



COOPERATIVE STRATEGIES FOR IMPERFECT CSI SCENARIOS
BASED ON DISTRIBUTED ALAMOUTI:

MATLAB CODE

Memòria del Projecte Fi de Carrera
d' Enginyeria Tècnica de Telecomunicació.
Especialitat de Sistemes Electrònics.

realitzat per

Marc Barceló Lladó

i dirigit per

José López Vicario

Bellaterra, 18 de Setembre de 2008

Contents

Chapter 3: Reference model	4
Chapter 5: Cooperative System from an Information Theory Viewpoint	5
<i>Without distances</i>	5
<i>Relays Position</i>	6
<i>Taking Distances into Account</i>	10
<i>Feedback Error</i>	13
Chapter 6: Real Protocol	20
<i>Without Feedback Error</i>	20
<i>With Feedback Error</i>	26
<i>Feedback Error Probability Aware Protocol</i>	33
<i>Feedback Error Probability and Average SNR Aware Protocol Criterion</i>	38
<i>Feedback Error Probability and Average SNR Aware Protocol</i>	43
<i>Number of Relays and Average SNR Aware Protocol</i>	48
<i>Protocol Allocation Protocol Criterion</i>	54
<i>Power Allocation Protocol</i>	58
<i>Rho Analysis</i>	62
<i>Correlated Channels</i>	63
<i>Correlated Channel Protocol over Feedback Error Probability and Average SNR Aware Protocol</i>	71
<i>Correlated Channel Protocol over Number of Relays and Average SNR Aware Protocol</i>	77

Chapter 3: Reference model

```
I=10000;
N=12;

R=2;

%Best Relay

for snrm=1:30
    canales_outage=0;

    for aux=1:I

        r1=sqrt(0.5)*(randn(N,1)+j*randn(N,1));           %Canales Fuente-Relay
        h1=sqrt(0.5)*(randn(N,1)+j*randn(N,1));           %Canales Relay-Destino
        snrr1=(abs(r1).^2)*snrm;

        vect_outage=snrr1>=(2^(2*R)-1);

        if(sum(vect_outage)>0)
            vect_sec_link=sort(abs(h1).^2.*vect_outage,'descend');
            snrr2=snrm*vect_sec_link(1);
            canales_outage=canales_outage+(snrr2<(2^(2*R)-1));
        else
            canales_outage=canales_outage+1;
        end
    end

    prob_outage_Mejor(snrm)=canales_outage/I;
end

%Direct transmission

for snrm=1:30
    canales_outage=0;

    for aux=1:I

        r1=sqrt(0.5)*(randn(1,1)+j*randn(1,1));           %Canales Fuente-Destino

        snrr2=snrm*(abs(r1).^2);
        canales_outage=canales_outage+(snrr2<(2^(2*R)-1));

    end

    prob_outage_Directo(snrm)=canales_outage/I;
    vector_snrm(snrm)=snrm;

end
```

Chapter 5: Cooperative System from an Information Theory Viewpoint

Without distances

```
clear all
%close all

I=50000;
N=30;
R=2;

% %Alamouti

for snrm=1:30
    canales_outage=0;

    for aux=1:I

        r1=sqrt(0.5)*(randn(N,1)+j*randn(N,1));           %Canales Fuente-Relay
        h1=sqrt(0.5)*(randn(N,1)+j*randn(N,1));           %Canales Relay-Destino
        snrr1=(abs(r1).^2)*snrm;                           %SNR de los canales
        Fuente-Relay

        vect_outage=snrr1>=(2^(2*R)-1);

        if(sum(vect_outage)>=2)
            vect_sec_link=sort(abs(h1).^2.*vect_outage,'descend');
            snrr2=(snrm/2)*(sum(vect_sec_link(1:2)));
            canales_outage=canales_outage+(snrr2<(2^(2*R)-1));
        else
            canales_outage=canales_outage+1;
        end
    end

    end

    vector_snrm(snrm)=snrm;
    prob_outage(snrm)=canales_outage/I;

end

semilogy(vector_snrm,prob_outage,'b');
hold on;

%Best Relay

for snrm=1:30
    canales_outage=0;

    for aux=1:I

        r1=sqrt(0.5)*(randn(N,1)+j*randn(N,1));           %Canales Fuente-Relay
        h1=sqrt(0.5)*(randn(N,1)+j*randn(N,1));           %Canales Relay-Destino
        snrr1=(abs(r1).^2)*snrm;
```

```

vect_outage=snrr1>=(2^(2*R)-1);

if(sum(vect_outage)>0)
    vect_sec_link=sort(abs(h1).^2.*vect_outage,'descend');
    snrr2=snrm*vect_sec_link(1);
    canales_outage=canales_outage+(snrr2<(2^(2*R)-1));
else
    canales_outage=canales_outage+1;
end

end

vector_snrn(snrn)=snrm;
prob_outage(snrn)=canales_outage/I;

end

%semilogy(vector_snrn,prob_outage,'g');

```

Relays Position

Line

```

clear all;

I=30000;
N=10;
R=2;
d=100;

trozo=d/(2*(N-1));

for aux=1:N
    distancia(aux)=d/4+trozo*(aux-1);
end

for aux=1:N
    distancia(aux)=d/4+trozo*(aux-1);
end

for aux=1:N
    distancia2(aux)=distancia(N+1-aux);
end

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;

for snrm=1:30
    canales_outage=0;

```

```

snrml=10^(snrm/10);

varianza1(snrm)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
Ps=snrml*Pn/varianza1(snrm);

for aux=1:I
    for cont=1:N
        varianza2=((distancia(cont)/d0)^(-mu)*(landac/(4*pi*d0))^2;
        r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
    end

    for cont=1:N
        varianza3=((distancia2(cont)/d0)^(-mu)*(landac/(4*pi*d0))^2;
        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
    end

    snrr1=(abs(r).^2)*Ps/(Pn*2);

    vect_outage=snrr1>=(2^(2*R)-1);

    if(sum(vect_outage)>=2)
        vect_sec_link=sort(abs(h).^2.*vect_outage,'descend');
        snrr2=(Ps/(Pn*4))*(sum(vect_sec_link(1:2)));
        canales_outage=canales_outage+(snrr2<(2^(2*R)-1));
    else
        canales_outage=canales_outage+1;
    end
end

vector_snrml(snrm)=snrm;
prob_outage(snrm)=canales_outage/I;

end

posicion1=find(prob_outage<1e-3);
valor1=min(posicion1);
Ps_Alamouti=Pn*vector_snrml(valor1)/varianza1(valor1);

%Enlace directo

for snrm=1:30
    canales_outage=0;
    snrml=10^(snrm/10);
    varianza1(snrm)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1(snrm);

    for aux=1:I

        r=sqrt(0.5)*sqrt(varianza1(snrm))*(randn(1,1)+j*randn(1,1));
        snrr1=(abs(r).^2)*(Ps/Pn);

```

```

if snrr1<((2^R)-1)
    canales_outage=canales_outage+1;
end

end
prob_outage_directo(snm)=canales_outage/I;

end

%semilogy(vector_snm,prob_outage_directo,'b');

hold on;
semilogy(vector_snm,prob_outage,'k');

% xlabel('SNRM');
% ylabel('Probabilidad de Outage');

posicion2=find(prob_outage_directo<1e-3);
valor2=min(posicion2);
Ps_directo=Pn*vector_snm(valor2)/varianza1(valor2);

```

Rectangular

```

%close all;
clear all;

I=30000;
N=10;
R=2;
d=100;

trozo=d/(2*(N-1));

if rem(N,2)==0
    limite=N/2;
end

if rem(N,2)==1
    limite=N/2+0.5;
end

for aux=1:limite
    distancia(aux)=sqrt((d/4+trozo*(aux-1))^2+(d/4)^2);
end

for aux=(limite+1):N
    distancia(aux)=sqrt((d/4+trozo*(aux-limite-1))^2+(d/4)^2);
end

distancia(N)=sqrt((d/4+d/2)^2+(d/4)^2);
distancia(limite)=sqrt((d/4+d/2)^2+(d/4)^2);

for aux=1:limite
    distancia2(aux)=distancia(limite+1-aux);
end

for aux=(limite+1):N
    distancia2(aux)=distancia(N+1+limite-aux);
end

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;

```



```

Pn=1;

for snrm=1:30
    canales_outage=0;
    snrml=10^(snrm/10);

    varianzal(snrm)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianzal(snrm);

    for aux=1:I

        for cont=1:N

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

            end

            for cont=1:N

                varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
                h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

                end

                snrr1=(abs(r).^2)*Ps/(Pn*2);

                vect_outage=snrr1>=(2^(2*R)-1);

                if(sum(vect_outage)>=2)

                    vect_sec_link=sort(abs(h).^2.*vect_outage,'descend');
                    snrr2=(Ps/(Pn*4))*(sum(vect_sec_link(1:2)));
                    canales_outage=canales_outage+(snrr2<(2^(2*R)-1));

                else

                    canales_outage=canales_outage+1;

                end

            end

            vector_snrm(snrm)=snrm;
            prob_outage(snrm)=canales_outage/I;

        end

    end

    posicion1=find(prob_outage<1e-3);
    valor1=min(posicion1);
    Ps_Alamouti=Pn*vector_snrm(valor1)/varianzal(valor1);

    %Enlace directo

    for snrm=1:30
        canales_outage=0;
        snrml=10^(snrm/10);
        varianzal(snrm)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
        Ps=snrml*Pn/varianzal(snrm);
    end

```

```

for aux=1:I

r=sqrt(0.5)*sqrt(varianza1(snmr))*(randn(1,1)+j*randn(1,1));
snrr1=(abs(r).^2)*(Ps/Pn);

if snrr1<((2^R)-1)
    canales_outage=canales_outage+1;
end

end
prob_outage_directo(snmr)=canales_outage/I;

end

%semilogy(vector_snmr,prob_outage_directo,'b');

hold on;
semilogy(vector_snmr,prob_outage,'g');

xlabel('SNRM');
ylabel('Probabilidad de Outage');

posicion2=find(prob_outage_directo<1e-3);
valor2=min(posicion2);
Ps_directo=Pn*vector_snmr(valor2)/varianza1(valor2);

```

Taking Distances into Account

```

clear all
%close all

I=50000;
N=10;
R=2;
d=100;

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;

%Alamouti

for snrm=1:30
    canales_outage=0;
    snrml=10^(snrm/10);

    varianza1(snmr)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1(snmr);

```

```

for aux=1:I
    for cont=1:N
        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
    end

    for cont=1:N
        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
    end

    snrr1=(abs(r).^2)*Ps/(Pn*2);

    vect_outage=snrr1>=(2^(2*R)-1);

    if(sum(vect_outage)>=2)

        vect_sec_link=sort(abs(h).^2.*vect_outage,'descend');
        snrr2=(Ps/(Pn*4))*(sum(vect_sec_link(1:2)));
        canales_outage=canales_outage+(snrr2<(2^(2*R)-1));

    else

        canales_outage=canales_outage+1;

    end

end

vector_snrn(snrn)=snrn;
prob_outage_Alamouti(snrn)=canales_outage/I;

end

% posicion1=find(prob_outage<1e-3);
% valor1=min(posicion1);
% Ps_Alamouti=Pn*vector_snrn(valor1)/varianza1(valor1);

%Enlace directo

for snrm=1:30
    canales_outage=0;
    snrm1=10^(snrm/10);
    varianza1(snrn)=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrm1*Pn/varianza1(snrn);

for aux=1:I

r=sqrt(0.5)*sqrt(varianza1(snrn))*(randn(1,1)+j*randn(1,1));
snrr1=(abs(r).^2)*(Ps/Pn);

if snrr1<((2^R)-1)
    canales_outage=canales_outage+1;
end

```

```

end
prob_outage_Directo(snrn)=canales_outage/I;

end

%Best Relay

for snrm=1:30
    canales_outage=0;
    snrml=10^(snrm/10);

    varianza1(snrn)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1(snrn);

    for aux=1:I

        for cont=1:N

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

            end

            for cont=1:N

                varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
                h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

                end

                snrr1=(abs(r).^2)*Ps/(Pn*2);

                vect_outage=snrr1>=(2^(2*R)-1);

                if(sum(vect_outage)>=2)

                    vect_sec_link=sort(abs(h).^2.*vect_outage,'descend');
                    snrr2=(Ps/(Pn*2))*(sum(vect_sec_link(1)));
                    canales_outage=canales_outage+(snrr2<(2^(2*R)-1));

                else

                    canales_outage=canales_outage+1;

                end

            end

        end

        vector_snrn(snrn)=snrm;
        prob_outage_Mejor(snrn)=canales_outage/I;

    end

    % posicion1=find(prob_outage<1e-3);
    % valor1=min(posicion1);
    % Ps_Alamouti=Pn*vector_snrn(valor1)/varianza1(valor1);

    % hold on;
    % semilogy(vector_snrn,prob_outage_directo,'r');
    %
    % hold on;

```

```

% semilogy(vector_snr,prob_outage,'b');
%
% xlabel('SNRM');
% ylabel('Probabilidad de Outage');
%
% posicion2=find(prob_outage_directo<1e-3);
% valor2=min(posicion2);
% Ps_directo=Pn*vector_snr(valor2)/varianza1(valor2);

```

Feedback Error

```

clear all;
%close all;

I=100000;
N=10;
R=2;
d=100;

%Calculen les dist`ncies aleatorias de cada destination.

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
Probe=0.3;
L=10;

%Elegimos seg`n mejor SNR y usamos Alamouti

for snrm=1:30
canales_outage_Alamouti=0;
rate_medio_Alamouti=0;
rateslot=R;
snrml=10^(snrm/10);
Nslots=0;

varianza1(snrml)=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
Ps=snrml*Pn/varianza1(snrml);

for aux=1:I

for cont=1:N

varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

end

```

```

for cont=1:N
    varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
    h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
end

Nslots=Nslots+1;

snrr1=(abs(r).^2)*Ps/(Pn*2);

vect_outage=snrr1>=(2^(2*R)-1);

if(sum(vect_outage)>=2)

    %Miramos el feedback error

    vect_sec_link=abs(h).^2.*vect_outage;
    [maxim,pos1]=max(vect_sec_link);           %Cogemos el m+ximo
    vect_sec_link2(1)=vect_sec_link(pos1);    %Lo guardamos
    vect_sec_link(pos1)=0;                   %Ponemos su posicion
a 0
maximo
    [maxim2,pos2]=max(vect_sec_link);         %Cogemos el siguiente
    vect_sec_link(pos1)=vect_sec_link2(1);    %Volvemos a poner el
valor m+ximo en su posicion

    pos1=pos1-1; %Para que empiezen las posiciones en 0
    pos2=pos2-1;

    b1=dec2bin(pos1); %Pasamos las posiciones a binario
    b2=dec2bin(pos2);

    a=rand(1);
    b=rand(1);
    c=rand(1);
    dd=rand(1);

    l=0;

    l(4)=b1(length(b1));

    if pos1>1
        l(3)=b1(length(b1)-1);
    end

    if pos1>3
        l(2)=b1(length(b1)-2);
    end

    if pos1>7
        l(1)=b1(length(b1)-3);
    end

    l=l>'0';

```

```

if a<=Probe
    l(4)=not(l(4));

end

if b<=Probe && pos1>1
    l(3)=not(l(3));
end

if c<=Probe && pos1>3
    l(2)=not(l(2));
end

if dd<=Probe && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

l=0;

l(4)=b2(length(b2));

if pos2>1
    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=1>'0';

if a<=Probe
    l(4)=not(l(4));

end

if b<=Probe && pos1>1
    l(3)=not(l(3));
end

if c<=Probe && pos1>3
    l(2)=not(l(2));
end

if dd<=Probe && pos1>7
    l(1)=not(l(1));
end

```

```

%Convertimos binario a decimal

eleccion2=1(4)+2*1(3)+4*1(2)+8*1(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if N<eleccion2
    vect_sec_link(eleccion2)=0;
end

snrr2=(Ps/
(Pn*4))*(vect_sec_link(eleccion1)+vect_sec_link(eleccion2));

canales_outage_Alamouti=canales_outage_Alamouti+
(snrr2<(2^(2*R)-1));

if snrr2>=(2^(2*R)-1)
    rate_medio_Alamouti=rate_medio_Alamouti+rateslot;
else
    rate_medio_Alamouti=rate_medio_Alamouti+0;
end

else

canales_outage_Alamouti=canales_outage_Alamouti+1;
rate_medio_Alamouti=rate_medio_Alamouti+0;

end

end

vector_snrn(snrn)=snrn;
prob_outage_Alamouti(snrn)=canales_outage_Alamouti/I;
prob_error_paquete_Alamouti(snrn)=1-(1-
prob_outage_Alamouti(snrn))^L;
rate_Alamouti(snrn)=rate_medio_Alamouti/Nslots;

end

%hold on;
%semilogy(vector_snrn,prob_outage,'b');
%plot(vector_snrn,rate_Alamouti,'r');

%Eligiendo solamente 1

for snrm=1:30
    Nslots=0;
    rate_medio_Mejor=0;
    canales_outage_Mejor=0;
    snrml=10^(snrm/10);

```



```

varianza1(snrml)=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
Ps=snrml*Pn/varianza1(snrml);

for aux=1:I

    for cont=1:N

        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

        end

    for cont=1:N

        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

        end

snrr1=(abs(r).^2)*Ps/(Pn*2);
vect_outage=snrr1>=(2^(2*R)-1);

Nslots=Nslots+1;

if(sum(vect_outage)>=1)

    %Miramos los errores de selecci3n para 4 Relays 00,01,10,11.
    %Enviamos dos pares de bits para cada 1

    vect_sec_link=abs(h).^2.*vect_outage;
    [maxim,pos1]=max(vect_sec_link);           %Cogemos el m3ximo
    pos1=pos1-1;                               %Para que empiezen

    las posiciones en 0

    b1=dec2bin(pos1);                           %Pasamos las posiciones
    a binario

    a=rand(1);
    b=rand(1);
    c=rand(1);
    dd=rand(1);

    l=0;

    l(4)=b1(length(b1));

    if pos1>1

        l(3)=b1(length(b1)-1);
    end

    if pos1>3
        l(2)=b1(length(b1)-2);
    end

    if pos1>7
        l(1)=b1(length(b1)-3);
    end
end

```

```

l=l>'0';

if a<=Probe
    l(4)=not(l(4));
end

if b<=Probe && pos1>1
    l(3)=not(l(3));
end

if c<=Probe && pos1>3
    l(2)=not(l(2));
end

if dd<=Probe && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal
eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

snrr2=(Ps/(Pn*2))*(vect_sec_link(eleccion1));

canales_outage_Mejor=canales_outage_Mejor+
(snrr2<(2^(2*R)-1));

if snrr2>=(2^(2*R)-1)
    rate_medio_Mejor=rate_medio_Mejor+rateslot;
else
    rate_medio_Mejor=rate_medio_Mejor+0;
end

else

canales_outage_Mejor=canales_outage_Mejor+1;
rate_medio_Mejor=rate_medio_Mejor+0;

end

end

vector_snrn(snrn)=snrn;
prob_outage_Mejor(snrn)=canales_outage_Mejor/I;
prob_error_paquete_Mejor(snrn)=1-(1-prob_outage_Mejor(snrn))^L;
rate_relay_Mejor(snrn)=rate_medio_Mejor/Nslots;

end

%hold on;
%semilogy(vector_snrn,prob_outage,'g');
%plot(vector_snrn,rate_relay,'k');

```

```

xlabel('SNR media');
ylabel('Rate');

for snrm=1:30
    Nslots=0;
    rate_medio_Directo=0;
    canales_outage_Directo=0;
    snrml=10^(snrm/10);
    rateslot=R;

    varianzal(snrml)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianzal(snrml);

    for aux=1:I
        e=sqrt(0.5)*sqrt(varianzal)*(randn(1,1)+j*randn(1,1));

        snrrl=(abs(e).^2)*Ps/(Pn);

        vect_outage=snrrl>=(2^(2*R)-1);

        Nslots=Nslots+1;

        if(vect_outage==1)

            if snrrl>=(2^(2*R)-1)
                rate_medio_Directo=rate_medio_Directo+rateslot;
            else
                rate_medio_Directo=rate_medio_Directo+0;
            end

        else

            canales_outage_Directo=canales_outage_Directo+1;
            rate_medio_Directo=rate_medio_Directo+0;

        end

    end

    vector_snrml(snrml)=snrm;
    prob_outage_Directo(snrml)=canales_outage_Directo/I;
    prob_error_paquete_Directo(snrml)=1-(1-prob_outage_Directo(snrml))^L;
    rate_relay_Directo(snrml)=rate_medio_Directo/Nslots;

end

```

Chapter 6: Real Protocol

Without Feedback Error

```
clear all;
%close all;

I=100000;
N=10;
R=2;
d=100;

%Calculen les distancies aleatorias de cada destination.

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
Probe=0.3;
L=10;

%Elegimos segun mejor SNR y usamos Alamouti

for snrm=1:30
    canales_outage_Alamouti=0;
    rate_medio_Alamouti=0;
    rateslot=R;
    snrml=10^(snrm/10);
    Nslots=0;

    varianzal(snrm)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianzal(snrm);

    for aux=1:I

        for cont=1:N

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

            end

            for cont=1:N

                varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
                h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
```

```

end

Nslots=Nslots+1;

snrr1=(abs(r).^2)*Ps/(Pn*2);

vect_outage=snrr1>=(2^(2*R)-1);

if(sum(vect_outage)>=2)

    %Miramos el feedback error

    vect_sec_link=abs(h).^2.*vect_outage;
    [maxim,pos1]=max(vect_sec_link);           %Cogemos el m+ximo
    vect_sec_link2(1)=vect_sec_link(pos1);    %Lo guardamos
    vect_sec_link(pos1)=0;                   %Ponemos su posicion

a 0
maximo
    [maxim2,pos2]=max(vect_sec_link);         %Cogemos el siguiente
    vect_sec_link(pos1)=vect_sec_link2(1);   %Volvemos a poner el
valor m+ximo en su posicion

    pos1=pos1-1; %Para que empiezen las posiciones en 0
    pos2=pos2-1;

    b1=dec2bin(pos1); %Pasamos las posiciones a binario
    b2=dec2bin(pos2);

    a=rand(1);
    b=rand(1);
    c=rand(1);
    dd=rand(1);

    l=0;

    l(4)=b1(length(b1));

    if pos1>1

        l(3)=b1(length(b1)-1);
    end

    if pos1>3
        l(2)=b1(length(b1)-2);
    end

    if pos1>7
        l(1)=b1(length(b1)-3);
    end

    l=l>'0';

    if a<=Probe
        l(4)=not(l(4));
    end

end

if b<=Probe && pos1>1
    l(3)=not(l(3));
end

```

```

end

if c<=Probe && pos1>3
    l(2)=not(l(2));
end

if dd<=Probe && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

l=0;

l(4)=b2(length(b2));

if pos2>1
    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

if a<=Probe
    l(4)=not(l(4));

end

if b<=Probe && pos1>1
    l(3)=not(l(3));
end

if c<=Probe && pos1>3
    l(2)=not(l(2));
end

if dd<=Probe && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

```

```

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if N<eleccion2
    vect_sec_link(eleccion2)=0;
end

snrr2=(Ps/
(Pn*4))*(vect_sec_link(eleccion1)+vect_sec_link(eleccion2));

canales_outage_Alamouti=canales_outage_Alamouti+
(snrr2<(2^(2*R)-1));

if snrr2>=(2^(2*R)-1)
    rate_medio_Alamouti=rate_medio_Alamouti+rateslot;
else
    rate_medio_Alamouti=rate_medio_Alamouti+0;
end

else

canales_outage_Alamouti=canales_outage_Alamouti+1;
rate_medio_Alamouti=rate_medio_Alamouti+0;

end

end

vector_snrn(snrn)=snrn;
prob_outage_Alamouti(snrn)=canales_outage_Alamouti/I;
prob_error_paquete_Alamouti(snrn)=1-(1-
prob_outage_Alamouti(snrn))^L;
rate_Alamouti(snrn)=rate_medio_Alamouti/Nslots;

end

%hold on;
%semilogy(vector_snrn,prob_outage,'b');
%plot(vector_snrn,rate_Alamouti,'r');

%Eligiendo solamente 1

for snrm=1:30
    Nslots=0;
    rate_medio_Mejor=0;
    canales_outage_Mejor=0;
    snrml=10^(snrm/10);

    varianzal(snrn)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianzal(snrn);

    for aux=1:I

        for cont=1:N

```

```

        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

    end

    for cont=1:N

        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

    end

    snrr1=(abs(r).^2)*Ps/(Pn*2);

    vect_outage=snrr1>=(2^(2*R)-1);

    Nslots=Nslots+1;

    if(sum(vect_outage)>=1)

        %Miramos los errores de selecci-n para 4 Relays 00,01,10,11.
        %Enviamos dos pares de bits para cada 1

        vect_sec_link=abs(h).^2.*vect_outage;
        [maxim,pos1]=max(vect_sec_link);           %Cogemos el m#xximo
        pos1=pos1-1;                               %Para que empiezen
las posiciones en 0

        b1=dec2bin(pos1);                          %Pasamos las posiciones
a binario

        a=rand(1);
        b=rand(1);
        c=rand(1);
        dd=rand(1);

        l=0;

        l(4)=b1(length(b1));

        if pos1>1

            l(3)=b1(length(b1)-1);
        end

        if pos1>3
            l(2)=b1(length(b1)-2);
        end

        if pos1>7
            l(1)=b1(length(b1)-3);
        end

        l=l>'0';

        if a<=Probe
            l(4)=not(l(4));

```



```

end

if b<=Probe && pos1>1
    l(3)=not(l(3));
end

if c<=Probe && pos1>3
    l(2)=not(l(2));
end

if dd<=Probe && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

snrr2=(Ps/(Pn*2))*(vect_sec_link(eleccion1));

canales_outage_Mejor=canales_outage_Mejor+
(snrr2<(2^(2*R)-1));

if snrr2>=(2^(2*R)-1)
    rate_medio_Mejor=rate_medio_Mejor+rateslot;
else
    rate_medio_Mejor=rate_medio_Mejor+0;
end

else

canales_outage_Mejor=canales_outage_Mejor+1;
rate_medio_Mejor=rate_medio_Mejor+0;

end

end

vector_snrn(snrn)=snrn;
prob_outage_Mejor(snrn)=canales_outage_Mejor/I;
prob_error_paquete_Mejor(snrn)=1-(1-prob_outage_Mejor(snrn))^L;
rate_relay_Mejor(snrn)=rate_medio_Mejor/Nslots;

end

%hold on;
%semilogy(vector_snrn,prob_outage,'g');
%plot(vector_snrn,rate_relay,'k');

xlabel('SNR media');
ylabel('Rate');

for snrm=1:30

```

```

Nslots=0;
rate_medio_Directo=0;
canales_outage_Directo=0;
snrml=10^(snrm/10);
rateslot=R;

varianza1(snm)=(d/d0)^(-mu)*(landac/(4*pi*d0))^2;
Ps=snrml*Pn/varianza1(snm);

for aux=1:I

e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

snrr1=(abs(e).^2)*Ps/(Pn);

vect_outage=snrr1>=(2^(2*R)-1);

Nslots=Nslots+1;

if(vect_outage==1)

    if snrr1>=(2^(2*R)-1)
        rate_medio_Directo=rate_medio_Directo+rateslot;
    else
        rate_medio_Directo=rate_medio_Directo+0;
    end

else

    canales_outage_Directo=canales_outage_Directo+1;
    rate_medio_Directo=rate_medio_Directo+0;

end

end

vector_snm(snm)=snrm;
prob_outage_Directo(snm)=canales_outage_Directo/I;
prob_error_paquete_Directo(snm)=1-(1-prob_outage_Directo(snm))^(L);
rate_relay_Directo(snm)=rate_medio_Directo/Nslots;

end

```

With Feedback Error

```

clear all;
%close all;

I=300000;
N=7;
R=2;
d=100;

%Calculen les distncies aleatorias de cada destination.

distanciay=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciay.^2+distanciay.^2);

distanciay2=d-distanciay;

```

```

distancia2=sqrt(distanciay.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.

%PER Directo
for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo_Directo=0;
    error_simbolo_Mejor=0;
    error_simbolo_Alamouti=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

    varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1;

    for aux=1:I

        e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

        snr_directo=(abs(e).^2)*Ps/(Pn);
        Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
        outage_directo=Pesim_directo<Pesim;

        if outage_directo==0
            error_simbolo_Directo=error_simbolo_Directo+1;
        else
            error_simbolo_Directo=error_simbolo_Directo+0;
        end
    end

    vector_snrm(snrm)=snrm;
    prob_outage_Directo(snrm)=error_simbolo_Directo/I;
    prob_error_paquete_Directo(snrm)=1-(1-prob_outage_Directo(snrm))^L;

%Best Relay

for aux=1:I

    for cont=1:N

        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

    end

    for cont=1:N

        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
    end
end

```

```

end

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

%Miramos los relays que est#n en outage para transmitir al
Destination

vect_sec_link=abs(h).^2.*outage_relays;
snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
Pesim_relays2=Pesim_relays2.*outage_relays;
outage_relays2=Pesim_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

Pebit=Pesim_relays2/2;

%Calculamos las mejores posiciones

vect_sec_link=abs(h).^2.*outage_relays;
[maxim,pos1]=max(vect_sec_link);           %Cogemos el m#ximo
vect_sec_link2=vect_sec_link(pos1);       %Lo guardamos
vect_sec_link(pos1)=0;                   %Ponemos su posicion a 0
[maxim2,pos2]=max(vect_sec_link);        %Cogemos el siguiente maximo
vect_sec_link(pos1)=vect_sec_link2;

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

pos1=pos1-1; %Para que empiezen las posiciones en 0
pos2=pos2-1;

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

```

```

if pos1>1
    l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);
eleccion1=eleccion1+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if(sum(outage_relays)>0)

    snr_mejor=(Ps/(Pn*2))*vect_sec_link(eleccion1);
    Pesim_mejor=2*(1/2*erfc(sqrt(snr_mejor/2)))-
(1/2*erfc(sqrt(snr_mejor/2))).^2;
    outage_mejor=Pesim_mejor<Pesim;

    if outage_mejor==0
        error_simbolo_Mejor=error_simbolo_Mejor+1;
    else
        error_simbolo_Mejor=error_simbolo_Mejor+0;
    end

else
    error_simbolo_Mejor=error_simbolo_Mejor+1;
    rate_medio=rate_medio+0;
end
end

```

```

vector_snrn(snrn)=snrn;
prob_outage_Mejor(snrn)=error_simbolo_Mejor/I;
prob_error_paquete_Mejor(snrn)=1-(1-prob_outage_Mejor(snrn))^(L);

%Alamouti

for aux=1:I

    for cont=1:N

        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

    end

for cont=1:N

    varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
    h(cont,1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

end

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

%Miramos los relays que est#n en outage para transmitir al
Destination

vect_sec_link=abs(h).^2.*outage_relays;
snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
Pesim_relays2=Pesim_relays2.*outage_relays;
outage_relays2=Pesim_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

Pebit=Pesim_relays2/2;

%Calculamos las mejores posiciones

vect_sec_link=abs(h).^2.*outage_relays;
[maxim,pos1]=max(vect_sec_link); %Cogemos el mximo
vect_sec_link2=vect_sec_link(pos1); %Lo guardamos
vect_sec_link(pos1)=0; %Ponemos su posicion a 0
[maxim2,pos2]=max(vect_sec_link); %Cogemos el siguiente maximo

```

```

vect_sec_link(pos1)=vect_sec_link2;

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

pos1=pos1-1; %Para que empiezen las posiciones en 0
pos2=pos2-1;

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

if pos1>1
    l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1
    l(3)=b2(length(b2)-1);
end

```

```

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if N<eleccion2
    vect_sec_link(eleccion2)=0;
end

if(sum(outage_relays)>=2)

    snr_Alamouti=(Ps/
(Pn*4))* (vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
    Pesim_Alamouti=2*(2*(1/2*erfc(sqrt(snr_Alamouti/2)))-
(1/2*erfc(sqrt(snr_Alamouti/2))).^2);
    outage_Alamouti=Pesim_Alamouti<Pesim;

    if outage_Alamouti==0
        error_simbolo_Alamouti=error_simbolo_Alamouti+1;
    else
        error_simbolo_Alamouti=error_simbolo_Alamouti+0;
    end
else
    error_simbolo_Alamouti=error_simbolo_Alamouti+1;
end

```



```

end
end

vector_snrn(snrn)=snrn;
prob_outage_Alamouti(snrn)=error_simbolo_Alamouti/I;
prob_error_paquete_Alamouti(snrn)=1-(1-
prob_outage_Alamouti(snrn))^(L);
end

```

Feedback Error Probability Aware Protocol

```

clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculem les distancies aleatorias de cada destination.

distanciay=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciay.^2+distanciay.^2);

distanciay2=d-distanciay;
distancia2=sqrt(distanciay2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.

for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

    varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1;

    for aux=1:I

        for cont=1:N

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

            end

```

```

for cont=1:N
    varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
    h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
end

e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

snr_directo=(abs(e).^2)*Ps/(Pn*2);
Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
outage_directo=Pesim_directo<Pesim;

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

%Miramos los relays que est#n en outage para transmitir al
Destination

vect_sec_link=abs(h).^2.*outage_relays;
snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
Pesim_relays2=Pesim_relays2.*outage_relays;
outage_relays2=Pesim_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

%Miramos cual va a ser la velocidad necesaria

if outage_directo==1
    rate=4;
    Nslots=Nslots+1;
    Pebit=Pesim_directo/2;
else
    rate=2;
    Nslots=Nslots+1;
    Pebit=Pesim_relays2/2;
end

```

```

end

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<Pebit || b<Pebit || c<Pebit || dd<Pebit
    fed_error=1;
end

if outage_directo==0 %Miramos que pasa
cuando no se puede transmitir de forma directa

    if fed_error==0 %Si no tenemos
problemas con el feedback usamos solamente el mejor Relay

        if(sum(outage_relays)>0)

            %Todo esto ya est+ calculado de antes para poder
conocer
            %Pebit

vect_sec_link=sort(abs(h).^2.*outage_relays,'descend');
            snr_relays2=(Ps/(Pn*2))*vect_sec_link(1);
            Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
            outage_relays2=Pesim_relays2<Pesim;
% Pesim_relays2=sum(outage_relays2(1:end))/N;
% Pesim_relays2=1-Pesim_relays2;
% Pebit=Pesim_relays2/2;

            if outage_relays2==1
                rate_medio=rate_medio+rate;
            else
                rate_medio=rate_medio+0;
                error_simbolo=error_simbolo+1;
            end

        else
            error_simbolo=error_simbolo+1;
            rate_medio=rate_medio+0;
        end
    else %Si hay feedback error
usamos Alamouti

        if(sum(outage_relays)>=2)

            vect_sec_link=abs(h).^2.*outage_relays;
            [maxim,pos1]=max(vect_sec_link); %Cogemos el
m+ximo
            vect_sec_link2(1)=vect_sec_link(pos1); %Lo
guardamos
            vect_sec_link(pos1)=0; %Ponemos su
posicion a 0
            [maxim2,pos2]=max(vect_sec_link); %Cogemos el
siguiente maximo
            vect_sec_link(pos1)=vect_sec_link2(1); %Volvemos a
poner el valor m+ximo en su posicion

            pos1=pos1-1; %Para que empiezen las posiciones en 0
            pos2=pos2-1;

```

```

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

if pos1>1

    l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

```

```

        if a<=Pebit
            l(4)=not(l(4));
        end

        if b<=Pebit && pos1>1
            l(3)=not(l(3));
        end

        if c<=Pebit && pos1>3
            l(2)=not(l(2));
        end

        if dd<=Pebit && pos1>7
            l(1)=not(l(1));
        end

        %Convertimos binario a decimal

        eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

        eleccion1=eleccion1+1;
        eleccion2=eleccion2+1;

        %En caso de haber seleccionado un relay que no
existe
        %ponemos ese valor a 0.

        if N<eleccion1
            vect_sec_link(eleccion1)=0;
        end

        if N<eleccion2
            vect_sec_link(eleccion2)=0;
        end

        snr_Alamouti=(Ps/
(Pn*4))*(vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
        Pesim_Alamouti=2*2*(1/2*erfc(sqrt(snr_Alamouti/
2)))-(1/2*erfc(sqrt(snr_Alamouti/2))).^2;
        outage_Alamouti=Pesim_Alamouti<Pesim;

        if outage_Alamouti==1
            rate_medio=rate_medio+rate;
        else
            rate_medio=rate_medio+0;
            error_simbolo=error_simbolo+1;
        end

    else
        error_simbolo=error_simbolo+1;
        rate_medio=rate_medio+0;
    end

end
else
    error_simbolo=error_simbolo;
    rate_medio=rate_medio+rate;
end
end
end

```

```

vector_snrn(snrn)=snrn;
prob_error_simbolo(snrn)=error_simbolo/I;
prob_error_paquete(snrn)=1-(1-prob_error_simbolo(snrn))^(L);
vector_rate(snrn)=rate_medio/Nslots;
end

%hold on;
% semilogy(vector_snrn,prob_error_paquete,'g');
% ylabel('Error package probability');
% xlabel('SNR');
% figure;
% plot(vector_snrn,vector_rate);

```

Feedback Error Probability and Average SNR Aware Protocol Criterion

```

clear all;
%close all;

I=50000;
N=4;
d=100;

%Calculen les dist ncies aleatorias de cada destination.

distanciay=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciay.^2+distanciay.^2);

distanciay2=d-distanciay;
distancia2=sqrt(distanciay2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s mbolos en un paquete.
Ps=10;
Pesim=10^(-1);
contador1=0;

for snrn=7:9

    contador1=contador1+1;
    error_simbolo=0;
    snrml=10^(snrn/10);
    Pesim=10^(-2);
    contador2=0;
    fede=0.1:0.05:1;

    for Pebit=0.1:0.05:1

        contador2=contador2+1;
        varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    end
end

```

```

Ps=snrml*Pn/varianza1;
canales_outage_Alamouti=0;
canales_outage_Mejor=0;

for aux=1:I
    for cont=1:N
        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
    end

    for cont=1:N
        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
    end

    snr_relays=(abs(r).^2)*Ps/(Pn*2);
    Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
    outage_relays=Pesim_relays<Pesim;
    Pesim_relays=Pesim_relays.*outage_relays;

    a=rand(1);
    b=rand(1);
    c=rand(1);
    dd=rand(1);

    %Alamouti

    if(sum(outage_relays)>=2)

        vect_sec_link=abs(h).^2.*outage_relays;
        [maxim,pos1]=max(vect_sec_link);           %Cogemos el m+ximo
        vect_sec_link2(1)=vect_sec_link(pos1);    %Lo guardamos
        vect_sec_link(pos1)=0;                   %Ponemos su posicion a 0
        [maxim2,pos2]=max(vect_sec_link);        %Cogemos el siguiente
maximo
        vect_sec_link(pos1)=vect_sec_link2(1);    %Volvemos a poner el
valor m+ximo en su posicion

        pos1=pos1-1; %Para que empiezen las posiciones en 0
        pos2=pos2-1;

        b1=dec2bin(pos1); %Pasamos las posiciones a binario
        b2=dec2bin(pos2);

        l=0;

        l(4)=b1(length(b1));

        if pos1>1

            l(3)=b1(length(b1)-1);
        end
    end
end

```

```

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));

end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));

end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3

```



```

        l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

%En caso de haber seleccionado un relay que no existe
%ponemos ese valor a 0.

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if N<eleccion2
    vect_sec_link(eleccion2)=0;
end

snr_Alamouti=(Ps/
(Pn*4))*(vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
Pesim_Alamouti=2*(1/2*erfc(sqrt(snr_Alamouti/2)))-
(1/2*erfc(sqrt(snr_Alamouti/2))).^2;
outage_Alamouti=Pesim_Alamouti<Pesim;

if outage_Alamouti==0

    canales_outage_Alamouti=canales_outage_Alamouti+1;

end

else

    canales_outage_Alamouti=canales_outage_Alamouti+1;
end

if sum(outage_relays)>=1

    vect_sec_link=abs(h).^2.*outage_relays;

    [maxim,pos1]=max(vect_sec_link);
    pos1=pos1-1; %Para que empiezen las posiciones en 0

    b1=dec2bin(pos1); %Pasamos las posiciones a binario

    l=0;

    l(4)=b1(length(b1));

    if pos1>1

        l(3)=b1(length(b1)-1);
    end
end

```

```

    if pos1>3
        l(2)=b1(length(b1)-2);
    end

    if pos1>7
        l(1)=b1(length(b1)-3);
    end

    l=1>'0';

    if a<=Pebit
        l(4)=not(l(4));
    end

    if b<=Pebit && pos1>1
        l(3)=not(l(3));
    end

    if c<=Pebit && pos1>3
        l(2)=not(l(2));
    end

    if dd<=Pebit && pos1>7
        l(1)=not(l(1));
    end

    %Convertimos binario a decimal

    eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

    eleccion1=eleccion1+1;

    if N<eleccion1
        vect_sec_link(eleccion1)=0;
    end

    snr_relays2=(Ps/(Pn*2))*vect_sec_link(eleccion1);
    Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
    outage_relays2=Pesim_relays2<Pesim;

    if outage_relays2==0

        canales_outage_Mejor=canales_outage_Mejor+1;
    end

else
    canales_outage_Mejor=canales_outage_Mejor+1;
end

end
end

prob_outage_Mejor(contador1,contador2)=canales_outage_Mejor/I;
prob_outage_Alamouti(contador1,contador2)=canales_outage_Alamouti/I;

end

end

```

Feedback Error Probability and Average SNR Aware Protocol

```
clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculen les dist^ncies aleatorias de cada destination.

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.

for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

    varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1;

    for aux=1:I

        mejor=0;

        for cont=1:N

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

            end

        for cont=1:N

            varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
```

```

        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

        end

        e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

        snr_directo=(abs(e).^2)*Ps/(Pn);
        Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
        outage_directo=Pesim_directo<Pesim;

        %Calculamos el decoding set

        snr_relays=(abs(r).^2)*Ps/(Pn*2);
        Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
        outage_relays=Pesim_relays<Pesim;
        Pesim_relays=Pesim_relays.*outage_relays;

        if sum(outage_relays(1:end))==0
            Pesim_relays=inf;
        else
            Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
        end

        %Miramos los relays que est#n en outage para transmitir al
Destination

        vect_sec_link=abs(h).^2.*outage_relays;
        snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
        Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
        Pesim_relays2=Pesim_relays2.*outage_relays;
        outage_relays2=Pesim_relays2<Pesim;

        if sum(outage_relays2(1:end))==0
            Pesim_relays2=inf;
        else
            Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
        end

        %Miramos cual va a ser la velocidad necesaria

        if outage_directo==1
            rate=4;
            Nslots=Nslots+1;
            Pebit=Pesim_directo/2;
        else
            rate=2;
            Nslots=Nslots+1;
            Pebit=Pesim_relays2/2;
        end

        end

        a=rand(1);
        b=rand(1);
        c=rand(1);

```



```

        l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));

```

```

end

if b<=Pebit %&& pos1>1
    l(3)=not(l(3));
end

if c<=Pebit %&& pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit %&& pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if outage_directo==0 %Miramos que pasa
cuando no se puede transmitir de forma directa

    if mejor==1 %Mejor Relay

        if(sum(outage_relays)>0)

            if N<eleccion1
                vect_sec_link(eleccion1)=0;
            end

            snr_mejor=(Ps/(Pn*2))*vect_sec_link(eleccion1);
            Pesim_mejor=2*(1/2*erfc(sqrt(snr_mejor/2)))-
(1/2*erfc(sqrt(snr_mejor/2))).^2;
            outage_mejor=Pesim_mejor<Pesim;

            if outage_mejor==1
                rate_medio=rate_medio+rate;
            else
                rate_medio=rate_medio+0;
                error_simbolo=error_simbolo+1;
            end

        else
            error_simbolo=error_simbolo+1;
            rate_medio=rate_medio+0;
        end

    else %Usamos Alamouti

        if(sum(outage_relays)>=2)

            %En caso de haber seleccionado un relay que no
existe
            %ponemos ese valor a 0.

```

```

        if N<eleccion1
            vect_sec_link(eleccion1)=0;
        end

        if N<eleccion2
            vect_sec_link(eleccion2)=0;
        end

        snr_Alamouti=(Ps/
(Pn*4))* (vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
        Pesim_Alamouti=2*2*(1/2*erfc(sqrt(snr_Alamouti/
2)))-(1/2*erfc(sqrt(snr_Alamouti/2))).^2;
        outage_Alamouti=Pesim_Alamouti<Pesim;

        if outage_Alamouti==1
            rate_medio=rate_medio+rate;
        else
            rate_medio=rate_medio+0;
            error_simbolo=error_simbolo+1;
        end

    else
        error_simbolo=error_simbolo+1;
        rate_medio=rate_medio+0;
    end
end
else
    error_simbolo=error_simbolo;
    rate_medio=rate_medio+rate;
end
end

Pebit(snrn)=Pebit;
vector_snrn(snrn)=snrn;
prob_error_simbolo(snrn)=error_simbolo/I;
prob_error_paquete(snrn)=1-(1-prob_error_simbolo(snrn))^(L);
vector_rate(snrn)=rate_medio/Nslots;
end

%hold on;
% semilogy(vector_snrn,prob_error_paquete,'g');
% ylabel('Error package probability');
% xlabel('SNR');
% figure;
% plot(vector_snrn,vector_rate);

```

Number of Relays and Average SNR Aware Protocol

```

clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculen les dist`ncies aleatories de cada destination.

distanciay=d*rand(N,1);
distanciax=(d/2)*rand(N,1);

```



```

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.

for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

    varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1;

    for aux=1:I

        mejor=0;

        for cont=1:N

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

            end

            for cont=1:N

                varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
                h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

                end

            e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

            snr_directo=(abs(e).^2)*Ps/(Pn);
            Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
            outage_directo=Pesim_directo<Pesim;

            %Calculamos el decoding set

            snr_relays=(abs(r).^2)*Ps/(Pn*2);
            Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
            outage_relays=Pesim_relays<Pesim;
            Pesim_relays=Pesim_relays.*outage_relays;

            if sum(outage_relays(1:end))==0

```



```

%                               Segœn SNR

if outage_directo==0
    if N<=2
        mejor=1;
    else if N==3 && snrm<26
        mejor=1;
    else if N==4 && snrm<13
        mejor=1;
    else if N==5 && snrm<10
        mejor=1;
    end
    end
end
end

%

vect_sec_link=abs(h).^2.*outage_relays;
[maxim,pos1]=max(vect_sec_link);           %Cogemos el m+ximo
vect_sec_link2=vect_sec_link(pos1);       %Lo guardamos
vect_sec_link(pos1)=0;                    %Ponemos su posicion a 0
[maxim2,pos2]=max(vect_sec_link);         %Cogemos el siguiente
maximo    vect_sec_link(pos1)=vect_sec_link2; %Volvemos a poner el valor
m+ximo en su posicion

pos1=pos1-1; %Para que empiezen las posiciones en 0
pos2=pos2-1;

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

if pos1>1

    l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));

end

if b<=Pebit && pos1>1
    l(3)=not(l(3));

```

```

end

if c<=Pebit &&& pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit &&& pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));

end

if b<=Pebit &&& pos1>1
    l(3)=not(l(3));
end

if c<=Pebit &&& pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit &&& pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

```

```

    if outage_directo==0                                %Miramos que pasa
cuando no se puede transmitir de forma directa

        if mejor==1                                    %Mejor Relay

            if(sum(outage_relays)>0)

                if N<eleccion1
                    vect_sec_link(eleccion1)=0;
                end

                snr_mejor=(Ps/(Pn*2))*vect_sec_link(eleccion1);
                Pesim_mejor=2*(1/2*erfc(sqrt(snr_mejor/2)))-
(1/2*erfc(sqrt(snr_mejor/2))).^2;
                outage_mejor=Pesim_mejor<Pesim;

                if outage_mejor==1
                    rate_medio=rate_medio+rate;
                else
                    rate_medio=rate_medio+0;
                    error_simbolo=error_simbolo+1;
                end

            else
                error_simbolo=error_simbolo+1;
                rate_medio=rate_medio+0;
            end

        else                                            %Usamos Alamouti

            if(sum(outage_relays)>=2)

                %En caso de haber seleccionado un relay que no
existe
                %ponemos ese valor a 0.

                if N<eleccion1
                    vect_sec_link(eleccion1)=0;
                end

                if N<eleccion2
                    vect_sec_link(eleccion2)=0;
                end

                snr_Alamouti=(Ps/
(Pn*4))*(vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
                Pesim_Alamouti=2*2*(1/2*erfc(sqrt(snr_Alamouti/
2)))- (1/2*erfc(sqrt(snr_Alamouti/2))).^2;
                outage_Alamouti=Pesim_Alamouti<Pesim;

                if outage_Alamouti==1
                    rate_medio=rate_medio+rate;
                else
                    rate_medio=rate_medio+0;
                    error_simbolo=error_simbolo+1;
                end

            else
                error_simbolo=error_simbolo+1;
            end
        end
    end
end

```

```

        rate_medio=rate_medio+0;
    end
end
else
    error_simbolo=error_simbolo;
    rate_medio=rate_medio+rate;
end
end

Pebit(snm)=Pebit;
vector_snm(snm)=snm;
prob_error_simbolo(snm)=error_simbolo/I;
prob_error_paquete(snm)=1-(1-prob_error_simbolo(snm))^(L);
vector_rate(snm)=rate_medio/Nslots;
end

%hold on;
% semilogy(vector_snm,prob_error_paquete,'g');
% ylabel('Error package probability');
% xlabel('SNR');
% figure;
% plot(vector_snm,vector_rate);

```

Protocol Allocation Protocol Criterion

```

%clear all;
%close all;

%I=700000;
N=4;
%R=2;
d=100;

%Calculen les dist^ncies aleatorias de cada destination.

distanciay=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciay.^2+distanciay.^2);

distanciay2=d-distanciay;
distancia2=sqrt(distanciay2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.

%snm=8;
contador=1;

for beta=0:0.02:1

    rate_medio=0;
    fed_error=0;

```

```

Nslots=0;
error_simbolo_Directo=0;
error_simbolo_Mejor=0;
error_simbolo_Alamouti=0;
snrml=10^(snrm/10);
Pesim=10^(-2);
varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
Ps=snrml*Pn/varianza1;

for aux=1:I

    for cont=1:N

        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

    end

    for cont=1:N

        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

    end

    %Calculamos el decoding set

    snr_relays=(abs(r).^2)*Ps/(Pn*2);
    Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
    outage_relays=Pesim_relays<Pesim;
    Pesim_relays=Pesim_relays.*outage_relays;

    if sum(outage_relays(1:end))==0
        Pesim_relays=inf;
    else
        Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
    end

    %Miramos los relays que est#n en outage para transmitir al
Destination

    vect_sec_link=abs(h).^2.*outage_relays;
    snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
    Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
    Pesim_relays2=Pesim_relays2.*outage_relays;
    outage_relays2=Pesim_relays2<Pesim;

    if sum(outage_relays2(1:end))==0
        Pesim_relays2=inf;
    else
        Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
    end

    Pebit=Pesim_relays2/2;

    %Calculamos las mejores posiciones

    vect_sec_link=abs(h).^2.*outage_relays;
    [maxim,pos1]=max(vect_sec_link); %Cogemos el m#ximo

```

```

vect_sec_link2=vect_sec_link(pos1); %Lo guardamos
vect_sec_link(pos1)=0; %Ponemos su posicion a 0
[maxim2,pos2]=max(vect_sec_link); %Cogemos el siguiente maximo
vect_sec_link(pos1)=vect_sec_link2;

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

pos1=pos1-1; %Para que empiezen las posiciones en 0
pos2=pos2-1;

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

if pos1>1
    l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

```



```

        l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));

end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if N<eleccion2
    vect_sec_link(eleccion2)=0;
end

if(sum(outage_relays)>=2)

    snr_Alamouti=(Ps/(Pn*2))*((1-
beta)*vect_sec_link(eleccion1)+beta*vect_sec_link(eleccion2));
    Pesim_Alamouti=2*(2*(1/2*erfc(sqrt(snr_Alamouti/2)))-
(1/2*erfc(sqrt(snr_Alamouti/2))).^2);
    outage_Alamouti=Pesim_Alamouti<Pesim;

    if outage_Alamouti==0
        error_simbolo_Alamouti=error_simbolo_Alamouti+1;
    else
        error_simbolo_Alamouti=error_simbolo_Alamouti+0;
    end
end

```

```

        end
    else
        error_simbolo_Alamouti=error_simbolo_Alamouti+1;

    end
end

vector_beta(contador)=beta;
prob_outage_Alamouti(contador)=error_simbolo_Alamouti/I;
prob_error_paquete_Alamouti(contador)=1-(1-
prob_outage_Alamouti(contador))^(L);

    contador=contador+1;
end

```

Power Allocation Protocol

```

%clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculem les dist`ncies aleatorias de cada destination.

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s`mbolos en un paquete.

%snrm=8;
%contador=1;

for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo_Directo=0;
    error_simbolo_Mejor=0;
    error_simbolo_Alamouti=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

```

```

varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
Ps=snrml*Pn/varianza1;

for aux=1:I

    for cont=1:N

        varianza2=((distancia(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        r(cont,1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));

    end

for cont=1:N

    varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
    h(cont,1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));

end

e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

    snr_directo=(abs(e).^2)*Ps/(Pn);
    Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
    outage_directo=Pesim_directo<Pesim;

    %Calculamos el decoding set

    snr_relays=(abs(r).^2)*Ps/(Pn*2);
    Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
    outage_relays=Pesim_relays<Pesim;
    Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

    %Miramos los relays que est#n en outage para transmitir al
Destination

    vect_sec_link=abs(h).^2.*outage_relays;
    snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
    Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
    Pesim_relays2=Pesim_relays2.*outage_relays;
    outage_relays2=Pesim_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

Pebit=Pesim_relays2/2;

    %Calculamos las mejores posiciones

```

```

vect_sec_link=abs(h).^2.*outage_relays;
[maxim,pos1]=max(vect_sec_link);           %Cogemos el máximo
vect_sec_link2=vect_sec_link(pos1);       %Lo guardamos
vect_sec_link(pos1)=0;                     %Ponemos su posición a 0
[maxim2,pos2]=max(vect_sec_link);         %Cogemos el siguiente máximo
vect_sec_link(pos1)=vect_sec_link2;

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

pos1=pos1-1; %Para que empiecen las posiciones en 0
pos2=pos2-1;

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

if pos1>1
    l(3)=b1(length(b1)-1);
end

if pos1>3
    l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda selección

l=0;

l(4)=b2(length(b2));

```

```

if pos2>1
    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal
eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if N<eleccion1
    vect_sec_link(eleccion1)=0;
end

if N<eleccion2
    vect_sec_link(eleccion2)=0;
end

%Criteri de eleccio de beta

if snrm<=10
    beta=0;
else if snrm==11
    beta=0.15;
else if snrm<=13
    beta=0.25;
else if snrm<=15
    beta==0.45;
else

```

```

        beta=0.5;
    end
end
end
end

if outage_directo==0

    if(sum(outage_relays)>=2)

        snr_Alamouti=(Ps/(Pn*2))*((1-
beta)*vect_sec_link(eleccion1)+beta*vect_sec_link(eleccion2));
        Pesim_Alamouti=2*(2*(1/2*erfc(sqrt(snr_Alamouti/2)))-
(1/2*erfc(sqrt(snr_Alamouti/2))).^2);
        outage_Alamouti=Pesim_Alamouti<Pesim;

        if outage_Alamouti==0
            error_simbolo_Alamouti=error_simbolo_Alamouti+1;
        else
            error_simbolo_Alamouti=error_simbolo_Alamouti+0;
        end
    else
        error_simbolo_Alamouti=error_simbolo_Alamouti+1;
    end

else
    error_simbolo_Alamouti=error_simbolo_Alamouti+0;
end
end

vector_snrn(snrn)=snrn;
prob_outage_Alamouti(snrn)=error_simbolo_Alamouti/I;
prob_error_paquete_Alamouti(snrn)=1-(1-
prob_outage_Alamouti(snrn))^L;
end

```

Rho Analisis

```

clear all;

aux=0;
for v=1:200

    fc=2.5*10^9;
    c=3*10^8;
    landac=c/fc;
    aux=aux+1;
    fd=v/landac; %Frecuencia Doppler
    Tm=2*10^-3; %Tiempo de muestra
    ro(aux)=besselj(0,2*pi*fd*Tm); %Correlacion entre dos instantes
    temporales
end

vector=1:200;

```

```
plot(vector,ro);
```

Correlated Channels

```
clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculen les dist^ncies aleatorias de cada destination.

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.
v=1;%Velocidad en m/s de media
fd=v/landac; %Frecuencia Doppler
Tm=2*10^-3; %Tiempo de muestra
ro=besselj(0,2*pi*fd*Tm); %Correlaci-n entre dos instantes temporales
%ro=0.9;
r(N,1)=0;
h(N,1)=0;
w(1)=0; %Variables de correlaci-n
x(1)=0;
usado_Mejor=1;
usado_Alamouti1=1;
usado_Alamouti2=2;

%PER Directo

for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo_Directo=0;
    error_simbolo_Mejor=0;
    error_simbolo_Alamouti=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

    varianzal=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianzal;

    for aux=1:I
```

```

e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

snr_directo=(abs(e).^2)*Ps/(Pn);
Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
outage_directo=Pesim_directo<Pesim;

if outage_directo==0
    error_simbolo_Directo=error_simbolo_Directo+1;
else
    error_simbolo_Directo=error_simbolo_Directo+0;
end
end

vector_snrn(snrn)=snrn;
prob_outage_Directo(snrn)=error_simbolo_Directo/I;
prob_error_paquete_Directo(snrn)=1-(1-prob_outage_Directo(snrn))^(L);

%Best Relay

for aux=1:I

    for cont=1:N
        if aux==1
            varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
            w(cont,1)=r(cont,1)/sqrt(varianza2);
        else
            varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            canal1(cont,1)=ro*w(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
            r(cont,1)=canal1(cont,1)*sqrt(varianza2);
            w(cont,1)=canal1(cont,1);
        end
    end

end

for cont=1:N

    if aux==1
        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
        x(cont,1)=h(cont,1)/sqrt(varianza3);
    else
        varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
        canal2(cont,1)=ro*x(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
        h(cont,1)=canal2(cont,1)*sqrt(varianza3);
        x(cont,1)=canal2(cont,1);
    end

end

end

```



```

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

%Miramos los relays que est#n en outage para transmitir al
Destination

vect_sec_link=abs(h).^2.*outage_relays;
snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
Pesim_relays2=Pesim_relays2.*outage_relays;
outage_relays2=Pesim_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

Pebit=Pesim_relays2/2;

%Calculamos las mejores posiciones

vect_sec_link=abs(h).^2.*outage_relays;
[maxim,pos1]=max(vect_sec_link);           %Cogemos el m#ximo
vect_sec_link2=vect_sec_link(pos1);       %Lo guardamos
vect_sec_link(pos1)=0;                    %Ponemos su posicion a 0
[maxim2,pos2]=max(vect_sec_link);         %Cogemos el siguiente maximo
vect_sec_link(pos1)=vect_sec_link2;

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

pos1=pos1-1; %Para que empiezen las posiciones en 0
pos2=pos2-1;

b1=dec2bin(pos1); %Pasamos las posiciones a binario
b2=dec2bin(pos2);

l=0;

l(4)=b1(length(b1));

if pos1>1
    l(3)=b1(length(b1)-1);
end

if pos1>3

```

```

        l(2)=b1(length(b1)-2);
    end

    if pos1>7
        l(1)=b1(length(b1)-3);
    end

    l=l>'0';

    if a<=Pebit
        l(4)=not(l(4));
    end

    if b<=Pebit &&& pos1>1
        l(3)=not(l(3));
    end

    if c<=Pebit &&& pos1>3
        l(2)=not(l(2));
    end

    if dd<=Pebit &&& pos1>7
        l(1)=not(l(1));
    end

    %Convertimos binario a decimal

    eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);
    eleccion1=eleccion1+1;

    if N<eleccion1
        eleccion1=usado_Mejor;
    end

    if(sum(outage_relays)>0)

        snr_mejor=(Ps/(Pn*2))*vect_sec_link(eleccion1);
        Pesim_mejor=2*(1/2*erfc(sqrt(snr_mejor/2)))-
(1/2*erfc(sqrt(snr_mejor/2))).^2;
        outage_mejor=Pesim_mejor<Pesim;

        if outage_mejor==0
            error_simbolo_Mejor=error_simbolo_Mejor+1;
        else
            error_simbolo_Mejor=error_simbolo_Mejor+0;
        end

        usado_Mejor=eleccion1;

    else
        error_simbolo_Mejor=error_simbolo_Mejor+1;
        rate_medio=rate_medio+0;
    end
end

vector_snrn(snrn)=snrn;
prob_outage_Mejor(snrn)=error_simbolo_Mejor/I;
prob_error_paquete_Mejor(snrn)=1-(1-prob_outage_Mejor(snrn))^(L);

```

```

%Alamouti

for aux=1:I

    for cont=1:N
        if aux==1
            varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
            y(cont,1)=r(cont,1)/sqrt(varianza2);
        else
            varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            canal3(cont,1)=ro*y(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
            r(cont,1)=canal3(cont,1)*sqrt(varianza2);
            y(cont,1)=canal3(cont,1);
        end
    end

    for cont=1:N

        if aux==1
            varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
            z(cont,1)=h(cont,1)/sqrt(varianza3);
        else
            varianza3=((distancia2(cont)/d0)^(-mu))*(landac/(4*pi*d0))^2;
            canal4(cont,1)=ro*z(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
            h(cont,1)=canal4(cont,1)*sqrt(varianza3);
            z(cont,1)=canal4(cont,1);
        end
    end

end

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

%Miramos los relays que est#n en outage para transmitir al
Destination

```

```

vect_sec_link=abs(h).^2.*outage_relays;
snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
outage_relays2=Pesim_relays2.*outage_relays;
outage_relays2=outage_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

    %Miramos cual va a ser la velocidad necesaria

    if outage_directo==1
        rate=4;
        Nslots=Nslots+1;
        Pebit=Pesim_directo/2;
    else
        rate=2;
        Nslots=Nslots+1;
        Pebit=Pesim_relays2/2;
    end

    %Calculamos las mejores posiciones

    vect_sec_link=abs(h).^2.*outage_relays;
    [maxim,pos1]=max(vect_sec_link);           %Cogemos el m+ximo
    vect_sec_link2=vect_sec_link(pos1);       %Lo guardamos
    vect_sec_link(pos1)=0;                   %Ponemos su posicion a 0
    [maxim2,pos2]=max(vect_sec_link);        %Cogemos el siguiente maximo
    vect_sec_link(pos1)=vect_sec_link2;

    a=rand(1);
    b=rand(1);
    c=rand(1);
    dd=rand(1);

    pos1=pos1-1; %Para que empiezen las posiciones en 0
    pos2=pos2-1;

    b1=dec2bin(pos1); %Pasamos las posiciones a binario
    b2=dec2bin(pos2);

    l=0;

    l(4)=b1(length(b1));

    if pos1>1

        l(3)=b1(length(b1)-1);
    end

    if pos1>3
        l(2)=b1(length(b1)-2);
    end

    if pos1>7
        l(1)=b1(length(b1)-3);
    end

```

```

end

l=l>'0';

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit &&& pos1>1
    l(3)=not(l(3));
end

if c<=Pebit &&& pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit &&& pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal
eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion
l=0;

l(4)=b2(length(b2));

if pos2>1
    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));
end

if b<=Pebit &&& pos1>1
    l(3)=not(l(3));
end

if c<=Pebit &&& pos1>3
    l(2)=not(l(2));
end

```



```

    prob_error_paquete_Alamouti(snrml)=1-(1-
prob_outage_Alamouti(snrml))^(L);
end

```

Correlated Channel Protocol over Feedback Error Probability and Average SNR Aware Protocol

```

clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculemos las distancias aleatorias de cada destination.

distanciay=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciay.^2+distanciay.^2);

distanciay2=d-distanciay;
distancia2=sqrt(distanciay2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.
v=1; %Velocidad media
fd=v/landac; %Frecuencia Doppler
Tm=2*10^-3; %Tiempo de muestra
ro=besselj(0,2*pi*fd*Tm); %Correlaci3n entre dos instantes temporales
%ro=0.6;
r(N,1)=0;
h(N,1)=0;
w(1)=0; %Variables de correlaci3n
x(1)=0;
usado_Mejor=1;
usado_Alamouti1=1;
usado_Alamouti2=2;

for snrml=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo=0;
    snrml=10^(snrml/10);
    Pesim=10^(-2);

    varianzal=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianzal;

```

```

for aux=1:I
    mejor=0;

    for cont=1:N
        if aux==1
            varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
            w(cont,1)=r(cont,1)/sqrt(varianza2);
        else

            varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            canal1(cont,1)=ro*w(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
            r(cont,1)=canal1(cont,1)*sqrt(varianza2);
            w(cont,1)=canal1(cont,1);
        end
    end

    for cont=1:N

        if aux==1
            varianza3=((distancia2(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
            x(cont,1)=h(cont,1)/sqrt(varianza3);
        else

            varianza3=((distancia2(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
            canal2(cont,1)=ro*x(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
            h(cont,1)=canal2(cont,1)*sqrt(varianza3);
            x(cont,1)=canal2(cont,1);
        end
    end

end

e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

snr_directo=(abs(e).^2)*Ps/(Pn);
Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
outage_directo=Pesim_directo<Pesim;

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else

```



```

    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

    %Miramos los relays que est#n en outage para transmitir al
Destination

    vect_sec_link=abs(h).^2.*outage_relays;
    snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
    Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
    Pesim_relays2=Pesim_relays2.*outage_relays;
    outage_relays2=Pesim_relays2<Pesim;

    if sum(outage_relays2(1:end))==0
        Pesim_relays2=inf;
    else
        Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
    end

    %Miramos cual va a ser la velocidad necesaria

    if outage_directo==1
        rate=4;
        Nslots=Nslots+1;
        Pebit=Pesim_directo/2;
    else
        rate=2;
        Nslots=Nslots+1;
        Pebit=Pesim_relays2/2;
    end

    a=rand(1);
    b=rand(1);
    c=rand(1);
    dd=rand(1);

    %Seg#n feedback

    if outage_directo==0

        if snrm<7
            mejor=1;
        else if snrm<9 && Pebit<0.95
            mejor=1;
        else if snrm<10 && Pebit<0.4
            mejor=1;
        else if snrm<12 &&Pebit<0.35
            mejor=1;
        else if snrm<14 && Pebit<0.2
            mejor=1;
        end
        end
        end
        end
        end
    end
end

```



```

        l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l+'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

if a<=Pebit
    l(4)=not(l(4));

end

if b<=Pebit && pos1>1
    l(3)=not(l(3));
end

if c<=Pebit && pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit && pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

```

```

    if outage_directo==0                                %Miramos que pasa
cuando no se puede transmitir de forma directa

        if mejor==1                                    %Mejor Relay

            if(sum(outage_relays)>0)

                if N<eleccion1
                    eleccion1=usado_Mejor;
                end

                snr_mejor=(Ps/(Pn*2))*vect_sec_link(eleccion1);
                Pesim_mejor=2*(1/2*erfc(sqrt(snr_mejor/2)))-
(1/2*erfc(sqrt(snr_mejor/2))).^2;
                outage_mejor=Pesim_mejor<Pesim;

                if outage_mejor==1
                    rate_medio=rate_medio+rate;
                else
                    rate_medio=rate_medio+0;
                    error_simbolo=error_simbolo+1;
                end

            else
                error_simbolo=error_simbolo+1;
                rate_medio=rate_medio+0;
            end
            usado_Mejor=eleccion1;

        else                                            %Usamos Alamouti

            if sum(outage_relays)>=2

                %En caso de haber seleccionado un relay que no
existe
                %ponemos ele elegido anteriormente

                if N<eleccion1
                    if usado_Alamouti1==eleccion2
                        eleccion1=usado_Alamouti2;
                    else
                        eleccion1=usado_Alamouti1;
                    end
                end

                if N<eleccion2
                    if usado_Alamouti2==eleccion1
                        eleccion2=usado_Alamouti1;
                    else
                        eleccion2=usado_Alamouti2;
                    end
                end

                snr_Alamouti=(Ps/
(Pn*4))*(vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
                Pesim_Alamouti=2*2*(1/2*erfc(sqrt(snr_Alamouti/
2)))-(1/2*erfc(sqrt(snr_Alamouti/2))).^2;
                outage_Alamouti=Pesim_Alamouti<Pesim;

                if outage_Alamouti==1
                    rate_medio=rate_medio+rate;
                else
                    rate_medio=rate_medio+0;
                    error_simbolo=error_simbolo+1;
                end
            end
        end
    end

```

```

end
else
    error_simbolo=error_simbolo+1;
    rate_medio=rate_medio+0;
end
usado_Alamouti1=eleccion1;
usado_Alamouti2=eleccion2;
end
end
end
end
end

Pebit(snr)=Pebit;
vector_snr(snr)=snr;
prob_error_simbolo(snr)=error_simbolo/I;
prob_error_paquete(snr)=1-(1-prob_error_simbolo(snr))^(L);
vector_rate(snr)=rate_medio/Nslots;
end

%hold on;
% semilogy(vector_snr,prob_error_paquete,'g');
% ylabel('Error package probability');
% xlabel('SNR');
% figure;
% plot(vector_snr,vector_rate);

```

Correlated Channel Protocol over Number of Relays and Average SNR Aware Protocol

```

clear all;
%close all;

I=100000;
N=4;
%R=2;
d=100;

%Calculen las distancias aleatorias de cada destination.

distanciax=d*rand(N,1);
distanciay=(d/2)*rand(N,1);

distancia=sqrt(distanciax.^2+distanciay.^2);

distanciax2=d-distanciax;
distancia2=sqrt(distanciax2.^2+distanciay.^2);

fc=2.5*10^9;
c=3*10^8;
landac=c/fc;
d0=1;
mu=3;
Pn=1;
L=10; %Numero de s'mbolos en un paquete.
v=1; %Velocidad media
fd=v/landac; %Frecuencia Doppler
Tm=2*10^-3; %Tiempo de muestra

```

```

ro=besselj(0,2*pi*fd*Tm); %Correlaci3n entre dos instantes temporales
%ro=0.6;
r(N,1)=0;
h(N,1)=0;
w(1)=0; %Variables de correlaci3n
x(1)=0;
usado_Mejor=1;
usado_Alamouti1=1;
usado_Alamouti2=2;

for snrm=1:30

    rate_medio=0;
    fed_error=0;
    Nslots=0;
    error_simbolo=0;
    snrml=10^(snrm/10);
    Pesim=10^(-2);

    varianza1=((d/d0)^(-mu))*(landac/(4*pi*d0))^2;
    Ps=snrml*Pn/varianza1;

    for aux=1:I

        mejor=0;

        for cont=1:N
            if aux==1
                varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
                r(cont,
1)=sqrt(0.5)*sqrt(varianza2)*(randn(1,1)+j*randn(1,1));
                w(cont,1)=r(cont,1)/sqrt(varianza2);
            else
                varianza2=((distancia(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
                canal1(cont,1)=ro*w(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
                r(cont,1)=canal1(cont,1)*sqrt(varianza2);
                w(cont,1)=canal1(cont,1);
            end
        end

        for cont=1:N

            if aux==1
                varianza3=((distancia2(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
                h(cont,
1)=sqrt(0.5)*sqrt(varianza3)*(randn(1,1)+j*randn(1,1));
                x(cont,1)=h(cont,1)/sqrt(varianza3);
            else
                varianza3=((distancia2(cont)/d0)^(-mu))*(landac/
(4*pi*d0))^2;
                canal2(cont,1)=ro*x(cont,1)+sqrt(0.5)*sqrt(1-
ro^2)*(randn(1,1)+j*randn(1,1));
                h(cont,1)=canal2(cont,1)*sqrt(varianza3);
                x(cont,1)=canal2(cont,1);
            end
        end
    end
end

```

```

e=sqrt(0.5)*sqrt(varianza1)*(randn(1,1)+j*randn(1,1));

snr_directo=(abs(e).^2)*Ps/(Pn);
Pesim_directo=2*(1/2*erfc(sqrt(snr_directo/2)))-
(1/2*erfc(sqrt(snr_directo/2))).^2;
outage_directo=Pesim_directo<Pesim;

%Calculamos el decoding set

snr_relays=(abs(r).^2)*Ps/(Pn*2);
Pesim_relays=2*(1/2*erfc(sqrt(snr_relays/2)))-
(1/2*erfc(sqrt(snr_relays/2))).^2;
outage_relays=Pesim_relays<Pesim;
Pesim_relays=Pesim_relays.*outage_relays;

if sum(outage_relays(1:end))==0
    Pesim_relays=inf;
else
    Pesim_relays=sum(Pesim_relays(1:end))/sum(outage_relays(1:end));
end

%Miramos los relays que est#n en outage para transmitir al
Destination

vect_sec_link=abs(h).^2.*outage_relays;
snr_relays2=(Ps/(Pn*2)).*vect_sec_link;
Pesim_relays2=2*(1/2*erfc(sqrt(snr_relays2/2)))-
(1/2*erfc(sqrt(snr_relays2/2))).^2;
Pesim_relays2=Pesim_relays2.*outage_relays;
outage_relays2=Pesim_relays2<Pesim;

if sum(outage_relays2(1:end))==0
    Pesim_relays2=inf;
else
    Pesim_relays2=sum(Pesim_relays2(1:end))/
sum(outage_relays2(1:end));
end

%Miramos cual va a ser la velocidad necesaria

if outage_directo==1
    rate=4;
    Nslots=Nslots+1;
    Pebit=Pesim_directo/2;
else
    rate=2;
    Nslots=Nslots+1;
    Pebit=Pesim_relays2/2;
end

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

%Seg#n feedback

```



```

        l(2)=b1(length(b1)-2);
end

if pos1>7
    l(1)=b1(length(b1)-3);
end

l=l>'0';

    if a<=Pebit
        l(4)=not(l(4));

    end

    if b<=Pebit &&& pos1>1
        l(3)=not(l(3));
    end

    if c<=Pebit &&& pos1>3
        l(2)=not(l(2));
    end

    if dd<=Pebit &&& pos1>7
        l(1)=not(l(1));
    end

%Convertimos binario a decimal

eleccion1=l(4)+2*l(3)+4*l(2)+8*l(1);

%Ahora hacemos lo mismo para la segunda seleccion

l=0;

l(4)=b2(length(b2));

if pos2>1

    l(3)=b2(length(b2)-1);
end

if pos2>3
    l(2)=b2(length(b2)-2);
end

if pos2>7
    l(1)=b2(length(b2)-3);
end

l=l>'0';

a=rand(1);
b=rand(1);
c=rand(1);
dd=rand(1);

    if a<=Pebit
        l(4)=not(l(4));

    end

    if b<=Pebit &&& pos1>1
        l(3)=not(l(3));

```

```

end

if c<=Pebit %&& pos1>3
    l(2)=not(l(2));
end

if dd<=Pebit %&& pos1>7
    l(1)=not(l(1));
end

%Convertimos binario a decimal

eleccion2=l(4)+2*l(3)+4*l(2)+8*l(1);

eleccion1=eleccion1+1;
eleccion2=eleccion2+1;

if outage_directo==0 %Miramos que pasa
cuando no se puede transmitir de forma directa

    if mejor==1 %Mejor Relay

        if(sum(outage_relays)>0)

            if N<eleccion1
                eleccion1=usado_Mejor;
            end

            snr_mejor=(Ps/(Pn*2))*vect_sec_link(eleccion1);
            Pesim_mejor=2*(1/2*erfc(sqrt(snr_mejor/2)))-
(1/2*erfc(sqrt(snr_mejor/2))).^2;
            outage_mejor=Pesim_mejor<Pesim;

            if outage_mejor==1
                rate_medio=rate_medio+rate;
            else
                rate_medio=rate_medio+0;
                error_simbolo=error_simbolo+1;
            end

            else
                error_simbolo=error_simbolo+1;
                rate_medio=rate_medio+0;
            end
            usado_Mejor=eleccion1;

        else %Usamos Alamouti

            if sum(outage_relays)>=2

                %En caso de haber seleccionado un relay que no
existe
                %ponemos ele elegido anteriormente

                if N<eleccion1
                    if usado_Alamouti1==eleccion2
                        eleccion1=usado_Alamouti2;
                    else

```

```

        eleccion1=usado_Alamouti1;
    end

    if N<eleccion2
        if usado_Alamouti2==eleccion1
            eleccion2=usado_Alamouti1;
        else
            eleccion2=usado_Alamouti2;
        end

        snr_Alamouti=(Ps/
(Pn*4))* (vect_sec_link(eleccion1)+vect_sec_link(eleccion2));
        Pesim_Alamouti=2*2*(1/2*erfc(sqrt(snr_Alamouti/
2)))-(1/2*erfc(sqrt(snr_Alamouti/2))).^2;
        outage_Alamouti=Pesim_Alamouti<Pesim;

        if outage_Alamouti==1
            rate_medio=rate_medio+rate;
        else
            rate_medio=rate_medio+0;
            error_simbolo=error_simbolo+1;

        end

    else
        error_simbolo=error_simbolo+1;
        rate_medio=rate_medio+0;
    end

    usado_Alamouti1=eleccion1;
    usado_Alamouti2=eleccion2;

end
end
end
end

Pebit(snr)=Pebit;
vector_snr(snr)=snr;
prob_error_simbolo(snr)=error_simbolo/I;
prob_error_paquete(snr)=1-(1-prob_error_simbolo(snr))^(L);
vector_rate(snr)=rate_medio/Nslots;
end

%hold on;
% semilogy(vector_snr,prob_error_paquete,'g');
% ylabel('Error package probability');
% xlabel('SNR');
% figure;
% plot(vector_snr,vector_rate);

```