A Project Report on

Performance Evaluation of BLUE Estimation and BLUE equalisation on BPSK and QPSK over Rayleigh Channel

Communication Signal Processing and Algorithms (TE61003)

Submitted by

Lakshmi Sudha Rani Nimmakayala

Roll no: **22GS61R08**

November 2022

G.S Sanyal School of Telecommunications

Indian Institute of Technology Kharagpur

Kharagpur, West Bengal-721302

Contents:

1.Project Objective :

The aim of this project is to analyze the Performance of BPSK and QPSK over Rayleigh fading channel. The Best Linear Unbiased Estimator (BLUE) will be used to estimate the channel. The performance of the BPSK and QPSK is studied on the basis of Bit Error Rate (BER) and Mean Square Error (MSE) of the channel estimator. This project results are simulated using MATLAB R2022a.

1.1 Project Details

The system model considered for this project is shown in Fig 1.1. It consists of 1 transmit antenna and 1 receive antenna. We send random data that has equiprobable of 0's and 1's. Data is modulated with Binary Phase Shift Keying (BPSK) or Quadrature Phase Shift Keying (QPSK) with each bit having unit energy. These modulated symbols are arranged into packets along with pilots are transmitted over a Rayleigh fading channel. The Blue estimator is used to estimate the channel "h" with the help of known sequences called as pilots. Then received symbols are compensated with "$\hat{h}$" using BLUE equalization technique. BPSK or QPSK demodulation is performed and then BER vs SNR and MSE vs SNR is plotted.

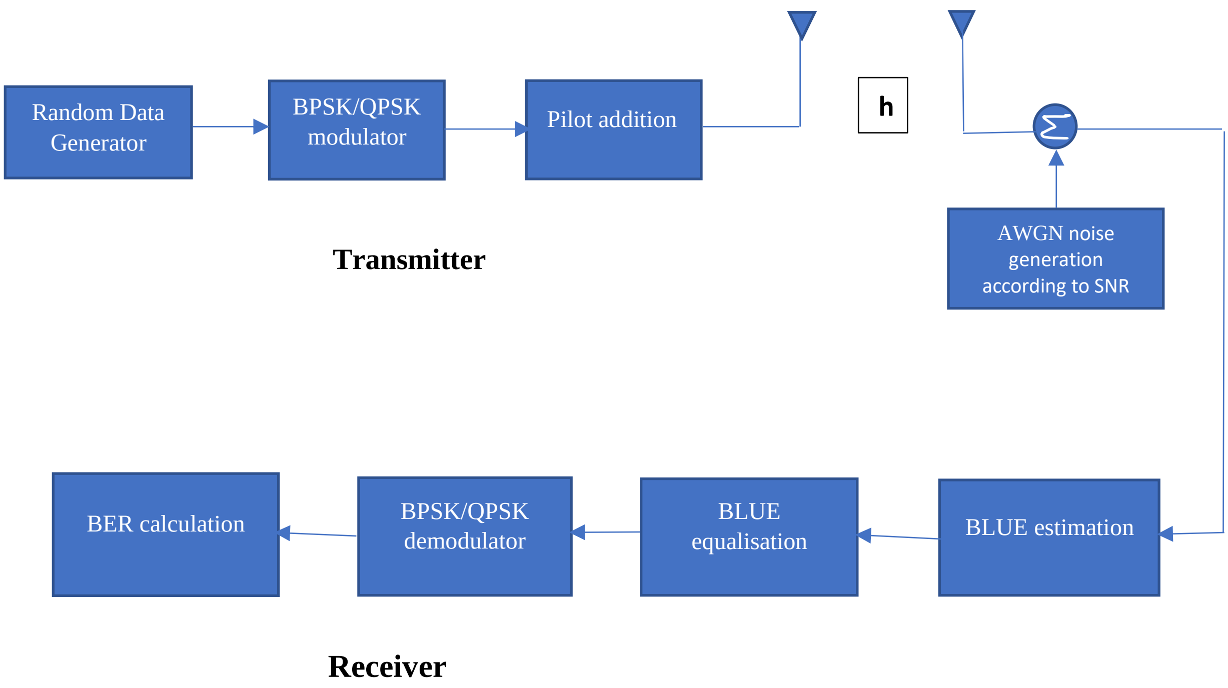

Fig.1.1 System Model

2. Introduction

2.1 IEEE Standard 802.11n pilots for system

The wireless system is a packet-switched system with random access protocol. This essentially means that the receiver has no priori knowledge about the arrival time of the packet. To perform synchronization of the system, a packet is preceded with a known sequence, i.e., a preamble. The preamble is carefully designed to provide sufficient information for the receiver to perform packet detection, frequency offset estimation, symbol timing acquisition, and channel estimation.

802.11n long training sequence, LTS=[1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0, 1, -1, -1, 1,1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1]

2.2 Best Linear Unbiased Estimator (BLUE) :

Sometimes we may not know the PDF of received data , in that case to estimate any parameter using MVUE (Optimal estimator) does not work. Then we go for suboptimal estimators.

BLUE is one of the suboptimal estimators in "classical estimators". It requires knowledge of only the mean and covariance of the data (first and second moments of the PDF). The only constraint is, estimator is to be linear in the data and unbiased.

For wireless BLUE Channel Estimation, we have data model given as $Y = hX + W$, where Y is received signal, X is transmitted pilot signal, h is unknown fading channel coefficients. w is additive complex white Gaussian noise.The mean of the Gaussian noise is 0, variance equal to $\boldsymbol{\sigma^2}$. Estimation of this unknown channel coefficient h, termed as Channel Estimation.

$$\hat{h} = \frac{X^T C^{-1} Y}{X^T C^{-1} X} \qquad \text{,as the noise is WGN means } C^{-1} = \boldsymbol{I/\sigma^2}$$

Hence, $\hat{h} = \frac{X^T Y}{X^T X}$

2.3 Best Linear Unbiased (BLUE) equalizer:

The first one, Zero Forcing channel equalization can be performed by dividing the received signal by the channel's discrete Fourier transform. This operation can be written as:

$$\boldsymbol{y_{zf\ equalized}} = \frac{\boldsymbol{y}}{\hat{h}} = \boldsymbol{x} + \frac{\boldsymbol{w}}{\hat{h}}$$

Thus, the equalized symbol will ideally equal the transmitted symbol plus noise term. Although the channel response is initially unknown, it can be determined with the use of training sequence. Second one, BLUE equalization is used to retrieve transmitted symbol at receiver end, however noise is associated with it. Hence we have to keep threshold detection after equalization.

$$X_{BLUE\_equalised} = \frac{\hat{h}^H Y}{\hat{h}^H \hat{h}}$$

3.Simulations and Results:

The simulation is done for transmission of 100 packets. Each packet has 500 data symbols. For the simulation, only 100 channel coefficients are taken from the distribution.The bit error rate for different scenarios are plotted and the effect of no estimation is depicted and discussed. The Mean square error for BLUE estimation is plotted for different SNR values.

3.1 MATLAB program:

3.11 BPSK :

```
%this code is written by Lakshmi Sudha Rani Nimmakayala with RollNo:22GS61R08

%wireless test bed on BLUE ESTIMATION and BLUE EQUALISATION FOR BPSK
% Date: 12-11-2022
%Version: 1

%Transmitter
clc;
clear all;
n_p=100; %number of packets
bits_packet=500; %number of bits per packet
len=bits_packet*n_p; %length of bits to be generated
LTS=[ 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1,
1, 1, 1, 0, 1, -1, -1, 1,1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1,
1, -1, 1, 1, 1, 1];
h=readmatrix("channel.csv");
h_Channel=h(:,1);%rayleigh channel coefficients(1st column)
ts=[];
%taking the Long training sequence 5-times
for j = 1:5
 ts = [ts LTS];
end
len_ts=length(ts);
ip = randn(1,len)<0;%binary sequence

%BPSK modulation
x = 2*ip -1; %binary sequence mapping to +1 and -1
Tx=[];
%creating transmit matrix for 100 packets
for i=1:n_p
 Tx(i,:)=[ts x((((len/n_p)*i)-(len/n_p)+1):(len/n_p)*i)];
end

SNRdB=0:15;
for j=1:length(SNRdB)
    SNR=10^(SNRdB(j)/10);

w=(1/sqrt(2*SNR))*(randn(n_p,len_ts+bits_packet)+1j*randn(n_p,len_ts+bits_packet))
;%complex noise matrix
    RX_AWGN=Tx+w; %received Signal in AWGN channel
```

```
    RX_RAY=h_Channel.*Tx+w; %received Signal in Rayleigh channel
    Rx_out_blue=[];
    Rx_out_AWGN=[];
    Rx_out_RAY=[];
    Rx_out_noe=[];
    temp=zeros(1,bits_packet);

    %RECEIVER side

    for k=1:n_p
        %AWGN channel
        for l=1:bits_packet
            if real(RX_AWGN(k,l+len_ts))>0
                temp(l)=1;
            else
                temp(l)=0;
            end
        end
        Rx_out_AWGN=[Rx_out_AWGN temp];
        % detection of received bits without Channel estimation
        for l=1:bits_packet
            if real(RX_RAY(k,l+len_ts))>0
                temp(l)=1;
            else
                temp(l)=0;
            end
        end
        Rx_out_noe=[Rx_out_noe temp];
        % dection of received bits with perfect CSI
        for l=1:bits_packet
            if real(RX_RAY(k,l+len_ts)/h_Channel(k))>0
                temp(l)=1;
            else
                temp(l)=0;
            end
        end
        Rx_out_RAY=[Rx_out_RAY temp];
        %Blue estimation
        h_blue(k)=sum(Tx(k,1:len_ts).*RX_RAY(k,1:len_ts))/sum(Tx(k,1:len_ts).^2);
        %blue equilisation using blue estimated channel coefficients

x_est(1:bits_packet)=(conj(h_blue(k)).*(RX_RAY(k,1+len_ts:end)))/(conj(h_blue(k)).
*(h_blue(k)));
        for l=1:bits_packet
            if real(x_est(l))>0
                temp(l)=1;
            else
                temp(l)=0;
            end
        end
        Rx_out_blue=[Rx_out_blue temp];
    end
    BER1(j)=sum(xor(ip,Rx_out_AWGN))/len; %BER for awgn
    BER2(j)=sum(xor(ip,Rx_out_noe))/len; %BER for no channel estimation
    BER3(j)=sum(xor(ip,Rx_out_RAY))/len; %BER for perfect CSI
    BER4(j)=sum(xor(ip,Rx_out_blue))/len; %BER for blue estimation,blue
equalisation
    BER_th(j)=0.5*erfc(sqrt(SNR)); %Theoretical BER for AWGN
    BER_rth(j)=0.5*(1-sqrt(SNR/(1+SNR))); %Theoretical BER for rayleigh
```

```
    mse(j)=mean(abs(h_blue-h_Channel.').^2); %mean square error for h and h_est
end

% graphs for BER vs SNR
figure;
semilogy(SNRdB,BER1,'r-*');
hold on;
semilogy(SNRdB,BER_th,'c');
grid on;
semilogy(SNRdB,BER_rth,'g');
semilogy(SNRdB,BER2,'k->');
semilogy(SNRdB,BER3,'b');
semilogy(SNRdB,BER4,'m+');
legend('AWGN simulation results','AWGN theoretical','rayleigh theoretical','No
channel estimation','equalisation with perfect CSI','BLUE
estimation&equalisation');
title ('BER vs SNR(dB) for BPSK');
xlabel('SNR(dB)');
ylabel('BER');
axis([0 15 10^-5 1]);

figure;
semilogy(SNRdB,mse,'k');
legend('mse');
grid on;
title ('mse vs SNR(dB)');
xlabel('SNR(dB)');
ylabel('mse');
```

3.12. QPSK:

```
%this code is written by Lakshmi Sudha Rani Nimmakayala with
%RollNo:22GS61R08
%wireless test bed on BLUE ESTIMATION and BLUE EQUALISATION FOR QPSK
% Date: 12-11-2022
%Version: 2

%Transmitter
clc;
clear all;
n_p=100; %number of packets
symbols_packet=500; %number of symbols per packet
len=symbols_packet*2*n_p; %length of binary sequence to be generated
LTS=[ 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1,
1, 1, 1, 0, 1, -1, -1, 1,1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1,
1, -1, 1, 1, 1, 1];
h=readmatrix("channel.csv");
h_Channel=h(:,1);%rayleigh channel coefficients(1st column)
ts=[];
%taking the Long training sequence 5-times
for j = 1:5
 ts = [ts LTS];
end
len_ts=length(ts);
ip = randn(1,len)<0;%binary sequence
```

```
%QPSK modulation
for n =1:length(ip)/2
     two_bits = ip(2*n-1:2*n);
     if (two_bits(1) == 0 && two_bits(2) == 0)
           symbol_temp = -0.7071-1i*0.7071;
     elseif(two_bits(1) == 0 && two_bits(2) == 1)
           symbol_temp = -0.7071+1i*0.7071;
     elseif(two_bits(1) == 1 && two_bits(2) == 1)
           symbol_temp = 0.7071+1i*0.7071;
     else
           symbol_temp = 0.7071-1i*0.7071;
     end
           x(n) = symbol_temp;
end
len_sym=length(x);
Tx=[];
%creating transmit matrix for 100 packets
for i=1:n_p
 Tx(i,:)=[ts x((((len_sym/n_p)*i)-(len_sym/n_p)+1):(len_sym/n_p)*i)];
end

SNRdB=0:15;
for j=1:length(SNRdB)
    SNR=10^(SNRdB(j)/10);

w=(1/sqrt(2*SNR))*(randn(n_p,len_ts+symbols_packet)+1j*randn(n_p,len_ts+symbols_pa
cket));%complex noise matrix
    RX_AWGN=Tx+w; %received Signal in AWGN channel
    RX_RAY=h_Channel.*Tx+w; %received Signal in Rayleigh channel
    Rx_out_blue=[];
    Rx_out_AWGN=[];
    Rx_out_RAY=[];
    Rx_out_noe=[];

    %RECEIVER side

    for k=1:n_p
        %AWGN channel
        for l=1:symbols_packet
            if (real(RX_AWGN(k,l+len_ts))>0 && imag(RX_AWGN(k,l+len_ts))>0)
                rx_bits=[1 1];
            elseif (real(RX_AWGN(k,l+len_ts))>0 && imag(RX_AWGN(k,l+len_ts))<0)
                rx_bits=[1 0];
            elseif (real(RX_AWGN(k,l+len_ts))<0 && imag(RX_AWGN(k,l+len_ts))>0)
                rx_bits=[0 1];
            else
                rx_bits=[0 0];
            end
            Rx_out_AWGN=[Rx_out_AWGN rx_bits];
        end
        %No channel estimation
        for l=1:symbols_packet
            if (real(RX_RAY(k,l+len_ts))>0 && imag(RX_RAY(k,l+len_ts))>0)
                rx_bits=[1 1];
            elseif (real(RX_RAY(k,l+len_ts))>0 && imag(RX_RAY(k,l+len_ts))<0)
                rx_bits=[1 0];
            elseif (real(RX_RAY(k,l+len_ts))<0 && imag(RX_RAY(k,l+len_ts))>0)
                rx_bits=[0 1];
            else
```

```
                rx_bits=[0 0];
            end
            Rx_out_noe=[Rx_out_noe rx_bits];
        end
        %equalisation with perfect CSI
        true_est=RX_RAY(k,len_ts+1:end)/h_Channel(k);
        for l=1:symbols_packet
            if (real(true_est(l))>0 && imag(true_est(l))>0)
                rx_bits=[1 1];
            elseif (real(true_est(l))>0 && imag(true_est(l))<0)
                rx_bits=[1 0];
            elseif (real(true_est(l))<0 && imag(true_est(l))>0)
                rx_bits=[0 1];
            else
                rx_bits=[0 0];
            end
            Rx_out_RAY=[Rx_out_RAY rx_bits];
        end
        %---------------------------------------------------------
         %Blue estimation
        h_blue(k)=sum(Tx(k,1:len_ts).*RX_RAY(k,1:len_ts))/sum(Tx(k,1:len_ts).^2);
        %blue equilisation using blue estimated channel coefficients

x_est(1:symbols_packet)=(conj(h_blue(k)).*(RX_RAY(k,1+len_ts:end)))/(conj(h_blue(k
)).*(h_blue(k)));
        for l=1:symbols_packet
            if (real(x_est(l))>0 && imag(x_est(l))>0)
                rx_bits=[1 1];
            elseif (real(x_est(l))>0 && imag(x_est(l))<0)
                rx_bits=[1 0];
            elseif (real(x_est(l))<0 && imag(x_est(l))>0)
                rx_bits=[0 1];
            else
                rx_bits=[0 0];
            end
            Rx_out_blue=[Rx_out_blue rx_bits];
        end
        %---------------------------------------------------------
    end
    BER1(j)=sum(xor(ip,Rx_out_AWGN))/len; %BER for awgn
    BER_th(j)=0.5*erfc(sqrt(SNR/2)); %Theoretical BER for AWGN
    BER2(j)=sum(xor(ip,Rx_out_noe))/len; %BER for no channel estimation
    BER3(j)=sum(xor(ip,Rx_out_RAY))/len; %BER for perfect CSI
    BER4(j)=sum(xor(ip,Rx_out_blue))/len; %BER for blue estimation,blue
equalisation
end
% graphs for BER vs SNR
figure;
semilogy(SNRdB,BER1,'r-*');
hold on;
semilogy(SNRdB,BER_th,'c');
grid on;
semilogy(SNRdB,BER2,'k->');
semilogy(SNRdB,BER3,'b');
semilogy(SNRdB,BER4,'m+');
legend('AWGN simulation results','AWGN theoretical','No channel
estimation','equalisation using perfect CSI','BLUE estimation&equalisation');
title ('BER vs SNR(dB) for QPSK');
xlabel('SNR(dB)');
```

```
ylabel('BER');
axis([0 15 10^-5 1]);
```

3.2 Simulation Results:

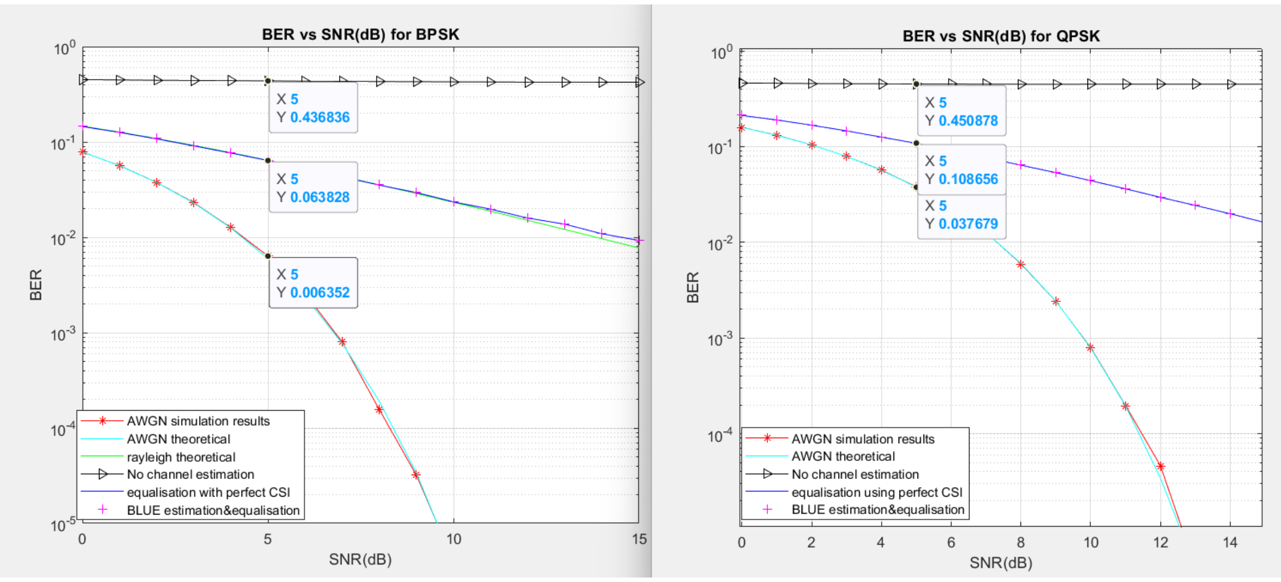

Fig.3.2a: BER of BPSK and QPSK over AWGN channel and Rayleigh channel

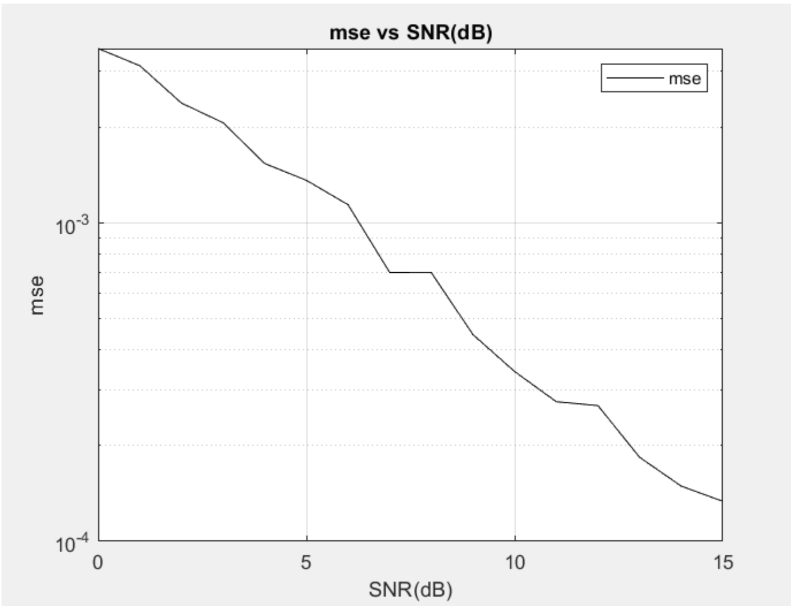

Fig. 3.2b: MSE of channel estimation error

4. Discussion :

Fig.3.2a. is the graph of BER vs SNR for BPSK and QPSK over AWGN and Rayleigh channels(these graphs are generated from BPSK code and QPSK code are kept side by side for comparison purpose). For AWGN channel, BER for BPSK in simulation is almost matching with the theoretical values till

SNR of 9 dB and after that the simulation BER is becoming zero which is probably due to limited number of symbols considered. In Rayleigh channel, the simulation was performed for three different cases: (i) Without any equalization (ii)Zero forcing Equalization with the original channel coefficients 'h'. (iii)BLUE Equalization with the BLUE estimated channel coefficients. All the three cases are compared with the theoretical BER values for Rayleigh channel. Equalization with estimated and original channel coefficients resulted in almost identical BER plot which indicates that the estimation carried out was very efficient and both the plots are very close to theoretical plot. BER plot for the case of 'No equalization' is completely away from the theoretical plot which emphasizes the need of channel estimation and removing the channel effect .

Comparison between BPSK and QPSK: It is evident from the Fig.3.2a. that QPSK error is more than BPSK error (at SNR=5dB, at highlighted points, we can see BER for BPSK and QPSK). But the advantage of higher order modulation schemes is we can transmit more bits per symbol which increase the data rate at the cost of high probability of error.

In Fig.3.2b. Mean Square Error of BLUE estimator and original channel coefficients is plotted against different SNR values. Mean Square Error of BLUE estimator and original channel coefficients is plotted against different SNR values. We can observe MSE of channel effect decreases with the increase of SNR. For getting a minimum mean square error, the design of estimator and the operating SNR value plays crucial role.

5.Conclusion:

The BLUE estimator for estimating Rayleigh channel coefficients is designed and the effect of Rayleigh channel on communication system is compared with the AWGN channel in the BPSK modulation scheme and QPSK modulation scheme. The effect of no estimation and equalization on the performance of the system is observed. The mean square error is observed for various values of SNR. From the results, we can conclude that estimation and equalization of unknown parameters is must before the detection of data, and better the estimator performance better the system performance for successful communication.

6.References

- Fundamentals of Statistical Signal Processing: Estimation Theory, Steven M.kay, University of Rhode Island.
- https://mentor.ieee.org/802.11/dcn/04/11-04-0931-00-000n-itri-preamble-spec.doc