delim $$

NAME

   Bkaufman, Bmitra − Erlang blocking probabilities for multiple traffic classes

SYNOPSIS

   #include <**erlang.h**>

   double Bkaufman(int j, int J, int C, int ∗d, double ∗a);

   double Bmitra(int j, int J, double C, int ∗d, double ∗a);

DESCRIPTION

   These functions give the blocking probability for a traffic class *j,* ($1 <= j <= J$), from number of classes *J.* *C* is the total capacity of the link. Each traffic class is characterized by it's bandwidth. *d[]* is a pointer to the list of these values. *a[]* is a list of traffic intensities for each class.

   The difference in these functions is in the implementation. **Bkaufman()** is implemented using the Kaufman and Roberts recursion formula:

   $$B_k = \sum_{d = C - d_k + 1}^{C} Q(d) / \sum_{d=0}^{C} Q(d)$$

   where Q(d) is defined as

   $$d \; Q(d) = \sum_{k = 1}^{K} a_k d_k Q(d - d_k)$$

   and Q(0) = 1. For negative arguments Q = 0.

   This function is useful for small values of C, but can't handle much larger values without overflowing. Evaluation time for this function increases as *C* increases.

   Bmitra() is the Poisson limit approximation to the same function and has accuracy of 1.2 %. This function has relatively constant evaluation time. Newton's method is used to find the value of z, which is needed in function. When $z = 1$ this function has a singularity.

BUGS

   Bkaufman() overflows very easily, especially with many classes.

SEE ALSO

   Berl_i(3), Berl_d, Xerl(3), Aerl(3)