Simulation of Adaptive Bit Loading for Wireless Network Channels

Submitted by:

Nithin Dinker Kamath
nkamath@usc.edu

ABSTRACT

My project involved the study and analysis of Adaptive Bit Loading Technique used in Wireless Channels using Frequency Division Multiplexing (FDM). To understand this technique I simulated the same on MatLab platform. The reason for using MatLab was the ease in with which matrices can be manipulated. Hence I could store the parameters for the Sub Channel as a matrice.

My simulation consisted of two parts.

a. Generate a random wireless channel. Divide it into Sub Channels and then transform it into frequency domain using Fast Fourier Transform and Single Value Decomposition
b. Do Adaptive Bit Loading. The parameter of the Sub Channel which indicates the whether the Sub Channel is "Good" or "Bad" was the gain it provided.

The Algorithm which was used do Energy Allocation was Chow Algorithm and to do optimized Bit Allocation was Campello.

1 INTRODUCTION

Wireless channels are inherently heterogeneous, i.e. that is amount of fading and noise varies considerably from sub channel to sub channel. The reason for this is maybe various: atmospheric conditions, frequency spectrum selected, selective fading or any random phenomenon.

Frequency Division Multiplexing (FDM) has been a widely-used technique for signal transmission in frequency selective channels. In essence, FDM divides the channel bandwidth into sub channels and transmits multiple relatively low rate signals by carrying each signal on a separate carrier frequency.

Hence, some sub channels may have good gain: that is, provide relatively less attenuation, distortion and noise interference, while other sub channels may be bad because the sub channel Signal to Noise Ratio (SNR) may be high.

2 EFFECTS OF SNR

While it is not really means that the bits traveling through sub channel that has low SNR will not be received, the probability that such receptions being bad or corrupt is high. Also, other factors which effects successful reception are

a. Frequency Selected
b. Guard Band
c. Power Allocated
d. Modulation Scheme

In FDM, the data packets are split into smaller packets, each of which is transmitted on a particular sub channel. Hence if a particular sub channels fails, on the receiver side, it might be required to retransmit the packet again. Hence there are lots of re-transmissions.
Another variation of FDM is known as OFDM, in which the sub channel frequencies are selected such a way that they are orthogonal to each other. That is they have a phase division of 90^0 between. Even though this reduces Inter Symbol Interference, the loss because of the channel is still very much present.

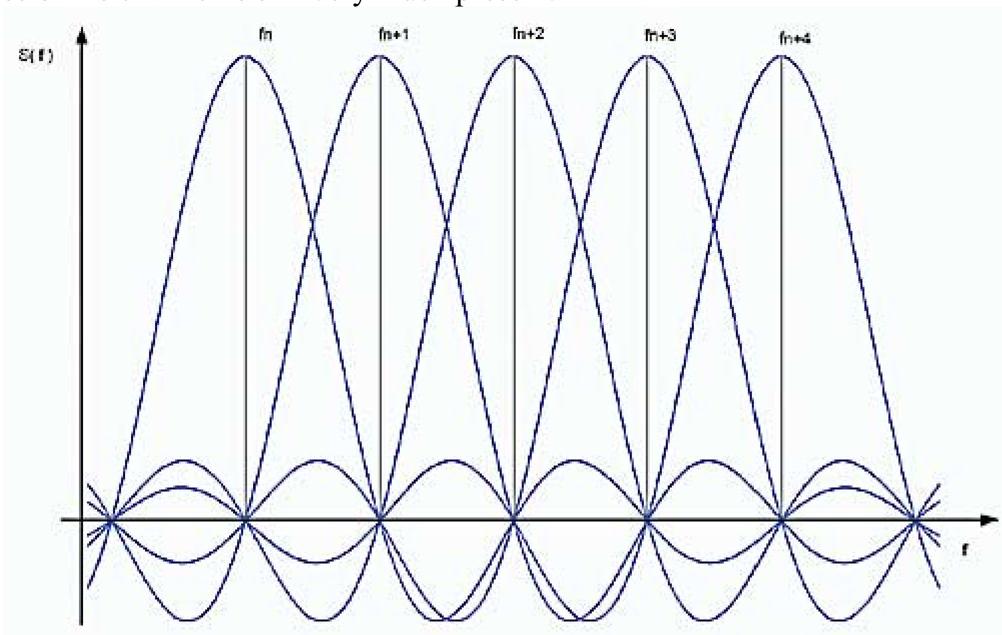

Figure 1: OFDM Channel

3 ADAPTIVE BIT LOADING

One of the possible solutions for this is using Adaptive Bit Loading. What this technique does that

a. More number of bits are made to go through better part of the channel. (Bit Allocation)
b. More energy is allocated to sub channels which are corrupt. (Energy Allocation)

The reason of allocating more bits to better sub channel is that, since these channels have better SNR, large number of bits can be received with high degree of error-freeness. At the same time, by allocating more energy to poorer sub carriers, you are increasing signal strength of such sub channels so much that they can overcome noise that will be added by the channel.
Hence the advantages are two pronged:

a. Send more bits through better sub channels so large number of bits are received error free.
b. Give more energy to bad sub channels so that it improves the chance of successful reception.

4 SIMULATION

4.1 Channel Creation

A random channel was created. The parameters that were assumed for the channel were: The channel provided exponential fading. That is the signal strength decayed exponentially with distance.

Code:
```
H_int = 1/sqrt(2)*(randn(length(A), 1) + j*randn(length(A), 1));
```

Then the channel was transformed from Time Domain to Frequency Domain using Finite Fourier Transform and Single Value Decomposition

Code:
```
H_f = zeros(1, (N));
h_f = fft(H_int2(1:1:(N-1)+1,1));
for k = 1:(N)
    H_f(1,1+(k-1)) = h_f(k);
end


for i = 1:N
    [Utmp Stmp Vtmp] = svd(H_f(:,(i-1)+1:i));
    S=[S Stmp];
end
```

Now the parameter S will store the gain of the sub channel. Since S is a matrix, S(i) is the gain of the ith sub channel.

4.2 Adaptive Bit Loading

Adaptive Bit Loading was done using two algorithm

1. Chow Algorithm [1]
2. Campello Algorithm [2]

Initially each sub channel was allocated energy using Chow Allocation and at the same time it was allocated maximum number of bits allowable per sub channel. Then bit allocation was done using Campello Algorithm.

Before running the algorithm, sub channels SNR was determined.

Code:
```
SNR = abs((subchan_gains.^2)./(noise));
```

From Chow's Algorithm, compute the energy for the ith sub channel based on the number of bits

$ei(b(i)) = (2^{b(i)} - 1)/SNR(i)$

Implementing Chow's Algorithm

Code:
```
for i = 1:num_subc
    bits_alloc(i) = M;                    %Allocating Maximum Number of
Bits allowable Sub Channel to each Sub Channel
```

```
    energy_alloc(i)  = (2^bits_alloc(i)-1)/SNR(i) ; %Allocating energy
to each Sub Channel
end
```

Then I compiled a matrix of energy increments. That is extra energy required by a sub channel to transmitting n bits to transmit (n+1) bits. This was calculated using
$\Delta ei(b) = ei(n) - ei(n - 1) = (2^{n-1})/SNR(i)$
This was stored in a matrix so that it can be referenced while implementing Campello's Algorithm

Code:
```
energytable = abs((1./SNR)'*(2.^([1:M+1]-1)));
energytable(:,M+1) = Inf*ones(size(energytable(:,M+1)));
for i = 3:2:M
    energytable(:,i) = (energytable(:,i) +energytable(:,i+1))/2;
    energytable(:,i+1) = energytable(:,i);
end
```

Campello's Algorithm
B' = 0
for n (1 to N) where N = number of sub channels
    while (B' != B); where B = maximum bits allowable per sub channel
        if (B > B)
            n = arg $\max_{1<j<N} \Delta ej(bj)$
            B = B – 1
            b(n)=b(n) – 1
        else
            n = arg $\max_{1<j<N} \Delta ej(bj+1)$
            B = B + 1
            b(n)=b(n) + 1

Implementation of Campello's Algorithm

Code:
```
bt = M*num_subc;
while (bt ~= total_bits)
    if (bt > total_bits)
        max_val = 0;
        max_ind = ceil(rand(1)*num_subc);
        for i = 1:num_subc
            if(bits_alloc(i) ~= 0)
                temp = energytable(i,bits_alloc(i));
            else
                temp = 0;
            end
            if(temp > max_val)
                max_val = temp;
                max_ind = i;
            end
        end
        if(bits_alloc(max_ind) > 0)
            bits_alloc(max_ind) = bits_alloc(max_ind) -1;
```

```
            energy_alloc(max_ind) = energy_alloc(max_ind) - max_val;
            bt = bt-1;
        end
    else
        min_val = Inf;
        min_ind = ceil(rand(1)*num_subc);
        for i = 1:num_subc
            if(bits_alloc(i) ~=0 & bits_alloc(i) <9)
                temp = energytable(i,bits_alloc(i) + 1);
            else
                temp = Inf;
            end
            if(temp < min_val)
                min_val = temp;
                min_ind = i;
            end
        end
        if(bits_alloc(min_ind) < 8)
            bits_alloc(min_ind) = bits_alloc(min_ind) + 1;
            if (min_val == inf)
                min_val = energytable(min_ind,bits_alloc(min_ind));
            end
            energy_alloc(min_ind) = energy_alloc(min_ind) +min_val;
            bt = bt+1;
        end
    end
end
```

5 SIMULATION RESULTS

The following input parameters were used for the simulation

Code:

```
A = [1 1/exp(1) 1/exp(2) 1/exp(3)];        %Power Delay Profile
Nchannel = 64;                             %Number of Sub Channels
Noise = 1e-3                               %Noise in the Channel
Bsub = 8;                                  %Maximum Number of Bits per Sub
Channel
Bmax = Nchannel*Bsub;                      %Maximum Channel Capacity
Btotal = Bmax/2;                           %Total Number of Bits to be
Transmitted
```

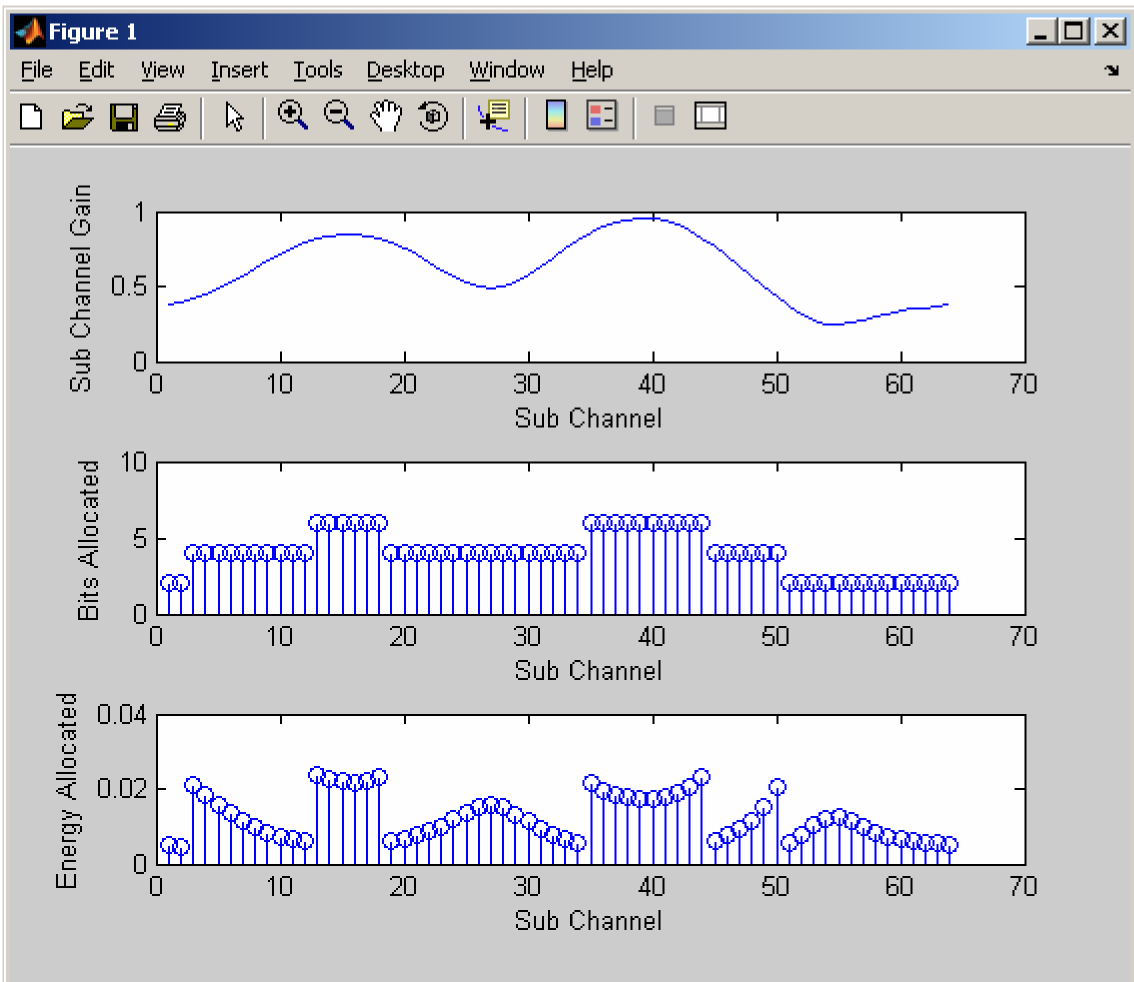

Figure 2: Simulation Result

6 FACTORS AFFECTING ADAPTIVE BIT LOADING

The various factors which effect bit loading is

a. Bits transmitted. If the number of bits transmitted is very close to the maximum channel capacity, then the efficiency of this technique decreases. The reason being, we have to allocate maximum number of bits possible to the sub channel.
b. Input Power. The lower the input power given the lesser the energy allocated per sub channel.

7 REQUIREMENTS AND OVERHEADS OF ADAPTIVE BIT LOADING

Requirements include

a. Transmitter and Receiver: The transmitter and receiver should be able to handle multiple modulation schemes. This is because the modulation scheme used to transmitted 2 bit may be different for one used to transmit 6 or 8 bits. Also they should be adaptive since they have to switch between different reception
b. Instantaneous Channel Knowledge: We mush have instantaneous channel knowledge to implement this technique. This is obtained by transmitting a “pilot” signal at regular intervals.

Of the overheads

a. Overhead due to transmission of pilot signals
b. Overhead due to reconfiguration of transmitters and receivers
c. Overhead due to synchronization

8 ADVANTAGES OF ADAPTIVE BIT LOADING

The advantages of using Adaptive Bit Loading are

a. Reduction of BER: One of the obvious advantages is that since the probability of successful reception of bits is increased because of bit and energy allocation lower BERs can be achieved. If along with Adaptive Bit Loading, Error Control is done, further reduction of BER is possible
b. Reduction in Retransmissions: Because more and more bits are successfully received, the need to re transmit data is reduced
c. Higher Data Transfer Rates: Higher data rates can be achieved because of the above two reasons
d. Energy Conservation: Energy is conserved because lesser energy is allocated to good sub channels. Also at the same time if a sub channel is extremely bad and is not transmitting any bits because 0 bits were allocated to it during bit eallocation, no power needs to be supplied to such sub channel. Also because less number of re transmissions occur, the overhead because of this is reduced

9 APPLICATIONS IN DATA NETWORKS

Adaptive Bit Loading can be implemented in Wireless Data Networks which uses FDM or OFDM to transmit data. In fact, it can be implemented at any node in the network which sends data over multiple channels. However to implement this the node should know exact channel it uses to communicate with another node. Also both transmitting node and receiving node should know how exactly the bits are allocated over multiple channels. Higher data rates can be acehived because of this technique. However there will be some necessary overheads. Like transmitting some data to find the channel status.

One of the other application is in Ad-Hoc Networks. A Mobile Ad-Hoc Network is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves, i.e., routing functionality will be incorporated into mobile nodes.

Now in this case, the nodes needs to continuously need to establish and dis connect links to other nodes. Hence, we can determine the channel status during such establishment and implement Adaptive Bit Loading.

Advantages of using Adaptive Bit Loading in such Networks are:

a. No additional overhead in determining th channel condition as this can be found out during link establishment
b. Energy conservation is one of the important concerns of such networks

10 COMMERCIAL APPLICATIONS

Commercially Adaptive Bit Loading is implemented in various wireless standards that use OFDM or FDM to transmit in a wireless data network. It is been used in various mobile networks.

CONCLUSION

While providing error free data transfer is the responsibility of Data Link and Network Layer of the OSI Model, these layers have no control over the channel's behavior. While we can always use parity checks and error correcting codes, there is always a chance that the bits maybe corrupted during transmission.

While this problem is not that apparent when the nodes are connected using physical medium like cables, when we use wireless channel for connection this becomes very apparent.

Wireless channels have unpredictable behavior and are very dynamic. They can be easily influenced by external conditions.

What Adaptive Bit Loading does, is get channel information and use this information. It tries to provide reliability into the channel.

Adaptive Bit Loading is very useful in wireless because wireless channels are time varying and very dynamic. By adapting the transmission system to match the current channel parameter, it tries to negate this dynamism.

There are some additional costs of implementation of this system but compared to reliability that it can provide, it is an acceptable overhead.

Adaptive Bit Loading provides Low BER and High Data Rates while at the same time reduces the energy requirement. Hence, it is very useful for wireless and mobile networks, especially Ad- Hoc ones, where the channel is very dynamic and varying.

REFERENCES

[1]“A Practical Discrete Multitone Transceiver Loading Algorithm for Data Transmission Over Spectrally Shaped Channels,"
PS Chow, JM Cioffi, JAC Bingham - IEEE Transactions on Communications, 1995
[2]“Discrete Bit Loading for Multicarrier Modulation Systems,"
J Campello - Information Theory IEEE Proceedings, 1998
“Third-Generation Systems and Intellignet Wireless Networking - Smart Antennas and Adaptive Modulation”
JS Blogh, L Hanzo
“OFDM Wireless LANs: A Theoretical and Practical Guide”
J Heiskala, J Terry