $X_A(t)$ (and the same goes for $X_B(t)$) are IID exponentially distributed random variable. Recall that a geometric sum of exponential random variables is an exponential random variable.

Splitting a Poisson Process (cont.)

Regular Splitting

If the splitting is not random, but **regular**, for example, the first packet that arrives at Switch X is forwarded to A the second to B, the third to A, the fourth to B, etc.

In this case, the packet stream from X to A (the X-A Process) will not follow a Poisson.

It will follow a stochastic process which is a point process where the inter-arrival times are Erlang distributed with parameters λ and 2.

Homework: Explain why the inter-arrival times are Erlang distributed with parameters λ and 2.

Poisson Sampling

The properties of the Poisson process, namely, independence and time-homogeneity, make the Poisson process, also known as **pure chance process**, able to randomly inspect other continuous-time stochastic processes in a way that the sample it provides gives us enough information on what is called **time-averages**.

In other words, its inspections are not biased.

Examples of time-averages are the proportion of time a process $X(t)$ is in state i, i.e., the proportion of time during which $X(t) = i$, or the overall mean of the process defined by

$$E[X(t)] = \frac{\int_0^T X(t)dt}{T}$$

for an arbitrarily large T.

Poisson Sampling (cont'd)

We are interested in generating exactly k sample points that follow a Poisson process during time T. How can it be done efficiently?

The pure chance nature of the Poisson process implies that at any point in time (or more precisely, for any arbitrarily small interval of length Δt), the probability of an occurrence is the same, and it is independent of any other occurrences.

This property gives rise to the following interesting and useful property.

If we generate k **uniform** independent random deviates within time interval $[0,\ T]$ and we order them in increasing order of their values, the resulting point process is statistically equivalent to a Poisson process that has exactly k occurrences during interval $[0,\ T]$.

This gives rise to an efficient way to generate exactly k sample points that follow a Poisson process during time T.

Markov Modulated Poisson Process (MMPP)

The stochastic process called **Markov modulated Poisson process (MMPP)** is a point process that behaves as a Poisson process with parameter λ_i for a period of time that is exponentially distributed with parameter δ_i.

Then, it moves to mode j where it behaves like a Poisson process with parameter λ_j for a period of time that is exponentially distributed with parameter δ_j.

The parameters δ_is are called **mode duration parameters**.

In general, the MMPP can have an arbitrary number of modes, so it requires a transition probability matrix as an additional set of parameters to specify the probability that it moves to mode j given that it is in mode i.

However, we are mostly interested in the simplest case of MMPP – the two mode MMPP denoted MMPP(2) and defined by only four parameters: λ_0, λ_1, δ_0, and δ_1.

MMPP (cont'd)

The MMPP(2) behaves as a Poisson process with parameter λ_0 for a period of time that is exponentially distributed with mode duration parameter δ_0.

Next, it moves to mode 1 where it behaves like a Poisson process with rate λ_1 for a period of time that is exponentially distributed with parameter δ_1.

Then, it switches back to mode 0, etc. alternating between the two modes 0 and 1.

Interrupted Poisson Process (IPP)

A special case of MMPP(2), where $\lambda_0 = 0$, is called the **Interrupted Poisson Process (IPP)**.

The IPP is characterized by three parameters λ, δ_0, and δ_1.

It behaves as a Poisson process with parameter $\lambda > 0$ during an active mode for a period of time that is exponentially distributed with parameter δ_1.

Next, it is interrupted with no arrivals at all during a nonactive mode for a period of time that is exponentially distributed with parameter δ_0.

Then, it switches back to an active mode, etc. alternating between the two modes active and nonactive.

Markov Chains

Markov chains are certain discrete space stochastic processes which are amenable for analysis and hence are very popular for analysis, traffic characterization and modeling of queueing and telecommunications networks and systems. They can be classified into two groups:

1. discrete-time Markov chains
2. continuous-time Markov chains.

Discrete-time Markov Chains

A discrete-time Markov chain is a discrete-time stochastic process

$$\{X_n, n = 0,\ 1,\ 2,\ \ldots\}$$

with the Markov property; namely, that at any point in time n, the future evolution of the process is dependent only on the state of the process at time n, and is independent of the past evolution of the process. The state of the process can be a scalar or a vector.

For simplicity, we will mainly discuss the case where the state of the process is a scalar.

The discrete-time Markov chain $\{X_n,\}$ at any point in time may take many possible values.

Discrete-time Markov Chains (cont'd)

The set of these possible values is finite or countable and it is called the state space of the Markov chain, denoted by Θ.

A **time-homogeneous Markov chain** is a process in which

$$P(X_{n+1} = i \mid X_n = j) = P(X_n = i \mid X_{n-1} = j) \quad \text{for } n = 0,\ 1,\ 2,\ \ldots,$$

$i, j \in \Theta$.

Henceforth, for simplicity of exposition, we consider that all the states are elements of Θ and we do not explicitly write it.
We will also only consider Markov chains which are time-homogeneous.

Discrete-time Markov Chains (cont'd)

A discrete-time time-homogeneous Markov chain is characterized by the property that, for any n, given X_n, the distribution of X_{n+1} is fully defined regardless of states that occur before time n.

That is,

$$P(X_{n+1} = j \mid X_n = i)$$

$$= P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, \ \ldots).$$

Discrete-time Markov Chains (cont'd)

A Markov chain is characterized by the so-called **Transition Probability Matrix** denoted **P** which is a matrix of one step transition probabilities P_{ij}, namely,

$$\mathbf{P} = [P_{ij}]$$

where

$$P_{ij} = P(X_{n+1} = j \mid X_n = i) \ \text{ for all } n.$$

We can observe in the latter that the event $\{X_{n+1} = j\}$ depends only on the state of the process at X_n and the transition probability matrix **P**.

Since the $P_{ij}s$ are probabilities and since when you transit out of state i, you must enter some state, all the entries in **P** are non-negatives, less or equal to 1, and the sum of entries in each row of **P** must add up to 1.

Discrete-time Markov Chains (cont'd)

Example:

Consider the following Transition probability matrix:

$$\mathbf{P} = \begin{bmatrix} 0.2 & 0.1 & 0.7 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}.$$

Assume $\Theta = \{0, 1, 2\}$.

Then, $P_{0,0} = 0.2$, $P_{0,1} = 0.1$, $P_{0,2} = 0.7$, $P_{1,0} = 0.4$,$P_{1,1} = 0.3$, $P_{1,2} = 0.3$, $P_{2,0} = 0.2$, $P_{2,1} = 0.2$, and $P_{2,2} = 0.6$.

In this case, for example:

$$P_{1,2} = P(X_{n+1} = 2 \mid X_n = 1) = 0.3 \text{ for all } n,$$

or

$$P_{2,1} = P(X_{n+1} = 1 \mid X_n = 2) = 0.2 \text{ for all } n.$$

Discrete-time Markov Chains (cont'd)

A state is called **positive recurrent** if the mean time (number of steps) to return to the state is finite (this implies also that the probability to return to the state in finite number of steps is positive).

A Markov chain is said to be **stable** if all the states in its state space are positive recurrent.

A state i is said to be **accessible** from another state j if there is a positive probability for the Markov chain to reach state i at some time in the future if it is now in state j.

A Markov chain is said to be **irreducible** if all the states in its state space are accessible from each other.

Discrete-time Markov Chains (cont'd)

State i is said to be **aperiodic** if $P_{i,i} > 0$.

A Markov chain is said to be **aperiodic** if all the states in its state space are aperiodic.

We denote by π_j, $j = 0,\ 1,\ 2,\ \ldots,$ the components of the vector $\mathbf{\Pi}$. The π_j is the steady-state probability of the Markov chain to be at state j.

This means that if we simulate a Markov chain for sufficiently long time to be in steady state and we pick a random time, the probability that the Markov chain is in state j is π_j.

Discrete-time Markov Chains (cont'd)

For an irreducible, aperiodic and stable Markov chain, the steady-state probabilities can be obtained by solving the following steady-state equations:

$$\pi_j = \sum_{i=0}^{\infty} \pi_i P_{ij} \text{ for all } j,$$

or in matrix notation

$$\mathbf{\Pi} = \mathbf{\Pi P},$$

and also the normalizing equation

$$\sum_{j=0}^{\infty} \pi_j = 1.$$

When the state space Θ is finite, one of the steady state equation is redundant and is replaced by the normalizing equation.

Discrete-time Markov Chains (cont'd)

In many real life applications, the state of the system either stays unchanged or sometimes increases by one, and at other times decreases by one, and no other transitions are possible.

Such a discrete-time Markov chain $\{X_n\}$ is called in various ways, e.g., **Discrete-Time Birth-Death Chains**.

We use the term **birth-and-death process** for both the discrete and continuous time versions.

In this case, $P_{ij} = 0$ if $|i - j| > 1$ and $P_{ij} > 0$ if $|i - j| \leq 1$, for $i \geq 0$ and $j \geq 0$.

By the first equation of the steady state equations, we obtain:

$$\pi_0 P_{01} = \pi_1 P_{10}.$$

Substituting the latter in the second equation of the steady state equations, we obtain

$$\pi_1 P_{12} = \pi_2 P_{21}.$$

Discrete-time Markov Chains (cont'd)

Homework: Derive the last two equations.

Solution:

$$\pi_0 = \pi_0 P_{00} + \pi_1 P_{10}$$

$$\pi_0(1 - P_{00}) = \pi_1 P_{10}$$

Since $P_{00} + P_{01} = 1$, we obtain

$$\pi_0 P_{01} = \pi_1 P_{10}$$

Discrete-time Markov Chains (cont'd)

Next,

$$\pi_1 = \pi_0 P_{01} + \pi_1 P_{11} + \pi_2 P_{21}.$$

From the last two equations we obtain

$$\pi_1 = \pi_1(P_{11} + P_{10}) + \pi_2 P_{21}$$

$$\pi_1(1 - P_{11} - P_{10}) = \pi_2 P_{21}$$

and since $P_{10} + P_{11} + P_{12} = 1$, we obtain

$$\pi_1 P_{12} = \pi_2 P_{21}.$$

This completes the derivation of the two equations.

Discrete-time Markov Chains (cont'd)

Continuing in the same way, we obtain

$$\pi_i P_{i,i+1} = \pi_{i+1} P_{i+1,i}, \quad i = 0, 1, 2, \ldots \ .$$

These equations are called **detailed balance equations**. They together with the normalizing equation

$$\sum_{i=0}^{\infty} \pi_i = 1$$

constitute a set of steady-state equations for the steady-state probabilities.

They are simpler than the original equations.

Homework: Solve the detailed balance equations together with the normalizing equations for the π_i, $i = 0, 1, 2, \ldots$.
Guide: Recursively, write all π_i, $i = 0, 1, 2, 3, \ldots$ in terms of π_0. Then use the normalizing equation and isolate π_0.

Discrete-time Multi-dimensional Markov Chains

So far, we discussed single dimensional Markov chains.

If the state space is made of finite vectors instead of scalars, we can easily convert them to scalars and proceed with the above described approach.

For example, if the state-space is (0,0) (0,1) (1,0) (1,1) we can simply change the names of the states to 0,1,2,3 by assigning the values 0, 1, 2 and 3 to the states (0,0), (0,1), (1,0) and (1,1), respectively.

All we need to do is to consider a 4×4 transition probability matrix as if we have a single dimension Markov chain.

Continuous-time Markov Chains

A continuous-time Markov chain is a continuous-time stochastic process $\{X_t, t \geq 0\}$.

At any point in time $t > 0$, X_t describes the state of the process which is discrete.

We will consider only continuous-time Markov chain where X_t takes values that are nonnegative integers.

The time between changes in the state of the process is exponentially distributed.

In other words, the process stays constant for an exponential time duration before changing to another state.

Continuous-time Markov Chains (cont'd)

In general, a continuous-time Markov chain $\{X_t, t \geq 0\}$ is defined by the property that for all real numbers $s \geq 0$, $t \geq 0$ and $v \geq 0$, and integers $i \geq 0$, $j \geq 0$ and $k \geq 0$,

$$P(X_{t+s} = j \mid X_t = i, X_v = k_v, v \leq t) = P(X_{t+s} = j \mid X_t = i).$$

That is, the probability distribution of the future values of the process X_t, represented by X_{t+s}, given the present value of X_t and the past values of X_t denoted X_v, is independent of the past and depends only on the present.

Continuous-time Markov Chains (cont'd)

A general continuous-time Markov chain can also be defined as a continuous-time discrete space stochastic process with the following properties.

1. Each time the process enters state i, $i \geq 0$, it stays at that state for an amount of time which is exponentially distributed with parameter δ_i before making a transition into a different state.
2. When the process leaves state i, it enters state j, $j \geq 0$, with probability denoted P_{ij}. The set of P_{ij}s must satisfy the following:

$$\begin{array}{ll} (1) & P_{ii} = 0 \ \text{ for all } i \\ (2) & \sum_j P_{ij} = 1. \end{array}$$

Continuous-time Markov Chains (cont'd)

Two examples of a continuous-time Markov chain

Example 1: A **Poisson process** with rate λ.

The state at time t, $\{X_t, t \geq 0\}$ can be the number of occurrences by time t which is the counting process $N(t)$.

In this example of the Poisson counting process $X_t = N(t), t \geq 0$ increases by one after every exponential time duration with parameter λ.

Continuous-time Markov Chains (cont'd)

Example 2: The so-called **pure birth process** $\{X_t, t \geq 0\}$. It is a generalization of the counting Poisson process.

Again $\{X_t, t \geq 0\}$ increases by one every exponential amount of time but here, instead of having a fixed parameter λ for each of these exponential intervals, this parameter depends of the state of the process and it is denoted δ_i.

In other words, when $X_t = i$, the time until the next occurrence in which $\{X_t, t \geq 0\}$ increases from i to $i + 1$ is exponentially distributed with parameter δ_i.

If we set $\delta_i = \lambda$ for all i, we have the Poisson counting process.

Continuous-time Markov Chains (cont'd)

As in the case of the discrete-time Markov chain, in many real-life applications, the state of the system in one point in time sometimes increases by one, and at other times decreases by one, but never increase or decrease by more than one at one time instance.

Such a continuous-time Markov chain $\{X_t, t \geq 0\}$, as its discrete-time counterpart, is called a **birth-and-death process**.

In such a process, the time between occurrences in state i is exponentially distributed, with parameter δ_i, and at any point of occurrence, the process increases by one (from its previous value i to $i+1$) with probability υ_i and decreases by one (from i to $i-1$) with probability $\vartheta_i = 1 - \upsilon_i$.

The transitions from i to $i+1$ are called **births** and the transitions from i to $i-1$ are called **deaths**.

Recall that the mean time between occurrences, when in state i, is $1/\delta_i$.

Continuous-time Markov Chains (cont'd)

Hence, the birth rate in state i, denoted b_i, is given by

$$b_i = \begin{cases} \delta_i \upsilon_i & \text{for } i \geq 1 \\ \delta_i & \text{for } i = 0, \end{cases}$$

and the death rate (d_i) is given by

$$d_i = \begin{cases} \delta_i \vartheta_i & \text{for } i \geq 1 \\ 0 & \text{for } i = 0. \end{cases}$$

Summing up these two equations gives the intuitive result that the total rate at state i is equal to the sum of the birth-and-death rates.

Namely,

$$\delta_i = b_i + d_i, \text{ for } i \geq 0,$$

and therefore the mean time between occurrences is

$$\frac{1}{\delta_i} = \frac{1}{b_i + d_i}, \text{ for } i \geq 0.$$

Continuous-time Markov Chains (cont'd)

Birth-and-death processes apply to queueing systems where customers arrive one at a time and depart one at a time.

Consider for example a birth-and-death process with the death rate higher than the birth rate.

Such a process could model, for example, a stable single-server queueing system.

Homework: Show the following:

$$\vartheta_i = \frac{d_i}{b_i + d_i}, \text{ for } i \geq 0$$

and

$$\upsilon_i = \frac{b_i}{b_i + d_i}, \text{ for } i \geq 0.$$

First Passage Time

An important problem that has applications in many fields, such as biology, finance and engineering, is how to derive the distribution or moments of the time it takes for the process to transit from state i to state j.

In other words, given that the process is in state i find the distribution of a random variable representing the time it takes to enter state j for the first time.

This random variable is called the **first passage time from i to j**.

Let us derive the mean of the first passage time from i to j in a birth-and-death process for the case $i < j$.

Continuous-time Markov Chains (cont'd)

To solve this problem we start with a simpler one. Let U_i be the mean passage time to go from i to $i+1$. Then

$$U_0 = \frac{1}{b_0}.$$

and

$$U_i = \frac{1}{\delta_i} + \vartheta_i[U_{i-1} + U_i], \quad \text{for } i \geq 1.$$

Continuous-time Markov Chains (cont'd)

Explanation:

Notice that U_{i-1} is the mean passage time to go from $i-1$ to i, so $U_{i-1}+U_i$ is the mean passage time to go from $i-1$ to $i+1$. Notice that U_i the mean passage time to go from i to $i+1$ is equal to the mean time the process stays in state i (namely $1/\delta_i$), plus the probability to move from i to $i-1$, times the mean passage time to go from $i-1$ to $i+1$.

Notice that the probability of moving from i to $i+1$ is not considered because if the process moves from i to $i+1$ when it completes its sojourn in state i then the process reaches the target (state $i+1$), so no further time needs to be considered.

Continuous-time Markov Chains (cont'd)

Therefore,

$$U_i = \frac{1}{b_i + d_i} + \frac{d_i}{b_i + d_i}[U_{i-1} + U_i], \quad i \geq 1,$$

or

$$U_i = \frac{1}{b_i} + \frac{d_i}{b_i}U_{i-1}, \quad i \geq 1.$$

Now we have a recursion by which we can obtain $U_0, U_1, U_2, \ \ldots$, and the mean first passage time between i and j, for $j > i$, is given by the sum

$$\sum_{k=i}^{j-1} U_k.$$

Homework: Let $b_i = \lambda$ for $i \geq 0$ and $d_i = \mu$ for all i for $i \geq 1$, derive a closed form expression for U_i.

Continuous-time Markov Chains (cont'd)

As in the case of the discrete-time Markov chain, define a continuous-time Markov chain to be called **irreducible** if there is a positive probability for any state to reach every state.

We define a state in a continuous-time Markov chain to be called **positive recurrent**, if the process visits and then leaves that state, the random variable that represents the time it returns to that state has finite mean.

As for discrete-time Markov chains, a continuous-time Markov chain is said to be **stable** if all its states are positive recurrent.

Henceforth we only consider continuous-time Markov chains that are irreducible, aperiodic and stable.

Continuous-time Markov Chains (cont'd)

As for discrete-time Markov chains, π_j is the steady-state probability of the continuous-time Markov chain to be at state j.

We shall now describe how the steady-state probabilities π_j can be obtained.

Again we define $P_{ij}, i \neq j$ as the probability that given that the process now is in state i, it will reach state j next. Then we can define the matrix $\mathbf{P} = [P_{ij}]$ (with zeros in the diagonal).

Continuous-time Markov Chains (cont'd)

We now construct the matrix **Q** which is called the **infinitesimal generator** of the continuous-time Markov chain. The matrix **Q** is a matrix of one step infinitesimal **rates** Q_{ij} defined by

$$Q_{ij} = \delta_i P_{ij} \text{ for } i \neq j \tag{1}$$

and

$$Q_{ii} = -\sum_{j \neq i} Q_{ij}. \tag{2}$$

Remarks:

- The state-space can be finite or infinite and hence the matrices **P** and **Q** can also be finite or infinite.
- Q_{ij} is the product of the rate to leave state i and the probability of transition to state j from state i which is the rate of transitions from i to j.

Continuous-time Markov Chains (cont'd)

To obtain the steady-state probabilities π_js, we solve the following set of steady-state equations:

$$0 = \sum_i \pi_i Q_{ij} \ \text{ for all } j$$

and the *normalization equation* that ensures that the sum of the steady-state probabilities is equal to one is

$$\sum_j \pi_j = 1.$$

Denoting $\mathbf{\Pi} = [\pi_0, \pi_1, \pi_2, \ \ldots]$, the steady state equations above can be written as

$$0 = \mathbf{\Pi Q}.$$

Continuous-time Markov Chains (cont'd)

The quantity $\pi_i Q_{ij}, i \neq j$ which is the steady-state probability of being in state i times the infinitesimal rate of a transition from state i to state j is called the **probability flux** from state i to state j.

The steady state equations guaranty that the total probability flux from all states into any given state j is equal to the total probability flux out of state j to all other states. These steady state equations are called **global balance equations**.

The j column in the $\mathbf{Q}$ matrix corresponds to the jth global balance equation, where the infinitesimal rates Q_{ij} for $i \neq j$ represent rates into state j and Q_{jj} is the total (negative) rate out of state j as it is equal to the negative sum of the other elements in the jth row.

Continuous-time Markov Chains (cont'd)

To explain the equality of the global balance equations, consider a long period of time L. Assuming the process returns to all states infinitely many times, during a long time period L, the number of times the process moves into state j is equal (in the limit $L \to \infty$) to the number of times the process moves out of state j.

Similar to the case of discrete-time Markov chains, the set of steady-state equations is dependent and one of the equations so it is redundant in the finite state space case, and it is replaced by the normalization equation.

For continuous-time birth-and-death processes, $Q_{ij} = 0$ for $|i - j| > 1$. As in the discrete-time case, under this special condition, the global balance equations can be simplified to the **detailed balance equations**.

Continuous-time Markov Chains (cont'd)

We start with the first steady state equation and using the condition $Q_{ij} = 0$ for $|i - j| > 1$, we obtain

$$\pi_0 Q_{01} = \pi_1 Q_{10}$$

The second equation is

$$\pi_1 [Q_{10} + Q_{12}] = \pi_0 Q_{01} + \pi_2 Q_{21}.$$

Then we obtain

$$\pi_1 Q_{12} = \pi_2 Q_{21}.$$

In a similar way, by repeating the process, we obtain the following detailed balance equations.

$$\pi_i Q_{i,i+1} = \pi_{i+1} Q_{i+1,i} \quad i = 0, 1, 2, \ \ldots \ .$$

Continuous-time Markov Chains (cont'd)

Continuous-time Multi-dimensional Markov Chains

The extension discussed earlier regarding multi-dimensional discrete-time Markov chains applies also to the case of continuous-time Markov chains.

If the state-space is made of finite vectors instead of scalars, as discussed, there is a one-to-one correspondence between vectors and scalars, so a multi-dimensional continuous-time Markov chain can be converted to a single-dimension continuous-time Markov chain and we proceed with the above described approach that applies to the single dimension.

Note that the above described MMPP is a Multi-Dimensional continuous time Markov chain.

Continuous-time Markov Chains (cont'd)

Solutions by Successive Substitutions

When the matrix $\mathbf{Q}$ is large but finite, there is a need to solve a set of steady-state equations efficiently.

An efficient way that normally works well for solving such equations is to use a method of successive substitutions which is applicable to both discrete-time and continuous-time Markov chains, but in our description of the method we consider for example a finite set of steady-state equations of a continues-time Markov chain of the form

$$0 = \mathbf{\Pi Q}$$

where $\mathbf{\Pi} = [\pi_0, \pi_1, \pi_2, \pi_3, \ldots, \pi_k]$ and $\mathbf{Q}$ is the infinitesimal generator $(k+1) \times (k+1)$ matrix of the continuous time Markov chain, and we also have the normalization equation.

Continuous-time Markov Chains (cont'd)

The method can be described as follows.

First, isolate the first element of the vector $\mathbf{\Pi}$; in this case, it is the variable π_0 in the first equation.

Next, isolate the second element of the vector $\mathbf{\Pi}$, namely, π_1 in the second equation, and then keep isolating all the variables of the vector $\mathbf{\Pi}$. This leads to the following vector equation for $\mathbf{\Pi}$

$$\mathbf{\Pi} = \mathbf{\Pi}\hat{\mathbf{Q}},$$

where $\hat{\mathbf{Q}}$ is different from the original $\mathbf{Q}$ because of the algebraic operations we performed when we isolated the elements of the $\mathbf{\Pi}$ vector.

Continuous-time Markov Chains (cont'd)

Then, perform the successive substitution operations by first setting arbitrary initial values to the vector $\mathbf{\Pi}$; substitute them in the right-hand side of the above equations and obtain different values at the left-hand side which are then substituted back in the right-hand side, etc.

For example, the initial setting can be $\mathbf{\Pi} = 1$ without any regards to the normalization equation.

When the values obtained for $\mathbf{\Pi}$ are sufficiently close to those obtained in the previous subsection, say, within a distance no more than 10^{-6}, stop.

Finally, normalize the vector $\mathbf{\Pi}$ obtained in the last iteration. This is the desired solution.

The Curse of Dimensionality

In many applications, the **Q** matrix is too large, so it may not be possible to solve the steady-state equations in reasonable time. Actually, the case of a large state-space (or large **Q** matrix) is common in practice.

This is often occur when the application lead to a Markov-chain model that is of high dimensionality. Consider for example a 49 cell mobile network, and assume that every cell has 23 voice channels.

Assuming Poisson arrivals and exponential call time duration and cell sojourn times.

Then this cellular mobile network can be modeled as a continuous time Markov chain with each state representing the number of busy channels in each cell.

Continuous-time Markov Chains (cont'd)

In this case, the number of states is equal to 24^{49}, so a numerical solution of the steady-state equations is computationally prohibitive.

When an exact numerical solution is not attainable, we often rely on simulations or approximations which we will discuss later.

Renewal Process

An informal way to describe a **renewal process** is a generalization of the Poisson process where the inter-arrival times are not necessarily exponentially distributed but are still positive IID random variables with finite mean.

Because of this generalization a renewal process is not necessarily memoryless.

There are discrete-time and continuous-time renewal processes. In a discrete-time renewal process the inter-arrival times take only positive integer values while in a continuous-time renewal processes the inter-arrival times are positive real valued.

Note: The definition of a renewal process is sometimes extended to cases that allow a zero value for the inter-arrival times to occur with positive probability. This implies more than one arrival at the same point in time.

IPP – a Renewal Process

The IPP is a renewal process because the time from an arrival (called arrival A) until the next arrival (called arrival B) is independent of the evolution of the process before event A.

Therefore, the inter-arrival times are independent.

In fact, because of the memoryless property of the exponential distribution of the inter-arrival times during the active mode, the distribution of the inter-arrival times is equal to the distribution of the time from any moment the system is in the active mode until the next arrival.

Therefore, the inter-arrival times are IID.

IPP – a Renewal Process (cont'd)

Because of the underlying structure of the IPP, the inter-arrival times are also continuous and positive.

Let X be a random variable representing the inter-arrival time of the IPP.

As mentioned above, because of the memoryless property of exponential distribution, X is equal to the time until the next arrival given that now the system is in active mode.

IPP – a Renewal Process (cont'd)

Then X can be described by the following recursive equation.

$$X = Y + \alpha(Z + X),$$

where Y is an exponentially distributed random variable with parameter $\lambda + \delta_1$ representing the time from the last arrival until the next event that can be either an arrival or change of mode from active to nonactive,

Z is an exponentially distributed random variable with parameter δ_0 representing the duration of a nonactive period,

X on the right-hand side represents the time until the next arrival from the moment the system is again in the active mode,

and α is the probability that the next event after an arrival is a change of mode from active to nonactive, given by

$$\alpha = \frac{\delta_1}{\lambda + \delta_1}.$$

IPP – a Renewal Process (cont'd)

Then

$$X = \frac{Y + \alpha Z}{1 - \alpha}.$$

$$X = \frac{Y + \frac{\delta_1}{\lambda+\delta_1} Z}{1 - \frac{\delta_1}{\lambda+\delta_1}} = \frac{(\lambda + \delta_1) Y + \delta_1 Z}{\lambda}.$$

Thus,

$$X = \frac{\lambda + \delta_1}{\lambda} Y + \frac{\delta_1}{\lambda} Z.$$

Taking expectations on both sides gives

$$E[X] = \frac{\lambda + \delta_1}{\lambda} E[Y] + \frac{\delta_1}{\lambda} E[Z].$$

IPP – a Renewal Process (cont'd)

By the definitions of the random variables Y and Z, we have

$$E[Y] = \frac{1}{\lambda + \delta_1},$$

and

$$E[Z] = \frac{1}{\delta_0}.$$

After substituting the latter two equations in the previous one, we obtain

$$E[X] = \frac{\lambda + \delta_1}{\lambda}\left(\frac{1}{\lambda + \delta_1}\right) + \frac{\delta_1}{\lambda}\left(\frac{1}{\delta_0}\right) = \frac{\delta_0 + \delta_1}{\lambda \delta_0}.$$

IPP – a Renewal Process (cont'd)

Let us now obtain $E[X]$ **in a different way**.

Consider the two-state continuous-time Markov chain of the mode process which alternates between state 0 (non-active mode) and state 1 (active mode).

Let π_i be the probability of being in state i, $i = 0, 1$.

The 2×2 **Q** matrix is $q_{00} = -\delta_0$, $q_{01} = \delta_0$, $q_{10} = \delta_1$, $q_{11} = -\delta_1$. That is,

$$\mathbf{Q} = \begin{bmatrix} -\delta_0 & \delta_0 \\ \delta_1 & -\delta_1 \end{bmatrix}.$$

IPP – a Renewal Process (cont'd)

This gives rise to the steady state equations $\mathbf{\Pi Q} = 0$, which are two dependent equations, so from one of them we have

$$\pi_0\delta_0 = \pi_1\delta_1.$$

We also have the normalizing equation

$$\pi_0 + \pi_1 = 1.$$

Solving these equations, we obtain

$$\pi_0 = \frac{\delta_1}{\delta_0 + \delta_1}$$

and

$$\pi_1 = \frac{\delta_0}{\delta_0 + \delta_1}.$$

IPP – a Renewal Process (cont'd)

The arrival rate of the IPP is given by

$$\lambda_{IPP} = \lambda\pi_1 + 0 \times \pi_0 = \frac{\lambda\delta_0}{\delta_0 + \delta_1}.$$

Therefore, the mean inter-arrival time $E[X]$ is obtained by

$$E[X] = \frac{1}{\lambda_{IPP}} = \frac{\delta_0 + \delta_1}{\lambda\delta_0}.$$

This is consistent with the previous result.

Chapter 3: General Queueing and Teletraffic Concepts

© Moshe Zukerman

June 3, 2020

Why Queueing modelling?

- There are queues everywhere: routers, computers, mobile phones, banks, shops, supermarkets, hospitals, airports, roads, trains, libraries, toilets, etc. etc.
- Cornerstone of efficiency – the study of queues helps in making systems more efficient.
- Resource/facility dimensioning – how many servers or how much link capacity we need to meet customers quality of service (QoS) – tradeoffs between cost and QoS. Problem: minimize cost subject to meeting QoS requirements.
- Traffic flow management/routing (roads, airplanes, Internet, etc.) to avoid congestion.
- Scheduling and prioritization of tasks, jobs, patients, packets, programs, etc.

Bridging the Gap between Theory and Practice through Simplification

- In practice, queueing systems and networks often involve complex characteristics and processes that not amenable to analysis.
- Nevertheless, insight can be gained using simpler queueing models.
- Modelling simplification is made when the aim is to analyze a complex queueing system or network, such as the Internet, where packets on their ways to their destinations arrive at a router where they are stored and then forwarded according to addresses in their headers.
- One element in the simplification process is the single-server queue (SSQ).
- Queueing analyses explain traffic and management processes and their effect on queueing performance to help improve efficiency.

Kendall's Notation

In 1953, Kendall proposed notation of the following structure: A/S/k

A – time between arrivals; Examples: D (Deterministic), M (Markovian/Memoryless - Poisson process), G (General), GI (General and independent), and Geom (Geometric).
S – service time distribution; Examples: D (Deterministic), M (Markovian/Memoryless - exponential), G (General), GI (General and independent), and Geom (Geometric).
k – number of servers.

This has since been extended to A/S/k/N/Pop/Disc

N – total waiting room (buffer size); number of places in the buffer including those customers that are being served. Default: infinity.
Pop – size of the total population from where the arrivals are coming. Default: infinity.
Disc – the queue discipline; Examples: First In First Out (FIFO), Last In First Out (LIFO), Processor Sharing (PS), ...). Default: FIFO.

Kendall's Notation (cont'd)

If default are used for *Pop* (if it is infinite), we use a dash "-" instead of "/" before the queue discipline (Disc).
If *Pop* is finite, no more than *Pop* customers will occupy the buffer even if N is infinite, so we do not use the default option.
However, if both *Pop* and N are infinite, we use the default for both, and, again, we use a dash "-" instead of "/" before the queue discipline (Disc).

Examples:

M/M/1-PS represents the case of Poisson arrivals, exponential service times, one server (SSQ), unlimited buffer, infinite sources, and processor sharing service discipline.

M/M/1/N represents the case of Poisson arrivals, exponential service times, single server, N buffer places, infinite sources, and FIFO service discipline.

Kendall’s Notation (cont’d)

More Examples:

D/D/1 denotes an SSQ with infinite buffer and population, FIFO queue discipline, where both the inter-arrival times and the service times are deterministic. This means that the inter-arrival times are all equal to each other and the service times are all equal to each other.

M/M/1 denotes an SSQ with a Poisson arrival process and exponential service times, with infinite buffer and population, and FIFO service discipline.

GI/M/1 is an SSQ which is generalization of M/M/1 where the arrival process is a renewal process and not necessarily Poisson.

Kendall's Notation (cont'd)

More Examples:

G/G/1 is a further generalization of GI/M/1 where the service times are not necessarily exponentially distributed and may even depend on each other, and the inter-arrival times may also be dependent. G/G/1 is the most general infinite buffer FIFO SSQ considered in queueing theory where both the arrival and service processes are general.

M/M/k/k represents the case of Poisson arrivals, exponential service times, k servers, k buffer places, infinite sources, and FIFO service discipline. No buffer places are available beyond the k places allocated to customers that are being served. If all k servers are busy (serving k customers) any additional arriving customer will be blocked.

M/G/k/k represents a generalization of M/M/k/k to the case of generally distributed service times. As M/M/k/k, M/G/k/k represents a k-server queue with Poisson arrivals and FIFO service discipline, without additional waiting room except at the servers.

Kendall's Notation (cont'd)

More Examples:

M/G/k/N (with $N \geq k$) represents k-server queue with Poisson arrivals and generally distributed service times which is a generalization of M/G/k/k where a waiting room that can accommodate up to $N - k$ customers is added.

M/G/1-PS is a generalization of M/M/1-PS where the service requirements are generally distributed. As M/M/1-PS, M/G/1-PS represents a single server processor sharing queue with Poisson arrivals. Notice that in an M/G/1-PS queue (and its special case M/M/1-PS), although the service time of a customer/packet starts immediately upon arrival it may continue for a longer time than its service requirement, because the server capacity is always shared among all customers/packets in the system.

Kendall's Notation (cont'd)

More Examples:

M/G/1/N-PS is a single server processor sharing queue with Poisson arrivals and a generally distributed service time that applies to all the customers, and the waiting room can accommodate up to N customers including the customer in service. A special case of the latter is the M/M/1/N-PS queue where the service time distribution is exponential.

M/G/k/N/Pop is a finite source queueing model where the population of customers is limited to Pop with FIFO service policy, Poisson arrivals, generally distributed service times, k servers, and the buffer size is limited to N.

Utilization

Utilization ($\hat{U}$) is the proportion of time that a server is busy on average.

$0 \leq \hat{U} \leq 1$

It is an important measure for queueing systems performance and efficiency. In many cases, cost incur all the time but income is received when server or facility is busy.

With multiple identical servers, the average utilization is considered.

Therefore, $\hat{U} = 0$ for M/M/∞ because in this case, the mean number of busy servers is finite and the mean number of idle servers is infinite.

Utilization (cont'd)

Utilization of G/G/1

Consider a G/G/1 queue.
Random variable S is the service time; $E[S] = 1/\mu$ (μ is the service rate.)
λ is the arrival rate.
Assume that $\mu > \lambda$ so that the queue is *stable*.

$$\text{Then,} \quad \hat{U} = \frac{\lambda}{\mu}.$$

Explanation: let L be an arbitrarily long period of time. The average number of customers arrived within time period L is: λL. The average number of customers that has been served during time period L is equal to $\mu \hat{U} L$. Note that during L, customers are being served only during $\hat{U}$ proportion of L when the system is not empty.
Since L is arbitrarily long and the queue is stable, these two values can be considered equal. Thus, $\mu \hat{U} L = \lambda L$. Hence, $\hat{U} = \frac{\lambda}{\mu}$.

Utilization (cont'd)

Another Explanation

Again, consider a very long period of time L. The assumption of L being arbitrarily long is required for both explanations to ensure that the edge effects (i.e., effects of a customer being in service at the beginning or the end of the time period L) are negligible.

Again, there are on average λL arrivals during period L. These arrivals require on average a total service time of

$$\lambda L \times \frac{1}{\mu}.$$

Therefore, the proportion of time the server is busy ($\hat{U}$) is obtained by

$$\hat{U} = \frac{\lambda L \times \frac{1}{\mu}}{L} = \frac{\lambda}{\mu}.$$

Little's Formula

Two Forms of Little's Formula for G/G/1

λ – arrival rate of customers to the system

$E[Q]$ – the stationary mean queue-size including the customer in service

$E[D]$ – the mean delay (system waiting time) of a customer from the moment it arrives until its service is completed

$$E[Q] = \lambda E[D]$$

$E[N_Q]$ – the mean number of customers in the queue in steady-state excluding the customer in service

$E[W_Q]$ – the mean delay of a customer, in steady-state, from the moment it arrives until its service commences (waiting time in the queue).

$$E[N_Q] = \lambda E[W_Q]$$

Little's Formula (cont'd)

Little's formula applies to **a wide range of systems** and not only to G/G/1.

An intuitive (non-rigorous) explanation: Consider a customer that just left the system (completed service). This customer sees behind his/her back on average $E[Q]$ customers. Who are these customers? They are the customers that had been arriving during the time that our customer was in the system. Their average number is $\lambda E[D]$.

Little's Formula (cont'd)

The amusement park explanation (Bertsekas, 2002) Consider an amusement park where customers arrive at a rate of λ per time unit. Assume that the park is in stationary condition. An arriving customer spends on average time $E[D]$ at various sites, and then leaves. The park charges one dollar per unit time a customer spends in the park. The mean queue size $E[Q]$ is the mean number customers in the park in steady state. Under these assumptions, the rate at which the park earns its income is $E[Q]$ per unit time. Let L be an arbitrarily long period of time. The mean number of customers that arrive during L is λL. Because L is arbitrarily long, we can assume that all the customers that arrive during L also left during L. Since a customer on average pays $E[D]$ dollars for its visit in the park, and there are on average λL customers visiting during L, the total income earned by the park on average during L is $\lambda L E[D]$. Therefore, the rate at which the park earns its income in steady state per unit time is $\lambda L E[D]/L = \lambda E[D]$. We also know that this rate is equal to $E[Q]$. Therefore, $\lambda E[D] = E[Q]$.

Little's Formula (cont'd)

A graphical proof of Little's formula for G/G/1 (Bertsekas and Gallager, 1992):

Consider a stable G/G/1 queue that starts empty at time $t = 0$.
$A(t)$ – the number of arrivals up to time t
$D(t)$ – the number of departures up to time t
$Q(t)$ – the queue-size (number in the system) at time t

$$Q(t) = A(t) - D(t), \qquad t \geq 0.$$

L – arbitrarily long period of time, so all arrivals during L left during L.

$$E[Q] = \frac{1}{L}\int_0^L Q(t)dt.$$

D_i – the time spent in the system by the ith customer.

$$\int_0^L Q(t)dt = \sum_{i=1}^{A(L)} D_i.$$

Little's Formula (cont'd)

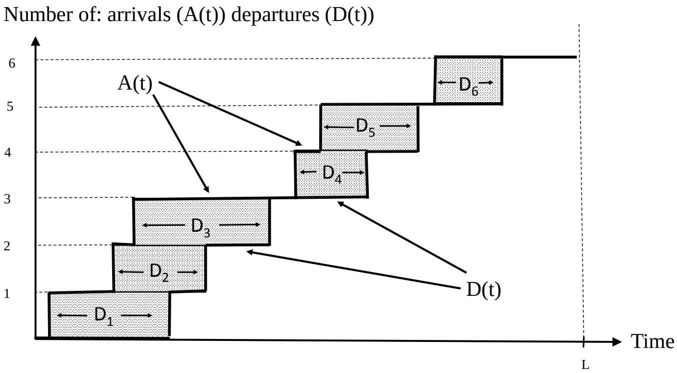


Figure 1: Graphical illustration for the proof of Little's formula for G/G/1.

Little's Formula (cont'd)

Therefore,

$$\frac{1}{L}\int_0^L Q(t)dt = \frac{1}{L}\sum_{i=1}^{A(L)} D_i$$

and realizing that

$$\lambda = \frac{A(L)}{L},$$

and

$$E[D] = \frac{1}{A(L)}\sum_{i=1}^{A(L)} D_i,$$

we obtain

$$E[Q] = \frac{1}{L}\int_0^L Q(t)dt = \frac{A(L)}{L}\frac{1}{A(L)}\sum_{i=1}^{A(L)} D_i = \lambda E[D].$$

Delay and Loss Systems

Delay system: A queueing system with infinite buffer, such as, G/G/1 or G/G/2, where an arriving customer that finds all servers busy will wait in the queue until it is served, or system such as G/G/1-PS where the service time and overall delay is increased with increasing congestion.

Loss system: A system without additional buffer space beyond what is available at the servers, e.g, M/G/k/k, where an arriving customer that finds all servers busy is blocked and cleared from the system. If the call is not blocked, it immediately starts its service upon its arrival. In loss systems, the term *holding time*, is often used for the time spent in the system by a call if it is served by a dedicated server. Let h be the mean holding time. As our holding time definition is associate with loss systems without waiting time, h is also the mean service time, namely,

$$h = E[S] = \frac{1}{\mu}.$$

The term holding time has been often used in telephony to describe the average time the phone call holds a circuit.

Delay and Loss Systems (cont'd)

Systems like M/G/1/k and M/G/1/k-PS are called **delay-loss systems**, such systems are characterized by both effects of delay and loss taking place. If a customer arrives and all servers are busy but the buffer (waiting room) is not full, it may suffer some additional delay, but if the buffer is full, the customer will be rejected and lost.

Finally, there are systems that are **neither delay nor loss systems**. These are **theoretical** systems where the number of servers is infinite. In such systems an arriving customer will always receive service at the full service rate upon arrival and will never be delayed or blocked.

Delay and Loss Systems (cont'd)

- Increasing delay and/or blocking probability which adversely affects users QoS is used by systems to save resources.
- System designers should always consider tradeoff between cost and customers QoS.
- A common optimization problem solved by businesses is:
  minimize: cost
  subject to: meeting customers QoS requirements.

Traffic

Quantity of traffic is measured in units of **erlangs** (named after the Danish mathematician Agner Krarup Erlang).

To measure the quantity of traffic, we consider the arrival rate of customer calls as well as the server time the calls require.

If a quantity of traffic A [erlangs] are offered to a system then A is given by

$$A = \frac{\lambda}{\mu} = \lambda h.$$

In other words, the quantity of traffic in erlangs is the product of the call arrival rate (λ) and the mean time that a single server will serve a call ($1/\mu$), i.e., the mean time that a call occupies a server.

Traffic (cont.)

Notice that A [erlangs] is the mean number of arrivals per mean service time.

Therefore, one erlang is one arrival per mean service time, so if one erlang is admitted, it will require one server on average forever.

Accordingly, if A traffic is admitted, then A is the mean number of servers required in steady state by this traffic load, and it is also the mean number of simultaneous calls/customers served in progress.

If we have N users, and the ith user generates A_i erlangs, the total traffic generated by the N users is

$$A = \sum_{i=1}^{N} A_i \quad \text{[erlang]}.$$

Traffic (cont.)

Question: Consider a wireless system that provides channels each of which can serve one phone call. There are 100 users making phone calls. Each user makes on average one phone call per hour and the average duration of a phone call is three minutes. What is the total traffic in erlangs that the 100 users generate?

Answer: In such questions, it is important first to choose a consistent time unit. Here it is convenient to choose minutes. Accordingly, the arrival rate of each user is $\lambda_i = 1/60 \ \text{calls per minute}$. The mean call duration (or holding time) is 3 minutes, so $A_i = 3/60 = 1/20 \ [\text{erlang}]$, and the total traffic is $100 \times (1/20) = 100/20 = 5$ [erlang].

Traffic (cont.)

The quantity of traffic measured in erlangs is also called *traffic intensity*. Then another related concept is *traffic volume*. Traffic volume is measured in units of erlang-hour (or erlang-minute, or call-hour or call-minute, etc.) and it is a measure of the traffic processed by a facility during a given period of time. Traffic volume is the product of the traffic intensity and the given time period, namely,

$$\textit{Traffic Volume} = \textit{Traffic Intensity} \times \textit{Time Period.}$$

Let $A, \lambda, \mu, \text{and } h$ be the traffic intensity, arrival rate, service rate and holding time, respectively. Then

$$A = \frac{\lambda}{\mu} = \lambda h.$$

Traffic (cont.)

Let a_T be the number of arrivals during the given period of time T. Then the relevant arrival rate can be estimated by

$$\lambda = \frac{a_T}{T}.$$

Let V_T be the traffic volume during T. Then

$$V_T = AT = \frac{a_T h T}{T} = a_T h.$$

This leads to a different (but equivalent) definition of traffic volume, namely, the product of the number of calls during T and the mean holding time, and this explains the traffic volume units of call-hour or call-minute. They are also called erlang-hour or erlang minute. The latter result for V_T can be illustrated by the following example.

Traffic (cont.)

Example

Consider a period of time of three hours and during this period of time, 120 calls have arrived and their average holding time is three minutes, then the traffic volume is $3 \times 120 = 360$ call-minute, or 360 erlang-minute, or $360/60 = 6$ erlang-hour. Then, the traffic intensity in erlangs is obtained by dividing the traffic volume in erlang-hour by the number of hours in the given period of time. In our case, the traffic intensity is $6/3 = 2$ [erlangs].

Offered and Carried Traffic

There are two traffic related concepts called: *offered traffic* and *carried traffic*. The offered traffic is defined as the mean number of arrivals (of customers, calls or packets) per mean service time. Accordingly, it is equal to the ratio λ/μ which is identical to the definition of traffic discussed in the previous section. It is common to use the notation A that we used for traffic, for the offered traffic as well.

The carried traffic is the average number of calls that are being served in the system. In a system such as M/G/k/k every call is served by one server, so the carried traffic is also the mean number of servers required to serve the customers, calls or packets that admitted to the system. It is also equal to the proportion of the offered traffic that is admitted to the system. Both offered and carried traffic are measured in *erlangs*. The relationship between offered and carried traffic is given by

$$[\text{offered traffic}](1 - P_b) = [\text{carried traffic}]$$

Offered and Carried Traffic (cont.)

Therefore, in systems where blocking does not occur, such as delay systems and systems where the number of servers is infinite, we have that $P_b = 0$, the offered and carried traffic are equal to each other.

As mentioned, another term to describe traffic which is often used is the *traffic intensity*. Then *offered traffic intensity* and *carried traffic intensity* are synonymous to offered traffic and carried traffic, respectively. In cases of infinite buffer/capacity systems, such as M/M/1 and M/M/∞, the term traffic intensity is used to describe both offered traffic and carried traffic. Accordingly, ρ is called traffic intensity in the M/M/1 context and A is the traffic intensity in an M/M/∞ system. Others use the term traffic intensity in multiservice system for the offered load per server. To avoid confusion, we will only use the term traffic intensity in the context of a single server queue with infinite buffer, in which case, traffic intensity is always equal to $\rho = \lambda/\mu$.

Work Conservation

Another important concept in queuing theory is the concept of *work conservation*. A queuing system is said to be work conservative if a server is never idle whenever there is still work to be done. For example, M/M/1 and M/M/1/7 are work conservative. However, a stable M/M/3 is not work conservative because a server can be idle while there are customers served by other servers.

Poisson Arrivals See Time Averages (PASTA)

Many of the queueing models we consider in this course involve Poisson arrival processes. The PASTA property is important for analysis and simulations of such queueing models. Let us further explain and prove this important property.

The PASTA property implies that arriving customers in steady state will find the number of customers in the system obeying its steady-state distribution. In other words, the statistical characteristics (e.g., mean, variance, distribution) of the number of customers in the system observed by an arrival is the same as those observed by an independent Poisson inspector.

PASTA (cont'd)

In addition to the assumption of Poisson arrivals, for PASTA to be valid we also need the condition that arrivals after time t are independent of the queue size at time t, $Q(t)$. For example, if we have a single-server queue with Poisson arrivals and the service times have the property that the service of a customer must always terminate before the next arrival, then the arrivals always see an empty queue, and, of course, an independent arrival does not.

However, in all the queueing systems that we study, this condition holds because normally the server cannot predict the exact time of the next arrival because of the pure chance nature of the Poisson process.

Bit-rate Versus Service Rate

In many telecommunications design problems, we know the bit-rate of an output link and we can estimate the load on the system in terms of the arrival rate of items of interest that require service. Examples for such items include packets, messages, jobs, or calls.

To apply queueing theory for evaluation of quality of service measures, it is necessary, as an intermediate step, to calculate the *service rate* of the system which is the number of such items that the system can serve per unit time.

For example, if we know that the average message size is 20 Mega Bytes [MB] and the bit-rate of the output link of our system (the capacity in bit-rate of the output link) of is 8 Gigabits per second [Gb/s] and we are interested in the service rate of the link in terms of messages per second, we first calculate the message size in bits which is $20 \times 8 = 160$ Mega bits [Mb] or 160×10^6 bits. Then the service rate is

$$\frac{8 \times 10^9}{160 \times 10^6} = 50 \text{ messages per second.}$$

Queueing Models and Performance Analyses

We discuss various queueing models that are amenable to analysis.

The analysis is simplest for **deterministic queues** where the inter-arrival and service times are deterministic (fixed values).

Afterwards, we will consider the so-called **Markovian queues**. These queues are characterized by the Poisson arrival process, independent exponential service times and independence between the arrival process and the service times. They are denoted by M in the first two positions (i.e., $\mathrm{M}/\mathrm{M}/\cdot/\cdot$). Because of the memoryless property of Markovian queues, these queues are amenable to analysis. In fact, they are all continuous-time Markov chains with the state being the *queue-size* defined as the number in the system n and the time between state transitions is exponential. The reason that these time periods are exponential is that at any point in time, the remaining time until the next arrival, or the next service completion, is a competition between various exponential random variables.

Queueing Modeling and Performance Analyses (cont'd)

We will also discuss the **M/G/1 queue** and **queues with non FIFO service disciplines including PS and LIFO**. Note also that few results for **G/G/1** has already been mentioned.

We will also discuss **queueing networks**, and some **queueing models where certain traffic streams have priority over others**.

Performance evaluation methods will mainly focus on **analyses**, but also **simulations** will be discussed.

Applications will be considered with particular emphasis on **telecommunications applications**.

Section 4.1: How to calculate Confidence Intervals

June 3, 2020

Preliminaries

Regardless of how long we run a simulation involving random processes, we will never obtain the exact mathematical result of a steady-state measure we are interested in.

To assess the error of our simulation, we begin by running a certain number, say n, of simulation experiments and obtain n observed values, denoted $a_1, a_2, \ \ldots, \ a_n$, of the measure of interest.

Sample Mean and Sample Variance

$\bar{a}$ is the observed mean of the n observations.

$$\bar{a} = \frac{1}{n} \sum_{i=1}^{n} a_i.$$

σ_a^2 is the observed variance of the n observations.

$$\sigma_a^2 = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \bar{a})^2.$$

Confidence Interval

The confidence interval of $\bar{a}$, with confidence α, $0 \leq \alpha \leq 1$, is

$$(\bar{a} - U_r, \bar{a} + U_r),$$

where

$$U_r = \left\{t_{(1-\alpha)/2,(n-1)}\right\} \frac{\sigma_a}{\sqrt{n}}.$$

$t_{(1-\alpha)/2,(n-1)}$

$t_{(1-\alpha)/2,(n-1)}$ is the appropriate percentage point for Student's t-distribution with $n-1$ degrees of freedom.

The $t_{(1-\alpha)/2,(n-1)}$ values are available in standard tables.

Example:

We are interested in 95% confidence.

This means $\alpha = 0.95$, so $1-\alpha = 0.05$, and $(1-\alpha)/2 = 0.025$.

For $n = 6$, we use $t_{0.025,5} = 2.57$.

For $n = 11$, we use $t_{0.025,10} = 2.23$.

Microsoft (MS) Excel$^{\mathrm{TM}}$ TINV

Microsoft (MS) Excel$^{\mathrm{TM}}$ provides the function *TINV* whereby $TINV(1-\alpha, n-1)$ gives the appropriate constant based in t-distribution for confidence α and $n-1$ degrees of freedom.

Then the confidence interval of $\bar{a}$, with confidence α, $0 \leq \alpha \leq 1$, is

$$(\bar{a} - U_r, \bar{a} + U_r),$$

where

$$U_r = TINV(1-\alpha, n-1)\frac{\sigma_a}{\sqrt{n}}.$$

TINV examples

Let us now consider the above-mentioned two examples of $n = 6$ and $n = 11$.

Using MS Excel$^{\text{TM}}$, $TINV(0.05, 5) = 2.57$ and $TINV(0.05, 10) = 2.23$.

That is, if we are interested in 95% confidence and we have $n = 6$ observations, we will use $TINV(0.05, 5) = 2.57$ to obtain the confidence interval.

If we have $n = 11$ observations, we will use $TINV(0.05, 10) = 2.23$.

More values for $t_{(1-\alpha)/2,(n-1)}$, or equivalently, $TINV(1 - \alpha, n - 1)$, for 95% confidence ($\alpha = 0.95$) and various n values, are provided in the following tables.

Table 1

α	n	$t_{(1-\alpha)/2,(n-1)}$ $(= TINV(1-\alpha, n-1))$
0.95	5	2.78
0.95	6	2.57
0.95	7	2.45
0.95	8	2.36
0.95	9	2.31
0.95	10	2.26
0.95	11	2.23
0.95	12	2.20

Table 2

α	n	$t_{(1-\alpha)/2,(n-1)}$ $(= TINV(1-\alpha, n-1))$
0.95	13	2.18
0.95	14	2.16
0.95	15	2.14
0.95	16	2.13
0.95	17	2.12
0.95	18	2.11
0.95	19	2.10
0.95	20	2.09
0.95	21	2.09

Tradeoff between *n* and accuracy

Generally, the larger the number of observations (n), the smaller is the 95% confidence interval, i.e., the smaller is U_r.

As certain simulations are very time consuming, a decision needs to be made on the tradeoff between time and accuracy.

In many cases, when the simulations are not very time consuming, we can increase the number of observations until required accuracy (length of confidence interval) is achieved.

Homework 1

Choose a set of 12 different real numbers to represent outcomes of multiple measurements of the same quantity, and calculate the confidence interval.

Homework 2

Generate 10 uniform (0,1) deviates compute their average. Repeat this 11 times. These repetitions will result in 11 different estimations of $E[X]$ where X is a uniform (0,1) random variable. Use these 11 estimations to obtain the confidence interval for the estimation of $E[X]$ based on the 10 deviates. Then repeat this exercise five more times, increasing the sample size (number of deviates) using 100, 1,000, 10,000, 100,000 and 1,000,000 uniform (0,1) deviates. Observe the length of the confidence interval as you increase the sample size.

Then, repeat this homework for the case where X is a uniform (a, b) random variable for a range of values a and b.

(Delayed) Homeworks 3 and 4

We will learn in Chapter 8 how to perform a Markov chain simulation of M/M/k/k and in Chapter 19, we will learn to perform a Markov chain simulation of a mobile network.
After you learn these, use confidence intervals for the M/M/k/k and mobile cellular network simulations.

Section 4.2: Discrete Event Simulation of a G/G/1 Queue

© Moshe Zukerman

June 3, 2020

Initial Comments

Much of the time in this course will be spent on **analytical methods** for queueing performance evaluation. However, analytical approaches are limited to cases that are amenable to analysis. Although analyses are important as they provide valuable insight, many practical cases require solutions based on **computer simulations**. In addition, computer simulations provide alternative solutions that can validate various approximations that rely on analyses. It is always important to solve problems in more than one way, so the different approaches are used to validate each other. We will therefore also extensively discuss various simulation approaches.

Here, we will present an example of how to simulate a G/G/1 queue using an approach called **Discrete Event Simulation**.

Although the example presented here is for a G/G/1 queue, the principles can be easily extended to multi server and/or finite buffer queues.

Step 1: Generate Two Sequences of Inter-arrival Times and Service Times

Notice that if one generates sequences of independent random deviates for the inter-arrival times and service times, the resulted queue model will be GI/GI/1 which is a special case of G/G/1.

To generate a sequence dependent random variables, you may use the inter-arrival times of MMPP, or exponential autoregressive process mentioned in Chapter 21 in <classnotes.pdf>. Alternatively, you may think of a different way to generate sequences of dependent deviates.

Homework: generate dependent sequences for the inter-arrival and service times.

Fill in the Table

In writing a computer simulation for G/G/1, we aim to fill in a table like the following for several 100,000s or millions arrivals (rows).

arrival time	service duration	queue-size on arrival	service starts	service ends	delay
1	4	0	1	5	4
3	6	1	5	11	8
4	4	2			
12	2				
16	5				
21	1				

Guide on Filling in the Table

- The arrival times and the service durations values are readily obtained from the inter-arrival and service time sequences.
- Assuming that the previous rows are already filled in, the "queue-size on arrival" is obtained by comparing the arrival time of the current arrivals and the values in the "service ends" column of the previous rows. In particular, the queue size on arrival is equal to the number of customers that arrive before the current customer (previous rows) that their "service ends" time values are greater than the arrival time value of the current arrival.

Guide on Filling in the Table (cont'd)

- The "service starts" value is the maximum of the "arrival time" value of the current arrival and the "service end" value of the previous arrival. Also notice that if the queue size on arrival of the current arrival is equal to zero, then the service start value is equal to the "arrival time" value of the current arrival, and if the queue size on arrival of the current arrival is greater than zero, then the service start value is equal to the "service end" value of the previous arrival.
- The "service ends" value is simply the sum of the "service starts" and the "service duration" values of the current arrival.
- The "delay" value is the difference between the "service ends" and the "arrival time" values.

How to Evaluate Queueing Performance Measures from the Results of the Table

- Using the results obtained in the last column, generating a histogram, we can estimate the **delay distribution** in steady-state.
- We can also estimate the moments of the delay distribution using these results including the **mean delay**.
- Having the mean delay, by Little's formula, we can obtain the **mean queue size**.
- However, in general, the "queue-size on arrival" values for all the customers do not provide directly, on their own, the steady-state queue-size distribution and moments.
- Only for the special case of M/G/1 (including M/M/1 and M/D/1), due to PASTA, the "queue-size on arrival" values can be used directly to obtain the steady-state queue-size distribution and moments.

Two ways to Evaluate the Queue Size Distribution and Moments

1. **Use a separate independent Poisson inspector.** If the arrival process does not follow a Poisson process, a separate independent Poisson inspector is required. In such a case, we generate a Poisson process: $t_1, t_2, t_3, \ldots,$ and for each t_i, $i = 1, 2, 3, \ldots$ we can invoke the queue-size at time t_i, denoted Q_i, in a similar way to the one we obtained the "queue-size on arrival" values. The Q_i values are then used to evaluate the queue-size distribution and moments.
2. **Consider both the queue-size measurements and the time they remain at that level**. Record the total time spent in each state. If there was an event (arrival or departure) at time t_j when the G/G/1 queue entered state i and the next event (arrival or departure) at t_k when the G/G/1 queue exited state i, then the period $t_k - t_j$ is added to a counter recording the total time spent in the state i.

Homework

1. Fill in the above table by hand. Then, modify the values in the columns "arrival time" and "service duration" and again fill in the table.
2. Write a computer simulation for a P/P/1 queue (a single-server queue with Pareto inter-arrival and service time distributions) to derive estimates for the mean and distribution of the delay and of the queue-size. Perform the simulations for a wide range of parameter values. Compute confidence interval.
3. Repeat the simulations, of the previous homework, for a wide range of parameter values, for a U/U/1 queue, defined as a single-server queue with Uniform inter-arrival and service time distributions, and for an M/M/1 queue. For the M/M/1 queue, verify that your simulation results are consistent with respective analytical results which we will discuss later (but you can read ahead). For the U/U/1 queue, use the Poisson inspector approach and the "time recording" approach and verify that the results are consistent.

Homework (cont'd)

4. Discuss the accuracy of your estimations in the different cases.
5. Use the principles presented here for a G/G/1 queue simulation to write a computer simulation for a G/G/k/k queue. In particular, focus on the cases of an M/M/k/k queue and a U/U/k/k queue, defined as a k-server system without additional waiting room where the inter-arrival and service times are uniformly distributed, and compute results for the blocking probability for these two cases. For a meaningful comparison use a wide range of parameter values.

Chapter 5: Deterministic Queueing Models

© Moshe Zukerman

June 4, 2020

Preliminaries

- We consider here the simple case where inter-arrival and service times are deterministic.
- To avoid ambiguity, we assume that if an arrival and a departure occur at the same time, the departure occurs first. Such an assumption is not required for Markovian queues where the probability of two events occurring at the same time is zero, but it is needed for deterministic queues.
- Unlike many of the Markovian queues that we study in this book, for deterministic queues, steady-state queue size distribution does not exist because the queue size deterministically fluctuate according to a certain pattern.
- Therefore, for deterministic queues, we will use the notation $P(Q = n)$, for the proportion of time that there are n customers in the queue, or equivalently, $P(Q = n)$ is the probability of having n in the queue at a randomly (uniformly) chosen point in time.

Preliminaries (cont'd)

- Accordingly, the mean queue size $E[Q]$ will be defined by

$$E[Q] = \sum_{n=0}^{\infty} nP(Q = n).$$

- We will use the term blocking probability P_b to represent the proportion of packets that are blocked.
- To derive performance measures such as mean queue size, blocking probability and utilization, in such deterministic queues, we follow the queue-size process, for a certain transient period, until we discover a pattern (cycle) that repeats itself. Then, we focus on a single cycle and obtain the desired measures of that cycle.

Unstable D/D/1

- If $\lambda > \mu$, the D/D/1 queue is unstable.
- In this case, the queue size constantly grows and approaches infinity as $t \to \infty$.
- Also in this case, since there are always packets in the queue waiting for service, the server is always busy, thus the utilization is equal to one.

Stable D/D/1

- Let us consider now a stable D/D/1 queue, assuming $\lambda \leq \mu$.
- Notice that for D/D/1, given our above assumption that if an arrival and a departure occur at the same time, the departure occurs first, the case $\lambda = \mu$ will also be stable.
- Assume that the first arrival occurs at time $t = 0$.
- The service time of this arrival will terminate at

$$t = \frac{1}{\mu}.$$

Stable D/D/1 (cont'd)

- Then, another arrival will occur at time

  $$t = \frac{1}{\lambda}$$

  which will be completely served at time

  $$t = \frac{1}{\lambda} + \frac{1}{\mu}.$$

  etc. etc.

- This gives rise to a deterministic cyclic process where the queue-size takes two values: 0 and 1 with transitions from 0 to 1 in points of time $n(1/\lambda)$, $n = 0, 1, 2,\ \ldots$, and transitions from 1 to 0 in points of time $n(1/\lambda) + 1/\mu$, $n = 0, 1, 2,\ \ldots$.

Stable D/D/1 (cont'd)

- Each cycle is of time-period $1/\lambda$ during which there is a customer to be served for a time-period of $1/\mu$, and there is no customer for a time-period of $1/\lambda - 1/\mu$. Therefore, the utilization is given by

  $$\hat{U} = \frac{\frac{1}{\mu}}{\frac{1}{\lambda}} = \frac{\lambda}{\mu}$$

  which is consistent with what we know about the utilization of G/G/1.

Stable D/D/1 (cont'd)

$$E[Q] = \hat{U}$$

Explanation 1: As each one of the customers that enters the system is served before the next one arrives, $E[Q]$ of D/D/1, denoted $E[Q]_{D/D/1}$, is equal to $E[N_s]$, and therefore, it is also equal to $\hat{U}$.

Explanation 2: Q alternates between the values 1 and 0, spending a time-period of $1/\mu$ at state 1, then a time-period of $1/\lambda - 1/\mu$ at state 0, then again $1/\mu$ time at state 1, etc. If we pick a random point in time, the probability that there is one in the queue is given by $P(Q = 1) = (1/\mu)/(1/\lambda)$, and the probability that there are no customers in the queue is given by $P(Q = 0) = 1 - (1/\mu)/(1/\lambda)$.
Therefore, $E[Q] = 0P(Q = 0) + 1P(Q = 1) = (1/\mu)/(1/\lambda) = \hat{U}$.

Stable D/D/1 (cont'd)

$$E[Q]_{\mathrm{G/G/1}} \geq E[Q]_{\mathrm{D/D/1}}$$

The subscript G/G/1 refers to any stable G/G/1 queue, with λ being the arrival rate and μ the service rate.
The subscript D/D/1 refers to a stable D/D/1 queue, with λ being the arrival rate and μ the service rate.
This can be shown using Little's formula $E[Q] = \lambda E[D]$ as follows.

$$E[Q]_{\mathrm{G/G/1}} = \lambda E[D]_{\mathrm{G/G/1}} = \lambda \left(\frac{1}{\mu} + E[W_Q]_{\mathrm{G/G/1}} \right) \geq \lambda \frac{1}{\mu} = E[Q]_{\mathrm{D/D/1}}.$$

D/D/k

If $\lambda > k\mu$, the D/D/k queue is unstable.

In this case, the queue size constantly increases and approaches infinity as $t \to \infty$, and since there are always more than k packets in the queueing system, all k servers are constantly busy, thus the utilization is equal to one.

If $\lambda \leq k\mu$, the queue is stable. Notice again that given our assumption that if an arrival and a departure occur at the same time, the departure occurs first, in the case $\lambda = k\mu$, the D/D/k queue is stable.

Stable D/D/k

If $\lambda \leq k\mu$, there must exist an integer $\hat{n}$, $1 \leq \hat{n} \leq k$ such that

$$(\hat{n} - 1)\mu < \lambda \leq \hat{n}\mu,$$

or equivalently

$$\frac{\hat{n} - 1}{\lambda} < \frac{1}{\mu} \leq \frac{\hat{n}}{\lambda}.$$

$\hat{n}$ is given by

$$\hat{n} = \left\lceil \frac{\lambda}{\mu} \right\rceil .$$

$\lceil x \rceil$ is called the **ceiling** of x; it is defined as the smallest integer greater or equal to x.

Stable D/D/k (cont'd)

To prove the latter, notice that

$$\frac{\hat{n}}{\lambda} = \frac{\left\lceil \frac{\lambda}{\mu} \right\rceil}{\lambda} \geq \frac{\frac{\lambda}{\mu}}{\lambda} = \frac{1}{\mu}.$$

Also,

$$\frac{\hat{n} - 1}{\lambda} = \frac{\left\lceil \frac{\lambda}{\mu} \right\rceil - 1}{\lambda} < \frac{\frac{\lambda}{\mu}}{\lambda} = \frac{1}{\mu}.$$

In addition, the integer $\hat{n}$ cannot be larger than k because it will violate the inequality

$$\frac{\hat{n} - 1}{\lambda} < \frac{1}{\mu}. \quad \square$$

Stable D/D/k (cont'd)

The above inequalities imply that if the first arrival arrives at $t = 0$, there will be additional $\hat{n} - 1$ arrivals before the first customer leaves the system.

Therefore, the queue-size increases incrementally taking the value j at time $t = (j - 1)/\lambda$, $j = 1, 2, 3, \ldots, \hat{n}$.

When the queue reaches $\hat{n}$ for the first time, which happens at time $(\hat{n} - 1)/\lambda$, **the cyclic behavior starts**.

Then, at time $t = 1/\mu$ the queue-size reduces to $\hat{n} - 1$ when the first customer completes its service.

Stable D/D/k (cont'd)

Next, at time $t = \hat{n}/\lambda$, the queue-size increases to $\hat{n}$ and decreases to $\hat{n} - 1$ at time $t = 1/\lambda + 1/\mu$ when the second customer completes its service.

This cyclic behavior continuous forever whereby the queue-size increases from $\hat{n} - 1$ to $\hat{n}$ at time points $t = (\hat{n} + i)/\lambda$, and decreases from $\hat{n}$ to $\hat{n} - 1$ at time points $t = i/\lambda + 1/\mu$, for $i = 0,\ 1,\ 2,\ \ldots\ .$
The cycle length is $1/\lambda$ during which the queue-size process is at state $\hat{n}$, $1/\mu - (\hat{n} - 1)/\lambda$ of the cycle time, and it is at state $\hat{n} - 1$, $\hat{n}/\lambda - 1/\mu$ of the cycle time.

Thus,

$$P(Q = \hat{n}) = \frac{\lambda}{\mu} - (\hat{n} - 1)$$

and

$$P(Q = \hat{n} - 1) = \hat{n} - \frac{\lambda}{\mu}.$$

Stable D/D/k (cont'd)

The mean queue-size $E[Q]$, can be obtained by

$$E[Q] = (\hat{n} - 1)P(Q = \hat{n} - 1) + \hat{n}P(Q = \hat{n})$$

which after some algebra gives

$$E[Q] = \frac{\lambda}{\mu}.$$

Homework: Perform the algebraic operations that lead to the last $E[Q]$ result.

This result is consistent with Little's formula.

Stable D/D/k (cont'd)

As customers are served as soon as they arrive, the time each of them spends in the system is the service time $1/\mu$ - multiplying it by λ, gives by Little's formula the mean queue size.

Since $E[Q]$ in D/D/k gives the number of busy servers, the utilization (which in this case is the average utilization per server) is given by

$$\hat{U} = \frac{\lambda}{k\mu}.$$

D/D/∞

The above equations derived for $E[Q]$ and $\hat{U}$ of D/D/k apply also to D/D/∞ for finite λ and μ.

Notice that as $k \to \infty$, we obtain $\hat{U} = 0$.

This is consistent with the explanation that because in D/D/∞ there are infinite number of servers and the mean number of busy servers is finite, the average utilization per server must be equal to zero.

D/D/k/k

In D/D/k/k there is no waiting room beyond those available at the servers.

Recall that to avoid ambiguity, we assume that if an arrival and a departure occur at the same time, the departure occurs first.

Therefore, if $\lambda \leq k\mu$, then we have the same queue behavior as in D/D/k as no losses will occur.

The interesting case is the one where $\lambda > k\mu$ and this is the case we focus on.

D/D/k/k – The case: $1/\mu > k/\lambda$

Having $\lambda > k\mu$, or $1/\mu > k/\lambda$, implies that

$$\tilde{n} = \left\lceil \frac{\lambda}{\mu} \right\rceil - k$$

satisfies

$$\frac{k + \tilde{n} - 1}{\lambda} < \frac{1}{\mu} \leq \frac{k + \tilde{n}}{\lambda}.$$

Homework: Prove the last statement.
Guide: Notice that

$$\frac{k + \tilde{n}}{\lambda} = \frac{\left\lceil \frac{\lambda}{\mu} \right\rceil}{\lambda} \geq \frac{\frac{\lambda}{\mu}}{\lambda} = \frac{1}{\mu}.$$

Also,

$$\frac{k + \tilde{n} - 1}{\lambda} = \frac{\left\lceil \frac{\lambda}{\mu} \right\rceil - 1}{\lambda} < \frac{\frac{\lambda}{\mu}}{\lambda} = \frac{1}{\mu}.$$

The D/D/k/k Process and Its Cycles

Again, consider an empty system with the first arrival occurring at time $t = 0$.

There will be additional $k - 1$ arrivals before all the servers are busy.

Notice that because $1/\mu > k/\lambda$, no service completion occurs before the system is completely full.

Then $\tilde{n}$ additional arrivals will be blocked before the first customer completes its service at time $t = 1/\mu$ at which time the queue-size decreases from k to $k - 1$.

The D/D/k/k Process and Its Cycles

Next, at time $t = (k + \tilde{n})/\lambda$, the queue-size increases to k and reduces to $k - 1$ at time $t = 1/\lambda + 1/\mu$ when the second customer completes its service.

This behavior of the queue-size alternating between the states k and $k - 1$ continues until all the first k customers complete their service which happens at time $t = (k - 1)/\lambda + 1/\mu$ when the kth customer completes its service, reducing the queue-size from k to $k - 1$.

Next, an arrival at time $t = (2k + \tilde{n} - 1)/\lambda$ increased the queue-size from $k - 1$ to k. Notice that the point in time $t = (2k + \tilde{n} - 1)/\lambda$ is an end-point of a cycle that started at $t = (k - 1)/\lambda$.

The D/D/k/k Process and Its Cycles (cont'd)

This cycles comprises two parts: the first is a period of time where the queue-size stays constant at k and all the arrivals are blocked, and the second is a period of time during which no losses occur and the queue-size alternates between k and $k-1$.

Then a new cycle of duration $(k+\tilde{n})/\lambda$ starts and this new cycle ends at $t=(3k+2\tilde{n}-1)/\lambda$.

In general, for each $j=1,\ 2,\ 3,\ \ldots,$ a cycle of duration $(k+\tilde{n})/\lambda$ starts at $t=(jk+(j-1)\tilde{n}-1)/\lambda$ and ends at $t=((j+1)k+j\tilde{n}-1)/\lambda$.

Blocking Probability for D/D/k/k

In every cycle, there are $k + \tilde{n}$ arrivals out of which $\tilde{n}$ are blocked. The blocking probability is therefore

$$P_b = \frac{\tilde{n}}{k + \tilde{n}}.$$

Since

$$k + \tilde{n} = \left\lceil \frac{\lambda}{\mu} \right\rceil,$$

the blocking probability is given by

$$P_b = \frac{\left\lceil \frac{\lambda}{\mu} \right\rceil - k}{\left\lceil \frac{\lambda}{\mu} \right\rceil}.$$

Mean Queue Size for D/D/k/k

Let $A = \lambda/\mu$, the mean-queue size is obtained using Little's formula to be given by

$$E[Q] = \frac{\lambda}{\mu}(1 - P_b) = \frac{kA}{\lceil A \rceil}.$$

Utilization for D/D/k/k

As in D/D/k, since every customer that enters a D/D/k/k system does not wait in a queue, but immediately enters service, the utilization is given by

$$\hat{U} = \frac{E[Q]}{k} = \frac{A}{\lceil A \rceil}.$$

Summary of Results

The following table summarizes the results on D/D/1, D/D/k and D/D/k/k. Note that we do not consider the cases $\lambda = k\mu$, $k = 1, 2, 3, \ldots$, for which the corresponding results for the cases $\lambda < k\mu$, $k = 1, 2, 3, \ldots$, are applicable assuming that if a departure and an arrival occur at the same time, the departure occurs before the arrival.

Model	Condition	$E[Q]$	$\hat{U}$
D/D/1	$\lambda < \mu$	λ/μ	λ/μ
D/D/1	$\lambda > \mu$	∞	1
D/D/k	$\lambda < k\mu$	$A = \lambda/\mu$	A/k
D/D/k	$\lambda > k\mu$	∞	1
D/D/k/k	$\lambda < k\mu$	A	A/k
D/D/k/k	$\lambda > k\mu$	$kA/\lceil A\rceil$	$A/\lceil A\rceil$

Chapter 6: The M/M/1 Queue

© Moshe Zukerman

October 24, 2019

M/M/1 Preliminaries

- M/M/1 is a single server queue (SSQ)
- the arrival process follows a Poisson process with parameter λ
- service times are assumed to be IID and exponentially distributed with parameter μ, and are independent of the arrival process.

As M/M/1 is a special case of G/G/1, all the results that are applicable to G/G/1 are also applicable to M/M/1.

For example,

- $\hat{U} = \rho = \lambda/\mu$,
- $\pi_0 = 1 - \lambda/\mu = 1 - \rho$
- Little's formula.

M/M/1 is the simplest Markovian queue; it has only a single server and an infinite buffer.

The M/M/1 Queue-size Process as a Continuous-time Markov Chain on the States: 0, 1, 2, ...

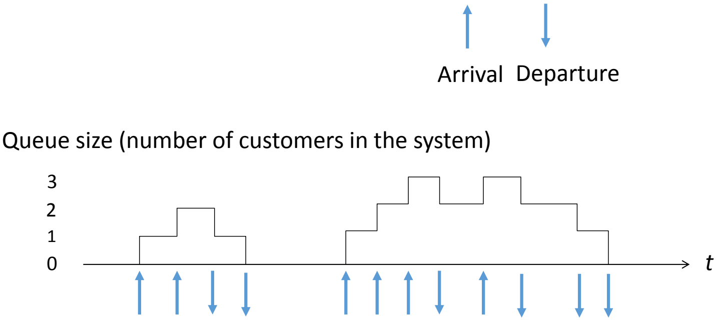


Figure 1: Queue size evolution of M/M/1.

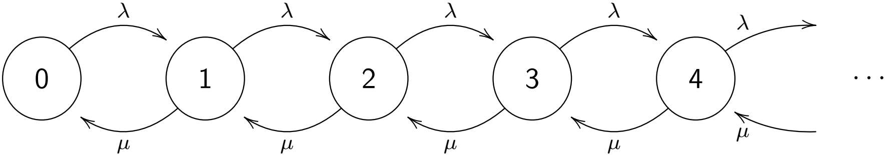
The M/M/1 Queue-size Process as a Continuous-time Markov Chain on the States: 0, 1, 2, . . . (cont'd).

If an M/M/1 queue-size process is at state 0, it will stay in state 0 for a period of time that is exponentially distributed with parameter λ then it moves to state 1.

If an M/M/1 queue-size process is at state n, for $n \geq 1$, it will also stay there an exponentially distributed amount of time, but this time, there is a competition between two exponential random variables: one is the time until the next arrival - with parameter λ, and the other is the time until the next departure - with parameter μ.
The minimum of the two is also exponential, but with parameter $\lambda + \mu$, and this minimum is the time the process stays in state n, for $n \geq 1$.
We also know that after spending an exponential amount of time with parameter $\lambda + \mu$, the process will move to state $n + 1$ with probability $\lambda/(\lambda + \mu)$ and to state $n - 1$ with probability $\mu/(\lambda + \mu)$.

State Transition Diagram, Steady State Equations, Reversibility, and the Output Process

The following diagram is the **state transition diagram** of M/M/1.



The following are the **global balance** steady-state equations for M/M/1.

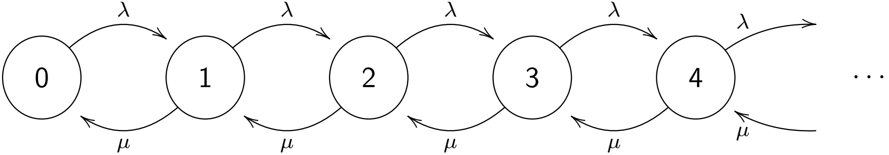
$\pi_0\lambda = \pi_1\mu$

$\pi_1(\lambda + \mu) = \pi_2\mu + \pi_0\lambda$

$$\text{and in general, for } i \geq 1: \quad \pi_i(\lambda + \mu) = \pi_{i+1}\mu + \pi_{i-1}\lambda.$$

$$\text{The } \textbf{normalizing equation} \text{ is } \sum_{j=0}^{\infty} \pi_j = 1.$$

State Transition Diagram, Steady State Equations, Reversibility, and the Output Process (cont'd)



The following are the **detailed balance** steady-state equations for M/M/1.

$$\pi_i \lambda = \pi_{i+1}\mu, \text{ for } i = 0,\ 1,\ 2,\ \ldots\ .$$

Homework: Show how to obtain the M/M/1 detailed balance steady state equations from the M/M/1 global balance steady state equations.

The equivalence of global balance and detailed balance in M/M/1 lead to:

- Reversibility
- The output process is Poisson

(M/M/1 is a birth-and-death process which also implies reversibility.)

Derivation of the M/M/1 Steady State Probabilities

Using $\rho = \lambda/\mu$, so we obtain,
$\pi_1 = \rho\pi_0$
$\pi_2 = \rho\pi_1 = \rho^2\pi_0$
$\pi_3 = \rho\pi_2 = \rho^3\pi_0$
and in general:

$$\pi_i = \rho^i\pi_0 \text{ for } i = 0,\ 1,\ 2,\ \ldots\ .$$

As M/M/1 is a special case of G/G/1, we have $\pi_0 = 1 - \rho$, so

$$\pi_i = \rho^i(1-\rho) \text{ for } i = 0,\ 1,\ 2,\ \ldots\ .$$

Homework: Show that $\pi_0 = 1 - \rho$ by summing up the π_is and equating the sum to 1.

Four Performance Measures: $E[Q]$, $E[N_Q]$, $E[D]$, $E[W_Q]$

Random variable Q is the queue-size in steady-state. Its **mean** is

$$E[Q] = \sum_{i=0}^{\infty} i\pi_i = \frac{\rho}{1-\rho}.$$

Homework: Derive the last result for the mean queue size $E[Q]$.

We know: $E[Q] = E[N_Q] + \rho$, where $E[N_Q]$ is the mean number of customers in the queue (excluding the one in service). This gives

$$E[N_Q] = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}.$$

By Little's formula, the **mean delay** is obtained by

$$E[D] = \frac{E[Q]}{\lambda} = \frac{\rho}{(1-\rho)\lambda} = \frac{1}{\mu-\lambda}.$$

Mean waiting time in the queue (excluding the time in service) is

$$E[W_Q] = E[D] - 1/\mu = \frac{1}{\mu-\lambda} - \frac{1}{\mu} = \frac{\rho}{\mu-\lambda}.$$

Mean Delay of Delayed Customers

$\hat{D}$ = The delay of a delayed customer including the service time.
$\hat{W}_Q$ = The delay of a delayed customer in the queue excluding the service time.

To obtain $E[\hat{W}_Q]$, we use Little's formula where we consider the queue (without the server) as the system and the arrival rate of the delayed customers which is $\lambda\rho$. Thus

$$E[\hat{W}_Q] = \frac{E[N_Q]}{\lambda\rho} = \frac{1}{\mu - \lambda},$$

and

$$E[\hat{D}] = E[\hat{W}_Q] + \frac{1}{\mu} = \frac{1}{\mu - \lambda} + \frac{1}{\mu}.$$

Mean Delay of Delayed Customers (cont'd)

Now, let us check the latter using the Law of Iterated Expectation as follows:

$$\begin{aligned} E[D] &= (1-\rho)[\text{Mean delay of a non-delayed customer}] \\ &+ \quad \rho[\text{Mean delay of a delayed customer}] \\ &= (1-\rho)\frac{1}{\mu} + \rho\left(\frac{1}{\mu-\lambda}+\frac{1}{\mu}\right) = \frac{1}{\mu-\lambda}. \end{aligned}$$

and we observe that consistency is achieved. Notice that this consistency check is an alternative way to obtain $E[\hat{D}]$.

Homework: Derive $E[\hat{D}]$ using the Law of Iterated Expectation.

Delay Distribution

M/M/1 delay is exponentially distributed because it comprises a geometric number of phases each of which is exponentially distributed. The number of phases P_h is equal to $Q+1$ (Q in the queue at the time of arrival plus own service time). We know that

$$P(Q=i)=\pi_i=\rho^i(1-\rho).$$

Since $P_h=Q+1$, we have

$$P(P_h=n)=P(Q+1=n)=P(Q=n-1)=\rho^{n-1}(1-\rho).$$

Therefore, P_h is geometrically distributed with mean

$$E[P_h]=\frac{1}{1-\rho}.$$

Delay Distribution (cont'd)

The mean delay equals the mean number of phases times the mean service time $1/\mu$. Thus,

$$E[D] = \frac{E[P_h]}{\mu} = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu 1 - \lambda}.$$

This is consistent with the previous result obtained by Little's formula. Substituting $1/E[D] = \mu - \lambda$ as the parameter of exponential density, the density of the delay distribution is obtained to be given by

$$\delta_D(x) = \begin{cases} (\mu - \lambda)e^{(\lambda-\mu)x} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$Var[D] = \frac{1}{(\mu-\lambda)^2} = \frac{1}{\mu^2(1-\rho)^2}.$$

Mean Busy Period

The *busy period* is the time elapsed from the moment a customer arrives at an empty system until the first time the system is empty again.

The end of a busy period is the beginning of the so called *idle period* - a period during which the system is empty.

Let T_B and T_I be the busy and the idle periods, respectively.

$$E[T_I] = \frac{1}{\lambda}$$

$$\frac{E[T_B]}{E[T_B] + E[T_I]} = \rho.$$

$$\frac{E[T_B]}{E[T_B] + \frac{1}{\lambda}} = \rho.$$

$$E[T_B] = \frac{1}{\mu - \lambda}. \quad \text{(Same as mean delay!!!)}$$

Mean Busy Period (cont'd)

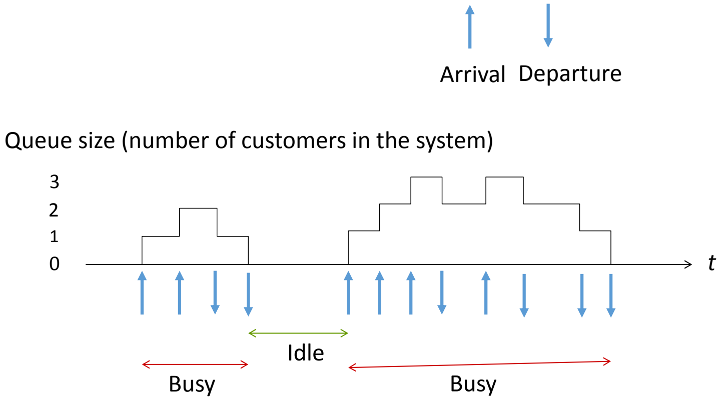


Figure 2: Busy and idle periods in M/M/1.

Mean Busy Period (cont'd)

Explanation of this potential counter-intuitive result

We have noticed that for the M/M/1 queue, the mean busy period is equal to the mean delay of a single customer. This may seem counter-intuitive.

However, we can realize that there are many busy periods each of which is made of a single customer service time. It is likely that for the majority of these busy periods (service times), their length is shorter than the mean delay of a customer.

Mean Busy Period (cont'd)

Explanation using a Last In First Out (LIFO) queue

Consider two queues M/M/1 FIFO and M/M/1 LIFO, both with the same parameters arrival and service rates λ and μ, respectively. They are both birth-and-death processes with the same parameters so their respective queue size processes are statistically the same. Then, by Little's formula their respective mean delays are also the same.

The mean delay of a LIFO customer is equal to the mean busy period created if the customer arrives at an empty queue, and therefore, the mean delay must be equal to the mean busy period in M/M/1 with FIFO service policy.

First Passage Time in M/M/1

As M/M/1 is a continuous-time Markov chain, the busy period is also the first passage time from state 1 to state 0, and the mean idle period is the first passage time from state 0 to state 1.

Homework:
Derive an expression for the mean first passage time for M/M/1 from state n to state 0 and from state 0 to state n, for $n \geq 3$.

Dimensioning μ Based on Meeting Required Mean Delay

A problem that often arises in practice is associated with resource allocation and dimensioning. If the demand is given, what is the minimal (least cost) service rate (or link capacity) such that a prespecified delay requirement is met. A simple version of this problem is the following.

Given a required mean delay and the arrival rate, assume that M/M/1 conditions hold, and we are asked to find the smallest value of μ, called μ^*, such that required mean delay ($E[D]_R$) is met. To find μ^*, we solve

$$E[D]_R = \frac{1}{\mu^* - \lambda}$$

$$\mu^* = \frac{1 + \lambda E[D]_R}{E[D]_R}.$$

This will give the lowest (cheapest) service rate (μ^*) that meets the delay requirement ($E[D]_R$). If $\mu < \mu^* \Rightarrow$ too much delay. If $\mu > \mu^* \Rightarrow$ too expensive. Therefore, μ^* is the optimal choice for service rate.

Dimensioning λ Based on Meeting Required Mean Delay

A second dimensioning problem is the following. Given service rate μ and the required mean delay $(E[D]_R)$, find the highest possible arrival rate λ^* that meets the delay requirement $(E[D]_R)$.

Now we need first to check if a feasible solution exists. That is, check if

$$\frac{1}{\mu} \leq E[D]_R. \qquad \textbf{This is the feasibility condition}.$$

Notice that the delay (D) includes the service time, so if the required mean delay is lower than the mean service time, no feasible solution exists, we cannot satisfy this requirement.

On the other hand, if the feasibility condition holds, then we solve

$$E[D]_R = \frac{1}{\mu - \lambda^*}$$

and we obtain

$$\lambda^* = \frac{\mu E[D]_R - 1}{E[D]_R}.$$

Homework

Consider again the problem of dimensioning μ based on meeting required mean delay.

Show that μ^* satisfies the condition

$$\frac{1}{\mu^*} \leq E[D]_R.$$

Effect of Rising Bit-rate on Link Efficiency and QoS

$$\text{From } \ E[D] = \frac{1}{\mu - \lambda}, \ \text{ we obtain } \ (\mu - \lambda)E[D] = 1.$$

Dividing both sides by $\mu E[D]$ and isolating ρ, we obtain

$$\rho = 1 - \frac{1}{\mu E[D]}.$$

- For constant $E[D]$, ρ approaches one as μ (and also λ) approaches infinity.
- For a fixed ρ and arbitrarily large μ, the delay is arbitrarily small.
- Potential for improving Quality of Service (QoS) and efficiency (lower cost) with increased link speed.

Multiplexing

Multiplexing: a method by which multiple traffic streams from possibly different sources share a common transmission resource.

Multiplexing improves efficiency and it is related to efficiency gain from rising traffic and link speed discussed earlier.

Multiplexing N traffic streams each of Poisson arrivals with rate λ implies Poisson arrivals with rate $N\lambda$. If the service rate increases to $N\mu$ and service times are exponentially distributed. Then, the mean delay is

$$E[D] = \frac{1}{N(\mu - \lambda)},$$

which is N times smaller then under the M/M/1 case with arrival rate λ and service rate μ.

Multiplexing (cont'd)

We know that in M/M/1, the delay statistics (mean and distribution) are a function of what we call the *spare capacity* (or *mean net input*) which is the difference between the service rate and the arrival rate.

If the arrival rate increases from λ to $N\lambda$, and we aim to find the service rate μ^* such that the delay-related QoS measure is just met, we will need to make sure that the spare capacity is maintained, that is

$$\mu - \lambda = \mu^* - N\lambda, \quad \text{or} \quad \mu^* = \mu + (N-1)\lambda.$$

$$\text{Multiplexing Gain } = M_{mg} = \frac{N\mu - \mu^*}{N\mu} = \frac{N-1}{N}(1-\rho).$$

The last equality is obtained by substituting μ^*.

- The multiplexing gain is positive for all $N > 1$.
- The multiplexing gain increases with N.
- The multiplexing gain is bounded above by $1-\rho$. If $\rho \approx 1$, $M_{mg} \approx 0$.
- In the limiting condition as $N \to \infty$, the multiplexing gain approaches its bound $1-\rho$.

Time Division Multiple Access (TDMA) versus Full MUltipleXing (FMUX)

Consider N users each transmitting packets at an average rate of R_u [bits/second].

The average packet size is S_u [bits] for any of the users.

Each user generates packets at rate $\hat{\lambda}$ [packets/second].

Thus, $\hat{\lambda} = R_u/S_u$.

TDMA

Each of the users obtains a service rate of B_u [bits/sec].

Packet sizes are assumed to be exponentially distributed with mean S_u [bits], so the service rate in packets/second denoted $\hat{\mu}$ is given by $\hat{\mu} = B_u/S_u$.

The packet service time is therefore exponentially distributed with parameter $\hat{\mu}$. Letting $\hat{\rho} = \hat{\lambda}/\hat{\mu}$, the mean queue size under TDMA, is given by

$$E[Q_{TDMA}] = \frac{\hat{\rho}}{1-\hat{\rho}},$$

and the mean delay is

$$E[D_{TDMA}] = \frac{1}{\hat{\mu}-\hat{\lambda}}.$$

FMUX

For FMUX, the total arrival rate is $N\hat{\lambda}$ and the service rate is $N\hat{\mu}$, so in this case, the ratio between the arrival and service rate remains the same, so the mean queue size that only depends on this ratio remains the same

$$E[Q_{FMUX}] = \frac{\hat{\rho}}{1-\hat{\rho}} = E[Q_{TDMA}].$$

However, we can observe an N-fold reduction in the mean delay:

$$E[D_{FMUX}] = \frac{1}{N\hat{\mu} - N\hat{\lambda}} = \frac{E[D_{TDMA}]}{N}.$$

A telecommunication provider aims to meet packet delay requirement of its N customers. Assume that TDMA provides satisfactory packet delay. Assume that the M/M/1 assumptions hold. Then the provider does not need a total capacity of $N\hat{\mu}$ for the FMUX alternative. It is sufficient to allocate $\hat{\mu} + (N-1)\hat{\lambda}$.

Dimensioning Based on Delay Distribution

Previously we considered dimensioning based on average delay. Now we aim for a dimensioning meeting a requirement based on percentile of the delay distribution; e.g., to require that no more than 1% of the packets will experience over 100 millisecond delay.

In the context of the M/M/1 model, we define two dimensioning problems.

First problem

For a given λ, $t \geq 0$ and α, find minimal μ value, denoted μ^*, such that

$$P(D > t) = e^{-(\mu^* - \lambda)t} \leq \alpha.$$

The solution is:

$$\mu^* = \lambda - \frac{\ln(\alpha)}{t}.$$

Dimensioning Based on Delay Distribution (cont'd)

Second problem

For a given μ, $t \geq 0$ and α, find maximal λ value, denoted λ^*, such that

$$P(D > t) = e^{-(\mu - \lambda^*)t} \leq \alpha.$$

Again, we first must make sure that this problem has a feasible solution, because the delay includes the service time and can never be less than the service time. That is, for certain parameter values, even if the arrival rate is very low, the delay requirements cannot be met, simply because the service time requirements exceeds the total delay requirements.

To find the feasible range set $\lambda^* = 0$, and obtain

$$\mu > \frac{-\ln(\alpha)}{t}.$$

Dimensioning Based on Delay Distribution (cont'd)

If a solution is feasible, we solve the second optimization problem by solving for λ^* the equation

$$P(D > t) = e^{-(\mu-\lambda^*)t} = \alpha,$$

and we obtain

$$\lambda^* = \frac{\ln(\alpha)}{t} + \mu.$$

A Markov-chain Simulation of M/M/1

If we are not interested in performance measures that are associated with times (such as delay distribution). If our aim is to evaluate queue size statistics or blocking probability, we can avoid tracking the time and use a *Markov chain simulation*.

We collect the relevant information about the process at PASTA time-points without even knowing what are the times at these points.

We simulate the evolution of the state of the process based on the transition probability matrix and collect information on the values of interest at selective PASTA points without being concerned about the time.

A Markov-chain Simulation of M/M/1 (cont'd)

Simulation to evaluate the mean queue size of M/M/1

Variables and input parameters: Q = queue size; $\hat{E}(Q)$ = estimation for the mean queue size; N = number of Q-measurements taken so far which is also equal to the number of arrivals so far; $MAXN$ = maximal number of Q-measurements taken; μ = service rate; λ = arrival rate.

Define function: $I(Q) = 1$ if $Q > 0$; $I(Q) = 0$ if $Q = 0$.
Define function: $R(01)$ = a uniform $U(0,1)$ random deviate. A new value for R(01) is generated every time it is called.
Initialization: $Q = 0$; $\hat{E}[Q] = 0$; $N = 0$.

```
1. If $R(01) \leq \lambda/(\lambda + I(Q)\mu)$, then $N = N + 1$,
$\hat{E}(Q) = [(N-1)\hat{E}(Q) + Q]/N$, and $Q = Q + 1$;
else, $Q = Q - 1$.
2. If $N < MAXN$ go to 1; else, print $\hat{E}(Q)$.
```

Only two IF statements!!

A Markov-chain Simulation of M/M/1 (cont'd)

Comment: The operation $Q = Q + 1$ is performed after the Q measurement is taken. This is done because we are interested in Q values seen by arrivals just before they arrive. If we include the arrivals after they arrive we violate the PASTA principle.

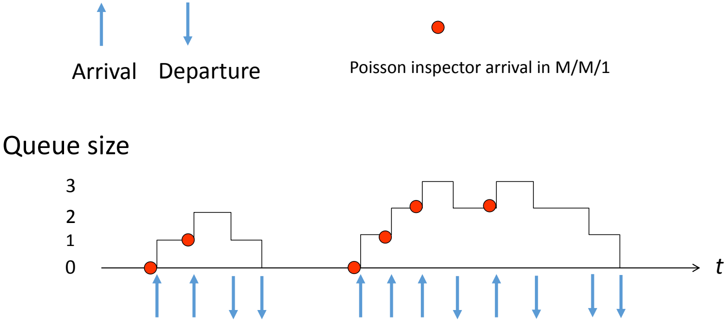


Figure 3: Poisson inspector in M/M/1.

A Markov-chain Simulation of M/M/1 (cont'd)

More Comments:

If the condition $R(01) \leq \lambda/(\lambda + I(Q)\mu)$ holds we have an arrival. Otherwise, we have a departure. This condition is true with probability $\lambda/(\lambda + I(Q)\mu)$. If $Q = 0$ then $I(Q) = 0$ in which case the next event is an arrival with probability 1. If the system is empty no departure can occur, so the next event must be an arrival.

If $Q > 0$, the next event is an arrival with probability $\lambda/(\lambda + \mu)$ and a departure with probability $\mu/(\lambda + \mu)$. We have here a competition between two exponential random variables: one (arrival) with parameter λ and the other (departure) with parameter μ.

A Markov-chain Simulation of M/M/1 (cont'd)

Departure
In a case of a departure, all we do is decrementing the queue size; namely, $Q = Q - 1$. We do not record the queue size at these points because according to PASTA arrivals see time-averages. (Notice that due to reversibility, if we measure the queue size immediately **after** departure points we will also see time-averages.)

Homework: Simulate an M/M/1 queue using a Markov-chain simulation to evaluate the mean queue-size for the cases of Section 4.2 in classnotes.pdf. Compare the results with the results obtain analytically and with those obtained using the G/G/1 simulation principles. In your comparison consider accuracy (closeness to the analytical results) the length of the confidence intervals and running times.

Chapter 7: M/M/∞

© Moshe Zukerman

August 3, 2019

M/M/∞ Preliminaries

In M/M/∞,

- there are infinite number of servers
- the arrival process follows a Poisson process with parameter λ
- service times are assumed to be IID and exponentially distributed with parameter μ, and are independent of the arrival process.
- Because the number of servers is infinite, the buffer capacity is unlimited and arrivals are never blocked.
- Offered traffic is equal to the carried traffic

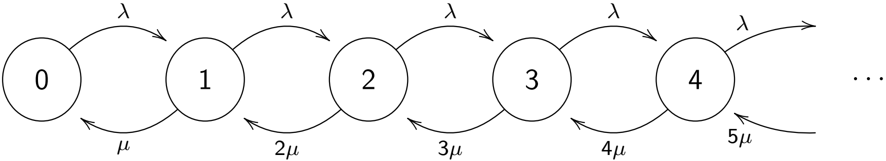
Recall: The offered traffic is in units of erlangs, and it is given by

$$A = \frac{\lambda}{\mu}.$$

$$\text{Also, by Little's formula, } E[Q] = \frac{\lambda}{\mu} = A.$$

State Transition Diagram

The state transition diagram of M/M/∞ is similar to that of M/M/1 except that the rate downwards from state n ($n = 1, 2, 3, \ldots$) is $n\mu$ rather than μ reflecting the fact at state n there are n servers serving the n customers. The state transition diagram of M/M/∞ is:



Steady-State Equations

The queue evolution of M/M/∞ follows a continuous-time Markov chain and a birth-and-death process – implying reversibility.

The steady-state probabilities π_i (for $i = 0, 1, 2, \ldots$) of having i customers in the system satisfy the following detailed balance equations:

$\pi_0 \lambda = \pi_1 \mu$
$\pi_1 \lambda = \pi_2 2\mu$
$\ldots$
and in general:

$$\pi_n \lambda = \pi_{n+1}(n+1)\mu, \text{ for } n = 0,\ 1,\ 2,\ \ldots\ .$$

Normalizing equation:

$$\sum_{j=0}^{\infty} \pi_j = 1.$$

Solving the Steady-State Equations

Using the A notation we write
$\pi_1 = A\pi_0$
$\pi_2 = A\pi_1/2 = A^2\pi_0/2$
$\pi_3 = A\pi_2/3 = A^3\pi_0/(3!)$
and in general:

$$\pi_n = \frac{A^n\pi_0}{n!} \text{ for } n = 0,\ 1,\ 2,\ \ldots\ .$$

Summing up and equating the sum of the π_ns to 1, we obtain

$$1 = \sum_{n=0}^{\infty} \frac{A^n\pi_0}{n!}.$$

Solving the Steady-State Equations (cont'd)

By the definition of Poisson random variable, we obtain

$$1 = \sum_{i=0}^{\infty} e^{-\lambda} \frac{\lambda^i}{i!}.$$

Thus,

$$e^{\lambda} = \sum_{i=0}^{\infty} \frac{\lambda^i}{i!}$$

which is also the well-known Maclaurin series expansion of e^{λ}. Therefore,

$$1 = \pi_0 e^{A},$$

Solving the Steady-State Equations (cont'd)

or

$$\pi_0 = e^{-A}.$$

Thus,

$$\pi_n = \frac{e^{-A}A^n}{n!} \text{ for } n = 0,\ 1,\ 2,\ \ldots\ .$$

We observe that the distribution of the number of busy channels (simultaneous calls or customers) in an M/M/∞ system is Poisson with parameter A.

Insensitivity

The above M/M/∞ results for π_i, $i = 0, 1, 2 \ \ldots$ and for the mean number of busy servers **are insensitive** to the shape of the service time (holding time) distribution.

All we need to know is the mean service time and the results are insensitive to higher moments of the service time.

This makes the model far more robust which allows us to use its analytical results for many applications where the service time is not exponential.

This insensitivity result is valid for M/G/∞, but it is not valid for many other systems including M/M/1, M/G/1, M/G/k and others.

This insensitivity property is valid also for the M/G/k/k system.

We will now provide intuitive (non-rigorous) explanations for the insensitivity property of M/G/∞.

Insensitivity (cont'd)

The insensitivity property of M/G/∞ with respect to the mean occupancy is already observed via Little's formula. A more direct explanation follows.

Consider an arbitrarily long period of time L and also consider the queue size process that represents the number of busy servers at any point in time between 0 and L. The average number of busy servers is obtained by the area under the queue size process function divided by L. This area is closely approximated by the number of arrivals during L which is λL times the mean holding (service) time of each arrival $(1/\mu)$.

Therefore, the mean number of busy servers, which is also equal to the mean number of customers in the system (queue size), is equal to $A = \lambda/\mu$ (notice that the L is canceled out here). Since all the traffic load enters the system (A) is also the carried traffic load.

Insensitivity (cont'd)

The words "closely approximated" are used here because there are some customers that arrive before L and receive service after L and there are other customers that arrive before time 0 and are still in the system after time 0. However, because we can choose L to be arbitrarily long, their effect is negligible.

Since in the above discussion, we do not use moments higher than the mean of the holding time, the mean number of busy servers (or mean queue size) is insensitive to the shape of the holding-time distribution and it is only sensitive to its mean.

Insensitivity (cont'd)

We now explain why the distribution of the number of busy servers in M/G/∞ is Poisson with parameter A which is also insensitive to the holding time distribution.

We know that the arrivals follow a Poisson process. Poisson process normally occurs in nature by having a very large number of independent sources each of which generates occasional events (arrivals) - for example, a large population of customers making phone calls. These customers are independent of each other.

In M/G/∞, each one of the arrivals generated by these customers is able to find a server and its arrival time, service time and departure time is independent of all other arrivals (calls). Therefore, the event that a customer occupies a server at an arbitrary point in time in steady-state is also independent of the event that any other customer occupies a server at that point in time.

Insensitivity (cont'd)

Therefore, the server occupancy events are also due to many sources generating occasional events.

Recall that a Binomial random variable with parameters n and p approaches Poisson with parameter λ as n approaches ∞ and p approaches zero and $np = \lambda$. (Here the Poisson parameter is A instead of λ.)

This explains the Poisson distribution of the server occupancy. From the above discussion, we know that the mean number of servers is equal to A, so we always have, in M/G/∞, in steady-state, a Poisson distributed number of servers with parameter A which is independent of the shape of the service-time distribution.

A Multi-access Model

An interesting application of the M/M/∞ system is the following multi-access problem (see Problem 3.8 in the book "Data Networks" by Bertsekas and Gallager). Consider a stream of packets that their arrival times follow a Poisson process with parameter λ. If the inter-arrival times of any pair of packets (not necessarily a consecutive pair) is less than the transmission time of the packet that arrived earlier out of the two, these two packets are said to collide. Assume that packets have independent exponentially distributed transmission times with parameter μ. What is the probability of no collision?

A Multi-access Model (Cont'ed)

Notice that a packet can collide with any one or more of the packets that arrived before it. In other words, it is possible that it may not collide with its immediate predecessor, but it may collide with a packet that arrived earlier. However, if it does not collide with its immediate successor, it will not collide with any of the packets that arrive after the immediate successor.

Therefore, the probability that an arriving packet will not collide on arrival can be obtained to be the probability of an M/M/∞ system to be empty, that is, e^{-A}. While the probability that its immediate successor will not arrive during its transmission time is $\mu/(\lambda+\mu)$. The product of the two, namely $e^{-A}\mu/(\lambda+\mu)$, is the probability of no collision.

Birth Rate Evaluation

Another application of the M/M/∞ system (or M/G/∞ system) is to the following problem. Consider a city with population 3,000,000, and assume that (1) there is no immigration in and out of the city, (2) the birth rate λ in constant (time independent), and (3) life-time expectancy μ^{-1} in the city is constant. It is also given that average life-time of people in this city is 78 years. How to compute the birth rate?

Using the M/M/∞ model (or actually the M/G/∞ as human lifetime is not exponentially distributed) with $E[Q] = 3,000,000$ and $\mu^{-1} = 78$, realizing that $E[Q] = A = \lambda/\mu$, we obtain,
$\lambda = \mu E[Q] = 3,000,000/78 = 38461$ new births per year or 105 new births per day.

Homework

Consider an M/M/∞ queue, with $\lambda = 120$ [call/s], and $\mu = 3$ [call/s]. Find the steady state probability that there are 120 calls in the system. This should be done by a computer. Use ideas presented when we discussed how to to compute probabilities from a Poisson probability distribution.

Chapter 8: M/M/k/k - The Erlang B System Including Revisions and Extensions

© Moshe Zukerman

April 12, 2024

The Exponential Random Variable (with Parameter μ) (Revision)

Probability density function

$$f(x) = \begin{cases} \mu e^{-\mu x} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Cumulative distribution function

$$F(x) = \int_0^x \mu e^{-\mu s} ds = 1 - e^{-\mu x} \qquad x \geq 0.$$

Complementary distribution function:

$$\bar{F}(x) = e^{-\mu x} \qquad x \geq 0.$$

An important application of the exponential random variable is the time until the next event.

The Memoryless Property of the Exponential Distribution (Revision)

A continuous random variable is called memoryless if for any $t \geq 0$ and $s \geq 0$,

$$P(X > s + t \mid X > t) = P(X > s).$$

A proof that the exponential random variable is memoryless:

$$\begin{aligned}
P(X > s + t \mid X > t) &= \frac{P(\{X > s + t\} \cap \{X > t\})}{P(X > t)} \\
&= \frac{P(X > s + t)}{P(X > t)} \\
&= \frac{e^{-\mu(s+t)}}{e^{-\mu t}} \\
&= e^{-\mu s} = P(X > s).
\end{aligned}$$

Distribution of The Minimum of Multiple Exponential Random Variables (Revision)

Let X_1 and X_2 be independent and exponentially distributed random variables with parameters μ_1 and μ_2, respectively.
Define: $X = \min[X_1, X_2]$.
The distribution of X is exponential with parameter $\mu_1 + \mu_2$.
Proof:
The complementary distribution of X is:

$$\begin{aligned} P(X > t) = P(\min[X_1 X_2] > t) = P(X_1 > t \cap X_2 > t) \\ = e^{-\mu_1 t} e^{-\mu_2 t} = e^{-(\mu_1+\mu_2)t}. \end{aligned}$$

In general, if $X_1, X_2, \ldots, X_k$ are k independent and exponentially distributed random variables with parameters $\mu_1, \mu_2, \ldots, \mu_k$, respectively, the distribution of $\min[X_1, X_2, \ldots, X_k]$ is exponential with parameter $\mu_1 + \mu_2 + \cdots + \mu_k$.

The Probability that One Random Variable is the Minimum of Multiple Exponential Random Variables (Revision)

Let $X_1, X_2, \ldots, X_k$ be k independent and exponentially distributed random variables with parameters $\mu_1, \mu_2, \ldots, \mu_k$. Then,

$$Prob(X_1 < \min[X_2, X_3, \ldots, X_k]) = \frac{\mu_1}{\mu_1 + \mu_2 + \ldots + \mu_k}.$$

A Proof for the case $k = 2$

$$P(X_1 < X_2) = P(X_2 > X_1) = \int_0^\infty (e^{-\mu_2 t})\mu_1 e^{-\mu_1 t} dt = \frac{\mu_1}{\mu_1 + \mu_2}.$$

The Probability that One Random Variable is the Minimum of Multiple Exponential Random Variables (Revision) (cont'd)

To see the latter note that

$$\int_0^\infty (\mu_1 + \mu_2) e^{-(\mu_1+\mu_2)t} dt = 1.$$

and

$$\int_0^\infty e^{-\mu_2 t} \mu_1 e^{-\mu_1 t} dt = \int_0^\infty \frac{\mu_1}{\mu_1 + \mu_2} (\mu_1 + \mu_2) e^{-(\mu_1+\mu_2)t} dt = \frac{\mu_1}{\mu_1 + \mu_2}.$$

Homework: Extend the proof to the general case of any $k = 1, 2, 3, \ \ldots$
Hint: Consider two random variables: X_1 and $\min[X_2, X_3, \ldots, X_k]$.

The Poisson Random Variable (Revision)

A Poisson random variable X with parameter λ has the following probability function:

$$P(X = i) = e^{-\lambda}\frac{\lambda^i}{i!} \qquad i = 0,\ 1,\ 2,\ 3,\ \ldots\ .$$

$$E[X] = Var[X] = \lambda.$$

Sum of Poisson Random Variables (Revision)

Question:
What is the probability distribution of the random variable $Y = X_1 + X_2$, where X_1 and X_2 are two independent Poisson random variables with parameters λ_1 and λ_2, respectively?

Answer:
The probability distribution of $Y = X_1 + X_2$ is Poisson with parameter $\lambda_1 + \lambda_2$.

In general:
Let $Y = X_1 + X_2 +, \ldots, X_k$, where $X_1, X_2, \ldots, X_k$ are k independent Poisson random variables with parameters $\lambda_1, \lambda_2, \ldots, \lambda_k$, respectively. The probability distribution of the random variable Y is Poisson with parameter $\lambda_1 + \lambda_2 + \ldots + \lambda_k$.

Sum of Poisson Random Variables (Revision) (cont'd)

A Proof for the case $k = 2$

$$
\begin{aligned}
P(Y=k) &= P(X_1+X_2=k) = \sum_{i=0}^{k} P(\{X_1=i\}\cap\{X_2=k-i\}) \\
&= \sum_{i=0}^{k} P_{X_1}(i)P_{X_2}(k-i) = \sum_{i=0}^{k} \frac{\lambda_1^i}{i!}e^{-\lambda_1}\frac{\lambda_2^{k-i}}{(k-i)!}e^{-\lambda_2} \\
&= e^{-(\lambda_1+\lambda_2)}\sum_{i=0}^{k}\frac{\lambda_1^i\lambda_2^{k-i}}{i!(k-i)!} = \frac{e^{-(\lambda_1+\lambda_2)}}{k!}\sum_{i=0}^{k}\frac{k!\lambda_1^i\lambda_2^{k-i}}{i!(k-i)!} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{k!}\sum_{i=0}^{k}\binom{k}{i}\lambda_1^i\lambda_2^{k-i} = \frac{e^{-(\lambda_1+\lambda_2)}(\lambda_1+\lambda_2)^k}{k!}.
\end{aligned}
$$

Show by induction the general case for any integer $k > 0$.

Poisson Process (Revision)

- The Poisson process is a pure-chance point process on the real line.
- The points on the real line (that normally represents time) are events (e.g. call arrivals). Events are random and independent of each other.
- The parameter λ is the **rate** of the Poisson process.
- The time between consecutive events is exponentially distributed with parameter λ.
- The mean time between consecutive events is equal to $\frac{1}{\lambda}$.
- The number of events in an interval time T is Poisson distributed with parameter λT.
- The mean number of events in an interval time T is equal to λT.
- The variance of the number of events in an interval time T is also equal to λT.
- The number of events in disjoint intervals are independent random variables, so the number of future events is independent of the past.

Superposition of Poisson Processes (Revision)

Consider two Poisson processes: one with parameter λ_1 and the other with parameter λ_2.

The superposition of these two processes is a new point process that comprises all the points of both processes.

This superposition is a Poisson process with rate $\lambda = \lambda_1 + \lambda_2$.

Superposition of Poisson Processes (Revision) (cont'd)

Notice that for any time interval T, the number of events in the process, which is the superposition of the two processes, will be the sum of the number of events occured in the two processes. Therefore, it is Poisson distributed with parameter $\lambda_1 T + \lambda_2 T = \lambda T$.

Notice also that the time between two consecutive events in the process, which is the superposition of the two processes, will be the minimum of two exponential random variables: one with parameter λ_1 and the other with parameter λ_2. We know that this minimum is exponentially distributed with parameter $\lambda = \lambda_1 + \lambda_2$.

In general: the superposition of k Poisson processes with parameters $\lambda_1, \lambda_2, \ldots, \lambda_k$, is a Poisson process with rate $\lambda = \lambda_1 + \lambda_2, \ldots, \lambda_k$.

Again, show it by induction.

Erlang B System (M/M/k/k)

The Erlang B system is a system of k channels (or servers) where calls arrive according to a Poisson process with parameter λ. If a call arrives and finds that all the channels are busy, the call is blocked and cleared from the system. Otherwise, it will use one of the channels for a period of time which is called the **holding time**. The holding time is assumed to be exponential with parameter μ. Therefore, the mean of the holding time is given by

$$E[\text{holding time}] = \frac{1}{\mu}.$$

The **traffic offered to the system** or the **offered traffic** (denoted A) is the product of the arrival rate by the mean holding time. It is measured by a dimensionless unit called erlang (in honor of Agner Krarup Erlang), and it is given by

$$A = \frac{\lambda}{\mu}\ [erlangs].$$

Erlang B System (cont'd)

Define π_n as the steady state probability that exactly n channels are busy, $n = 0, 1, 2, \ldots, k$.

π_k is the probability that k channels are busy. It is also the probability that an arriving call is blocked. It is therefore called the **blocking probability** which is an important system performance measure in many application. It is also an important Quality of Service (QoS) measure in cellular networks.

π_0 is the probability that all the channels are idle; namely, it is the probability that the system is empty.

The **carried traffic** is given by $(1 - \pi_k)A$.

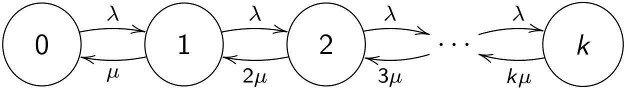
The **lost traffic** is the difference between the offered traffic and the carried traffic and it is given by

$$A - (1 - \pi_k)A = \pi_k A.$$

Erlang B System (cont'd)

The Erlang B system is said to be in state n if exactly n channels are busy. Accordingly, π_n is the steady state probability that the Erlang B system is in state n.

The following diagram is the **state transition diagram** of the Erlang B system.

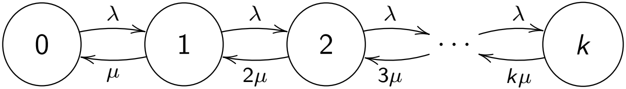


When the system is in **state** 0, the next event must be an arrival.

However, when the system is in state n, for $k \geq n > 0$, the next event can be either an arrival or a departure.

When the system in is state k, and an arrival occurs, it will be blocked and will not change the system state. Therefore, in state k, the next event that will change the system state **must be** a departure.

Erlang B System (cont'd)



For any state n, for $k \geq n \geq 0$, the time until the next arrival is exponentially distributed with parameter λ.

Therefore, when the system is in state 0, the time until the next **event** (which must be an arrival) is exponentially distributed with parameter λ.

When the system is in state n, for $k \geq n > 0$, the time until the next departure is the minimum of n exponential random variables each of which with parameter μ. Therefore, the time until the next **event** is exponentially distributed with parameter $n\mu + \lambda$.

Erlang B System (cont'd)

It is given that the system is in state n, for $k \geq n \geq 0$. (Notice that the case $n = 0$ is now included.)

What is the probability that the next event is an arrival?

What is the probability that the next event is a departure?

Erlang B System (cont'd)

Answers

The probability that the next event is an arrival is

$$\frac{\lambda}{n\mu + \lambda}.$$

The probability that the next event is a departure is

$$\frac{n\mu}{n\mu + \lambda}.$$

Notice that these apply to the case $n = 0$ where the next event is an arrival with probability 1.

Continuous-time Markov Chain

- A **Markov chain** is a random process that at any point in time can be in a discrete number of states and the time it stays in the same state follows a memoryless distribution. A Markov chain can be continuous-time or discrete-time. In this lecture, we focus on continuous-time Markov chains.
- A continuous-time Markov chain stays in state i exponentially distributed amount of time with parameter μ_i.
- After spending an exponential amount of time in state i, it will transit to state j with transition probability p_{ij}.
- Because the exponential distribution is memoryless and the transition probabilities are independent of the past evolution of the process, at any point in time, the future evolution of the process depends only on the present state.
- Because of their special properties Markov chains are easy to analyse and to simulate.

Markov Chain Simulation of the Erlang B System

An Erlang B system is a continuous-time Markov chain. We will now show how we can program a Markov chain simulation for an Erlang B system to estimate the blocking probability.

Variables and input parameters:

k = number of servers
Q = number of customers in the system (queue size)
B_p = estimation for the blocking probability
N_a = number of customer arrivals counted so far
N_b = number of blocked customers counted so far
$MAXN_a$ = maximal number of customer arrivals (It is used for the stopping condition.)
μ = service rate
λ = arrival rate
$R(01)$ = a uniform $U(0,1)$ random deviate (A new value for $R(01)$ is generated every time It is called.)

Markov Chain Simulation of the Erlang B System (cont'd)

A Pseudo code for the estimation of the blocking probability

```
Initialization: $Q = 0$; $N_a = 0$, $N_b = 0$.

1. If $R(01) \leq \lambda/(\lambda + Q\mu)$, then $N_a = N_a + 1$; if $Q = k$ then $N_b = N_b + 1$,
   else $Q = Q + 1$;
   else, Q = Q - 1.

2. If $N_a < MAXN_a$ go to 1; else, print $B_p = N_b/N_a$.
```

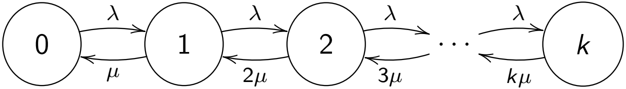
Note: The condition

$$R(01) \leq \frac{\lambda}{\lambda + Q\mu}$$

indicates that the next event is an arrival. If this condition does not hold we will have a departure (service completion).

The program has only two IF statements: the first checks if the next event is an arrival or a departure, and the second is a stopping criterion.

Steady-state (Detailed Balance) Equations of the Erlang B System



Let us write a set of linear equations for $\pi_n, \quad n = 0, 1, 2, \ldots, k$.

We begin with the so-called **normalizing equation** based on the fact that the sum of the steady-state probabilities must be equal to one. That is,

$$\sum_{n=0}^{k} \pi_n = 1.$$

Then, we write a set of steady-state equations that equate the rate of transitions from state n to state $n+1$ with the rates of transitions from state $n+1$ to state n, for $n = 0, 1, 2, \ldots, k-1$.

$$\pi_n \lambda = \pi_{n+1}(n+1)\mu, \text{ for } n = 0,\ 1,\ 2,\ \ldots, k-1.$$

The Erlang B Formula

This set of steady-state equations are solved by recursively writing $\pi_1, \pi_2, \pi_3, \ldots, \pi_k$ in terms of π_0, and then using the normalizing equation to obtain π_0. Finally, all the other π_n are obtained including π_k that gives the blocking probability.

For details on these derivations, see: <classnotes.pdf> available on Canvas.

In addition to π_k, a common notation for the blocking probability is $E_k(A)$. Using this notation makes it clear that we assume an Erlang system with offered traffic A and k channels.

Solution of the steady-state equations leads to the following well-known Erlang B formula:

$$E_k(A) = \pi_k = \frac{\frac{A^k}{k!}}{\sum_{n=0}^{k} \frac{A^n}{n!}}.$$

The Erlang B Formula (cont'd)

Erlang B recursion

For systems with large k, it is difficult to compute the blocking probability using the above formula. Then, the following recursion is useful.

$$E_m(A) = \frac{AE_{m-1}(A)}{m + AE_{m-1}(A)}, \text{ for } m = \ 1,\ 2,\ \ldots, k.$$

When $m = 0$, there are no channels available, and therefore all the calls are blocked, namely,

$$E_0(A) = 1.$$

The above two equations give rise to a simple recursive algorithm by which the blocking probability can be calculated for a large k.

For details on the derivations of the Erlang B recursion, see: <classnotes.pdf> available on Canvas.

Insensitivity of the Erlang B formula

An important property of the Erlang B formula is that **it is insensitive to the holding time distribution**. Although we have assumed that the holding times are exponentially distributed, the blocking probability result of the Erlang B formula holds true even if the holding times are not exponentially distributed. We only need know the mean holding time, then, we can use Erlang B formula to obtain the exact blocking probability.

Homework:
Simulate an M/M/k/k queue based on the Markov-chain simulation principles to evaluate the blocking probability for a wide range of parameter values. Provide confidence intervals. Compare the simulation results with results obtained by the Erlang B Formula. Do the Erlang B results fall within the confidence intervals of the simulations?

The Special Case: M/M/1/1

Homework: Derive a formula for the blocking probability of M/M/1/1 by **Erlang B Formula** and by **the Erlang B recursion**.

Another way to obtain a formula for the blocking probability of M/M/1/1 is using **Little's formula**. The M/M/1/1 system can have at most one customer in it. Therefore, its mean queue size is given by

$$E[Q] = 0\pi_0 + 1\pi_1 = \pi_1$$

which is also its blocking probability. The arrival rate into the system (made only of successful arrivals) is equal to $\lambda(1 - E[Q])$. The mean time a customer stays in the system is $1/\mu$. Then, by Little's formula,

$$\frac{\lambda(1 - E[Q])}{\mu} = E[Q].$$

Isolating $E[Q]$, the blocking probability is given by

$$\pi_1 = E[Q] = \frac{A}{1 + A}.$$

Properties of Erlang B Formula

- The blocking probability $E_k(A)$ monotonically increases as the offered traffic A increases (for a fixed k).
- The blocking probability $E_k(A)$ monotonically decreases as the number of servers k increase (for a fixed A).
- If (1) the ratio A/k stays fixed, (2) $A/k \leq 1$, and (3) k and A both approach infinity, then $E_k(A)$ approaches zero.
- If (1) the ratio A/k stays fixed, (2) $A/k > 1$, and (3) k and A both approach infinity, then $E_k(A)$ approaches $1 - k/A$.

See <classnotes.pdf> for further discussions, proofs, homework and references.

M/M/k/k Utilization

The mean number of busy circuits in an M/M/k/k system fed by A erlangs using Little's formula is given by

$$E[Q] = (1 - E_k(A))\lambda \times \frac{1}{\mu} = (1 - E_k(A))A.$$

Accordingly, the **utilization** of an M/M/k/k system is given by

$$\hat{U} = \frac{[1 - E_k(A)]A}{k}.$$

M/M/k/k Dimensioning

The dimensioning problem in M/M/k/k is as follows. Find the minimal number of circuits/servers – integer k, so that for a given A, the value of $E_k(A)$ is below a certain value (e.g. 0.01). Recall that the value of $E_k(A)$ is a QoS measure of customers or Grade of Service (GoS) for link capacity dimensioning which eventually affect QoS.

Taking advantage of the monotonicity of Erlang formula, we can also solve the dimensioning problem of finding the required minimal k.

We simply keep incrementing the number of servers/circuits k and calculate in each case the blocking probability. When the desired blocking probability (e.g., 1%) is reached, we have our answer.

Normally, it is impossible to find k that gives exactly, e.g. 1% blocking probability, so we choose k such that $E_k(A) < 1\%$, but $E_{k-1}(A) > 1\%$.

M/M/k/k Dimensioning (cont'd)

The following table provides the minimal values of k obtained for various values of A such that the blocking probability is no more than 1%, and the utilization in each case. Observe the increase in utilization with the traffic.

A	k	$E_k(A)$	Utilization
20	30	0.0085	66.10%
100	117	0.0098	84.63%
500	527	0.0095	93.97%
1000	1029	0.0099	96.22%
5000	5010	0.0100	98.81%
10000	9970	0.0099	99.30%

Homework: Write a computer program to reproduce the results of the above table.

M/M/k/k Dimensioning (cont'd)

Dimensioning Simply by $k = A$ for Heavy Traffic

Notice that for the case of $A = 10,000$ erlangs, to maintain no more than 1% blocking, k value less than A is sufficient.

Recall that the carried traffic is not A but $A(1 - E_k(A))$.

For $A \geq 10,000$, dimensioning simply by $k = A$ will mean no more than 1% blocking and no less than 99% utilization - not bad for such a simple rule of thumb! This also implies that if the system capacity is much larger than individual service requirement, very high efficiency (utilization) can be achieved without a significant compromise on QoS.

Let us now further examine the case $k = A$.

M/M/k/k under Critical Loading ($k = A$)

If we maintain $k = A$ and we increase both k and A, the blocking probability decreases, the utilization increases, and interestingly, the product $E_k(A)\sqrt{A}$ approaches a constant, denoted $\tilde{C}$, that does not depend on A or k.

A	k	$E_k(A)$	Utilization	$E_k(A)\sqrt{A}$
10	10	0.215	78.5 %	0.679
100	100	0.076	92.4%	0.757
1000	1000	0.025	97.5%	0.785
10000	10000	0.008	99.2%	0.79365
20000	20000	0.00562	99.438%	0.79489
50000	50000	0.00356	99.644%	0.79599

Homework: Reproduce the results of the above table.

Explanation of Low Blocking and High Utilization in a Large M/M/k/k System with $k = A$

In such a case the standard deviation of the traffic is very small relative to the mean.

Therefore, the traffic behaves close to deterministic (with very little variations).

If 100 liters per second of water are offered, at a constant rate, to a pipe that has capacity of 100 liters per second, then the pipe can handle the offered load with very small losses.

Insensitivity of the Erlang B Formula

The distribution and the mean of the number of busy servers is insensitive to the shape of the service time distribution (although it is still sensitive to the mean of the service time) in the cases of M/G/∞ and M/G/k/k.

For M/G/k/k, also the blocking probability is insensitive to the shape of the service time distribution.

However, this insensitivity property does not extend to the arrival process. We still require a Poisson arrival process for the Erlang B formula to apply.

If we have a more burtsy arrival process (e.g. Poisson batch arrivals), we will have more losses than predicted by the Erlang B formula, and if we have a smoother arrival process than Poisson, we will have less losses than predicted by the Erlang B formula.

Comparison D/D/1/1 versus M/M/1/1

We illustrate that Erlang B does not apply if the arrival process is not Poisson by comparing M/M/1/1 versus D/D/1/1, where each of these two systems is fed by A erlangs, and that $A < 1$.

Arrivals into the D/D/1/1 system with $A < 1$ will never experience losses because the inter-arrivals are longer that the service times, so the service of a customer is always completed before the arrival of the next customer.

Accordingly, by Little's formula: $E[Q] = A$, and since $E[Q] = 0 \times \pi_0 + 1 \times \pi_1$, we have that $\pi_1 = A$ and $\pi_0 = 1 - A$.

In this case, the blocking probability P_b is equal to zero and not to π_1.

The utilization will be given by $\hat{U} = \pi_1 = A$.

Comparison D/D/1/1 versus M/M/1/1 (cont'd)

By contrast, for the M/M/1/1 system,
$P_b = E_1(A) = E[Q] = \pi_1 = A/(1+A)$, so $\pi_0 = 1 - \pi_1 = 1/(1+A)$.
The utilization is obtained by $\hat{U} = \pi_1 = A/(1+A)$, or

$$\hat{U} = (1 - \pi_1)A = [1 - A/(1+A)]A = A/(1+A).$$

This comparison is summarized in the following table:

	M/M/1/1	D/D/1/1
π_0	$1/(1+A)$	$1-A$
π_1	$A/(1+A)$	A
$\hat{U}$	$A/(1+A)$	A
P_b	$A/(1+A)$	0
$E[Q]$	$A/(1+A)$	A

The results for the two systems are different.

Two Types of Customers

Problem

Consider two classes of customers (packets). Class i customers, $i = 1, 2$ arrive following an independent Poisson process at rate of λ_i each of which requires independent and exponentially distributed service time with parameter μ_i, for $i = 1, 2$. There are k independent servers without waiting room (without additional buffer). The aim is to derive overall blocking probability.

Solution

The combined arrival process of all the customers is a Poisson process with parameter $\lambda = \lambda_1 + \lambda_2$.

The probability of an arbitrary customer to belong to the first class is

$$p = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{\lambda_1}{\lambda}.$$

Two Types of Customers (Cont'd)

Therefore, the service time of an arbitrary customer has hyperexponential distribution because with probability p it is exponentially distributed with parameter μ_1, and with probability $1 - p$, it is exponentially distributed with parameter μ_2.

Accordingly, by the Law of iterated expectation, the mean service time (holding time) is given by

$$E[S] = \frac{p}{\mu_1} + \frac{1-p}{\mu_2}$$

so $A = \lambda E[S]$, and the blocking probability is $E_k(A)$.

Two Types of Customers (Cont'd)

Furthermore, let

$$A_i = \frac{\lambda_i}{\mu_i} \qquad i = 1, 2$$

and observe that

$$E[S] = \left(\frac{\lambda_1}{\lambda}\right)\left(\frac{1}{\mu_1}\right) + \left(\frac{\lambda_2}{\lambda}\right)\left(\frac{1}{\mu_2}\right) = \frac{A_1 + A_2}{\lambda}.$$

Then

$$A = \lambda E[S] = A_1 + A_2.$$

Homework: Extend the blocking probability derivation to the case of n types of customers.

Preemptive Priorities

We now extend the M/M/k/k loss system to the case where the arriving calls (customers) are of m priority types. Where priority 1 represents the highest priority and priority m represents the lowest priority.

In general, if $i < j$ then priority i arrival may preempt a priority j customer upon its arrival.

The arrival process of priority i customers follows a Poisson process with rate λ_i, for $i = 1, 2, 3, \ldots, m$.

The service time of all the customers is exponentially distributed with parameter μ.

The offered traffic of priority i customers is given by

$$A_i = \frac{\lambda_i}{\mu}, \quad i = 1, 2, 3, \ldots, m.$$

Preemptive Priorities (Cont'd)

Let $P_b(i)$ be the blocking probability of priority i customers.

Because the priority 1 traffic access the system regardless of low priority loading, for the case $i = 1$, we have

$$P_b(1) = E_k(A_1).$$

To obtain $P_b(i)$ for $i > 1$, we first observe that because of the memoryless of the exponential distribution, the blocking probability of all the traffic of priority i and higher priorities, namely, the traffic generated by priorities $1, 2, \ldots, i$, is given by

$$E_k(A_1 + A_2 + \ldots, A_i).$$

Preemptive Priorities (Cont'd)

Next, we observe that the lost traffic of priority i, $i = 1, 2, 3, \ldots, m$, is given by the lost traffic of priorities $1, 2, 3, \ldots, i$ minus the lost traffic of priorities $1, 2, 3, \ldots, i-1$, namely,

$$\begin{aligned}[\text{Lost traffic of priority } i\ ] = & (A_1 + A_2 + \ldots, A_i) E_k (A_1 + A_2 + \ldots, A_i) \\ & -(A_1 + A_2 + \ldots, A_{i-1}) E_k (A_1 + A_2 + \ldots, A_{i-1}).\end{aligned}$$

Therefore, the value of $P_b(i)$ for $i > 1$, can be obtained as the ratio of the lost traffic of priority i to the offered traffic of priority i, that is,

$$P_b(i) = \frac{\left(\sum_{j=1}^{i} A_j\right) E_k \left(\sum_{j=1}^{i} A_j\right) - \left(\sum_{j=1}^{i-1} A_j\right) E_k \left(\sum_{j=1}^{i-1} A_j\right)}{A_i}.$$

Overflow Traffic of M/M/k/k

In many practical situations traffic that cannot be admitted to a k server group overflows to another server group. In such a case the overflow traffic is not Poisson, but it is more bursty than a Poisson process. That is, the variance of the number of arrivals in an interval is higher than the mean number of arrivals in that interval.

It is therefore important to characterize such overflow traffic by its variance and its mean.

In particular, consider an M/M/k/k queueing system with input offered traffic A and let M [Erlangs] be the traffic overflowed from this k-server system.

$$M = AE_k(A).$$

Overflow Traffic of M/M/k/k (Cont'd)

Let V be the variance of the overflow traffic. Namely, V is the variance of the number of busy servers in an infinite server systems to which the traffic overflowed from our M/M/k/k is offered.

The following result known as Riordan Formula, can be used to obtain V.

$$V = M\left(1 - M + \frac{A}{k + 1 + M - A}\right).$$

Note that M and V of the overflow traffic are completely determined by k and A.

The variance to mean ratio of a traffic stream is called *Peakedness*. In our case, the peakedness of the overflow traffic is denoted Z and is given by

$$Z = \frac{V}{M},$$

and it normally satisfies $Z > 1$.

Overflow Traffic of M/M/k/k (Cont'd)

Consider two loss systems called the primary and the secondary systems.

The number of servers in the primary system is k_1 and the number of servers in the secondary system is k_2.

Assume that the offered traffic to the primary server group is equal to A and the overflow traffic of the primary system is the offered traffic of the secondary system.

Then the blocking probability of the secondary system is obtained by

$$P_b(\text{secondary}) = \frac{E_{k_1+k_2}(A)}{E_{k_1}(A)}.$$

In this case we also know the mean and variance of the traffic offered to the secondary system because they are statistics M and V of the overflow traffic of the primary system.

Multi-server Loss Systems with Non-Poisson Input

Consider a generalization of an M/M/k/k system to the case where the arrival process is not Poisson.

In particular, given a multi-server loss system with k servers loaded by non-Poisson offered traffic with mean M and variance V, find the blocking probability.

This problem does not have an exact solution, but reasonable approximations are available.

We will present two approximations:

1. Hayward Approximation
2. Equivalent Random Method (ERM).

Hayward Approximation

The **Hayward approximation** is based on the following result.

A multi-server system with k servers fed by traffic with mean M and variance V has a similar blocking probability to that of an M/M/k/k system with offered load $\frac{M}{Z}$ and $\frac{k}{Z}$ servers. Hence, the blocking probability is approximated by

$$P_b(k, M, V) \approx E_{\frac{k}{Z}}\left(\frac{M}{Z}\right)$$

where $\frac{k}{Z}$ is rounded to an integer.

To be conservative, use for the number of servers:

$$\left\lfloor \frac{k}{Z} \right\rfloor .$$

Equivalent Random Method (ERM)

Again we have a loss system with k servers and the mean and variance of the offered traffic are M and V are known.

This offered traffic could have come from or overflowed from various sources, the idea of ERM is to assume that the offered traffic comes from a primary loss system with N_{eq} servers and offered traffic A_{eq}.

We know M and V, but we need to find N_{eq} and A_{eq}. Then we can use the result obtained for the blocking probability $P_b(k, M, V)$.

Equivalent Random Method (ERM)

Equivalent Notations

	M/M/k/k overflow to a secondary system	ERM
(mean) offered traffic - primary	A	A_{eq}
number of servers - primary	k_1	N_{eq}
number of servers - secondary	k_2	k
(mean) offered traffic - secondary	M	M
variance offered traffic - secondary	V	V
blocking probability of secondary/non-Poisson system	$P_b(secondary)$	$P_b(k, M.V)$

Equivalent Random Method (ERM)

$$P_b(k, M, V) \approx \frac{E_{N_{eq}+k}(A_{eq})}{E_{N_{eq}}(A_{eq})},$$

where

$$A_{eq} \approx V + 3Z(Z-1),$$

and

$$N_{eq} = \frac{A_{eq}(M+Z)}{M+Z-1} - M - 1,$$

where

$$Z = \frac{V}{M}.$$

Chapter 9: M/M/k

March 24, 2025

M/M/k – a generalization of M/M/1

The M/M/k queue is a generalization of the M/M/1 queue to the case of k servers.

As in M/M/1, for an M/M/k queue, the buffer is infinite and the arrival process is Poisson with rate λ.

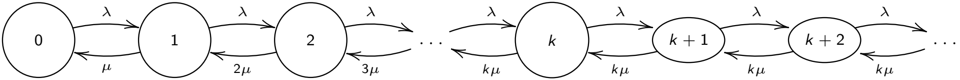
The service time of each of the k servers is exponentially distributed with parameter μ.

As in the case of M/M/1 we assume that the service times are independent and are independent of the arrival process.

State Transitions of M/M/k

The M/M/k queue is said to be in state n, $n \geq 0$ if there are exactly n customers in the system (including in the service and waiting in the queue). Accordingly, π_n is the steady state probability that the M/M/k queue is in state n.

The following diagram is the **state transition diagram** of the M/M/k queue.



As in M/M/1 and in M/M/k/k, when the system is in **state** 0, the next event must be an arrival.

As in M/M/1, when the system is in state n, for $n > 0$, the next event can be either an arrival or a departure.

Steady State Equations of M/M/k

Letting $A = \lambda/\mu$, and assuming the stability condition $\lambda < k\mu$, or $A < k$, the M/M/k queue gives rise to the following steady-state equations:

$$\pi_1 = A\pi_0$$
$$\pi_2 = A\pi_1/2 = A^2\pi_0/2$$
$$\pi_3 = A\pi_2/3 = A^3\pi_0/(3!)$$
$$\ldots$$
$$\pi_k = A\pi_{k-1}/k = A^k\pi_0/(k!)$$
$$\pi_{k+1} = A\pi_k/k = A^{k+1}\pi_0/(k!k)$$
$$\pi_{k+2} = A\pi_{k+1}/k = A^{k+2}\pi_0/(k!k^2)$$
$$\ldots$$
$$\pi_{k+j} = A\pi_{k+j-1}/k = A^{k+j}\pi_0/(k!k^j) \text{ for } j = 1,\ 2,\ 3,\ \ldots$$

Steady State Equations of M/M/k (cont'd)

and in general:

$$\pi_n = \frac{A^n \pi_0}{n!} \text{ for } n = 0,\ 1,\ 2,\ \ldots,\ k-1$$

and

$$\pi_n = \frac{A^n \pi_0}{k! k^{n-k}} \text{ for } n = k,\ k+1,\ k+2,\ \ldots\ .$$

Solution of the Steady State Equations

To obtain π_0, we sum up both sides of the steady state equations, and because the sum of the π_ns equals one, we obtain an equation for π_0, which its solution is

$$\pi_0 = \left(\sum_{n=0}^{k-1} \frac{A^n}{n!} + \frac{A^k}{k!} \frac{k}{(k-A)} \right)^{-1}.$$

Substituting the latter in the above steady state equations, we obtain the steady-state probabilities π_n, $n = 0,\ 1,\ 2,\ \ldots$.

Erlang C Formula

Of special interest is the so called Erlang C formula. It represents the proportion of time that all k servers are busy and is given by:

$$C_k(A) = \sum_{n=k}^{\infty} \pi_n = \frac{A^k}{k!}\frac{k}{(k-A)}\pi_0 = \frac{\frac{A^k}{k!}\frac{k}{(k-A)}}{\sum_{n=0}^{k-1}\frac{A^n}{n!} + \frac{A^k}{k!}\frac{k}{(k-A)}}.$$

This leads to the following relationship between Erlang B and Erlang C formulae.

$$C_k(A) = \frac{kE_k(A)}{k - A[1 - E_k(A)]}.$$

Comparisons of Erlang B and C Results

In the following table, we observe significant differences between $C_k(A)$ and $E_k(A)$ results as the ratio A/k increases. Clearly, when $A/k > 1$, the M/M/k queue is unstable.

A	k	$E_k(A)$	$C_k(A)$
20	30	0.0085	0.025
100	117	0.0098	0.064
500	527	0.0095	0.158
1000	1029	0.0099	0.262
5000	5010	0.0100	0.835
10000	9970	0.0099	unstable

Notation - Revision

$Q =$ the total number of customers in the system (waiting in the queue and being served);

$N_Q =$ the total number of customers waiting in the queue (this does not include those customers being served);

$N_s =$ the total number of customers that are being served;

$D =$ the total delay in the system (this includes the time a customer waits in the queue and in service);

$W_Q =$ the time a customer waits in the queue (this excludes the time a customer spends in service);

$S =$ a random variable representing the service time.

$\hat{D} =$ the delay of a delayed customer including the service time.

$\hat{W_Q} =$ the delay of a delayed customer in the queue excluding the service time.

Queueing Performance Formulae

$$E[Q] = E[N_Q] + E[N_s]$$

$$E[D] = E[W_Q] + E[S].$$

$$E[S] = \frac{1}{\mu}.$$

To obtain $E[N_s]$ for the M/M/k queue, we use Little's formula for the system made of servers to obtain the mean number of busy servers given by

$$E[N_s] = \frac{\lambda}{\mu} = A.$$

Queueing Performance Formulae (cont'd)

To obtain $E[N_Q]$, we consider two mutually exclusive and exhaustive events: $\{Q \geq k\}$, and $\{Q < k\}$. Recalling the Law of Iterated Expectation, we have

$$E[N_Q] = E[N_Q \mid Q \geq k]P(Q \geq k) + E[N_Q \mid Q < k]P(Q < k).$$

To derive $E[N_Q \mid Q \geq k]$, we notice that the evolution of the M/M/k queue during the time when $Q \geq k$ is equivalent to that of an M/M/1 queue with arrival rate λ and service rate $k\mu$. The mean queue size of such M/M/1 queue is equal to $\rho/(1-\rho)$ where $\rho = \lambda/(k\mu) = A/k$. Thus,

$$E[N_Q \mid Q \geq k] = \frac{A/k}{1 - A/k} = \frac{A}{k - A}.$$

Therefore, since $E[N_Q \mid Q < k] = 0$ and $P(Q \geq k) = C_k(A)$, we obtain

$$E[N_Q] = C_k(A)\frac{A}{k - A}.$$

Queueing Performance Formulae (cont'd)

$$E[Q] = E[N_Q] + E[N_s] = C_k(A)\frac{A}{k-A} + A.$$

Therefore, by Little's formula

$$E[W_Q] = \frac{C_k(A)\frac{A}{k-A}}{\lambda} = \frac{C_k(A)}{\mu k - \lambda}.$$

Notice the physical meaning of $E[W_Q]$. It is the ratio between the probability of having all servers busy and the spare capacity of the system. The mean delay is readily obtained by adding the mean service time to $E[W_Q]$. Thus,

$$E[D] = \frac{C_k(A)}{\mu k - \lambda} + \frac{1}{\mu}.$$

Delay Factor

Another useful measure is the so-called *delay factor*. It is defined as the ratio of the mean waiting time in the queue to the mean service time.

Namely, it is given by

$$D_F = \frac{E[W_Q]}{1/\mu} = \frac{\frac{C_k(A)}{\mu k-\lambda}}{\frac{1}{\mu}} = \frac{C_k(A)}{k-A}.$$

The rationale to use delay factor is that in some applications users that require long service time may be willing to wait longer time in the queue in direct proportion to the service time.

Mean Delay of Delayed Customers

As in the M/M/1 case, to obtain $E[\hat{W}_Q]$, we use Little's formula where we consider the queue (without the servers) as the system and the arrival rate of the delayed customers which in the present case is $\lambda C_k(A)$. Therefore,

$$E[\hat{W}_Q] = \frac{AC_k(A)}{\lambda C_k(A)(k-A)} = \frac{1}{k\mu - \lambda}.$$

$$E[\hat{D}] = E[\hat{W}_Q] + \frac{1}{\mu} = \frac{1}{k\mu - \lambda} + \frac{1}{\mu}.$$

As in M/M/1, we check the latter using the Law of Iterated Expectation:

$$\begin{aligned} E[D] &= (1 - C_k(A))E[S] + C_k(A)E[\hat{D}] \\ &= (1 - C_k(A))\frac{1}{\mu} + C_k(A)\left(\frac{1}{k\mu - \lambda} + \frac{1}{\mu}\right) = \frac{C_k(A)}{\mu k - \lambda} + \frac{1}{\mu}. \end{aligned}$$

This consistency check is an alternative way to obtain $E[\hat{D}]$.

Dimensioning

Potential dimensioning problems are of finding, for a given A, the smallest k such that $C_k(A)$, or the mean delay, is lower than a given value.

This is done by realizing that the value of $C_k(A)$ or $E[D]$ decreases as k increases.

The dimensioning problem with respect to $C_k(A)$ can be solved in an analogous way to the M/M/k/k dimensioning problem.

Having the $C_k(A)$ value for each k value, one can also obtain the minimal k such that the mean delay is bounded by a given value.

A similar procedure can be used to find the minimal k such that delay factor requirement is met.

Utilization

The utilization of an M/M/k queue is the ratio of the mean number of busy servers to k, therefore the utilization of an M/M/k queue is obtained by

$$\hat{U} = \frac{E[N_s]}{k} = \frac{A}{k}.$$

Chapter 13: Processor Sharing

August 3, 2019

Processor Sharing Preliminaries

In a processor sharing (PS) queueing system the server capacity is shared equally among all the customers that are present in the system.

This model is applicable to a time-shared computer system where a central processor serves all the jobs present in the system simultaneously at an equal service rate. Another important application of PS is for a multiplicity of TCP connections that share a common bottleneck.

The Internet router at the bottleneck simultaneously switches (serves) the flows generated by the users, while TCP congestion control mechanism guarantees that the service rate obtained by the different flows are equal.

Processor Sharing Preliminaries (Cont'd)

As any of the other queueing models considered in this course, the PS model is only an approximation for the various real-life scenarios.

It does not consider overheads and wastage associated with various real life operations of computer systems, and therefore it may be expected to underestimate queueing delay.

If the server capacity to render service is μ [customers per time-unit] and there are i customers in the system, each of the customers is served at the rate of μ/i.

As soon as a customer arrives, its service starts.

The M/M/1-PS queue

The M/M/1-PS queue is characterized by Poisson arrivals and exponentially distributed service-time requirement, as the ordinary (FIFO) M/M/1 queue), but its service regime is assumed to be processor sharing.

In particular, we assume that the process of the number of customers i in the system is a continuous time Markov chain, where customers arrive according to a Poisson process with parameter λ [customers per time-unit] and that the service time required by an arriving customer is exponentially distributed with parameter μ.

We also assume the stability condition of $\lambda < \mu$.

The M/M/1-PS queue (Cont'd)

Let us consider now the transition rates of the continuous-time Markov chain for the number of customers in the system associated with the M/M/1-PS model.

Firstly, we observe that the transition rates from state i to state $i + 1$ is λ as in the M/M/1 model.

We also observe that the rates from state i to state $i + j$ for $j > 1$ and from state i to state $i - j$ for $j > 1$ are all equal to zero (again, as in M/M/1).

The latter is due to the fact that the probability of having more than one event, arrival or departure, occurred at the same time is equal to zero.

The M/M/1-PS queue (Cont'd)

To derive the rates from state i to state $i-1$ for $i \geq 1$ notice that at state i, assuming that no arrivals occur, the time until a given customer completes its service is exponentially distributed with rate μ/i.

Therefore, the time until the first customer out of the i customers that completes its service is the minimum of i exponential random variables each of which with rate μ/i, which is exponentially distributed with rate $i(\mu/i) = \mu$.

Therefore, the transition rates from state i to state $i-1$ is equal to μ (again, as in M/M/1).

The M/M/1-PS queue (Cont'd)

These imply that the process of number of customers in the system associated with the M/M/1-PS model is statistically the same as the continuous-time Markov chain that describes the M/M/1 (FIFO) queue.

Therefore, the queue size steady-state distribution $\{\pi_i\}$ and the mean queue-size $E[Q]$ obtained for M/M/1 are also applied to the M/M/1-PS model. That is,

$$\pi_i = \rho^i(1-\rho) \text{ for } i = 0,\ 1,\ 2,\ \ldots$$

and

$$E[Q] = \frac{\rho}{1-\rho}.$$

The M/M/1-PS queue (Cont'd)

By Little's formula the result obtained for the mean delay $E[D]$ in Eq. (**??**) is also applicable to the M/M/1-PS model:

$$E[D] = \frac{1}{(1-\rho)\mu} = \frac{1}{\mu - \lambda}.$$

However, the delay distribution of M/M/1 given by Eq. (**??**) does not apply to M/M/1-PS.

Having obtained the mean delay for a customer in the M/M/1-PS queue, an interesting question is what is the mean delay of a customer that requires amount of service x. Here x represents the time that the customer spends in the system to complete its service assuming that there are no other customers being served and all the server capacity can be dedicated to it. Let $E[D|x]$ be the mean delay of such customer conditional on its service requirement x.

The M/M/1-PS queue (Cont'd)

For an M/M/1 queue under the FIFO discipline, denoted in this section as M/M/1-FIFO, the time a customer waits in the queue is not a function of its service requirement x because it depends only on service requirements of other customers. Only after the customer completes its waiting time in the queue, x will affect its total delay simply by being added to the waiting time in the queue.

In particular, for M/M/1-FIFO,

$$E[D|x] = E[W_Q] + x = \frac{\rho}{\mu - \lambda} + x,$$

where the last equality is obtained by substituting the value of $E[W_Q]$ obtained for M/M/1.

The M/M/1-PS queue (Cont'd)

By comparison, in the case of the M/M/1-PS queue, for a customer that requires service time of x from a dedicated server, its mean delay in the system from the moment it arrives until its service is complete has linear relationship with x. That is,

$$E[D|x] = cx,$$

for some constant c.

That is, under PS, the mean delay of a message has linear relationship with its required service time.

The implication of this is that, on average, if a customer require twice as much service than another customer, its mean delay will be twice that of the delay of the other customer.

The M/M/1-PS queue (Cont'd)

Note that as soon as a message arrives, its service starts. It does not need to wait in a queue to the start of a service, but the service rate changes with the load.

We know that under our stability assumption, the process of the number of customers in the system is a stable and stationary continuous time Markov chain.

It is a birth-and-death process because the transitions are only up by one or down by one.

Therefore, the infinitesimal service rate obtained by a test customer will also follow a stable and stationary continuous time Markov chain. This provides intuitive explanation to the linear relationship of $E[D|x]$.

The M/M/1-PS queue (Cont'd)

To obtain the parameter c, we first obtain the mean delay of an average size message and then invoke the linearity of $E[D|x]$.

This is done by taking the mean with respect to x on both sides of $E[D|x] = cx$ and invoking the law of iterated expectations, we obtain

$$E[D] = c\frac{1}{\mu},$$

and we obtain

$$\frac{1}{(1-\rho)\mu} = c\frac{1}{\mu}.$$

Thus,

$$c = \frac{1}{1-\rho},$$

The M/M/1-PS queue (Cont'd)

so by the latter and $E[D|x] = cx$, we obtain

$$E[D|x] = \frac{x}{1-\rho}.$$

Comparing the two approaches PS and FIFO, PS is better for small messages and FIFO is better for large messages. Normally, the customer / message does not choose the system.

PS is used to avoid a system that allows jobs or customers that have very large service demands to cause extreme congestion and large delay for small messages.

The M/M/1-PS queue (Cont'd)

Although the queue-size steady-state probabilities for M/M/1-PS and for M/M/1-FIFO are the same, which implies that $E[Q]$ and $E[D]$ (by Little's formula) for both are also equal, the delay variances and distributions for the two queues are different. **The variance of the delay** for M/M/1-PS queue is given by (see references in the book)

$$Var[D] = \frac{1}{(\mu-\lambda)^2}\left(\frac{2+\rho}{2-\rho}\right) = \frac{1}{\mu^2(1-\rho)^2}\left(\frac{2+\rho}{2-\rho}\right).$$

Comparing this with M/M/1 delay variance, we observe that the variance of the delay for the M/M/1-PS queue is larger than that for M/M/1-FIFO, for $1 > \rho > 0$, by the factor of

$$\left(\frac{2+\rho}{2-\rho}\right),$$

which increases from 1 to 3 as ρ increases from 0 to 1.

The M/M/1-PS queue (Cont'd)

Intuitive Explanation

Intuitively, under PS, the delay of the very long messages (or customers that require very long service time) is longer that under FIFO. This is because under PS, a very long message not only needs to wait until the messages that it finds in the system at the time of its arrival complete their service, which is the case under FIFO; it also has to wait until many of the messages that arrive *after* it arrives complete their service.

This effect is more pronounced as ρ increases.

Since the range of message delay values is between nearly zero (experienced by very short messages that arrive at an empty queue) and the longest delays experienced by the longest messages, the variance of the delay under M/M/1-PS is longer than under M/M/1-FIFO.

Insensitivity

One important property of a processor sharing queue is that the queue size distribution, and therefore, the mean number of customers in the system $E[Q]$, the mean delay of a customer $E[D]$, and the mean delay of a customer with service requirement x, $E[D(x)]$, are insensitive to the shape of the distribution of the service-time requirements of the customers. In other words, these results apply also to the M/G/1-PS model.

However, the insensitivity property does not apply to the delay distribution.

Finally, notice the similarity between the M/G/1-PS and the M/G/∞ models. They are both insensitive to the shape of the distribution of the service time requirement in terms of the queue size distribution, mean delay and mean number of customers in the system, but not in terms of the delay distribution.

Chapter 16: M/G/1 and Extensions

© Moshe Zukerman

April 15, 2022

M/G/1 Preliminaries

- M/G/1 is a single server queue (SSQ)
- the arrival process follows a Poisson process with parameter λ
- service times are assumed to be IID and with mean $E[S] = 1/\mu$, and standard deviation σ_S and are independent of the arrival process.

As M/G/1 is a special case of G/G/1, all the results that are applicable to G/G/1 are also applicable to M/G/1. For example,

- $\hat{U} = \rho = \lambda E[S] = \lambda/\mu$.
- The probability that there is no-one in the system is $\pi_0 = 1 - \rho$
- Little's formula.

M/M/1 is a special case of M/G/1. In M/G/1 the service times are no longer exponentially distributed. This introduces significant complexity as we can no longer use the Markov chain model.

M/D/1 is another special case of M/G/1 for the case where the service times are deterministic.

Pollaczek Khinchine (PK) Formula: Residual Service Approach (Bertsekas and Gallager, 1992)

The waiting time in the queue of an arriving customer to an M/G/1 queue is the remaining service time of the customer in service plus the sum of the service times of all the customers in the queue ahead of the arriving customer.

Therefore, the mean waiting time in the queue is given by

$$E[W_Q] = E[R] + \frac{E[N_Q]}{\mu},$$

where $E[R]$ denotes the mean (unconditional) residual service time.

Question: What is $E[R]$ for M/M/1?

PK Formula: Residual Service Approach (Cont'd)

Answer:
For M/M/1,

$$E[R] = \frac{\rho}{\mu}$$

,
Why?

Define the following two events:

Event $A =$ There is one customer in service.

Event $B =$ There is no customer in service (the system is empty).

By the law of iterated expectation,

$$E[R] = P(A) \times E[R \mid A] + P(B) \times E[R \mid B] = \rho \times \frac{1}{\mu} + (1 - \rho) \times 0 = \frac{\rho}{\mu}.$$

PK Formula: Residual Service Approach (Cont'd)

By Little's formula

$$E[N_Q] = \lambda E[W_Q]$$

and

$$E[W_Q] = E[R] + \frac{E[N_Q]}{\mu},$$

we obtain

$$E[W_Q] = \frac{E[R]}{1-\rho}.$$

Now all that remains is to obtain $E[R]$.

PK Formula: Residual Service Approach (Cont'd)

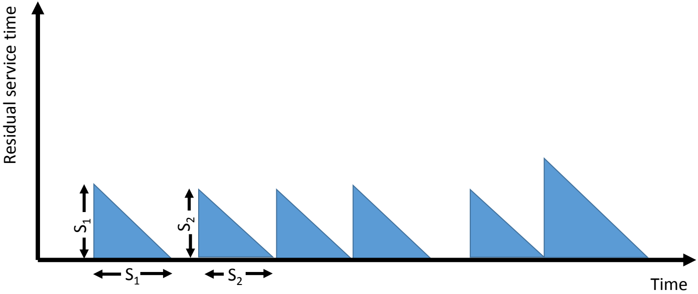


Figure 1: $R(t)$ – the residual service time as a function of time.

$$E[R] = \frac{1}{T}\int_0^T R(t)dt = \frac{1}{T}\sum_{i=1}^{S(T)} \frac{1}{2}S_i^2 = \frac{1}{2}\frac{S(T)}{T}\frac{1}{S(T)}\sum_{i=1}^{S(T)} S_i^2.$$

PK Formula: Residual Service Approach (Cont'd)

Let $T \to \infty$, we obtain

$$E[R] = \frac{1}{2}\lambda\overline{S^2},$$

where $\overline{S^2}$ is the second moment of the service time.
Thus,

$$E[W_Q] = \frac{\lambda\overline{S^2}}{2(1-\rho)}$$

and

$$E[D] = \frac{\lambda\overline{S^2}}{2(1-\rho)} + 1/\mu.$$

PK Formula: Residual Service Approach (Cont'd)

Recalling that $\sigma_s^2 = \overline{S^2} - (1/\mu)^2$, Eq. the latter equation leads to the well known Pollaczek Khinchine formula for the mean delay in an M/G/1 system:

$$E[D] = \frac{\lambda(\sigma_s^2 + \mu^{-2})}{2(1-\rho)} + \frac{1}{\mu} = \frac{\rho + \lambda\mu\sigma_s^2}{2(\mu - \lambda)} + \frac{1}{\mu}.$$

Using Little's formula, we obtain the Pollaczek Khinchine formula for the mean number of customers in an M/G/1 system:

$$E[Q] = \rho + \frac{\rho^2 + \lambda^2\sigma_s^2}{2(1-\rho)}.$$

Observe that according to the PK formula, if we have two M/G/1 queueing systems, where they both have the same arrival and service rates, but for one the variance of the service time is higher than that of the other, the one with the higher variance will experience higher mean queue size and delay.

Special Cases: M/M/1 and M/D/1

Now let us consider the special case of exponential service time – i.e., the M/M/1 case. To obtain $E[Q]$ for M/M/1, we substitute $\sigma_s^2 = 1/\mu^2$ in the PK formula, and after some algebra, we obtain

$$E[Q] = \frac{\rho}{1-\rho}$$

which is consistent with what we have learnt on M/M/1.
Another interesting case is the M/D/1 queue in which case we have: $\sigma_s^2 = 0$. Substituting the latter in the PK formula, we obtain

$$E[Q] = \frac{\rho}{1-\rho} \times \frac{2-\rho}{2}.$$

Because the second factor, namely $(2-\rho)/2$, we see that $E[Q]$ for M/M/1 is higher than $E[Q]$ for M/D/1, with the same arrival and service rates, by a factor in the range 1 to 2, and this factor monotonically decreases with ρ.

Mean Busy Period of M/G/1

We have defined and discussed the concept of busy period in the context of the M/M/1 queue. The same analysis applies to M/G/1, and we obtain:

$$E[T_B] = \frac{1}{\mu - \lambda}.$$

What we learn from this is that the mean busy period is insensitive to the shape of the service time distribution. In other words, the mean busy periods of M/M/1 and M/G/1 systems are the same if the mean arrival and service rates are the same.

M/G/1-LIFO

Unlike the M/G/1-FIFO, but similarly to M/G/1-PS, the M/G/1-LIFO is insensitive to the shape of the service time distribution.

We already know that the queue size process of M/M/1 is the same as that of its M/M/1-LIFO equivalence. Therefore, they also have the same mean queue size and delay.

Due to the insensitivity of M/G/1-LIFO, the M/M/1 results for $E[Q]$, $E[D]$ and queue size distribution are applicable also to M/G/1-LIFO.

M/G/1-LIFO (Cont'd)

Specifically, if we are given an M/G/1-LIFO queue with arrival rate λ and mean service time $1/\mu$, denote $\rho = \lambda/\mu$, then the queue size distribution is given by:

$$\pi_i = \rho^i(1-\rho) \text{ for } i = 0,\ 1,\ 2,\ \ldots\ .$$

The mean queue size is given by

$$E[Q] = \frac{\rho}{1-\rho}$$

and the mean delay is given by

$$E[D] = \frac{1}{\mu - \lambda}.$$

M/G/1-LIFO (Cont'd)

To show why M/G/1-LIFO queue is insensitive and satisfies the above equations, notice that an arriving customer that upon its arrival finds i customers in the system will be served only during time when the system is in state $i+1$.

Furthermore, all the customers served when the system is in state $i+1$ will be customers that have arrived when the system is in state i.

Therefore, the time the system spends in the state $i+1$ comprises exactly the service times of the customers that arrive when the system in state i.

M/G/1-LIFO (Cont'd)

Now consider a long interval of time T. As we denote by π_i the proportion of time that the system is in state i, then during a long time interval T, the mean number of arrivals in state i is $\lambda \pi_i T$ and their total service time is equal to $\lambda \pi_i T(1/\mu) = \rho \pi_i T$.

Accordingly, the proportion of time the system is in state $i+1$ is given by

$$\pi_{i+1} = \frac{\rho \pi_i T}{T} = \rho \pi_i .$$

Since the latter holds for $i = 0, 1, 2, \ldots$, then we observe that the queue-size distribution of M/G/1-LIFO obeys the steady-state equations of M/M/1 **regardless of the shape of the service time distribution**.

M/G/1 with m priority classes

Let us consider an M/G/1 queueing system with m priority classes.

Let λ_j and μ_j be the arrival and service rate of customers belonging to the jth priority class for $j = 1, 2, 3, \ldots, m$.

The mean service time of customers belonging to the jth priority class is therefore equal to $1/\mu_j$.

The second moment of the service time of customers belonging to the jth priority class is denoted $\overline{S^2(j)}$.

We assume that priority class j has higher priority that priority class $j + 1$, so Class 1 represents the highest priority class and Class m the lowest.

M/G/1 with m priority classes (Cont'd)

For each class j, the arrival process is assumed to be Poisson with parameter λ_j, and the service times are assume mutually independent and independent of any other service times of customers belonging to the other classes, and are also independent of any inter-arrival times.

Let $\rho_j = \lambda_j/\mu_j$.

We assume that $\sum_{j=1}^{m} \rho_j < 1$.

We will consider two disciplines: *nonpreemptive* and *preemptive resume*.

M/G/1 with m nonpreemptive priority classes

Under this regime, a customer in service will complete its service even if a customer of a higher priority class arrive while it is being served.

Let $E[N_Q(j)]$ and $E[W_Q(j)]$ represent the mean number of class j customers in the queue excluding the customer in service and the mean waiting time of a class j customer in the queue (excluding its service time), respectively.

Further let R be the residual service time (of all customers of all priority classes). As we derived $E[R]$ for M/G/1, we obtain

$$E[R] = \frac{1}{2}\sum_{j=1}^{m} \lambda_j \overline{S^2(j)}.$$

Homework: Derive this equation.

M/G/1 with m nonpreemptive priority classes (Cont'd)

As in M/G/1, we have for the highest priority,

$$E[W_Q(1)] = E[R] + \frac{E[N_Q(1)]}{\mu_1}$$

and similar to M/G/1, we obtain

$$E[W_Q(1)] = \frac{E[R]}{1 - \rho_1}.$$

Regarding the second priority, $E[W_Q(2)]$ is the sum of the mean residual service time $E[R]$, the mean time it takes to serve the Class 1 customers in the queue $E[N_Q(1)]/\mu_1$, the mean time it takes to serve the Class 2 customers in the queue $E[N_Q(2)]/\mu_2$, and the mean time it takes to serve all the Class 1 customers that arrives during the waiting time in the queue for the Class 2 customer $E[W_Q(2)]\lambda_1/\mu_1 = E[W_Q(2)]\rho_1$.

M/G/1 with m nonpreemptive priority classes (Cont'd)

Putting it together

$$E[W_Q(2)] = E[R] + \frac{E[N_Q(1)]}{\mu_1} + \frac{E[N_Q(2)]}{\mu_2} + E[W_Q(2)]\rho_1.$$

By the latter and Little's formula for Class 2 customers, namely,

$$E[N_Q(2)] = \lambda_2 E[W_Q(2)],$$

we obtain

$$E[W_Q(2)] = \frac{E[R] + \rho_1 E[W_Q(1)]}{1 - \rho_1 - \rho_2}.$$

These lead to

$$E[W_Q(2)] = \frac{E[R]}{(1 - \rho_1)(1 - \rho_1 - \rho_2)}.$$

M/G/1 with *m* nonpreemptive priority classes (Cont'd)

Also (show it as a homework)

$$E[W_Q(3)] = \frac{E[R]}{(1-\rho_1-\rho_2)(1-\rho_1-\rho_2-\rho_3)},$$

and in general (show it as a homework)

$$E[W_Q(j)] = \frac{E[R]}{(1-\sum_{i=1}^{j-1}\rho_i)(1-\sum_{i=1}^{j}\rho_i)}.$$

The mean delay for a jth priority class customer, denoted $E[D(j)]$, is given by

$$E[D(j)] = E[W_Q(j)] + \frac{1}{\mu_j} \text{ for } j = 1, 2, 3, \ldots, m.$$

M/G/1 with m preemptive resume priority classes

In this case, an arriving customer of priority j never waits for a customer of a lower priority class (of Class i for $i > j$) to complete its service.

Therefore, when we derive the delay of a customer of priority j, we can ignore all customers of class i for all $i > j$.

Accordingly, the mean delay of a priority j customer satisfies the following equation

$$E[D(j)] = \frac{1}{\mu_j} + \frac{R(j)}{1 - \sum_{i=1}^{j} \rho_i} + E[D(j)] \sum_{i=1}^{j-1} \rho_i$$

where $R(j)$ is the mean residual time of all customers of classes $i = 1, 2, \ldots, j$ given by

$$R(j) = \frac{1}{2} \sum_{i=1}^{j} \lambda_i \overline{S^2(i)}.$$

M/G/1 with m preemptive resume priority classes (Cont'd)

The first term of the equation for $E[D(j)]$ is simply the mean service time of a jth priority customer.

The second term is the mean time it takes to clear all the customers of priority j or higher that are already in the system when a customer of Class j arrives.

It is merely the equation that gives the mean time of waiting in the queue in an M/G/1 queueing system where we replace ρ by $\sum_{i=1}^{j} \rho_i$ which is the total traffic load offered by customers of priority j or higher.

From the point of view of the jth priority customer, the order of the customers ahead of it will not affect its mean delay, so we can "mix" all these customers up and consider the system as M/G/1.

M/G/1 with *m* preemptive resume priority classes (Cont'd)

The last term of the equation for $E[D(j)]$ is the mean total work introduced to the system by customers of priorities higher than j that arrive during the delay time of our j priority customer.

Notice that we use the ρ_is there because $\rho_i = \lambda_i(1/\mu_i)$ representing the product of the mean rate of customer arrivals and the mean work they bring to the system for each priority class i.

The equation for $E[D(j)]$ leads to

$$E[D(1)] = \frac{(1/\mu_1)(1-\rho_1) + R(1)}{1-\rho_1},$$

and

$$E[D(j)] = \frac{(1/\mu_j)\left(1-\sum_{i=1}^{j}\rho_i\right) + R(j)}{\left(1-\sum_{i=1}^{j-1}\rho_i\right)\left(1-\sum_{i=1}^{j}\rho_i\right)}.$$

M/G/1 with *m* preemptive resume priority classes (Cont'd)

Do the homework problems in the book and observe the following from the solution of last homework problem.

For the M/M/1 with priorities model, if the queues of all priorities are stable, and if the service rate is arbitrarily high, then the mean delay is arbitrarily low regardless of the utilization. **Then in such a case, there is no much benefit in implementing priorities.**

However, if for example, $\rho_1 + \rho_2 > 1$ but $\rho_1 < 1$, then priority 1 customers clearly benefit from having priority even if the service rate (and also arrival rate) is arbitrarily large.

Notice that we have observed similar results for M/M/1 without priorities.

Also notice that we consider here a scenario where the service rate is arbitrarily high and the utilization is fixed which means that the arrival rate is also arbitrarily high.

Section 19.1: Jackson Networks

© Moshe Zukerman

April 12, 2024

Preliminaries

Consider an open network of queues where every node has an infinite buffer queue with one or more servers.

Customers are arriving from the outside of the network to any of the nodes to obtain service and then may move to other nodes for further service until the service is completed and they leave the network.

This is equivalent to messages that are generated in one node and are transmitted to their destinations through other nodes (then the service is the transmission).

The Output Process

An important issue for such a queueing network is the statistical characteristics of the output of such queues because in queueing networks, output of one queue may be the input of another.

Burke's Theorem states that, in steady-state, the output (departure) process of M/M/1, M/M/k or M/M/∞ queue follows a Poisson process.

Because no traffic is lost in such queues, the arrival rate must be equal to the departure rate, then any M/M/1, M/M/k, or M/M/∞ queue with arrival rate of λ will have a Poisson departure process with rate λ in steady-state.

A Model of Two Queues in Series

Consider an example of a simple queueing network made of two identical single-server queues in series.

The output of the first queue is the input of the second queue.

All the customers that complete service at the second queue leave the system.

Assume that the customers that arrive into the first queue follow a Poisson process with parameter λ.

The service times required by each of the arriving customers at the two queues are independent and exponentially distributed with parameter μ.

The Two Queue Model and Acyclic Networks

Under this two queue model we have

- The amount of time a customer requires in the first queue is independent of the amount of time a customer requires in the second queue.
- the service times are both independent of the arrival process into the first queue.
- Since the output process of the first queue is Poisson with parameter λ, and since the first queue is clearly an M/M/1 queue, we have here two identical M/M/1 queues in series.

This is an example of a network of queues where Burke's theorem leads immediately to a solution for queue size and waiting time statistics. A class of networks that can be easily analyzed this way is the class of the so-called **acyclic networks**. These networks are characterized by the fact that a customer never goes to the same queue twice for service.

Networks That Are Not Acyclic

If the network is not acyclic, the independence between inter arrival times and between inter arrival and service times do not hold any longer.

This means that the queues are no longer Markovians.

To illustrate this let us consider a single server queue with feedback described as follows.

Normally, a single node does not constitute a network, however, this simple single queue example is sufficient to illustrate the feedback effect and related dependencies.

Single Queue with Feedback

Customers arrive into the system from the outside according to a Poisson process with parameter λ and the service time is exponentially distributed with parameter μ.

When the customer completes the service, the customer returns to the end of the queue with probability p, and with probability $(1-p)$, the customer leaves the system.

Now assume that λ is very small and μ is very large.

Also assume that $p > 0.99$.

This results in an arrival process which is based on very infrequent original arrivals (from the outside) each of which brings with it a burst of many feedback arrivals that are very close to each other.

Clearly, this is not a Poisson process.

Single Queue with Feedback (Cont'd)

Furthermore, the inter-arrivals of packets within a burst, most of which are feedback from the queue output, are very much dependent on the service times, so clearly we have dependence between inter-arrival times and service times.

Nevertheless, the so-called Jackson's Theorem extends the simple result applicable to an acyclic network of queues to networks that are not acyclic.

In other words, although the queues are not M/M/1 (or M/M/k or M/M/∞), they behave in terms of their queue-size statistics as if they are.

Jackson's Theorem

Consider a network of N single-server queues with infinite buffer in steady-state.

The Jackson theorem also applies to multi-server queues, but let us consider single-server queues for now.

For queue i, $i = 1, 2, 3, \ \ldots, N$, the arrival process from the outside is Poisson with rate r_i. We allow for $r_i = 0$ for some queues, but there must be at least one queue j, such that $r_j > 0$.

Once a customer completes its service in queue i, it continues to queue j with probability P_{ij}, $i = 1, 2, 3, \ \ldots, N$, or leaves the system with probability $1 - \sum_{j=1}^{N} P_{ij}$.

Notice that we allow for $P_{ii} > 0$ for some queues. That is, we allow for positive probability for customers to return to the same queue they just exited.

Jackson's Theorem (Cont'd)

Let λ_j be the total arrival rate into queue j.

These λ_j values can be computed by solving the following set of equations.

$$\lambda_j = r_j + \sum_{i=1}^{N} \lambda_i P_{ij}, \qquad j = 1, 2, 3, \ \ldots, N.$$

The above set of equations can be solved uniquely, if every customer eventually leaves the network.

This means that the routing probabilities P_{ij} must be such that there is a sequence of positive routing probabilities and a final exit probability that create an exit path of positive probability from each node.

Jackson's Theorem (Cont'd)

The service times at the jth queue are assumed exponentially distributed with parameter μ_j.

They are assumed to be mutually independent and also independent of the arrival process at that queue. Let ρ_j be defined by

$$\rho_j = \frac{\lambda_j}{\mu_j} \qquad \text{for } j = 1, 2, 3, \ \ldots, N.$$

Assume that

$$0 \leq \rho_j < 1 \qquad \text{for } j = 1, 2, 3, \ \ldots, N.$$

Before we proceeds with the derivation of the queueing performance results, let us summarize the conditions of Jackson Theorem which will henceforth be called *Jackson's Assumptions*.

Jackson's Assumptions

1. The N-node network, where each node has infinite buffer queue, is open (meaning that traffic can come from outside the network and goes out from the network) and any external arrivals (arrivals from outside the network) to node j follow a Poisson process with rate r_j. We allow some the external arrival rates to be zero, but we require that at least one of them to one queue must be positive.
2. All service times are exponentially distributed random variables that are independent of the arriving packets to queues and their service times in previous queues.
3. The service discipline at all queues is FIFO.
4. A customer completing service at queue i will either move to queue j with routing probability P_{ij} (note that i can be equal to j, so we have a self loop), or leave the system with probability $1 - \sum_{j=1}^{N} P_{ij}$ which is positive for some queues.
5. All the queues must be stable - namely,
   $0 \leq \rho_j < 1 \qquad \text{for } j = 1, 2, 3, \ \ldots, N.$

Jackson's Theorem – Key Result

Let Q_j be the queue-size of queue j. Then, based on Jackson's assumptions, according to Jackson's Theorem, in steady-state, we have that

$$P(Q_1 = k_1, Q_2 = k_2, \ \ldots, \ Q_N = k_N) = P(k_1)P(k_2)P(k_3) \cdot \ \ \ldots \ \ \cdot P(k_N),$$

where $P(k_i) = \rho_i^{k_i}(1-\rho_i), \qquad \text{for } i = 1, 2, 3, \ \ldots, N.$

Comments on Jackson's Theorem

Although Jackson theorem assumes that the arrival processes from the outside follow Poisson processes, it does not assume that the input into every queue follows a Poisson processes.

Therefore, it does not assume that the queues are independent M/M/1 (or M/M/k or M/M/∞) queues.

However, it turns out, according to Jackson theorem that the joint steady-state probability distribution of the queue sizes of the N queues is obtained as if the queues are independent M/M/1 (or M/M/k or M/M/∞) queues.

This result applies despite the fact that the network is not acyclic in which case we have demonstrated that the queues do not have to be M/M/1 queues.

Implications of Jackson's Theorem

Accordingly, the mean queue-size of the jth queue is given by

$$E[Q_j] = \frac{\rho_j}{1-\rho_j}.$$

The mean delay of a customer in the jth queue $E[D_j]$ defined as the time from the moment the customer joins the queue until it completes service, can be obtain by Little's formula as follows.

$$E[D_j] = \frac{E[Q_j]}{\lambda_j}.$$

Using Little's formula, by considering the entire queueing network as our system, we can also derive the mean delay of an arbitrary customer $E[D]$:

$$E[D] = \frac{\sum_{j=1}^{N} E[Q_j]}{\sum_{j=1}^{N} r_j}.$$

Check for the Single Queue with Feedback Model

We now confirm the result of Jackson's theorem for the case of a single queue with feedback by deriving its queueing statistics use Jacson's theorem and an alternative way.

In the first way, we use Jackson's Theorem, so we write

$$\lambda_1 = \lambda + p\lambda_1$$

and we obtain

$$\lambda_1 = \frac{\lambda}{(1-p)}.$$

Thus,

$$\rho = \frac{\lambda}{(1-p)\mu}.$$

Check for the Single Queue with Feedback Model (Cont'd)

In the second way, we consider an equivalent system with longer service durations, we let the feedback arrivals have preemptive resume priority over all other arrivals. This priority regime will not change the queue size statistics. Now we have that the service time comprises a geometric sum of exponential random variables which is also exponential.

Accordingly, the longer service durations are exponentially distributed with mean service time

$$\frac{1}{\mu} \times \frac{1}{(1-p)} = \frac{1}{\mu(1-p)}.$$

Thus,

$$\rho = \frac{\lambda}{(1-p)\mu}.$$

This is exactly the same result for ρ that we have obtained using Jackson's theorem. Since in M/M/1, the queue size distribution, and therefore $E[Q]$ are functions of only ρ, then they are equal for both cases. Then, the mean delay is obtained according to Little's formula by the ratio $E[Q]/\lambda$, so the mean delay is also the same for both cases.

Example of Two Queues with Feedback

Let us now consider a network of two-queue in series where all the traffic that completes service in queue 1 enters queue 2 and some of the traffic in queue 2 leaves the system while the rest enters queue 1.

This example is similar to the above mentioned example of a single queue with feedback.

Using our notation, let the arrivals from the outside follow Poisson processes with rates $r_1 = 10^{-8}$ and $r_2 = 0$ and let $\mu_1 = \mu_2 = 1$.

Further assume that the probability that a customer that completes service in queue 2 leaves the system is 10^{-3}, so it enters queue 1 with probability $1 - 10^{-3}$.

Example of Two Queues with Feedback (cont'd)

Accordingly,

$$\lambda_1 = r_1 + (1 - 10^{-3})\lambda_2$$

and

$$\lambda_2 = \lambda_1.$$

Thus,

$$\lambda_1 = 10^{-8} + (1 - 10^{-3})\lambda_1,$$

so

$$\lambda_1 = \lambda_2 = 10^{-5}$$

and

$$\rho_1 = \rho_2 = 10^{-5},$$

so

$$E[Q_1] = E[Q_2] = \frac{10^{-5}}{1 - 10^{-5}} \approx 10^{-5} + 10^{-10}$$

and

$$E[D_1] = E[D_2] \approx \frac{10^{-5} + 10^{-10}}{10^{-5}} = 1 + 10^{-5}.$$

Sanity Check for the Two Queues with Feedback Results

Recalling that the mean service time is equal to one, this means that small queueing delay is expected.

This result makes sense intuitively.

Although the feedbacked traffic is more bursty than Poisson, but most of the traffic here comprises the same packet that returns over and over again and it is impossible for the same packet to wait in the queue for itself to be served.

Because the arrival rate from the outside is very low, in rare occasion of a new arrival from the outside there may be a minor queueing delay incurred.

Extensions Beyond Jackson

For Jackson network, namely, an open network of M/M/1, M/M/k or M/M/∞ queues described above, an exact solution is available.

However, in most practical cases, especially when we have to deal with the so-called loss networks that comprise queues, such as M/M/k/k, where traffic is lost, we have to make additional modelling assumptions and to rely on approximations to evaluate performance measures, such as blocking probability, or carried traffic.

One approximation is the so-called Reduced-Load Erlang Fixed-Point Approximation which is reasonably accurate and useful for loss networks.

Section 19.2: Computation of Blocking Probability in Circuit Switched Networks by the Erlang Fixed-Point Approximation (EFPA)

March 29, 2023

Preliminaries

Consider a circuit switched network made of nodes (switching centers) that are connected by links.

Each link has a fixed number of circuits.

In order to make a call between two nodes: source and destination, a user should reserve a free circuit in each consecutive link of a path between the two nodes.

Such reservation is successful if and only if there exists a free circuit on each of the links of that path.

An important characteristic of a circuit-switched network is that once a user makes a reservation for a connection between a source and a destination the capacity for this connection is exclusively available for the user of this connection and no other users can utilize this capacity for the entire duration of this connection holding time.

EFPA

To evaluate the probability that a circuit reservation is blocked, we first make the following simplifying **EFPA assumptions**.

1. All the links are independent.
2. The arrival process of calls for each origin-destination pair is Poisson.
3. The arrival process seen by each link is Poisson.

Having made these assumptions, we now consider each link as an independent M/M/k/k system for which the blocking probability is readily available by the Erlang B formula. The assumption that the number of circuits on each link is the same (equal to k) is made only for simplicity of exposition, but it can be easily generalized, so it is not one of the three key EFPA assumptions.

a_j = total offered load to link j from all the routes that pass through link j.

Then the blocking probability on link j is obtained by

$$B_j = E_k(a_j).$$

EFPA (Cont'd)

Now that we have means to obtain the blocking probability on each link, we can approximate the blocking probability of a call made on a given route.

The route R is an ordered set of links.

$B(R) =$ blocking probability of a call made on route R.

$L_R =$ the set of links in route R.

Accordingly, the route blocking probability is given by

$$B(R) = 1 - \prod_{i \in L_R} (1 - B_i).$$

EFPA (Cont'd)

$A(R)$ = offered traffic on route R.

$a_j(R)$ = total traffic offered to link j from traffic that flow on route R.

Then, $a_j(R)$ can be computed by deducting from $A(R)$ the traffic lost due to congestion on links other than j. That is,

$$a_j(R) = A(R) \prod_{i \in L_R;\ i \neq j} (1 - B_i) \ \ \text{for } j \in L_R,$$

and $a_j(R) = 0$ if j is not in L_R.

This consideration to the reduced load due to blocking on other links gave rise to the name *reduced load approximation* to this procedure.

EFPA (Cont'd)

$\mathcal{R}$ = set of all routes.

Then the total offered traffic on link j is obtained by

$$a_j = \sum_{R \in \mathcal{R}} a_j(R).$$

These give a set of nonlinear equations that requires a fixed-point solution. Notice that Erlang B is non-linear.

To solve these equations, we start with an initial vector of B_j values; for example, set $B_j = 0$ for all j.

As we know the $A(R)$s, use the equation for $a_j(R)$ to obtain the $a_j(R)$ values.

EFPA (Cont'd)

Next, use the equation for a_j to obtain the a_j values, which can be substituted in the equation for B_j to obtain a new set of values for the blocking probabilities.

Then, the process repeats itself iteratively until the blocking probability values obtained in one iteration is sufficiently close to those obtained in the previous iteration.

Finally, having the resulted B_is, the $B(R)$ values for each route R are obtained.

EFPA (Cont'd)

The above solution based on the principles of the Reduced-Load and Erlang Fixed-Point Approximations can be applied to many systems and networks.

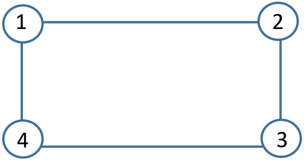
We have discussed an approach to evaluate blocking probability for circuit switched networks under the so-called **fixed routing** regime, where a call is offered to a route, and if it is rejected, it is lost and cleared from the system.

There are, however, various other regimes involving **alternate routing**, where rejected calls offered to a given route can overflow to other routes.

A similar Erlang fixed-point approximation can be used for circuit switching with alternate routing, but this is beyond the scope of this course.

Homework

Consider the 4-node ring network shown below.



Each of the four links has $k = 20$ circuits.

The traffic demands are as follows.

On Route: 4 → 1 → 2, the offered traffic is $A(4,1,2) = 10$ erlangs.
On Route: 4 → 1, the offered traffic is $A(4,1) = 11$ erlangs.
On Route: 1 → 2, the offered traffic is $A(1,2) = 9$ erlangs.
On Route: 2 → 3 → 4, the offered traffic is $A(2,3,4) = 9$ erlangs.
On Route: 2 → 3, the offered traffic is $A(2,3) = 12$ erlangs.
On Route: 3 → 4, the offered traffic is $A(3,4) = 11$ erlangs.

Find the blocking probability on routes 4 → 1 → 2, 4 → 1, 2 → 3, and 2 → 3 → 4.

Guide

Observe that the traffic on the first three routes is independent of the traffic on the last three routes. The two sets of routes use different links, so they can be treated separately as two separate networks.
We use the notation $[i, j]$ to denote a link between the end-nodes i and j.

We initially assume $B_{[i,j]} = 0$ for all four links [1,2], [2,3], [3,4], and [4,1].

The First Three Routes

Let's focus first on the traffic on the first three routes that only use only the links [4,1] and [1,2].

Because traffic in the routes (4,1) and (1,2) use a single link, for both, there is no loss due to blocking on other links. Therefore, we always have:

$a_{[4,1]}(4, 1) = 11$ erlangs and $a_{[1,2]}(1, 2) = 9$ erlangs.

Guide (cont'd)

However, for the route (4,1,2) this is not the case, and in general, we will need to take account of the reduced load. Only in the first step where all the B_js are equal to zero, we have
$a_{[4,1]}(4,1,2) = a_{[1,2]}(4,1,2) = 10$ erlangs.

Next, we update the blocking probabilities in links [4,1] and [1,2] and obtain
$B_{[4,1]} = E_{20}(11+10) = E_{20}(21) = 0.184111065$
$B_{[1,2]} = E_{20}(9+10) = E_{20}(19) = 0.133761441$

Now the reduced load is taken into account for the route (4,1,2), and we obtain:
$a_{[4,1]}(4,1,2) = 10 \times (1-0.133761441) = 8.662385587$
$a_{[1,2]}(4,1,2) = 10 \times (1-0.184111065) = 8.158889352$

Updating again the blocking probabilities in links [4,1] and [1,2], we obtain
$B_{[4,1]} = E_{20}(11+8.662385587) = E_{20}(19.66238559) = 0.150375622$
$B_{[1,2]} = E_{20}(9+8.158889352) = E_{20}(17.15888935) = 0.089462624$

Guide (cont'd)

Updating the loads, we obtain

$a_{[4,1]}(4,1,2) = 10 \times (1 - 0.089462624) = 9.105373758$

$a_{[1,2]}(4,1,2) = 10 \times (1 - 0.150375622) = 8.496243779$

Updating the blocking probabilities, we obtain

$B_{[4,1]} = E_{20}(11 + 9.105373758) = E_{20}(20.105373758) = 0.161552648$

$B_{[1,2]} = E_{20}(9 + 9.105373758) = E_{20}(18.10537376) = 0.111754835$

Updating the loads, we obtain

$a_{[4,1]}(4,1,2) = 10 \times (1 - 0.111754835) = 8.882451648$

$a_{[1,2]}(4,1,2) = 10 \times (1 - 0.161552648) = 8.384473517$

Updating the blocking probabilities, we obtain

$B_{[4,1]} = E_{20}(11 + 8.882451648) = E_{20}(19.882451648) = 0.155924918$

$B_{[1,2]} = E_{20}(9 + 8.384473517) = E_{20}(17.384473517) = 0.094653631$

Updating the loads, we obtain

$a_{[4,1]}(4,1,2) = 10 \times (1 - 0.094653631) = 9.053463687$

$a_{[1,2]}(4,1,2) = 10 \times (1 - 0.155924918) = 8.440750821$

Guide (cont'd)

Updating the blocking probabilities, we obtain
$B_{[4,1]} = E_{20}(11 + 9.053463687) = E_{20}(20.053463687) = 0.160241842$
$B_{[1,2]} = E_{20}(9 + 8.440750821) = E_{20}(17.440750821) = 0.095961813$
Updating the loads, we obtain
$a_{[4,1]}(4, 1, 2) = 10 \times (1 - 0.095961813) = 9.040381865$
$a_{[1,2]}(4, 1, 2) = 10 \times (1 - 0.160241842) = 8.397581581$
Updating the blocking probabilities, we obtain
$B_{[4,1]} = E_{20}(11 + 9.040381865) = E_{20}(20.040381865) = 0.159911528$
$B_{[1,2]} = E_{20}(9 + 8.397581581) = E_{20}(17.397581581) = 0.094957879$
Updating the loads, we obtain
$a_{[4,1]}(4, 1, 2) = 10 \times (1 - 0.094957879) = 9.050421212$
$a_{[1,2]}(4, 1, 2) = 10 \times (1 - 0.159911528) = 8.400884722$
Updating the blocking probabilities, we obtain
$B_{[4,1]} = E_{20}(11 + 9.050421212) = E_{20}(20.050421212) = 0.160165019$
$B_{[1,2]} = E_{20}(9 + 8.400884722) = E_{20}(17.400884722) = 0.095034591$

Guide (cont'd)

Reasonable accuracy has been achieved for $B_{[4,1]}$ and $B_{[1,2]}$.

To approximate the route blocking probabilities for routes (4,1) and (1,2), they are $B_{[4,1]}$ and $B_{[1,2]}$, respectively.

For route (4,1,2), the blocking probability approximation is $1-(1-B_{[4,1]})(1-B_{[1,2]})$.

The Remaining three Routes

Now repeat the procedure to approximate the blocking probability for the routes (2,3), (3,4) and (2,3,4).

Can you try writing a computer program to do it?

Analysis and Simulations of Cellular Mobile Networks

(It includes material on mobile network analysis from slides of the course ”Mobile Data Networks” by S. Chan, and on mobile network simulations from Section 19.3 of the book “Introduction to Queueing Theory and Stochastic Teletraffic Models” by M. Zukerman.)

© Sammy Chan and Moshe Zukerman

April 8, 2023

Performance Modeling Without Mobility

Assume that cell i of a cellular network has f_i channels and the traffic offered to it is given by A_i Erlangs, and follows a Poisson process for the call arrivals, with general distribution of call holding time. If there is no mobility of calls, then each cell operates independently, and the call blocking probability in cell i, denoted B_i, can be obtained by the Erlang B formula as follows:

$$B_i = E_{f_i}(A_i) = \frac{A_i^{f_i}/f_i!}{\sum_{j=0}^{f_i} A_i^j/j!}.$$

As mentioned, if f_i is too large, the Erlang B recursion can be used.

Performance Modeling with mobility

When mobility is considered, the interaction between neighbouring cells needs to be taken into account.

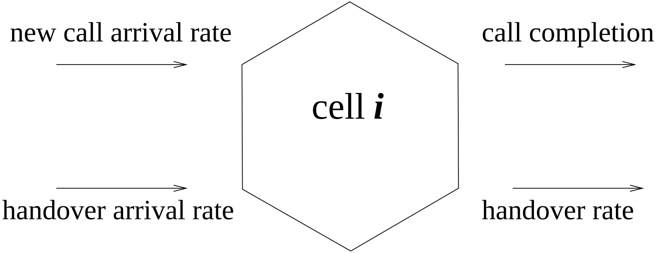


Figure 1: Traffic rates for handoff analysis.

Performance Modeling with Mobility (Cont'd)

Channel holding time

Define the following random variables:

T_μ = the duration of a new call = call holding time
T_δ = the time for which a mobile resides in a cell (cell dwelling time)
T_h = the channel holding time

A channel becomes free (is released) at cell i because of either:

1. the call has been completed normally, or
2. it is handed over to another cell.

$$T_h = \min[T_\mu, T_\delta].$$

Channel holding time (Cont'd)

Assuming that both T_μ and T_δ are exponentially distributed with parameters μ and δ, respectively. Then,

$$E[T_\mu] = \frac{1}{\mu},$$

and

$$E[T_\delta] = \frac{1}{\delta}.$$

Also, under this assumption, since T_h is the minimum of two exponential random variables, T_h is exponentially distributed with parameter $\mu + \delta$.

Therefore,

$$E[T_h] = \frac{1}{\mu + \delta}.$$

Performance Modeling with Mobility (Cont'd)

There are two approaches:

- Erlang fixed-point approximation (EFPA)
- Markov chain simulation

Under EFPA, we do not distinguish between blocking of new calls and dropping of handover calls, but in the simulations we do make this distinction.

We begin with EFPA.

Define the following parameters:

Arrival rate of new calls to cell i	$\lambda(i)$
Total call arrival rate to cell i (new calls plus handoff calls)	θ_i
Probability that a call will handover	p_h
Probability of handover from cell i to cell j	p_{ij}
Blocking probability for new and handoff calls at cell i (assuming they are equal)	B_i

EFPA – Derivation of p_h

p_h is the probability that a call will handover to another cell. In other words, p_h is the probability that the cell dwelling time is shorter than the call holding time.

$$p_h = P(T_\delta < T_\mu)$$

We assumed that T_δ and T_μ are exponentially distributed random variables with parameters δ and μ, respectively.

Therefore, p_h is the probability of one exponentially distributed random variable T_δ is the minimum of two exponential random variables T_δ and T_μ, and according to our previous derivation:

$$p_h = \frac{\delta}{\delta + \mu}.$$

EFPA – Probability of handover from cell i to cell j

According to our model, the event that a call will handover from cell i to cell j requires two events to occur:

1. the call will handover out from cell i, and
2. given that the cell handover out of cell i, it will handover to cell j.

Therefore,

$$p_{ij} = p_h P(i,j),$$

where
$P(i,j)$ is the probability of a handoff from cell i to cell j given that the call will handoff.

In a homogeneous network with hexagonal cells, each cell has six neighbors, so $P(i,j) = 1/6$.

EFPA – Total Arrival Rate

- The total arrival rate is given by the sum of the arrival rate of new calls and handover calls
- the rate of incoming calls can be considered to be the total rate of calls offered to other cell, times the probability that they will not be blocked there, times the probability that a call will handover to cell i, i.e., the total arrival rate is

$$\theta_i = \lambda(i) + \sum_{j \in Neib(i)} (1 - B_j)\theta_j p_{ji} \qquad i = 1, 2, \ldots, m, \tag{1}$$

where $Neib(i)$ is the set of neighbours of cell i and m is the total number of cells in the networks.

EFPA – How to Find θ_i and B_i?

- Using the approximation that the blocking probability B_i, at cell i is given by the Erlang B function, i.e.,

$$B_i = E_{f_i}(\theta_i/(\mu+\delta)), \quad i = 1, 2, \ldots, m, \tag{2}$$

  where $E_k(A)$ is the Erlang B function for a traffic of A erlangs offered to and Erlang B system with k channels.
- For a cellular network of m cells, this model gives two sets of m-dimensional nonlinear equations (Equations (1) and (2)) which can be solved numerically.

EFPA – How to find θ_i and B_i? (Cont'd)

Method of Successive Substitution

1. Guess an initial set of θ_i.
2. Substitute them into Equation (2) to obtain a set of B_i.
3. Substitute θ_i and B_i into Equation (1) to obtain a new set of θ_i.
4. Check if the relative difference between the old and new set of θ_i is smaller than a tolerance limit ϵ.
5. If yes, stop. Otherwise, repeat from Step (1) using the new set of θ_i.

EFPA – Overall average blocking probability in a cell

To obtain the estimation for the overall average blocking probability $P(B)$ (of new and handoff calls) in a cell, we weighted average the different B_i values (the blocking probability in individual cells) by the corresponding arrival rates as follows.

$$P(B) = \frac{\sum_1^m B_i\theta_i}{\sum_1^m \theta_i}.$$

Be reminded again that the overall blocking probability $P(B)$, includes blocking of new calls and dropping of handover calls.

EFPA – Blocking or dropping of individual calls

Consider a call made through a mobile station that from its starting time until its completion, the mobile station visits (or plan to visit) z cells, starting in cell i_1 then it continues to cell i_2, and then visits cells $i_3, i_4 \ \ldots, i_z$ in that order. (Note that we allow visits to the same cell, e.g., $i_2 = i_4$, but consecutive cells are different.)
To approximate its blocking probability, we assume independence between the cells, and we notice that the call will be successful with probability:

$$P(successful) = (1 - B_{i_1})(1 - B_{i_2})(1 - B_{i_3}) \cdots (1 - B_{i_z}) = \prod_{j=i_1}^{i_z} (1 - B_j).$$

Then the approximate probability of blocking or dropping of the call is:

$$P(blocking\ \ or\ \ dropping) = 1 - P(successful) = 1 - \prod_{j=i_1}^{i_z} (1 - B_j).$$

Markov chain simulation of a mobile cellular network

Again, we model a mobile cellular network as a network of Erlang B systems assuming:

1. the number of channels in each cell is fixed and equal to k,
2. new call generations in each cell follows a Poisson process,
3. call holding times are exponentially distributed,
4. times until handover occur in each cell are also exponentially distributed.

In the following we describe how to simulate such a network. This is also available in Section 19.4 in classnotes.pdf

Markov chain simulation of a mobile cellular network (Cont'd)

Variables and input parameters

m = number of cells (Erlang B Systems) in the network
$Q(i)$ = number of calls in progress (queue size) in cell i
B_p = estimation for the blocking probability
$N_a(i)$ = number of call arrivals counted so far in cell i
$N_b(i)$ = number of blocked new calls (arrivals) counted so far in cell i
$N_h(i)$ = number of call handovers counted so far into cell i
$N_d(i)$ = number of dropped calls that tried to handover so far into cell i
$MAXN_a$ = maximal number of call arrivals - used as a stopping criterion

Markov chain simulation of a mobile cellular network (Cont'd)

Variables and input parameters (Cont'd)

$\mu = 1/(\text{the mean call holding time})$

$\lambda(i) =$ arrival rate of new calls in cell i

$P(i,j) =$ the probability of a handoff from cell i to cell j given that the call will handoff

$\delta(i) =$ handover rate in cell i per call $= 1/$(mean time a call stays in cell i before it leaves the cell)

$P_B =$ Blocking probability estimation

$P_D =$ Dropping probability estimation of dropped handover calls

$Neib(i) =$ the set of neighboring cells of cell i

$|Neib(i)| =$ number of neighboring cells of cell i

Markov chain simulation of a mobile cellular network (Cont'd)

How to write the simulation

We will repeatedly consider $R(01)$ a uniform $U(0,1)$ random deviate. A new value for $R(01)$ is generated every time it is called.

To know if the next event is an arrival, we use the following IF statement. If

$$R(01) \leq \frac{\sum_{i=1}^{m} \lambda(i)}{\sum_{i=1}^{m} \lambda(i) + \sum_{i=1}^{m} Q(i)\mu + \sum_{i=1}^{m} Q(i)\delta(i)}$$

then the next event is an arrival. Else, to find out if it is a departure, i.e., a call termination in its own cell, we use the following IF statement (using the same $R(01)$ value as before). If

$$R(01) \leq \frac{\sum_{i=1}^{m} \lambda(i) + \sum_{i=1}^{m} Q(i)\mu}{\sum_{i=1}^{m} \lambda(i) + \sum_{i=1}^{m} Q(i)\mu + \sum_{i=1}^{m} Q(i)\delta(i)}$$

then the next event is a departure; else, it is a handover.

Markov chain simulation of a mobile cellular network (Cont'd)

How to write the simulation (Cont'd)

If the next event is an arrival, we need to know in which of the m cells it occurs. To find out, we use the following loop.
For $i = 1$ to m, do: If

$$R(01) \leq \frac{\sum_{j=1}^{i} \lambda(j)}{\sum_{j=1}^{m} \lambda(j)},$$

stop the loop. The arrival occurs in cell i and we increment the total number of arrivals in cell i so far by $N_a(i) = N_a(i) + 1$ and if $Q(i) < k$ then $Q(i) = Q(i) + 1$, else the number of lost calls needs to be incremented, namely, $N_b(i) = N_b(i) + 1$. (For all i values in the above loop, make sure that you use the same $R(01)$ value.)

How to write the simulation (Cont'd)

Now we check if this arrival is the $MAXN_a$ arrival. If it is then we end the simulation. In particular, if

$$\sum_{j=1}^{m} N_a(j) \geq MAXN_a,$$

the simulation ends, and we compute the blocking probability as follows.

$$P_B = \frac{\sum_{i=1}^{m} N_b(i)}{MAXN_a}.$$

In a similar way we can compute The dropping probability of handover calls as follows.

$$P_D = \frac{\sum_{i=1}^{m} N_d(i)}{\sum_{i=1}^{m} N_h(i)}.$$

How to write the simulation (Cont'd)

If the next event is a departure, we need to know in which of the m cells it occurs. To find out we use the following loop.
For $i = 1$ to m, do: If

$$R(01) \leq \frac{\sum_{j=1}^{i} Q(j)\mu}{\sum_{j=1}^{m} Q(j)\mu} = \frac{\sum_{j=1}^{i} Q(j)}{\sum_{j=1}^{m} Q(j)}.$$

Then stop the loop. The departure occurs in Cell i, so $Q(i) = Q(i) - 1$. (Again, for all i values in the above loop, use the same $R(01)$ value.) Note that we do not need to verify that $Q(i) > 0$ before decrementing it (why?). This is because if $Q(i) = 0$ the above loop will not generate a departure in Cell i.

Markov chain simulation of a mobile cellular network (Cont'd)

How to write the simulation (Cont'd)

If the next event is a handover, we need to know from which of the m cells it handovers out of. To find it out, we use the following loop.
For $i = 1$ to m, do: If

$$R(01) \leq \frac{\sum_{j=1}^{i} Q(j)\delta(j)}{\sum_{j=1}^{m} Q(j)\delta(j)}.$$

Then, stop the loop. The handover occurs out of cell i, so $Q(i) = Q(i) - 1$. Note that again we do not need to verify that $Q(i) > 0$. (Again, for all i values in the above loop, use the same $R(01)$ value.)

Markov chain simulation of a mobile cellular network (Cont'd)

How to write the simulation (Cont'd)

Then, to find out into which cell the call handover in, we use the following: For $j = 1$ to $|Neib(i)|$, $j \in Neib(i)$, do: If

$$R(01) \leq \frac{\sum_{n=1}^{j} P(i,n)}{\sum_{n=1}^{|Neib(i)|} P(i,n)} = \sum_{n=1}^{j} P(i,n), \quad n \in Neib(i),$$

the call handovers into cell j. (Again, for all j values in the above loop, use the same $R(01)$ value.) Then, increment $N_h(j)$. If cell j is full, namely $Q(j) = k$, the handover is dropped, so increment also $N_d(j)$. If $Q(j) < k$, increment $Q(j)$, i.e, $Q(j) = Q(j) + 1$.

Markov chain simulation of a mobile cellular network (Cont'd)

Explanation how we choose the next event

We use the following probabilities:

- the probability that the next event is an arrival, denoted P_{arr};
- the probability that the next event is a departure, denoted P_{dep};
- the probability that the next event is a handover, denoted P_{ho}.

They are given by

$$P_{arr} = \frac{\sum_{i=1}^{m} \lambda(i)}{\sum_{i=1}^{m} \lambda(i) + \sum_{i=1}^{m} Q(i)\mu + \sum_{i=1}^{m} Q(i)\delta(i)}.$$

$$P_{dep} = \frac{\sum_{i=1}^{m} Q(i)\mu}{\sum_{i=1}^{m} \lambda(i) + \sum_{i=1}^{m} Q(i)\mu + \sum_{i=1}^{m} Q(i)\delta(i)}.$$

$$P_{ho} = \frac{\sum_{i=1}^{m} Q(i)\delta(i)}{\sum_{i=1}^{m} \lambda(i) + \sum_{i=1}^{m} Q(i)\mu + \sum_{i=1}^{m} Q(i)\delta(i)}.$$

Explanation (Cont'd)

Then, generating $R(01)$, the condition for an arrival to be the next event is $R(01) \leq P_{arr}$.
If this condition does not hold, i.e., $R(01) > P_{arr}$, then, the condition for a departure to be the next event is $R(01) \leq P_{arr} + P_{dep}$.
Then, if this condition does not hold, i.e., $R(01) > P_{arr} + P_{dep}$, then, the next event must be a handover.

Markov chain simulation of a mobile cellular network (Cont'd)

Homework

Consider the 49-cell cellular network model with wrapped-around design depicted in Figure 2 (see next slide). Choose your own input parameters for the number of channels per cell, arrival rates and mean holding times. Use a Markov-chain simulation to approximate the overall blocking and dropping probabilities in this network.

For the same 49-cell network, obtain the blocking probability using EFPA.

Compare the results of the two approaches for a range of parameter values. Notice that the simulation provides separate results for the blocking and the dropping probabilities, while the EFPA provides a result for the combined blocking/dropping probability. You will need to make another step in the simulation to obtain the combined probability in order to make the two approaches comparable. Discuss and explain the differences in the results between the two approaches.

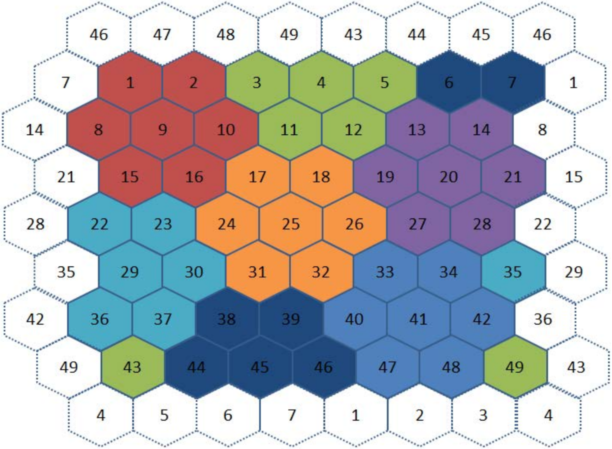


Figure 2: 49-cell hexagonal configuration network model with wrapped-around design.

Source: Figure 2 in J. Wu, E. W. M. Wong, J. Guo, and M. Zukerman, "Performance Analysis of Green Cellular Networks with Selective Base-Station Sleeping," Performance Evaluation, vol. 111, pp. 17-36, May 2017.

Channel Reservation – a single cell model

If a new call is unsuccessful due to blocking, that is not as bad as a handoff call being dropped. Therefore, handoff calls should have a higher priority to use the channels.

One possible way to assigning priority to handoff calls is to reserve some channels exclusively for handoff calls, as shown in Fig. 3.

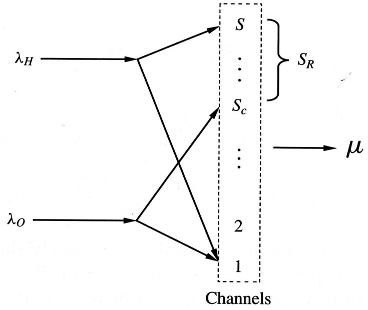


Figure 3: System model with reserved channels for handoff calls.

Channel Reservation – a single cell model (Cont'd)

There are totally S channels, S_R channels are reserved exclusively for handoff calls. Thus,

$S_c = S - S_R$ channels are shared by both types of calls.
λ_O is the mean arrival rate of new calls.
λ_H is the mean arrival rate of handoff calls.
$1/\mu$ is the mean duration of a call.

a Markov chain model

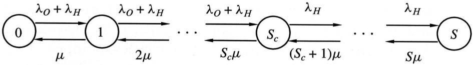


Figure 4: State transition diagram for the channel reservation model.

To simplify the analysis, assume that λ_H is known and given. The steady-state equations are:

$$\begin{cases} i\mu\pi_i = (\lambda_O + \lambda_H)\pi_{i-1}, & 0 \le i \le S_c \\ i\mu\pi_i = \lambda_H\pi_{i-1}, & S_c < i \le S \end{cases} \tag{3}$$

where π_i is the steady state probability of having i channels busy, $i = 0, 1, 2, \ldots, S$.

Channel Reservation – a single cell model (Cont'd)

Steady-state Probabilities

From (3), π_i can be expressed as π_0:

$$\pi_i = \begin{cases} \frac{(\lambda_O+\lambda_H)^i}{i!\mu^i}\pi_0, & 0 \le i \le S_c \\ \frac{(\lambda_O+\lambda_H)^{S_c}\lambda_H^{i-S_c}}{i!\mu^i}\pi_0, & S_c < i \le S \end{cases} \tag{4}$$

Then, using the normalization condition $\sum_{i=0}^{S}\pi_i = 1$, we have

$$\pi_0 = \left[\sum_{i=0}^{S_c}\frac{(\lambda_O+\lambda_H)^i}{i!\mu^i} + \sum_{i=S_c+1}^{S}\frac{(\lambda_O+\lambda_H)^{S_c}\lambda_H^{i-S_c}}{i!\mu^i}\right]^{-1} \tag{5}$$

Once π_0 is found, all π_i are solved.

Channel Reservation – a single cell model (Cont'd)

Blocking Probabilities

Since a new call is blocked when S_c or more channels are occupied, the blocking probability B_0 for a new call is given by

$$B_O = \sum_{i=S_c}^{S} \pi_i \tag{6}$$

On the other hand, a handoff call can access all channels, it is blocked only if all S channels are occupied. Thus, blocking probability or the forced termination probability of a handoff call is given by

$$B_H = \pi_S = \frac{(\lambda_O + \lambda_H)^{S_c} \lambda_H^{S-S_c}}{S! \mu^S} \pi_0 \tag{7}$$