NAME

Fmd1() − Virtual waiting time distribution function

IntFmd1() − Virtual waiting time distribution function for integral values of x

SumMd1() − Calculates the state probabilities of a M/D/1 queue

RecMd1() − Calculates the state probabilities of a M/D/1 queue using a recursive algorithm

SYNOPSIS

#include <queuel.h>

double Fmd1(double x, double rho);

double IntFmd1(int N, double rho);

double *SumMd1(double x, double rho);

double *RecMd1(int x, double rho);

DESCRIPTION

delim $$ These functions return the state probabilities or the virtual waiting time distribution of a M/D/1 queue.

Fmd1() is a model for the M/D/1 queuing system with Poisson arrivals and deterministic (constant) service time. Parameter *x* is the amount of unfinished work in the system. *Rho* is the load level of the system.

ALGORITHM

M/D/1 waiting time distribution is calculated using the following algorithm (Iversen):

> P{$W <= T + tau$}$ = e sup{lambda tau} sum from {0<=n<=T} {{(-{lambda tau}) sup n} over {n!}} P${$W <= T - n$},

where P{$W <= T - n$} is the waiting time for integral values of x:

> P{$W <= t$}$ = P${$0$}$ + P${$1$}$ + ... + P${$t$}.

The state probabilities can be calculated by using a general or a recursive algorithm (Fmd1() uses the recursive algorithm, because it is more accurate):

> $P(0)=1-A $
> $P(1)=(1-A)(e sup A -1) $
> $P(2)=(1-A)(-e sup A (1+A) + e sup {2A}) $
> $A={lambda h}.$ (If h=1, $A={lambda}).$

General algorithm:

> $P(i) = (1-A) sum from {1<=n<=i} (-1) sup {i-n} e sup {nA} ({{(nA) sup {i-n}} over {(i-n)!}} + {{(nA) sup {i-n-1}} over {(i-n-1)!}})$, i=2,3,...
>
> (The last term always equals ${e sup {iA}}.) $

Recursive algorithm:

> $P(i+1) = {1 over {P(0,h)}} ${$P(i)-$[$P(0)+P(1)$]$ P(i,h) - sum from {2<=n<=i} P(n) P(i-n+1,h)$}, i=3,4,...
>
> $(P(i,h)={{({lambda h}) sup i} over {i!}} e sup {-({lambda h})})$

ERRORS

When $ rho $ is close to 1, these functions might give inaccurate results.

SEE ALSO

COST 224: Performance evaluation and design of multiservice networks