```
clear
clc
close all
%
% Author: Anastasios Giovanidis,
% CNRS CR2, LTCI - Telecom ParisTech
% January 2016
%
% This is the main body of code, to evaluate the Total (and
Expected)
% Service Success Probability from the paper
% Chedia Jarray and Anastasios Giovanidis - "The Effects of Mobility
on
% the Hit Performance of Cached D2D Networks", to appear In
% WIOPT-SPASWIN'16.
%
% The program provides an evaluation plot of the total service
probability
% over a range of values for the mean lifespan of transmission nodes
% from (I) Simulations and (II) Theory.
%
% Two cases are considered for the object demands (A) Audio and (B)
Video.
% This is given as option 1.
%
% The choice of different probability distribution for the cache-
size is
% also given as option 2.
%
%-------------------------------------------------------------------
-
%------------------ Problem-Parameters
------------------------------
%-------------------------------------------------------------------
-
%
% Note: The choice of the parameters that follow guarantee
% that the inequality Pr[BW*Mt*ln(1+SNR(R_1))>=Z] >= Eps,
% provides the following bounds, when:
%
% Audio
% Z = MAXzA, Eps <= 61.15%
% Z = MINzA, Eps <= 99.40%
%
% Video
% Z = MAXzV, Eps <= 29.36%
% Z = MINzV, Eps <= 90.02%
%
% given that R_1 is the mean distance of the Nearest Transmitter and
% equal to the Poisson case to R_1=1/(lam*sqrt(2)).
%
%-------------------------------------------------------------------
-
% Simulation Window Parameters
```

```
Wx = 100;               % [km] window size x
Wy = Wx;                % [km] window size y
lam = 0.0025;           % [D2D users/m^2] density of users
%
% Communication Parameters
Ns = 10^(-11);          % [Watt/Hz] noise power
alpha = 4;              % [] path-loss exponent
P = 0.5;                % [Watt] transmission power
BW = 5*10^6;            % [Hz] Attributed bandwidth
%
% Object Parameters
F = 100;                % [objects] catalog size
%
% Audio Files
gam = 0.78;             % [] Zipf exp
MtA = 10;               % [sec] Mean lifespan time (for Audio)
MzA = 2e6;              % [bits] Mean (Exponential) File Size (for
Audio)
MAXzA = 20e6;           % [bits] Max File Size (Audio)
MINzA = 12e4;           % [bits] Min File Size (Audio)
%
% Video Files
MtV = 1000;             % [sec] Mean lifespan time (for Video)
MzV = 1e9;              % [bits] Mean (Exponential) File Size (for
Video)
MINzV = 5e8;            % [bits] Min File Size (Video)
MAXzV = 4e9;            % [bits] Max File Size (Video)
%
% Cache
K = 5;                  % Cache size (No of objects)
%
T = 1000;               % Number of iterations for the Simulations
%
%-----------------------------------------------------------------
-
% OPTION 1: CHOOSE if you wish to work with Audio or Video
choice = 2;
% Choice Audio:
if choice == 1
Mt = MtA;
Mz = MzA;
Mtvec = (1:2:100);
MAXz = MAXzA;
MINz = MINzA;
elseif choice ==2
%
% Choice Video:
Mt = MtV;
Mz = MzV;
Mtvec = (1:50:2000);
MAXz = MAXzV;
MINz = MINzV;
end
%
```

```
% If interested only in one object of interest
%ob = 1;
%-------------------------------------------------------------------
-
%------------------ Pre-Processing
--------------------------------
%-------------------------------------------------------------------
-
% Object file size
%
% OPTION 2: CHOOSE the distribution
% (1.EXP, 2.Uniform, 3.Pareto, 4. Weibull k<1, 5. Log-Normal,
% 6.Ascend Uni/Exp, 7.Descend Uni/Exp)
%
% zf is the Input file-size vector that is generated/sampled.
%
distr = 5;
if distr == 1
    zf1 = exprnd(Mz,1,F); % Exponential
    zf = zf1;
    % Dummys for EPserveth
    alphaP = 1;
    k = 1;
    mu = 1;
    sig = 1;
elseif distr==2
    zf2 = rand(1,F)*(MAXz-MINz)+MINz; % Uniform
    zf = zf2;
    % Dummys for EPserveth
    alphaP = 1;
    k = 1;
    mu = 1;
    sig = 1;
elseif distr==3 % Pareto
    %alpha = 100/99; beta = 0.1*10^6; % Audio: so that (1) EX =
10Mb,
    % (2) a>1 and the mean is finite, but not the variance, and
    % (3) z>b = 100 Kbit as in Uniform.
    alphaP = 20/19; beta = 0.05*10^9; % Video: so that (1) EX = 1Gb,
    % (2) a>1 and the mean is finite, but not the variance, and
    % (3) z>b = 50 Mbit as in Uniform.
    %zf5 = randp(1,F,alpha,beta);
    zf5 = beta*(1-rand(1,F)).^(-1/alphaP);
    zf = sort(zf5,'descend');
    % Dummys for EPserveth
    k = 1;
    mu = 1;
    sig = 1;
elseif distr==4 % Weibull k<1
    k = 0.1; mu = 276; % Video: so that (1) k<1, (2) EX = 1Gb
    zf6 = wblrnd(mu,k,1,F); % Weibull
    zf = sort(zf6,'descend');
    % Dummys for EPserveth
    alphaP = 1;
```

```
    sig = 1;
elseif distr==5
    mu = 5*log(10); sig = sqrt(8*log(10)); % Video: so that EX =
1Gb.
    zf7 = lognrnd(mu,sig,1,F); % log-normal
    zf = sort(zf7,'descend');
    % Dummys for EPserveth
    alphaP = 1;
    k = 1;
elseif distr==6
    zf3 = rand(1,F)*(MAXz-MINz)+MINz; %  Uniform asc.
    %zf3 = exprnd(Mz,1,F); %  Exp asc.
    zf = sort(zf3,'ascend');
    % Dummys for EPserveth
    alphaP = 1;
    k = 1;
    mu = 1;
    sig = 1;
elseif distr==7
    zf4 = rand(1,F)*(MAXz-MINz)+MINz; %  Uniform desc.
    %zf4 = exprnd(Mz,1,F); %  Exp desc.
    zf = sort(zf4,'descend');
    % Dummys for EPserveth
    alphaP = 1;
    k = 1;
    mu = 1;
    sig = 1;
end
%
% Popularity Zipf
f = [1:F];
af = f.^(-gam); const1 = sum(af);
af = af/const1;
%
% Memory-to-Catalogue-Size-Ratio
aKF = K/F;
%
% Object placement vector
% Policy 1: Truncated popularity heuristic policy to test the
Problem
B = 10;
bf = min(af(1:B)/sum(af(1:B))*K,1);
sum(bf);
bf = [bf zeros(1,F-B)];
% Policy 2: Most Popular Content policy
%B = K;
%bf = [ones(1,B), zeros(1,F-B)];
%-------------------------------------------------------------------
-
%------------------ Evaluation ---------------------------------
%-------------------------------------------------------------------
-
%
kk=0;
```

```
%
P_all_Ssave = zeros(length(Mtvec),F);
P_all_Tsave = zeros(length(Mtvec),F);
E_all_Tsave = zeros(length(Mtvec),F);
%
for Mt=Mtvec
    kk = kk+1
%tic
% SIMUL (Calls routine Pservesim)
    P_all_S = Pservesim(Wx,Wy,lam,P,Ns,BW,alpha,Mt,K,F,bf,zf,T);
    P_all_Ssave(kk,:) = P_all_S;
    % Case 1: Find the service-per-object
    %Pservice_ob_S(kk) = P_all_S(ob);
    % Case 2: Find the total-service
    Pservice_ob_S(kk) = sum(P_all_S.*af);
    %
% SERVICE_TH_ALL (Calls routine Pserveth)
    P_all_T = Pserveth(lam,P,F,bf,zf,alpha,Mt,Ns,BW);
    P_all_Tsave(kk,:) = P_all_T;
    % Case 1: Find the service-per-object if you want
    %Pservice_ob_T(kk) = P_all_T(ob);
    % Case 2: Find the total-service
    % Put zeros to avoid NaNs from integration
    P_all_T = [P_all_T(1:10) zeros(1,F-10)];
    %
    Pservice_ob_T(kk) = sum(P_all_T.*af);
    %toc
    %
% Expected Service (Calls routine EPserveth)
% OPTION 2(b): CHOOSE the distribution for the file-size, to give as
% input for the EPserveth routine.
c = distr;
if c<6
% The choice follows the initial choice, without the options
% distr = 6, distr = 7, which fall in the non-independent case.
(Asc/Desc.)
% Options:
%   Case c=1: EXPONENTIAL
%   Case c=2: UNIFORM (uses MINz, MAXz)
%   Case c=3: PARETO (uses alphaP, beta=MINz)
%   Case c=4: WEIBULL (uses k, mu)
%   Case c=5: LOG-NORMAL (uses mu, sig)
%   Dummy values are given in
%   OPTION 2 when needed... Observe that WEIBULL
%   and LOG-NORMAL use the same declared variable mu, which has a
different
%   function in each of the two cases.
%
%
    E_all_T =
EPserveth(lam,P,F,bf,alpha,Mt,Ns,BW,Mz,MINz,MAXz,alphaP,k,mu,sig,c);
    E_all_Tsave(kk,:) = E_all_T;
    % Put zeros to avoid NaN in integration.
    E_all_T = [E_all_T(1:10) zeros(1,F-10)];
```

```
    Eservice_ob_T(kk) = sum(E_all_T.*af);
end
end
%
% Illustration
% Plot Total Service Simulations/Analytical
figure(1), plot(Mtvec, Pservice_ob_S,'rv '), hold on,
plot(Mtvec, Pservice_ob_T,'bv-')
% Plot Expected Service Analytical
if c<6
    figure(2), hold on, plot(Mtvec, Eservice_ob_T, 'ko-')
end
%
% Save something...
% save PservicePARvidF200.mat zf5 Pservice_ob_S Pservice_ob_T
% save PserviceEXPvidF200.mat zf3 Pservice_ob_S Pservice_ob_T
% save PservicePARvidF200D.mat zf Pservice_ob_S Pservice_ob_T
% save PserviceDESCENDaud.mat zf4 zf Pservice_ob_S Pservice_ob_T
```