Exam page 3

Janis Hodorjonoks

May 30, 2019

434 Elementary Queueing Theory

To determine performance measures, we first compute

$$L_q = \sum_{n=c}^{K}(n-c)p_n = p_0\frac{(\lambda/\mu)^c\,\lambda/c\mu}{c!(1-\lambda/c\mu)^2}\left[1-\left(\frac{\lambda}{c\mu}\right)^{K-c+1} - \left(1-\frac{\lambda}{c\mu}\right)(K-c+1)\left(\frac{\lambda}{c\mu}\right)^{K-c}\right]$$

$$= p_0\frac{(c\rho)^c\rho}{c!(1-\rho)^2}\left[1-\rho^{K-c+1}-(1-\rho)(K-c+1)\rho^{K-c}\right],$$

where we have defined $\rho = \lambda/(c\mu)$. This allows us to obtain the average number of customers in the system (see also Exercise 11.6.1):

$$L = L_q + \left(c - \sum_{n=0}^{c-1}(c-n)p_n\right) = L_q + c - \sum_{n=0}^{c-1}(c-n)\frac{(\lambda/\mu)^n}{n!}p_0, \tag{11.12}$$

which is the number waiting in the queue plus the average number of busy servers. In the special case of $c = 1$, this reduces to

$$L = L_q + (1-p_0) = L_q + \frac{\lambda'}{\mu},$$

where $\lambda' = \lambda(1-p_K)$ is the effective arrival rate. The time spent in the system can now obtained from $W = L/\lambda'$. Finally, the mean time spent in the queue waiting for service to begin is found from $W_q = W - 1/\mu$ or $W_q = L_q/\lambda'$.

Erlang's Loss Formula

For the special case when $K = c$, i.e., the *M/M/c/c* queue, the so-called "blocked calls cleared with c servers" system, these results lead to another well-known formula associated with the name Erlang. The probability that there are n customers in the *M/M/c/c* queue is given as

$$p_n = \frac{(\lambda/\mu)^n/n!}{\sum_{i=0}^{c}(\lambda/\mu)^i/i!} = p_0\frac{(\lambda/\mu)^n}{n!}.$$

The formula for p_c is called "Erlang's loss formula" and is the fraction of time that all c servers are busy. It is written as $B(c, \lambda/\mu)$ and called "Erlang's B formula":

$$B(c,\lambda/\mu) = \frac{(\lambda/\mu)^c/c!}{\sum_{i=0}^{c}(\lambda/\mu)^i/i!} = p_0\frac{(\lambda/\mu)^c}{c!}.$$

Notice that the probability that an arrival is lost is equal to the probability that all channels are busy. Erlang's loss formula is also valid for the *M/G/c/c* queue. In other words, the steady-state probabilities are a function only of the mean service time, and not of the complete underlying cumulative distribution function. An efficient recursive algorithm for computing $B(c, \lambda/\mu)$ is given by

$$B(0,\lambda/\mu) = 1, \quad B(c,\lambda/\mu) = \frac{(\lambda/\mu)B(c-1,\lambda/\mu)}{c+(\lambda/\mu)B(c-1,\lambda/\mu)}.$$

We leave its proof as an exercise. It is also possible to express Erlang's C formula in terms of $B(c, \lambda/\mu)$ and again, we leave this as an exercise.

11.7 Finite-Source Systems—The *M/M/c//M* Queue

The *M/M/c//M* queue is a c-server system with a finite customer population. No longer do we have a Poisson input process with an infinite user population. There is a total of M customers and a customer is either in the system, or else is outside the system and is in some sense "arriving." When

To determine performance measures, we first compute

$$L_q = \sum_{n=c}^{K}(n-c)p_n = p_0 \frac{(\lambda/\mu)^c \lambda/c\mu}{c!(1-\lambda/c\mu)^2}\left[1-\left(\frac{\lambda}{c\mu}\right)^{K-c+1} - \left(1-\frac{\lambda}{c\mu}\right)(K-c+1)\left(\frac{\lambda}{c\mu}\right)^{K-c}\right]$$

$$= p_0 \frac{(cp)^c \lambda/c\mu}{c!(1-\lambda/c\mu)^2}\left[1-\rho^{K-c+1}-(1-\rho)(K-c+1)\rho^{K-c}\right],$$

where we have defined $\rho = \lambda/(c\mu)$. This allows us to obtain the average number of customers in the system (see also Exercise 11.6.1):

$$L = L_q + \left(c - \sum_{n=0}^{c-1}(c-n)p_n\right) = L_q + c - \sum_{n=0}^{c-1}(c-n)\frac{(\lambda/\mu)^n}{n!}p_0, \tag{11.12}$$

which is the number waiting in the queue plus the average number of busy servers. In the special case of $c = 1$, it reduces to

$$L = L_q + (1-p_0) = L_q + \frac{\lambda'}{\mu'}$$

where $\lambda' = \lambda(1-p_K)$ is the effective arrival rate. The time spent in the system can now obtained from $W = L/\lambda$. Finally, the mean time spent in the queue waiting for service to begin is found from $W_q = W - 1/\mu$ or $W_q = L_q/\lambda'$.

Erlang's Loss Formula

For the special case K = *c*, i.e., the *M/M/c/c* queue, the so-called "blocked calls cleared with *c* servers "system, the results lead to another well-known formula associated with the name of Erlang. The probability that are *n* customers in the *M/M/c/c* queue is given as

$$p_n = \frac{(\lambda/\mu)^c/n!}{\sum_{i=0}^{c}(\lambda/\mu)^i/i!} = p_0\frac{(\lambda/\mu)^n}{n!}.$$

The formula for p_c is called "Erlang's loss formula "and is the fraction of time that all *c* severs are busy. It is written as $B(c, \lambda/\mu)$ and called "Erlang's *B* formula ":

$$B(c,\lambda/\mu) = \frac{(\lambda/\mu)^c/c!}{\sum_{i=0}^{c}(\lambda/\mu)^i/i!} = p_0\frac{(\lambda/\mu)^c}{c!}.$$

Notice that the probability that an arrival is lost is equal to the probability that all channels are busy. Erlang's loss formula is also valid for the *M/G/c/c* queue. In other words, the steady-state probabilities are a function only of the mean service time, and not of the complete underlying cumulative distribution function. An efficient recursive algorithm for computing $B(c, \lambda/\mu)$ is given by

$$B(0,\lambda/\mu) = 1, \quad B(c,\lambda/\mu) = \frac{(\lambda/\mu)B(c-1,\lambda/\mu)}{c+(\lambda/\mu)B(c-1,\lambda/\mu)}.$$

We leave its proof as an exercise. It is also possible to express Erlang's C formula in terms of $B(c, \lambda/\mu)$ and again, we leave this as an exercise.

11.7 Finite-Source Systems—The *M/M/c//M* Queue

The *M/M/c//M* queue is a *c*-server system with a finite customer population. No longer do we have a Poisson input process with an infinite user population. There is a total of *M* customers and a customer is either in the system, or else is outside the system and is in some sense "arriving. "When

```
\documentclass[11pt]{book}
\usepackage{geometry}
\usepackage{fancyhdr}
\usepackage{amsmath}
\usepackage{tcolorbox}
\usepackage{xcolor}
\usepackage{times}
\usepackage{amssymb}
\usepackage{ragged2e}
\usepackage{parskip}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{textcomp}
\fontfamily{cmtt}\selectfont
    \geometry{a4paper,
 total={212mm,305mm},
 left=29mm,
 top=33mm,
 right=29mm,
 bottom=40mm}
\pagestyle{fancy}
\fancyhead[L]{\textbf{434 \hspace{5mm} \small{Elementary Queueing Theory}}}
\fancyfoot{}
\usepackage{xcolor}
\pagecolor{yellow!5}
\author{Janis Hodorjonoks}
\title{Exam page 3}
\setlength{\parskip}{2em}
\parindent=1cm
\renewcommand{\headrulewidth}{0.7pt}
\renewcommand{\baselinestretch}{0.8}
\begin{document}
\maketitle
\pagebreak

\parindent=0.5cm To determine performance measures, we first compute
    $$L_q = \sum^{K}_{n=c}(n - c)p_{n} = p_{0}
    \frac{(\lambda/\mu)^c \lambda/c\mu}
    {c!(1 - \lambda/c\mu)^2}\left[1-  \left
    (\frac{\lambda}{c\mu}\right)^{K-c+1}-
    \left(1-\frac{\lambda}{c\mu}\right)(K - c + 1)
    \left(\frac{\lambda}{c\mu}\right)
    ^{K - c}\right]$$ \\
    $$= p_{0}\frac{(cp)^{c}\lambda/c\mu}{c!(1 -
    \lambda/c\mu)^{2}}\left[1 - \rho^
    {K -c + 1}-(1 - \rho)(K - c + 1)\rho^{K - c}
    \right],$$\\
where we have defined $\rho=\lambda/(c\mu).$ This
allows us to obtain the average
number of customers in the system (see also
Exercise 11.6.1):
\begin{equation}\tag{11.12}
L =L_{q} + \left( c - \sum^{c-1}_{n=0}(c-n)p_{n}
\right) = L_{q} + c - \sum^{c-1}_{n=0}
(c-n)\frac{(\lambda/\mu)^{n}}{n!}p_{0},
\end{equation}
which is the number waiting in the queue plus the
average number of busy servers.
In the special case of $c = 1$, it reduces to
$$L = L_{q} + (1 - p_{0}) = L_{q} + \frac{\lambda'}
{\mu'}$$
where $\lambda' = \lambda(1 - p_{K})$ is the
effective arrival rate. The time spent
in the system can now obtained from $W = L/\lambda.
$ Finally, the mean time spent in
the queue waiting for service to begin is found
```

```
from $W_{q} = W - 1/\mu$ or
$W_{q} = L_{q}/\lambda'.$ \par

\noindent\textbf{Erlang's Loss Formula}\newline
For the special case K = \textit{c}, i.e., the
\textit{M/M/c/c} queue, the so-called
\textquotedblleft blocked calls cleared with \textit
{c} servers \textquotedblright
system, the results lead to another well-known
formula associated with the name of Erlang.
\linebreak The probability that are \textit{n}
customers in the \textit{M/M/c/c} queue is given as
$$p_{n}=\frac{(\lambda/\mu)^{c}/n!}{\sum^{c}_{i=0}
(\lambda/\mu)^{i}/i!}=p_{0}\frac{
(\lambda/\mu)^{n}}{n!}.$$
The formula for $p_{c}$ is called \textquotedblleft
Erlang's loss formula \textquotedblright
and is the fraction of time that all \textit{c}
severs are busy. It is written as
$B(c,\lambda/\mu)$ and called \textquotedblleft
Erlang's \textit{B} formula \textquotedblright:
$$B(c,\lambda/\mu)=\frac{(\lambda/\mu)^{c}/c!}
{\sum^{c}_{i=0}(\lambda/\mu)^{i}/i!}=p_{0}\frac{
(\lambda/\mu)^{c}}{c!}.$$
Notice that the probability that an arrival is
lost is equal to the probability that all
channels are busy. Erlang's loss formula is also
valid for the \textit{M/G/c/c} queue.
In other words, the steady-state probabilities
are a function only of the mean service
time, and not of the complete underlying cumulative
distribution function. An efficient
recursive algorithm for computing  $\,B(c,\lambda/
\mu)\,$ is given by
$$B(0,\lambda/\mu)=1,\quad B(c,\lambda/\mu)=\frac{
(\lambda/\mu)B(c-1,\lambda/\mu)}{c +
(\lambda/\mu)B(c-1,\lambda/\mu)}.$$
We leave its proof as an exercise. It is also
possible to express Erlang's C formula in
terms of $\,B(c,\lambda/\mu)\,$ and again, we
leave this as an exercise.\par
\noindent\textbf{\Large{11.7 Finite-Source Systems
\textthreequartersemdash The \textit
{M/M/c//M} Queue }}\par
\justify
The \textit{M/M/c//M} queue is a \textit{c}-server
system with a finite customer population
. No longer do we have a Poisson input process with
an infinite user population. There is a
total of \textit{M} customers and a customer is
either in the system, or else is outside the
system and is in some sense \textquotedblleft
arriving. \textquotedblright When
```