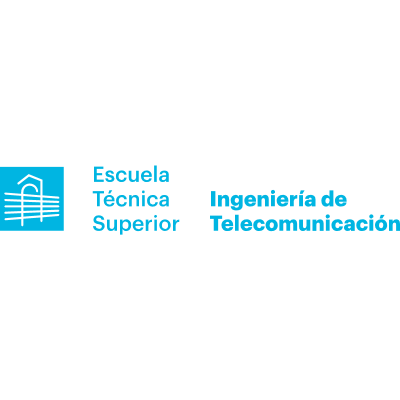


Diego Ismael Antolinos García Andrés Ruz Nieto

TRABAJO - MODELADO



# Código empleado

clc;

clear;

% Configuracion de parametros

t\_prop = 0.001; %tiempo de propagacion en s

bits\_fichero = 10000; % numero de bits a transmitir

R = 500000; %tasa de transmision en bits/s

p\_tx\_dB = 10; %potencia de transmisiÃ³n en dBm

p\_tx = 10^(p\_tx\_dB/10)\*0.001; %potencia de transmisiÃ³n en W

P\_b = ProbErrorBit(R, p\_tx);

Eb = 1000\*p\_tx/R; % energia por bit transmitido enmilijulios/bit

carga\_min = 100;

carga\_max = 4000;

carga\_incremento = 100;

cargas = carga\_min:carga\_incremento:carga\_max;

% ASIGNE UN VALOR A LA CABECERA DEL PAQUETE ENTRE 20 Y 120

L\_cabecera = 20; % longitud cabecera en bits

T = zeros(1,length(cargas));

E = zeros(1,length(cargas));

index = 0;

for L\_carga = cargas

index = index + 1;

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %%%%%%% AÑADA SU CODIGO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = ceil(bits\_fichero/L\_carga); % numero de paquetes dado bits\_fichero y L\_carga

entero = 1;

if(mod(bits\_fichero,L\_carga)~=0)

entero = 0;

end

% calculo de la longitud de los paquetes

L = L\_carga + L\_cabecera;

if(entero == 0)

L\_ult = mod(bits\_fichero,L\_carga) + L\_cabecera;

end

% calculo de las probabilidades de error

prob\_error = 1-(1-P\_b)^L;

if(entero == 0)

prob\_error\_ultimo = 1-(1-P\_b)^L\_ult;

end

% generar P

P = zeros(n+1);

for i = 1:n

P(i,i)=prob\_error;

P(i,i+1)=(1-prob\_error);

end

if(entero == 0)

P(n,n) = prob\_error\_ultimo;

P(n,n+1) = 1-prob\_error\_ultimo;

end

P(end,end) = 1;

M = P(1:n,1:n);

% generar g\_time

g\_time = zeros(n,1);

for i = 1:n

g\_time(i)= (L/R) + (2\*t\_prop);

end

if(entero == 0)

g\_time(n)= (L\_ult/R) + (2\*t\_prop);

end

% generar g\_energy

g\_energy = zeros(n,1);

for i = 1:n

g\_energy(i)= L\*Eb;

end

if(entero == 0)

g\_energy(n)= L\_ult\*Eb;

end

% obtener v\_time

v\_time = (eye(n)-M)\g\_time;

% obtener v\_energy

v\_energy = (eye(n)-M)\g\_energy;

T(index) = v\_time(1);

E(index) = v\_energy(1);

end

% obtener el valor de L\_carga que minimiza T

[valot\_T, indice\_T] = min(T);

L\_carga\_T\_min = cargas(indice\_T)

% obtener el valor de L\_carga que minimiza E

[valot\_E, indice\_E] = min(E);

L\_carga\_E\_min = cargas(indice\_E)

plot(cargas,T);

figure;

plot(cargas,E);

# Código empleado en la ampliación

clc;

clear;

% Configuracion de parametros

t\_prop = 0.001; %tiempo de propagacion en s

bits\_fichero = 10000; % numero de bits a transmitir

p\_tx\_dB = 10; %potencia de transmisiÃ³n en dBm

p\_tx = 10^(p\_tx\_dB/10)\*0.001; %potencia de transmisiÃ³n en W

carga\_min = 100;

carga\_max = 4000;

carga\_incremento = 100;

cargas = carga\_min:carga\_incremento:carga\_max;

Rvector = 100000:10000:600000;

% ASIGNE UN VALOR A LA CABECERA DEL PAQUETE ENTRE 20 Y 120

L\_cabecera = 20; % longitud cabecera en bits

T = zeros(length(cargas),length(Rvector));

E = zeros(length(cargas),length(Rvector));

index1 = 0;

index2 = 0;

for L\_carga = cargas

index1 = index1 + 1;

index2 = 0;

for R = Rvector

P\_b = ProbErrorBit(R, p\_tx);

Eb = 1000\*p\_tx/R; % energia por bit transmitido en milijulios/bit

index2 = index2 + 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %%%%%%% AÑADA SU CODIGO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = ceil(bits\_fichero/L\_carga); % numero de paquetes dado bits\_fichero y L\_carga

entero = 1;

if(mod(bits\_fichero,L\_carga)~=0)

entero = 0;

end

% calculo de la longitud de los paquetes

L = L\_carga + L\_cabecera;

if(entero == 0)

L\_ult = mod(bits\_fichero,L\_carga) + L\_cabecera;

end

% calculo de las probabilidades de error

prob\_error = 1-(1-P\_b)^L;

if(entero == 0)

prob\_error\_ultimo = 1-(1-P\_b)^L\_ult;

end

% generar P

P = zeros(n+1);

for i = 1:n

P(i,i)=prob\_error;

P(i,i+1)=(1-prob\_error);

end

if(entero == 0)

P(n,n) = prob\_error\_ultimo;

P(n,n+1) = 1-prob\_error\_ultimo;

end

P(end,end) = 1;

M = P(1:n,1:n);

% generar g\_time

g\_time = zeros(n,1);

for i = 1:n

g\_time(i)= (L/R) + (2\*t\_prop);

end

if(entero == 0)

g\_time(n)= (L\_ult/R) + (2\*t\_prop);

end

% generar g\_energy

g\_energy = zeros(n,1);

for i = 1:n

g\_energy(i)= L\*Eb;

end

if(entero == 0)

g\_energy(n)= L\_ult\*Eb;

end

% obtener v\_time

v\_time = (eye(n)-M)\g\_time;

% obtener v\_energy

v\_energy = (eye(n)-M)\g\_energy;

T(index1,index2) = v\_time(1);

E(index1,index2) = v\_energy(1);

end

end

% obtener el valor de L\_carga que minimiza T

min\_elemento\_T = min(T(:));

[L\_carga\_T, R\_T] = find(T==min\_elemento\_T);

L\_carga\_T\_min = cargas(L\_carga\_T)

R\_T\_min = Rvector(R\_T)

% obtener el valor de L\_carga que minimiza E

min\_elemento\_E = min(E(:));

[L\_carga\_E, R\_E] = find(E==min\_elemento\_E);

L\_carga\_E\_min = cargas(L\_carga\_E)

R\_E\_min = Rvector(R\_E)

figure

mesh(Rvector,cargas,T);

figure;

mesh(Rvector,cargas,E);

# Datos obtenidos

## Parte I

## Parte II (ampliación)

# Gráficas obtenidas

# 

Tiempo de envío respecto la carga del paquete

# 

Energía consumida respecto la carga del paquete



Energía respecto la carga del paquete y su velocidad de transmisión



Tiempo de envío respecto la carga del paquete y su velocidad de transmisión