

```

function [ EIF ]=
EPserveth(lam,P,F,bf,alpha,Mt,Ns,BW,Mz,MINz,MAXz,alphaP,k,mu,sig,c)
%EPserveth(lam,P,F,bf,zf,alpha,Mt,Ns,BW) function gives the expected
value
% of service for some file-size distribution.
%
% considers constant object lifespan with mean Mt,
% and takes as arguments the specific
% object placement vector bf
% node density lam
% path-loss exponent alpha
% transmitted signal power P
% catalog size F
% noise power W/Hz Ns
% bandwidth BW
% AND mean object size Mz
% AND minimum object size MINz
% AND maximum object size MAXz
% AND Pareto parameter alphaP
% AND Weibull parameter (k,mu)
% AND Log-normal parameter (mu,sig)
% AND case of distribution:
%
% Case 1: EXPONENTIAL
% Case 2: UNIFORM
% Case 3: PARETO
% Case 4: WEIBULL
% Case 5: LOG-NORMAL
%
%It returns the theoretical value of expected service probability
for
% the vector of F objects.
%
% Author: Anastasios Giovanidis,
% CNRS CR2, LTCI - Telecom ParisTech, Paris,
% February 2016.
%
if c == 1 % Exp
for ob = 1:F
    bz      =   bf(ob);
    Funy2   =   @(Z)((1-exp(-pi*lam*bz*(P)^(2/alpha)*gamma(1+2/
alpha).*((2.^(Z/Mt/BW)-1).^(-2/alpha))/(Ns*BW)^(2/alpha))).*(1/
Mz*exp(-Z/Mz)));
    EIF(ob) =   quadgk(Funy2,0,Inf);
    clear Funy2
end
%
elseif c == 2 % Unif
for ob = 1:F
    bz      =   bf(ob);
    Funy2   =   @(Z)((1-exp(-pi*lam*bz*(P)^(2/alpha)*gamma(1+2/
alpha).*((2.^(Z/Mt/BW)-1).^(-2/alpha))/(Ns*BW)^(2/alpha))).*(1/
(MAXz-MINz)));
    EIF(ob) =   quadgk(Funy2,MINz,MAXz);

```

```

clear Funy2
end
%
elseif c == 3 % Pareto
for ob = 1:F
    bz      =    bf(ob);
    Funy2   =    @(Z)((1-exp(-pi*lam*bz*(P)^(2/alpha)*gamma(1+2/
alpha).*(2.^(Z/Mt/BW)-1).^(-2/alpha))/(Ns*BW)^(2/
alpha))).*(alphaP./(Z.^(alphaP+1))*MINz.^alphaP));
    EIF(ob) =    quadgk(Funy2,MINz,Inf);
    clear Funy2
end
%
elseif c == 4 % Weibull
for ob = 1:F
    bz      =    bf(ob);
    Funy2   =    @(Z)((1-exp(-pi*lam*bz*(P)^(2/alpha)*gamma(1+2/
alpha).*(2.^(Z/Mt/BW)-1).^(-2/alpha))/(Ns*BW)^(2/alpha))).*(k/
mu*(Z/mu).^(k-1).*exp(-(Z/mu).^k));
    EIF(ob) =    quadgk(Funy2,0,Inf);
    clear Funy2
end
%
elseif c == 5 % Log-normal
for ob = 1:F
    bz      =    bf(ob);
    Funy2   =    @(Z)((1-exp(-pi*lam*bz*(P)^(2/alpha)*gamma(1+2/
alpha).*(2.^(Z/Mt/BW)-1).^(-2/alpha))/(Ns*BW)^(2/alpha))).*(1/sig/
sqrt(2*pi)./Z.*exp(-((log(Z)-mu).^2/2/sig^2)));
    EIF(ob) =    quadgk(Funy2,0,Inf);
    clear Funy2
end
end
%
end

```