```
%% =================  initialization  =====================
clc
close all
K=4; % number of relays
 %r0=1.0; %secrecy rate

 P_up=0.051;
trials=1000000; %for testing only
Delay_bound=10;
L=Delay_bound-1;    % L here is representing the mask size B^m, where in all
simulation we used Delay_bound << the buffer size
LS=Delay_bound-1;
out_count=0; P=1;%  snr_limit=31   snr_range=1:snr_limit;
z_array=zeros(K,2);  b=[0.4;0.4;0.4;0.4]; %SIC factor
max_link_c=0;
minimizing=0;
max2=0;LOOP=0;
Q=zeros(K,L+1);
QS=zeros(1,LS+1);
pa_ind_err=0;       %r0=2.0;

            %% =================  SNR(P) loop   =====================



             % for iid case  R-R
             var3=[10^(4/10),10^(4/10),10^(4/10),10^(4/10);...
                 10^(4/10),10^(4/10),10^(4/10),10^(4/10);...
                 10^(4/10), 10^(4/10), 10^(4/10),10^(4/10);...
                 10^(4/10), 10^(4/10), 10^(4/10),10^(4/10)];
             var1=[10^(15/10);10^(15/10);10^(15/10);10^(15/10)];  %for ind case
S-R
             var2=[10^(17/10);10^(17/10);10^(17/10);10^(17/10)];  %for ind case
R-D
             var5=[10^(8/10);10^(8/10);10^(8/10);10^(8/10)];   %for ind case  R-E
             var4=[10^(5/10)];   %for ind case  S-E




                 for r0=0.1:0.1:4
                     LOOP=LOOP+1;
                Q=zeros(K,L+1);
                QS=zeros(1,LS+1);
                packet_delays=zeros(K,L);
                packet_indicator=zeros(K,L);
                p_s_indicator=zeros(1,LS);
                c_o_s=0;    % the location of the oldest packet in the source buffer
                current_old=zeros(K,1); % buffer filled with data to the half before
start
                n_o_s=0;    % the location of the second oldest packet in the source
buffer
                next_old=zeros(K,1);
%               snr_loop_dB=snr_loop-1; % to start from 0 to 50 dB
%               P=10^(snr_loop_dB/10);
                out_count=0;
                dropped_bits=0;
                dropped_bits_source_only=0;
                rec_r0_count=0;num_packet=0;
            %%  ch_threshold=sqrt((2^(2*r0)-1)/P);  %%
                new_frame=1;
                %% ===============  communication trials for specific SNR(P)
===========
                for i=1:trials
                    Dj_link=zeros(K,2); % store the trnsmission links satified the
condition (7) for both hops
                    DJ1_link=zeros(K,K); % store the jamming links satified the
condition (8) for both hops forthe first hop
                    DJ2_link=zeros(K,K); % store the jamming links satified the
condition (8) for both hops for the second hop
                    %  Notice that : 1)it needs to be initialized in every
                    %  time slot; 2)linenum represents index of
                    %  transmission link, rownum represents index of
                    %  its corresponding jamming link
                    %%  ================    assign  ch  gain  for  all  links
======================
                        U(1,1)=                              sqrt(.5*var4(1,1))
*( complex(randn(1,1),randn(1,1))); %the channel gain S-E
                    for i2=1:K
                        X(i2,1)=                             sqrt(.5*var1(i2,1))
*( complex(randn(1,1),randn(1,1)));  %the channel gain S-R
                        X(i2,2)=                             sqrt(.5*var2(i2,1))
*( complex(randn(1,1),randn(1,1)));  %the channel gain R-D
                        V(i2,1)=                             sqrt(.5*var5(i2,1))
*( complex(randn(1,1),randn(1,1))); %the channel gain R-E
                       for i3=1:K
                         W(i2,i3)=                           sqrt(.5*var3(i2,i3))
*( complex(randn(1,1),randn(1,1))); %the channel gain R-R
```

```
                       end
                    end

                    y1=abs(U);y2=abs(X);y3=abs(V);y4=abs(W);
                    max=-10000;
                    max_id=[-5,0];   % [relay  ID,  1=relay  receives  or  -1=relay
transmits]
                    bit = 1;             % bit available for source to transmit
                    % =================      segment after considering souce delay
==========
                    p_s_indicator(2:L)=p_s_indicator(1:L-1); % shift the buffer
and the indicator
                    p_s_indicator(1)=0; %the new packet in source located in the
first position
                    if new_frame==1
                        QS(1,1)=QS(1,1)+1;
                        p_s_indicator(1)=1; %the new packet that just arrived
                        num_packet=num_packet+1;
                    end
                    if  new_frame==1
                        new_frame=0;
                    end
                    %%
==========================================================
                    c_o_s=0;
                    check=0;
                    for i4=LS:-1:1  %  to check the oldest packet position of the
source from the farrest position
                        if p_s_indicator(i4)==1&&check==0
                            c_o_s=i4;
                            check=1;
                        end
                    end
                        for i3=1:K
                            if Q(i3,1)<=L&&Q(i3,1)>=0&&QS(1,1)>0
                            % if QS(1,1)>0
                                Dj_link(i3,1)=1;
                            else
                                Dj_link(i3,1)=0;
                            end

                            if Q(i3,1)<=L&&Q(i3,1)>0
                                Dj_link(i3,2)=1;
                            else  %==0
```

```
                Dj_link(i3,2)=0;
            end
        end

        for i3=1:K
            if (y2(i3,1)^2)<(-2^(r0)*var4(1,1)*log(P_up))
                Dj_link(i3,1)=0;
            end
            if (y2(i3,2)^2)<(-2^(r0)*var5(i3,1)*log(P_up))
                Dj_link(i3,2)=0;
            end
        end
        %%
===============================================================
            max=0;
            max2=0;
            min1=10000;
            min_J2=10000;
            min_J1=10000;
            distance=-10000;
            minimizing=0;
            min_found=0;
            max_found=0;
            max_Q=-10000;
            max9=-10000;
            max_QS=c_o_s;
            min_Q=10000;
            max_sing_id=1;
            min_sing_id=1;
            max=0;
            c_count=0;
            C=zeros(1,K);
            flag=ones(K);
            for i3=1:K
               for i4=1:K
                   if i3==i4
                       flag(i3,i4)=0;
                   end
               end
            end
            for i3=1:K
                if Dj_link(i3,2)==1&&current_old(i3)>max_Q %find
the relay with the oldest packet
                    max_sing_id=i3;
```

```
                            max_found=1;
                            max_Q=current_old(i3);
                        end

                        if  Dj_link(i3,1)==1&&Q(i3,1)<min_Q    %find  the
relay with the least packet
                            min_found=1;
                            min_Q=Q(i3,1);
                        end
                    end

                    if max_found==1
                        max=y2(max_sing_id,2);
                        max_id(1,1)=max_sing_id;
                        max_id(1,2)=-1;

                    elseif max_found==0&&min_found==1
                        for i3=1:K
                            if Dj_link(i3,1)==1&&Q(i3,1)==min_Q
                                c_count=c_count+1;
                                C(c_count)=i3;
                            end
                        end

                        if c_count==1
                            min_sing_id=C(1);
                            min_Q=Q(C(1),1);
                        else
                            t7=randi(c_count,1);
                            min_sing_id=C(t7);
                            min_Q=Q(C(t7),1);
                        end
                        max=y2(min_sing_id,1);
                        max_id(1,1)=min_sing_id;
                        max_id(1,2)=1;

                    else %  max_found==0&&min_found==0

                      for i3=1:K
                            for i4=1:K
                              if Q(i3,1)==0
                                flag(i3,i4)=0;
                              end
                            end
```

```
                              end
                                for i3=1:K
                                    for i4=1:K

                                       if
flag(i3,i4)==1&&((y2(i3,1))^2>=(y4(i3,i4)^2*2^(r0)*var4(1,1)*(1-P_up)/(P_up*var5
(i4,1))*b(i3,1)))

                                         DJ1_link(i3,i4)=1;  %for one i3, we need
to store all the indexs of jammers which satisfy the condition (8)
                                       else
                                          DJ1_link(i3,i4)=0;
                                       end
%
                                       if
flag(i3,i4)==1&&((y2(i3,2))^2>=(y2(i4,2))^2*2^(r0)*var5(i3,1)*(1-P_up)/(P_up*var
5(i4,1)))
                                        % Dj_link(i3,2)=-1;
                                          DJ2_link(i3,i4)=1;
                                       else
                                         DJ2_link(i3,i4)=0;
                                       end
                                    end
                                end
                                      %%---------------------------------    the
optimal relay and jammer----------------
                                for i3=1:K
                                    for i4=1:K
                                       if
DJ2_link(i3,i4)==1&&current_old(i3)>max_Q %find the relay with the oldest packet

                                           max_sing_id=i3;
                                           max_found=-1;
                                           max_Q=current_old(i3);
                                       end
                                        if
DJ1_link(i3,i4)==1&&Q(i3,1)<min_Q  %find the relay with the least packet
                                           min_sing_id=i3;
                                           min_found=-1;
                                           min_Q=Q(i3,1);
                                        end
                                    end
                                end
                                for i4=1:K
```

```
                                    if
DJ2_link(max_sing_id,i4)==1&&(y4(max_sing_id,i4)<min_J2)
                                        min_J2=y4(max_sing_id,i4);
                                        min_J2_id=i4;   % the optimal jammer
                                    end
                                    if
DJ1_link(max_sing_id,i4)==1&&(((y4(max_sing_id,i4))^2*b(max_sing_id,1))<min_J1)

min_J1=(y4(max_sing_id,i4))^2*b(max_sing_id,1);
                                        min_J1_id=i4;   % the optimal jammer
                                    end
                                 end
                                 if max_found==-1
                                     max=y2(max_sing_id,2);
                                     max_id(1,1)=max_sing_id;
                                     max_id(1,2)=-1;
                                 elseif max_found==0&&min_found==-1
                                     max=y2(min_sing_id,1);
                                     max_id(1,1)=min_sing_id;
                                     max_id(1,2)=1;
                                 end
                                 %%-------------------------------------------------
-----------------------------------
                             end


            %% ============================================

                    %%      ==============              outage       probability
============================


                    if max_found==0&&min_found==0

                        out_count=out_count+1;
                        for i4=1:K
                            if current_old(i4)==L
                                % if the last element contains data thats mean will
be dropped now
                                dropped_bits=dropped_bits+1;
                                Q(i4,1)=Q(i4,1)-1;
                                if next_old(i4)==0
                                    current_old(i4)=0;  % % nothing found
                                    next_old(i4)=0;     % no need still remain =0
                                else
```

```
                                    current_old(i4)=next_old(i4)+1; % because the
next_old will be shifted bec of the delay
                                    check=1;
                                    i3=next_old(i4)-1;
                                    while check==1&&i3>0
                                        if packet_indicator(i4,i3)==1
                                            next_old(i4)=i3+1;      %    because
will be shifted bec of the delay
                                            check=0;
                                        end
                                        i3=i3-1;
                                    end
                                    if check==1
                                        next_old(i4)=0; % nothing found
                                    end
                                end

                            elseif current_old(i4)~=0
                                current_old(i4)=current_old(i4)+1;
                                if next_old(i4)==0
                                    next_old(i4)=0; % remain =0
                                else
                                    next_old(i4)=next_old(i4)+1;
                                end
                            else    % current_old==0
                                current_old(i4)=0;
                                next_old(i4)=0;
                            end

                            packet_indicator(i4,2:L)=packet_indicator(i4,1:L-1);
    %shift the buffer bec of the delay
                            packet_indicator(i4,1)=0;   %  after  the  shift  the
first element is empty bec of the outage
                        end
                        %%
======================================================================p_s_indic
ator
                        if c_o_s==LS
                            dropped_bits=dropped_bits+1;
                            dropped_bits_source_only=dropped_bits_source_only+1;
                            p_s_indicator(c_o_s)=0;
                            QS(1,1)=QS(1,1)-1;
                            new_frame=1; % the new packet will arrive in the next
time slot
```

```
                        else
                             new_frame=0;
                        end

                        %%
=======================================================================
                    else  %either trans. or recev.

                        i5=max_id(1,1);
                        if  current_old(i5)<L
                            Q(max_id(1,1),1)=Q(max_id(1,1),1)+max_id(1,2); % num
of data in the selected relay buffer:
                        % thats mean if the buffer contains already L data(full) and
sel for rec
                        % don't increment Q(1,) bec one packet will be dropped
                        % elseif Q(max_id(1,1),1)==L&&max_id(1,2)==-1
                        elseif current_old(i5)==L&&max_id(1,2)==-1  %trans.
                            Q(max_id(1,1),1)=Q(max_id(1,1),1)+max_id(1,2);
                        end                                             % either
incremented or decremented by 1
                        %%
=======================================================================
                        if  max_id(1,2)==1                %relay  received  data
successfully
                            %%
==========================================================


                            p_s_indicator(c_o_s)=0;
                            QS(1,1)=QS(1,1)-1;
                            new_frame=1;
                            %%
==========================================================
                            % i5=max_id(1,1);
                            if current_old(i5)==L
                                % if the last element contains data thats mean will
be dropped now
                                dropped_bits=dropped_bits+1;
                                if next_old(i5)==0
                                    current_old(i5)=c_o_s;  %  the  new  arrived
packet
                                    next_old(i5)=0;      % no need still remain =0
                                else
                                    current_old(i5)=next_old(i5)+1; % because the
next_old will be shifted bec of the delay
```

```
                                    check=1;
                                    i3=next_old(i5)-1;
                                    while check==1&&i3>0
                                        if packet_indicator(i5,i3)==1
                                            next_old(i5)=i3+1;      %    because
will be shifted bec of the delay
                                            check=0;
                                        end
                                        i3=i3-1;
                                    end
                                    if check==1
                                        next_old(i5)=c_o_s; %  the  new  arrived
packet
                                    end
                                end

                            elseif current_old(i5)~=0

                                current_old(i5)=current_old(i5)+1;
                                if next_old(i5)==0
                                    next_old(i5)=c_o_s; % the new arrived packet
                                else
                                    next_old(i5)=next_old(i5)+1;
                                end

                            else
                                current_old(i5)=c_o_s;
                                next_old(i5)=0;
                            end

    packet_indicator(max_id(1,1),2:L)=packet_indicator(max_id(1,1),1:L-1);     %
shift the buffer and the indicator
                            packet_indicator(max_id(1,1),1)=0;  %the    youngest
packet is null
                            packet_indicator(max_id(1,1),c_o_s)=1;  %the     new
packet that just arrived at source

                            for i4=1:K
                                if i4~=max_id(1,1)
                                    if current_old(i4)==L
                                        % if the last element contains data thats
mean will be dropped now
                                        dropped_bits=dropped_bits+1;
                                        Q(i4,1)=Q(i4,1)-1;
                                        if next_old(i4)==0
                                            current_old(i4)=0;  %% nothing found
                                            next_old(i4)=0;     % no need still
remain =0
                                        else
                                            current_old(i4)=next_old(i4)+1; %
because the next_old will be shifted bec of the delay
                                            check=1;
                                            i3=next_old(i4)-1;
                                            while check==1&&i3>0
                                                if packet_indicator(i4,i3)==1
                                                    next_old(i4)=i3+1;      %
because will be shifted bec of the delay
                                                    check=0;
                                                end
                                                i3=i3-1;
                                            end
                                            if check==1
                                                next_old(i4)=0; % nothing found
                                            end
                                        end

                                    elseif current_old(i4)~=0
                                        current_old(i4)=current_old(i4)+1;
                                        if next_old(i4)==0
                                            next_old(i4)=0; % remain =0
                                        else
                                            next_old(i4)=next_old(i4)+1;
                                        end
                                    else    % current_old==0
                                        current_old(i4)=0;
                                        next_old(i4)=0;
                                    end


    packet_indicator(i4,2:L)=packet_indicator(i4,1:L-1);    %shift the buffer bec
of the delay
                                    packet_indicator(i4,1)=0;
                                end
                            end


                        else                        %relay transmitted successfully
```

```
                            %%
=======================================================================
                            if c_o_s==LS
                                dropped_bits=dropped_bits+1;

    dropped_bits_source_only=dropped_bits_source_only+1;
                                p_s_indicator(c_o_s)=0;
                                QS(1,1)=QS(1,1)-1;
                                new_frame=1;
%                             else
%                                 new_frame=0;
                            end

                            %%
=======================================================================%%
                            packet_indicator(i5,current_old(i5))=0;     %    the
place of the transmitted packet is now empty
                            rec_r0_count=rec_r0_count+1;
                            if current_old(i5)==1
                                current_old(i5)=0;
                                next_old(i5)=0;
                            else    % current_old>1
                                if next_old(i5)==0
                                    current_old(i5)=0;  % no other packet in the
buffer
                                    next_old(i5)=0;     % no need still remain =0
                                else
                                    current_old(i5)=next_old(i5)+1; % because the
next_old will be shifted bec of the delay
                                    check=1;
                                    i3=next_old(i5)-1;
                                    while check==1&&i3>0
                                        if packet_indicator(i5,i3)==1
                                            next_old(i5)=i3+1;      %    because
will be shifted bec of the delay
                                            check=0;
                                        end
                                        i3=i3-1;
                                    end
                                    if check==1
                                        next_old(i5)=0; % nothing found
                                    end
                                end
                            end
    packet_indicator(max_id(1,1),2:L)=packet_indicator(max_id(1,1),1:L-1);      %
shift delay buff after Tx
                            packet_indicator(max_id(1,1),1)=0;

                            for i4=1:K
                                if i4~=max_id(1,1)
                                    if current_old(i4)==L
                                        % if the last element contains data thats
mean will be dropped now
                                        dropped_bits=dropped_bits+1;
                                        Q(i4,1)=Q(i4,1)-1;
                                        if next_old(i4)==0
                                            current_old(i4)=0;  %% nothing found
                                            next_old(i4)=0;     % no need still
remain =0
                                        else
                                            current_old(i4)=next_old(i4)+1; %
because the next_old will be shifted bec of the delay
                                            check=1;
                                            i3=next_old(i4)-1;
                                            while check==1&&i3>0
                                                if packet_indicator(i4,i3)==1
                                                    next_old(i4)=i3+1;      %
because will be shifted bec of the delay

                                                    check=0;
                                                end
                                                i3=i3-1;
                                            end
                                            if check==1
                                                next_old(i4)=0; % nothing found
                                            end
                                        end

                                    elseif current_old(i4)~=0
                                        current_old(i4)=current_old(i4)+1;
                                        if next_old(i4)==0
                                            next_old(i4)=0; % remain =0
                                        else
                                            next_old(i4)=next_old(i4)+1;
                                        end
                                    else    % current_old==0
                                        current_old(i4)=0;
                                        next_old(i4)=0;
                                    end



    packet_indicator(i4,2:L)=packet_indicator(i4,1:L-1);    %shift the buffer bec
of the delay
                                    packet_indicator(i4,1)=0;
                                end
                            end
                        end
                    end
                end
                 %% ============  end of 100 trials  =========================
                out_delay_sim_K2_D5_ind(LOOP)=out_count/trials;
            num_packet
                %drop_delay_sim_K2_D5_ind(LOOP)=dropped_bits/trials;
                drop_delay_sim_K2_D5_ind1(LOOP)=dropped_bits/num_packet;

    drop_source_delay_sim_K2_D5_ind(LOOP)=dropped_bits_source_only/trials;

                cap_delay_sim_K2_D5_ind(LOOP)=rec_r0_count*r0/trials;
%                 out_delay_sim_K2_D5_ind=out_count/trials;
%
%               drop_delay_sim_K2_D5_ind=dropped_bits/trials;
%
%               drop_source_delay_sim_K2_D5_ind=dropped_bits_source_only/trials;
%
%               cap_delay_sim_K2_D5_ind=rec_r0_count*r0/trials;
             end
            out_delay_sim_K2_D5_ind
           % drop_delay_sim_K2_D5_ind
            drop_delay_sim_K2_D5_ind1
            drop_source_delay_sim_K2_D5_ind
            cap_delay_sim_K2_D5_ind
%
            %% =================   end snr_loop    ========================
%
    semilogy(snr_range(1:snr_limit)-1,out_delay_sim_K2_D5_ind(1:snr_limit),'b-*'
)
%           axis([0 snr_limit-1 10^-3 1])
%           grid on
%
%           figure
%
```

```
%
    semilogy(snr_range(1:snr_limit)-1,drop_delay_sim_K2_D5_ind(1:snr_limit),'b-*
')
%           axis([0 snr_limit-1 10^-4 1])
%           grid on
%
%           figure
%
%
    semilogy(snr_range(1:snr_limit)-1,drop_source_delay_sim_K2_D5_ind(1:snr_limi
t),'b-*')
%           axis([0 snr_limit-1 10^-4 1])
%           grid on
%
%           figure
%
%
    plot(snr_range(1:snr_limit)-1,2*r0*cap_delay_sim_K2_D5_ind(1:snr_limit),'b-*
')
%             axis([0 snr_limit-1 10^-4 1])
%           grid on
```