



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

**“MODELAMIENTO Y SIMULACION DE ALGORITMOS ADAPTIVOS DE
ALOCACION DE BITS PARA SISTEMAS MULTIUSER”**

TESINA DE SEMINARIO

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por

DIEGO FRANCISCO AGUIRRE GONZALEZ

GARY EDGAR SERRANO CASTRO

Guayaquil – Ecuador

2011

AGRADECIMIENTO

Le agradecemos Al Ingeniero Hernán Córdova, profesor del seminario de graduación, por su ayuda y su invaluable apoyo académico durante el desarrollo del presente proyecto.

Diego Francisco Aguirre

Gary Edgar Serrano

DEDICATORIA

Dedico este proyecto a Dios por haberme guiado en el transcurso de toda mi vida, y por supuesto a mi amada madre, hermana y a la memoria de mi padre quienes han sido los protagonistas principales de mi éxito como profesional y quienes me han brindado su amor y apoyo incondicional en todo momento.

Gary Edgar Serrano

DEDICATORIA

Todo esto se dio gracias a la bendición de Dios y es por eso que dedico este proyecto a mi padre Saúl y mi madre Gladys quienes con mucha humildad y arduo trabajo me han guiado, apoyado y aconsejado en todo momento y sobre todo han confiado en mí.

A mis hermanos Javier y Leonardo que me han aconsejado siempre para bien. Y en especial a mi hermano Iván que desde el cielo siempre nos ha cuidado, protegido y guiado y me dejó un lindo recuerdo en vida por lo que siempre lo llevaré en mi mente.

TRIBUNAL DE SUSTENTACIÓN

Ing. Juan Carlos Avilés

Sub-Decano de la FIEC

Ing. Hernán Córdova

Profesor de seminario

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina de Seminario, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

Diego Francisco Aguirre G.

Gary Edgar Serrano C.

RESUMEN

Este proyecto se enfoca en el análisis y desarrollo de un algoritmo que nos permitirá entender con mayor facilidad lo que es la alocaión adaptiva de bit en un sistema multiuser OFDM.

Usaremos la tecnología OFDM debido a que esta se encargara de dividir el ancho de banda de nuestro sistema en múltiples subportadoras, las cuales podrán ser alocadas con mayor facilidad a los diferentes usuarios.

La solución que hemos planteado para el desarrollo de este proyecto es de tipo ***“Multi-user discrete bit-loading”***, la cual podremos ver desarrollada más adelante {3.3}, pero que con una sencilla explicación podemos decir que esta solución trata de encontrar la forma más optima de transmitir una cierta cantidad de bits, con el único propósito de minimizar la potencia total de transmisión.

El sistema a utilizar para el análisis de nuestra solución es el sistema VDSL2, debido a la alta velocidad de transmisión de datos en distancias cortas, en el cual hemos considerado el efecto del ruido blanco aditivo gaussiano y así mismo el efecto del FEXT que es la interferencia generada por el transmisor

a lado opuesto del receptor, como único disturbio en nuestra transmisión de datos.

ÍNDICE GENERAL

	Pág.
AGRADECIMIENTO	II
DEDICATORIA	III
DEDICATORIA	IV
TRIBUNAL DE GRADUACIÓN	V
DECLARACIÓN EXPRESA	VI
RESUMEN	VII
ÍNDICE GENERAL	IX
ABREVIATURAS	XI
INTRODUCCIÓN	XII

CAPITULO 1: COMPRESION DEL PROBLEMA

1.1 Introducción a la comprensión del problema.....	1
1.2 Definición de Alocaçión Adaptiva de bits.....	2

CAPITULO 2: DEFINICION DE PROBEMA

2.1 Cuál es el problema.....	4
2.2 Formulaci3n del problema	5
2.2.1. M3nima potencia para la asignaci3n de bits.....	10

CAPITULO 3: COMPRESION DE LA SOLUCION

3.1 Soluci3n General.....	12
---------------------------	----

3.2	Solución Principal.....	12
3.3	Desarrollo de la Solución Principal.....	13
3.3.1.	Declaración de Variables.....	14
3.3.2.	Inicialización.....	14
3.3.3.	Paso 1.....	14
3.3.4.	Paso 2.....	15
3.3.5.	Paso 3.....	16
3.3.6.	Paso 4.....	17
3.3.7.	Paso 5.....	17
3.3.8.	Paso 6.....	18
3.3.9.	Paso 7.....	18
3.3.10.	Paso 8.....	19
3.3.11.	Paso 9.....	19

CAPITULO 4: MODELAMIENTO Y SIMULACION

4.1	Escenario.....	20
4.2	Modelamiento.....	22
4.3	Simulación	23
4.3.1.	Observación 1.....	25
4.3.2.	Observación 2.....	27
4.3.3.	Observación 3.....	29
4.3.4.	Observación 4	30

CONCLUSIONES Y RECOMENDACIONES.....31

ANEXOS.....35

BIBLIOGRAFÍA.....57

ABREVIATURAS

<u>Abreviatura</u>	<u>Significado</u>
DSL	Línea Suscriptor Digital
VDSL	Very high bit-rate Digital Subscriber
AWGN	Additive White Gaussian Noise
FDM	Frequency Division Multiplexing
OFDM	Orthogonal frequency-division multiplexing
FEXT	Far End Crosstalk
NEXT	Near End Crosstalk
ISI	Inter Symbol Interference
SNR	Signal-to-Noise Ratio
DMT	multi-tono discreto
MA	Margin Adaptive
LAN	Local Area Network
PSD	Power Spectral Density

INTRODUCCIÓN

VDSL son las siglas de Very high bit-rate Digital Subscriber Line, es decir de muy alta tasa de transferencia. Se trata de una tecnología de acceso a internet de Banda Ancha, perteneciente a la familia de tecnologías xDSL la cual transmite datos a alta velocidad sobre distancias cortas utilizando pares trenzados de líneas de cobre con un rango de velocidad que depende de la longitud de la línea. La máxima velocidad de transmisión de la red al cliente está entre 51 y 55 Mbps sobre líneas de 300 metros de longitud. VDSL puede operar tanto en modo simétrico como en el asimétrico.

Como dijimos anteriormente VDSL transmite datos a alta velocidades sobre distancias cortas, es decir que la señal se atenúa a medida que recorre el cable telefónico. Llega más débil cuanto mayor es la longitud del par de cobre. Con una señal más débil, el ruido de fondo tendrá más protagonismo y reducirá el margen útil para transportar información, llamado relación señal-ruido, o SNR.

Para la transmisión de datos en Nuestro sistema VDSL usamos la Multiplexación por División de Frecuencias Ortogonales, (OFDM).

OFDM Es una tecnología de modulación multiportadora que transmite múltiples señales simultáneamente sobre un solo medio de transmisión, como lo son un cable o aire. Estos símbolos son transmitidos en paralelo sobre múltiples subportadoras que comparten el ancho de banda del sistema usando alguna forma de multiplexación por división de frecuencia (FDM), para una mejor explicación sobre OFDM podemos decir que esta se encarga de dividir el ancho de banda total del sistema de transmisión en un gran número de subcanales ortogonales con menor ancho de banda.

Se proponen métodos de asignación adaptativa de bits para poder aprovechar mejor la capacidad o recursos de un sistema y lograr un mejor desempeño, esto quiere decir que el esquema se va adaptando dinámicamente a las condiciones del canal.

CAPITULO 1

COMPRESION DEL PROBLEMA

1.1 INTRODUCCION A LA COMPRESION DEL PROBLEMA

La asignación adaptativa de bits consiste en hallar la forma óptima para la transmisión de datos, y esto lo podemos lograr distribuyendo de la mejor manera los recursos de un sistema de transmisión como lo son las subportadoras o a su vez a potencia de esta, y como sabemos esto va depender única y exclusivamente de las especificaciones o de los requerimientos del sistema.

La asignación adaptativa de bits es una parte importante dentro de OFDM ya que ésta tratará de explotar la información del canal para así asignar una mayor cantidad de bits en el subcanal y usuario con el menor costo posible con la finalidad de obtener la menor potencia de transmisión.

La diferencia entre ADSL y las otras Tecnologías DSL es que la velocidad de bajada (Downstream) y la de subida (Upstream) no son simétricas, es decir, ADSL trabaja en modo Asimétrico donde el Downstream es mayor que el Upstream.

1.2 DEFINICIONES DE ALOCACIÓN ADAPTIVA DE BITS

Como ya vimos anteriormente no existe una definición específica de lo que es alocación adaptiva de bits pero podemos concluir de que se trata de hallar un esquema o algoritmos que nos permita utilizar la mínima potencia del sistema con la única finalidad de que la tasa de transmisión de bits no decaiga de un valor establecido.

Otra forma de definirla sería hallar un esquema que permita distribuir las subportadoras a cada uno de los usuarios de tal forma que se pueda maximizar la tasa de transferencia de bits pero sin exceder un valor límite en la potencia total de transmisión.

Con esto tenemos dos conceptos básicos de lo que es alocación adaptiva de bits y a la vez una gran disyuntiva para obtener un desempeño óptimo en la transmisión de datos, es decir por una parte tenemos que tratar de minimizar la potencia estableciendo una cierta cantidad fija de bits para transmitir y por otra parte maximizar la tasa de transferencia de bits manteniendo fija la potencia total del sistema.

Para lograr esto necesitamos un algoritmo que nos permita realizar la asignación de subportadoras y bits a los diferentes usuarios con el único propósito de minimizar la potencia total del sistema o a su vez aumentar la tasa de transmisión de datos, debido a que si esta

asignación la hacemos de una manera aleatoria o sin ningún tipo de fundamentos técnicos es muy probable de que muchos bits se pierdan o que la potencia total aumente.

CAPITULO # 2

DEFINICION DEL PROBLEMA

2.1 CUAL ES EL PROBLEMA:

OFDM es la división y distribución de ancho de banda de un sistema, es decir manipula el ancho de banda dividiendo los datos en canales paralelos a lo largo de las subportadoras, que son espaciados (ortogonalidad) entre sí por llamadas frecuencias. La ortogonalidad de los canales proporciona un mejor y más efectivo uso del espectro con la ventaja de que al dividir en subportadoras se puede modular en forma independiente y poder adaptarse a las condiciones del canal.

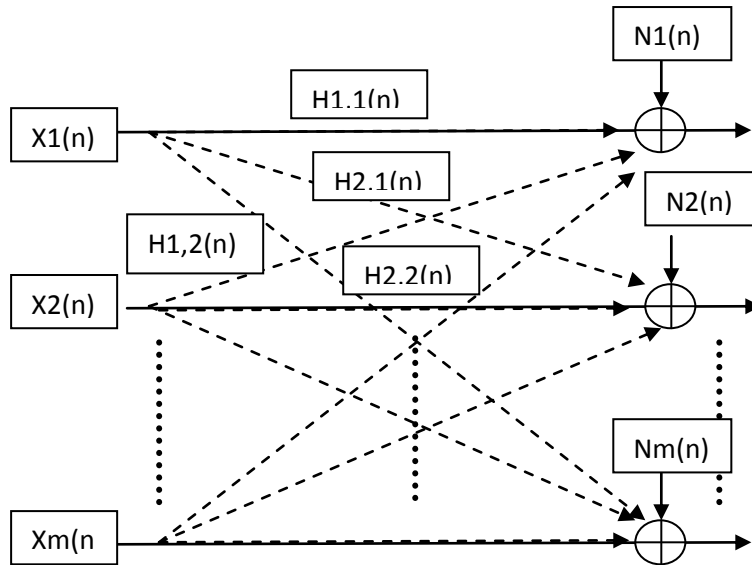
En la transmisión de datos, se pueden presentar un sin número de problemas, entre ellos y uno muy sensible es la potencia. El aumento o disminución de la misma es vital para mantener dentro de los parámetros aceptables una transmisión, eh aquí donde aparece la alocaación adaptiva de bits, que en este caso puntual nos ayudará a reducir la potencia de forma dinámica manteniendo la transmisión y

ajustando convenientemente los parámetros a utilizar basándonos en una estimación fiable del canal.

2.2 FORMULACION DEL PROBLEMA:

Cuando se realiza transmisión de datos en sistemas DSL, cada usuario produce campos electromagnéticos los cuales afectan a los demás, ya que debido a estos campos se generan señales **crosstalk (Diafonía)** entre ellos [2]. Existen dos tipos de diafonía, **NEXT (near-end crosstalk) o Paradiafonía** que es aquella interferencia generada por el transmisor que se encuentra en el mismo lado del receptor, **FEXT (far-end crosstalk) o telediafonía** en cambio es la interferencia generada por el transmisor a lado opuesto del receptor, pero en el desarrollo del proyecto usaremos un esquema de duplexación de división de frecuencia para que el NEXT pase desapercibido.

Debido a que aún existe FEXT, el canal de M usuarios será un canal con interferencia ISI (Inter-symbol Interference) [1], usando técnica DMT, el canal principal puede ser modulado en subcanales libres de interferenci ISI. En la figura (1) se muestra como se divide el canal principal.



La relación señal a ruido (SINR) para el usuario i en la subcanal n está dada por:

$$S = \frac{H_{i,i}^{(2)} P_i(n)}{N_i(n) + \sum_{j=1, j \neq i}^M H_{i,j}^2(n) P_j(n)} \quad (1)$$

Donde $H_{i,i}^{(2)}$ es la ganancia del canal, $P_i(n)$ es la señal de potencia y $N_i(n)$ la señal de ruido del usuario i en la subportadora n , entonces la SINR necesaria para que pueda transmitir el usuario i $b_i(n)$ bits en la subportadora n es:

$$\gamma_i(b_i(n)) = f_i(b_i(n), \overline{P_{e,i}}) \quad (2)$$

Donde $\overline{P_{e,i}}$ es la probabilidad media de error de símbolo del usuario i y la función f_i depende del esquema de modulación y codificación que se vaya a utilizar el cual se desarrollará mas adelante. De las ecuaciones (1) y (2) podemos decir que la potencia $P_i(n)$ del usuario i en la subportadora n para una distribución $b(n) = [b_1(n), b_2(n), \dots, b_M(n)]^T$ debe de satisfacer:

$$\frac{H_{i,i}^{(2)} P_i(n)}{N_i(n) + \sum_{j=1, j \neq i}^M H_{i,j}^2(n) P_j(n)} \geq f_i(b_i(n), \overline{P_e}) \text{ para } i = 1 \dots M \quad (3)$$

Un problema interesante en este proyecto es minimizar la potencia necesaria con el fin de encontrar una tasa de transmisión dada, el cual se conoce como “*Margin-adaptive loading*” para un único usuario [2], [3], pero como se busca utilizar multi-usuarios, el objetivo será minimizar el total de la potencia utilizada para satisfacer dicha tasa de transmisión [1]. Esta minimización puede ser formulada de la siguiente manera:

$$\text{minimización} \quad \sum_{i=1}^M \sum_{n=1}^N P_i(n) \quad (4)$$

$$\text{sujeto a } \sum_{i=1}^M \sum_{n=1}^N b_i(n) \geq B$$

$$b_i(n) \in Z_0^\infty$$

Donde B es el objetivo de tasa de transmisión de bits, Z_0^∞ es el conjunto de enteros no negativos. El problema de la minimización de la potencia total es modificado para tener en cuenta estas restricciones adicionales:

$$\text{minimización } \sum_{i=1}^M \sum_{n=1}^N P_i(n) \quad (5)$$

$$\text{sujeto a } \sum_{i=1}^M \sum_{n=1}^N b_i(n) \geq B$$

$$b_i(n) \in Z_0^{\bar{b}}$$

$$P_i(n) \leq \overline{P(n)}$$

Donde \bar{b} es el límite de bits permitido para cada usuario en cada subportadora y $\overline{P(n)}$ es la potencia límite para evitar que dicha potencia exceda el máximo permitido en cada subportadora.

Para reducir la complejidad del algoritmo óptimo descrito anteriormente, nosotros proponemos un algoritmo subóptimo llamado **“Multi-user discrete bit-loading”**, que es el extendido del algoritmo óptimo de Greedy para un único usuario [4].

El algoritmo de **Greedy** para un solo usuario, asigna 1 bit adicional a **la subportadora** que requiere de menor costo para transmitir, lo cual es la idea principal para desarrollar este algoritmo a diferencia que aquí se asigna un bit al **usuario y subportadora** que asignándole dicho bit se logra minimizar el costo $J(n, i)$ de transmisión. Sin embargo, se debe de tener en cuenta que al momento de asignar el bit no solo el usuario i experimenta un aumento de potencia, debido a la interferencia de los demás usuarios obliga a que cada uno de ellos traten de aumentar su potencia para poder mantener su SINR, por ello el costo $J(n, i)$ es escogido como el mínimo incremento de la potencia total de todos los usuarios. A continuación se presenta una función que nos ayudará a encontrar el costo $J(n, i)$.

$$P_{\Sigma}^*(n, b_n) = \min \sum_{i=1}^M P_i(n, b_n) \quad (6)$$

2.2.1 MINIMA POTENCIA PARA LA ASIGNACION DE BIT

La potencia $P_i(n, b_n)$ debe de satisfacer la ecuación (3), la cual se puede expresar en la siguiente matriz [1].

$$(I - A_n)x_n(b_n) \geq y_n \quad (7)$$

Donde:

$$\{A_n\}_{i,j} = \begin{cases} \frac{\gamma_i(b_i(n))H_{i,j}^2(n)}{H_{i,j}^2(n)} & , \text{para } i \neq j \\ 0 & , \text{para } i = j \end{cases} \quad (8)$$

$$x_n = [P_1(n, b_n) \dots \dots \dots P_M(n, b_n)]^T \quad (9)$$

$$y_n = \left[\frac{\gamma_1(b_1(n))N_1(n)}{H_{1,1}^2(n)} \dots \dots \frac{\gamma_M(b_M(n))N_M(n)}{H_{M,M}^2(n)} \right]^T \quad (10)$$

Donde $\gamma_i(b_i(n))$ se obtiene de la ecuación (2). Como A_n es una matriz irreducible utilizamos los eigen-valores $\lambda(A_n)$ [5], [6], lo cual nos muestra que las siguientes premisas son equivalentes:

- a). Existe una solución para la ecuación (7)
- b). $\lambda(A_n)$ son menores que 1

c). $(I - A_n)^{-1}$ existe y es positivo.

Además, $x_n^*(b_n) = (I - A_n)^{-1}y_n$ es una solución óptima de (7), lo que quiere decir que cualquier $x_n(b_n)$ que satisface la ecuación (7) es mayor o igual $x_n^*(b_n)$. Si algún usuario no transmite bit en una subportadora n la matriz A_n sería irreducible, sin embargo los resultados en [6] puede ser utilizados para una matriz irreducible, lo cual se explicará con mayores detalles en anexos.

CAPITULO # 3

COMPRESION DE LA SOLUCION

3.1 SOLUCION GENERAL

Para poder tener un poco más claro el tema de asignación adaptativa de bits se ha propuesto la siguiente solución:

1- Minimizar la potencia de transmisión

$$\text{minimización} \quad \sum_{i=1}^M \sum_{n=1}^N P_i(n)$$

$$\text{sujeto a} \quad \sum_{i=1}^M \sum_{n=1}^N b_i(n) \geq B$$

$$b_i(n) \in Z_0^{\bar{b}}$$

$$P_i(n) \leq \overline{P(n)}$$

3.2 SOLUCION PRINCIPAL

La asignación adaptativa de bits mediante bit-loading, consiste en distribuir de la mejor manera los recursos de un sistema, en este

caso hará la distribución de las subportadoras y los bits de asignación, de tal manera que se minimice la potencia total de transmisión manteniendo una tasa de bits constante.

Para reducir la complejidad de los algoritmos óptimos de tipo Greedy, nosotros en este proyecto hemos creído conveniente utilizar la solución sub-óptima ***Multi-user discrete bit-loading***, basándonos en que este algoritmo nos permite manipular la asignación de bit por bit a los diferentes usuarios y subportadoras con el mínimo costo, de tal manera que la potencia consumida sea la mínima.

Esta tecnología ofrece tasas de transferencia sensiblemente mayores a la anterior, usando la misma infraestructura con terminales especiales que permitan el nuevo ancho de banda, además este estándar contempla una serie de implementaciones que mejoran la supervisión de la conexión y la calidad de servicio (QoS) de los servicios demandados a través del bucle de abonado.

3.3 DESARROLLO DE LA SOLUCION PRINCIPAL

El algoritmo bit-loading para multi-usuarios nos permite hacer la asignación de bits a los usuarios y subportadoras con los costos mínimos y a continuación realizamos la descripción:

3.3.1 DECLARACION DE VARIABLES:

n = Subportadoras

i = Usuarios

b_n = Vector de distribución de bits.

X_n^* = Vector de Potencia para la solución óptima.

P_M = Señal de potencia de cada usuario.

F = Matriz de Saturación.

$J(n, i)$ = Costo de transmisión del usuario i en la subportadora n .

$P_i(n, e_i)$ = Potencia del usuario i en la subportadora n cuando ya se le asignó 1 bit más.

$\overline{P(n)}$ = Potencia límite permitido.

CM = Matriz de costos mínimo.

$K(n)$ = Matriz de usuarios que tienen el costo mínimo de transmisión.

$\widehat{J(n)}$ = Matriz de valores de cada costo mínimo.

3.3.2 INICIALIZACIÓN:

3.3.3 PASO 1.

$n = 1, \dots, N$; Subportadoras

$i = 1, \dots, M$; *Usuarios*

Realizamos la inicialización del vector bit-rate b_n debido a que en primera instancia no va a ver transmisión de datos:

$b_n = [0 \dots 0]^T$; **T= transpuesta**

Inicializamos la matriz X_n^* que es una matriz compuesta por los vectores que representan la potencia de cada usuario con respecto a cada subportadora:

$$X_n^*(b_n) = \left[\begin{array}{c|c|c} \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix} & P_1 & \dots & \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix} & P_M \end{array} \right]$$

3.3.4 PASO 2.

Inicializamos la matriz F que representa la matriz de saturación, cuyas filas representan las N subportadoras y las columnas los M usuarios, cuando un elemento de esta matriz esta en 0, me indica que ese usuario con dicha subportadora se le puede asignar un bit, caso contrario si este elemento esta en 1 me indica que a ese

usuario con dicha subportadora está saturada debido a que ya ha sido asignado un bit.

$$F = \begin{bmatrix} 1 & \dots & M \\ \vdots & \ddots & \vdots \\ N & \dots & \dots \end{bmatrix}; \text{ toda la matriz inicialmente en 0}$$

3.3.5 PASO 3.

Calculamos el costo mínimo de transmisión de 1 bit.

- a) Calculamos $X_n^*(e_i)$.
- b) Calculamos $J(n, i)$.
- c) Comprobar que $P_i(n, e_i) > \overline{P(n)}$ si cumple, entonces $F(n, i) = 1$.

Primero realizamos el cálculo de la matriz de potencias con respecto a e_i , luego por medio de: $J(n, i) = P_{\Sigma}^*(n + b_n, e_i) - P_{\Sigma}^*(n, b_n)$, calculamos el costo mínimo real de transmisión de 1 bit, debido a que restamos la potencia inicial obtenida de $X_n^*(b_n)$ con la potencia necesaria para transmitir cuando ya se hizo la asignación de 1 bit, gracias a este cálculo podemos obtener una matriz con los valores

de los costos mínimos de transmisión de cada usuario con relación a cada subportadora.

$$CM = \begin{bmatrix} 1 & \dots & M \\ \vdots & \ddots & \vdots \\ N & \dots & \end{bmatrix}; \text{Matriz de costos mínimos}$$

3.3.6 PASO 4.

Encontramos un usuario $K(n)$, que tiene el costo mínimo para cada subportadora.

$$K(n) = \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix}; \text{ donde cada elemento del vector } K(n), \text{ me representa el usuario con el costo}$$

mínimo de la matriz CM

$$\widetilde{J(n)} = \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix} \text{ donde cada elemento de este vector, me representa el valor del costo}$$

mínimo de la matriz CM

3.3.7 PASO 5.

Encuentro la subportadora m , que tenga el mínimo costo del vector

$$\widetilde{J(n)}$$

3.3.8 PASO 6.

Como ya encontramos la subportadora y el usuario con el mínimo costo, procedemos a asignar un bit a dicha subportadora m del usuario $K(m)$.

$$b_m = b_m + e_k(m)$$

3.3.9 PASO 7.

Actualizamos los valores con el nuevo bit asignado y realizamos los nuevos cálculos:

- a) Calculamos $X_m^*(b_m + e_i)$.
- b) Calculamos $J(m, i)$
- c) Comprobamos en la matriz de saturación, que cada vez que se asigne un bit a una subportadora m , este elemento de la matriz se sature, es decir $F(m, i) = 1$.
- d) Comprobar que $\max_k \{ P_k (n, b_m + e_i) \} > \overline{P(n)}$, si cumple, entonces $F(m, i) = 1$.

3.3.10 PASO 8.

Una vez actualizados todos los datos, volvemos a encontrar el usuario con el mínimo costo y la subportadora de dicho usuario para seguir asignando los bits como en el PASO 4.

3.3.11 PASO 9.

Comprobamos que la sumatoria total de los bits asignados no sea mayor a la tasa binaria total de transmisión. Si esto se cumple, regresamos al paso 5 y así sucesivamente hasta asignar todos los bits de la tasa total de transmisión.

CAPITULO # 4

MODELAMIENTO Y SIMULACION

4.1 ESCENARIO:

Una transmisión puede presentar un gran número de escenarios dependiendo de condiciones climáticas, interferencia, etc. Por ello hemos planteado un escenario en particular para fines didácticos y explicativos teniendo en cuenta que la importancia de este proyecto radica en el desarrollo de la solución del algoritmo, por lo tanto se ha planteado el siguiente escenario con respecto al perfil 12^a correspondiente a la tabla [8]:

Profile	Max. DS Power	Max. DS Freq.	Max. US Power	Max. US Freq.	U0 for Long Range	Typical Application
8c	11.5 dbm	8.5 MHz	14.5 dbm	5.2 MHz	Required	Cabinet (RT)
8d	14.5 dbm	8.5 MHz	14.5 dbm	5.2 MHz	Required	Cabinet (RT) Exchange (CO)
8a	17.5 dbm	8.5 MHz	14.5 dbm	5.2 MHz	Required	Exchange (CO)
8b	20.5 dbm	8.5 MHz	14.5 dbm	5.2 MHz	Required	Exchange (CO)
12a	14.5 dbm	8.5 MHz	14.5 dbm	12.0 MHz	Required	Cabinet (RT) Exchange (CO)
12b	14.5 dbm	8.5 MHz	14.5 dbm	12.0 MHz	Not required	Cabinet (RT) Exchange (CO)
17a	14.5 dbm	Not specified	14.5 dbm	Not specified	Not required	Cabinet (RT) MxU
30a	14.5 dbm	Not specified	14.5 dbm	Not specified	Not required	MxU

Tabla 1. VDSL Profiles with Frequencies correspondig to Band Plan 998

- El sistema a utilizar es VDSL2 (simulado en MATLAB) para la alocaación adaptiva de bits debido a su alta disponibilidad, ya que se puede acceder por medio de las líneas telefónicas de cobre.
- Se utilizará un $f_{max} = 12$ MHz de upstream que es el correspondiente a VDSL2, debido a que estamos trabajando en upstream no usamos la banda completa es decir nos enfocaremos en el rango de 4.5 y 8 como primera banda y 8.5 y 12 MHz como la segunda banda.
- Vamos a realizar el cálculo de la capacidad de cada banda por separado y finalmente la sumamos para encontrar la capacidad total del sistema.
- Pot max por usuario en upstream: 14.5 dBm.

- Distancia de transmisión = 0.2Km, 0.5Km, 0.7Km
- Número de usuarios 3.
- Nos hemos enfocado en el lado "upstream" con SNR-gap 12dB con 3.8 dB de ganancia de código.
- Se tomo 6dB de margen de ruido y una probabilidad de error de símbolo de 10^{-7}
- Se utilizo la formula FSAN (crosstalk summing) con el fin de tratar de evitar los peores escenarios posibles,

4.2 MODELAMIENTO

Uno de los puntos importantes en este proyecto es conocer el comportamiento del medio alámbrico a utilizar, por ello es imprescindible analizar la ganancia del canal, lo cual lograremos mediante la expresión:

$$|H|_{DB} = -k_r * d * \sqrt{w}$$

Donde:

Kr = constante del cable.

d = distancia (Km).

w = velocidad angular (rad/seg).

En este proyecto únicamente vamos a tomar en cuenta el FEXT como interferencia externa, para ello tenemos la siguiente función de transferencia:

$$|H_{FEXT}(f, L)| = k_{FEXT} * N^{0.6} * L * f^2 * |H(L, f)|^2$$

Donde:

$$k_{fext} = 8 \times 10^{-20}$$

N = número de disturbers

L = distancia (Km)

f = frecuencia

H = la ganancia del canal

4.4 SIMULACION

Grafica de la capacidad de cada usuario, usuario 1 a 200mts, usuario 2 a 500 mts y usuario 3 a 700mts.

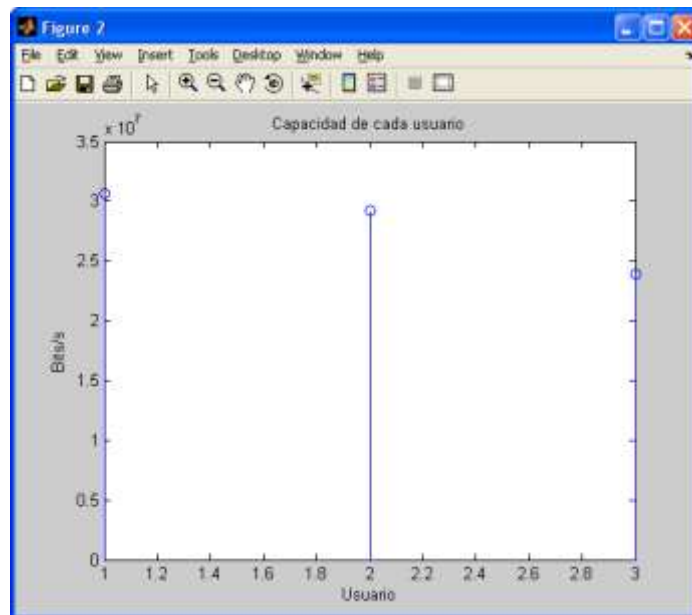


Fig. 4.1. Capacidad de los 3 usuarios a diferentes distancias.

Grafica de la capacidad de cada usuario sin el algoritmo, usuario 1 a 200mts, usuario 2 a 500 mts y usuario 3 a 7000mts.

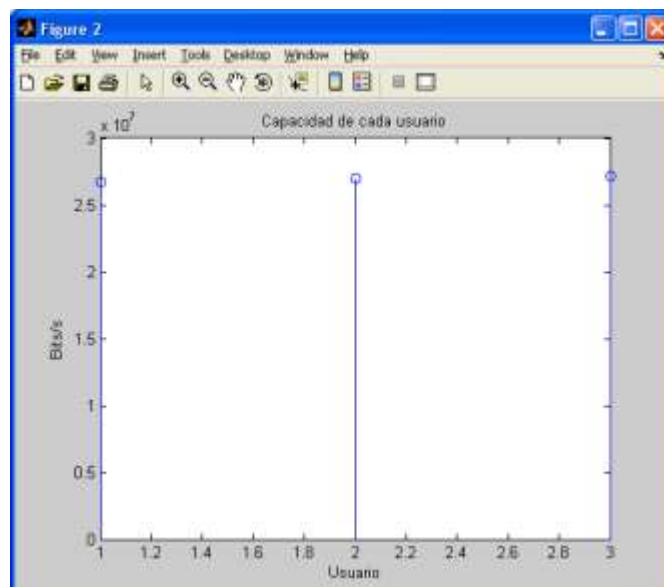


Fig. 4.2. Capacidad de los 3 usuarios a diferentes distancias sin algoritmo.

4.3.1 OBSERVACION 1:

Las figuras 4.1 y 4.2 muestran el esquema de asignación para 3 usuarios y la transmisión de datos sin usar el algoritmo respectivamente a diferentes distancias. Observando la variación de dicha capacidad a medida que aumenta la distancia.

Graficas de la distribución de bits (10000) para cada subportadora en los 3 usuarios.

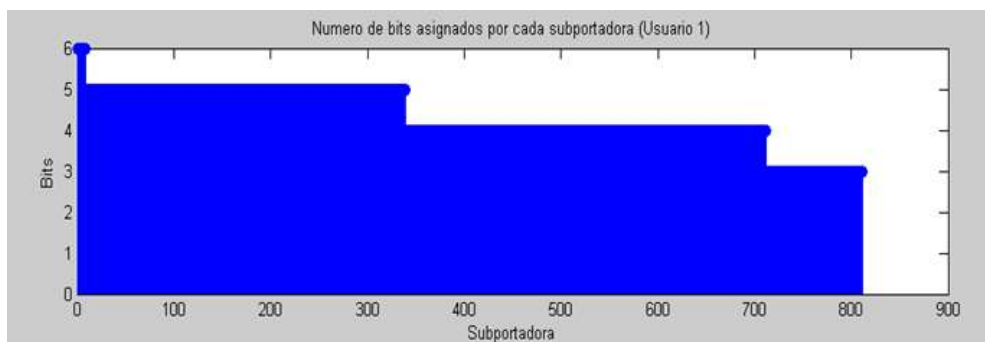


Fig. 4.3. Distribución de bits para el usuario 1 a 0.2km.

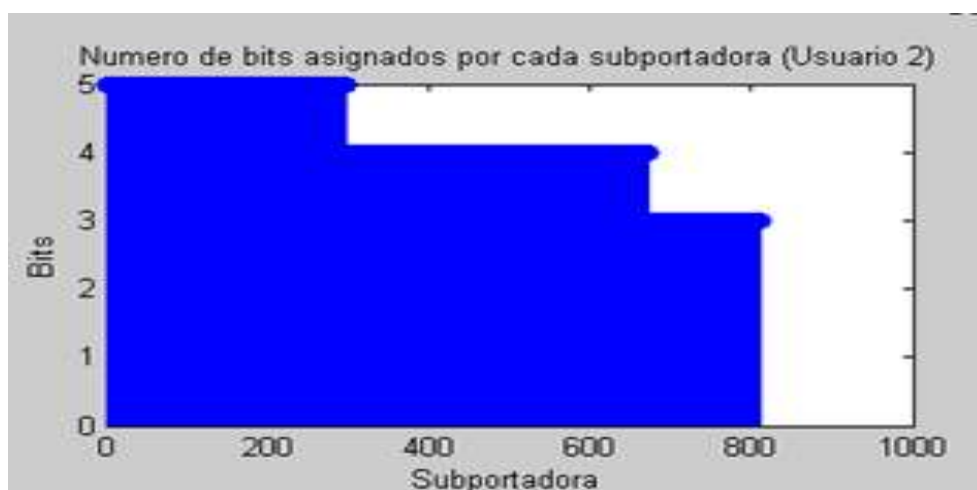


Fig. 4.4. Distribución de bits para el usuario 2 a 0.5km.

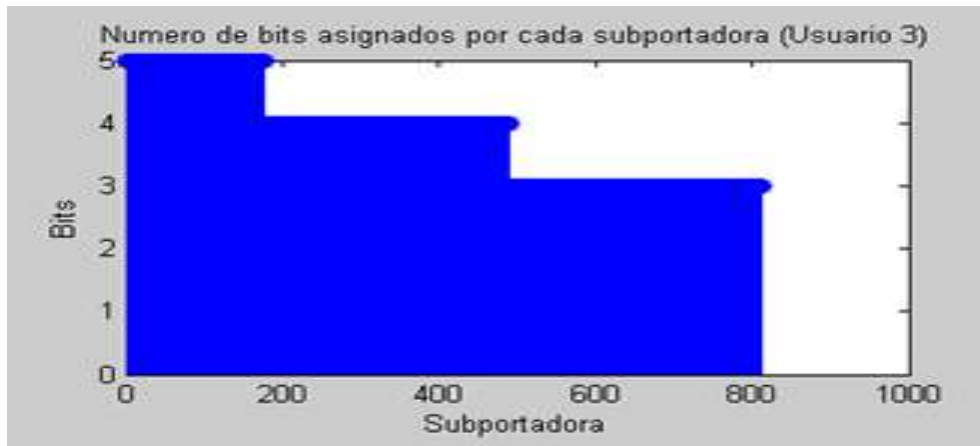


Fig. 4.5. Distribución de bits para el usuario 3 a 0.7km.

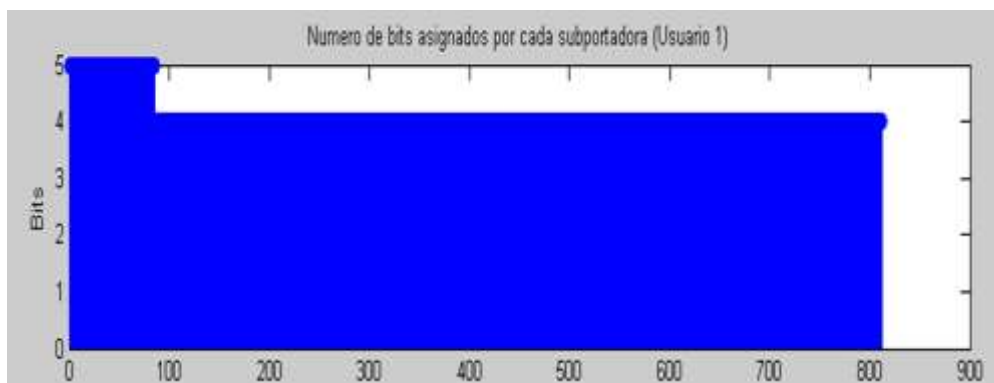


Fig. 4.6. Distribución de bits para el usuario 1 a 0.2km sin algoritmo.

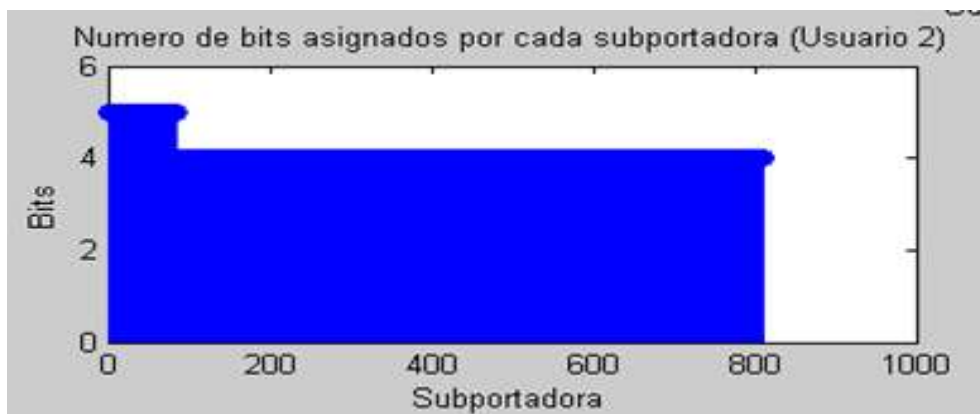


Fig. 4.7. Distribución de bits para el usuario 2 a 0.5km sin algoritmo.



Fig. 4.8. Distribución de bits para el usuario 3 a 0.7km sin algoritmo.

4.3.2 OBSERVACION 2:

Las figuras 4.3, 4.4 y 4.5, nos muestra la distribución de bits con respecto a las subportadoras en cada uno de los 3 usuarios a diferentes distancias de igual manera las figuras 4.6, 4.7 y 4.8 nos muestran la misma distribución pero esta vez sin incluir el algoritmo.

A continuación se muestra las gráficas de la distribución de bits para el usuario 1 a 200mts, usuario 2 a 500 y usuario 3 a 700mts.

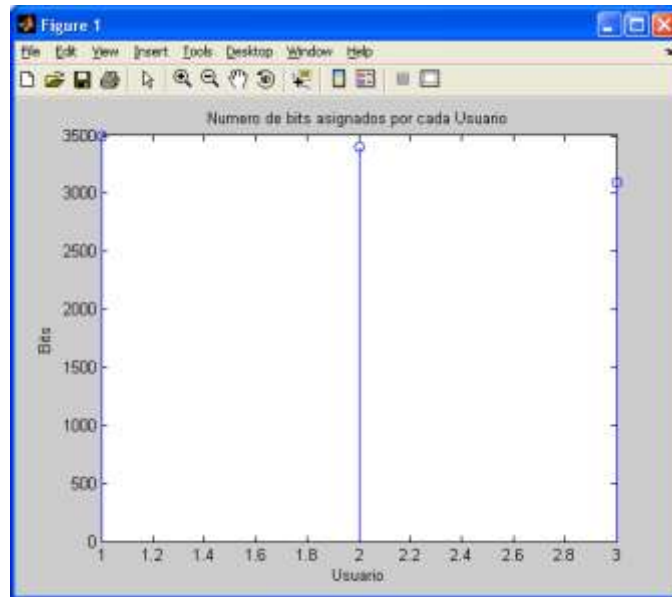


Fig. 4.9. Bits consumidos por cada uno de los 3 usuarios a diferentes distancias

A continuación se muestra las gráficas de la distribución de bits para el usuario 1 a 200mts, usuario 2 a 500 y usuario 3 a 700mts sin utilizar el algoritmo.

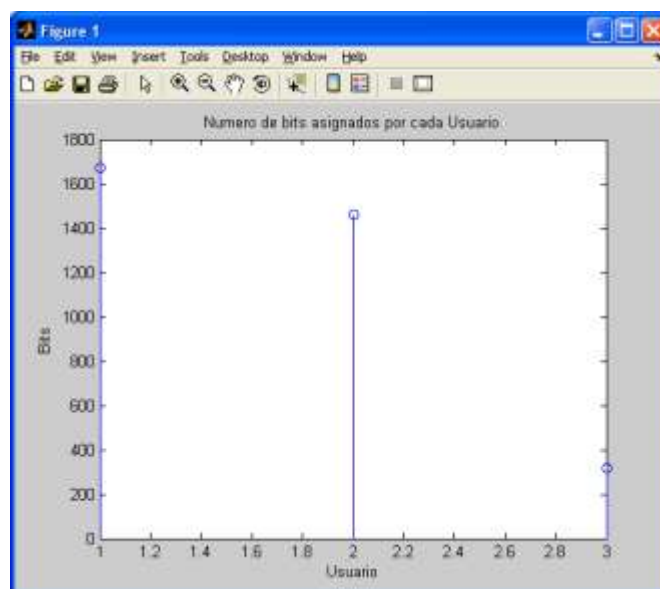


Fig. 4.10. Bits consumidos por cada uno de los 3 usuarios a diferentes distancias sin algoritmo

4.3.3 OBSERVACION 3:

Las figuras 4.9, 4.10 muestran el esquema de asignación para 3 usuarios con y sin el algoritmo respectivamente a distancias diferentes. Observamos la variación del número de bits asignados a cada usuario a medida que aumente la distancia.

A continuación se muestra las gráficas de la potencia total por usuario para el usuario 1 a 200mts, usuario 2 a 500 y usuario 3 a 600mts.

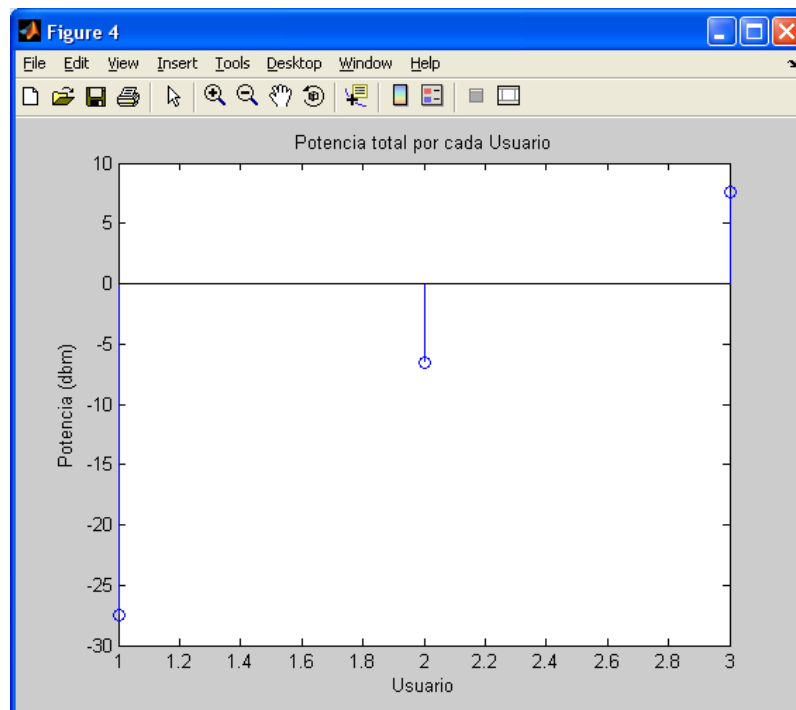


Fig. 4.11. Potencia consumida por cada uno de los 3 usuarios a diferentes distancias.

A continuación se muestra las gráficas de la potencia por usuario para el usuario 1 a 200mts, usuario 2 a 500 y usuario 3 a 700mts.

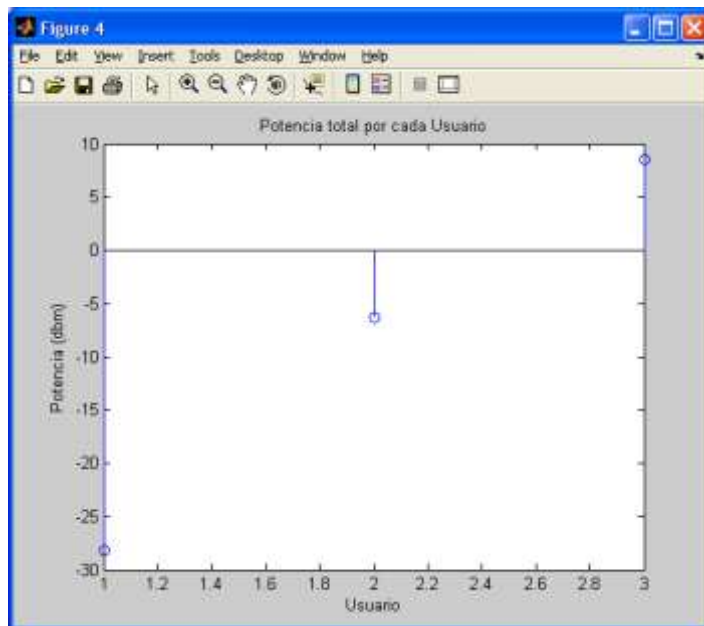


Fig. 4.12. Potencia consumida por cada uno de los 3 usuarios a diferentes distancias.

4.3.4 OBSERVACION 4:

Las figuras 4.11, 4.12 muestran el esquema de asignación para 3 usuarios con y sin algoritmo respectivamente a distancias diferentes. En estas gráficas nos podemos dar cuenta que la potencia total consumida por usuario se mantiene constante ya que transmitimos la misma cantidad de bits.

CONCLUSIONES

Y

RECOMENDACIONES

Como conclusiones tenemos:

- 1) Uno de los puntos importantes de tener en cuenta en este proyecto es que ha sido realizado basado en el desarrollo del paper "Multi-User Discrete Bit-Loading for DMT-based DSL System", comprendiendo cada uno de los pasos y desarrollos para poder dar una explicación detallada del funcionamiento del algoritmo.**
- 2) Como hemos comprobado en el desarrollo de este proyecto la alocaación adaptiva de bit no tiene un concepto predefinido ya que siempre dependerá de los parámetros que se quiera y se puedan manipular en un sistema, con el fin de optimizar la transmisión.**
- 3) El esquema de este algoritmo nos permite decidir a qué usuario y a qué subportadora le vamos a asignar un bit, esto dependiendo del costo mínimo que se requiera para transmitir dicho bit.**
- 4) La potencia teórica máxima de transmisión en un sistema VDSL2 por usuario en upstream es de 14.5 Dbm, pero según nuestras simulaciones hemos obtenidos valores menores al mencionado,**

- 5) esto debido a que en este algoritmo se trata de optimizar la potencia de transmisión.

- 6) Hemos decidido hacer el análisis con algoritmo y sin el mismo, esto ya que queremos demostrar la diferencia que se presenta cuando actúa el algoritmo y los beneficios que tenemos, lo cual lo demostramos al observar la graficas de la capacidad, ya que observamos que la capacidad cuando se usa el algoritmo es mayor que la capacidad cuando no se lo utiliza, demostrando así que el beneficio principal es el aumento de capacidad tomando en cuenta que trabajamos con la mínima potencia por usuario.

- 7) Se puede observar basándonos en los gráficos obtenidos que la potencia se mantiene constante ya sea con o sin algoritmo debido a que se transmiten igual cantidad de bits pero diferenciando en que cuando no se aplica el algoritmo la distribución de bits es mucho menor lo cual se ve reflejado en la capacidad como se describió anteriormente.

- 8) Gráficamente se pudo comprobar que efectivamente el número de bits transmitidos por cada suportadora no sobrepasa el valor máximo total que le as

9) Ignamos como *"bit cap"* realizando una distribución por usuario de mayor a menor (cuando se aplica el algoritmo) debido a que los bits se van asignando dependiendo del costo para transmitir.

10) De igual manera se observa que la transmisión de bits por suportadora es de mayor a menor basándonos en la conclusión anteriormente descrita.

Como recomendaciones tenemos:

1) Se recomienda antes de realizar la simulación tener en claro el escenario que se va a utilizar, tomando en cuenta el canal de transmisión para que no haya equivocaciones en los parámetros.

2) Se recomienda para un futuro desarrollo de este proyecto, se lo realice en un medio inalámbrico tomando en cuenta los parámetros y variaciones del sistema, con especial cuidado a la interferencia que se presenta en este medio de transmisión.

ANEXOS

A continuación se realizará una pequeña demostración de las ecuaciones para calcular la mínima potencia necesaria para transmitir un bit extra [7]:

De la ecuación (2) tenemos que:

$$\gamma_i(b_i(n)) = f_i(b_i(n), \overline{P_{e,i}})$$

Y de la ecuación (3) observamos la igualdad,

$$\frac{H_{i,i}^{(2)} P_i(n)}{N_i(n) + \sum_{j=1, j \neq i}^M H_{i,j}^2(n) P_j(n)} \geq f_i(b_i(n), \overline{P_e})$$

Por ende se puede decir que:

$$\gamma_i(b_i(n)) = \frac{H_{i,i}^2(n) P_i(n, b_n)}{N_i(n) + \sum_{j=1, j \neq i}^M H_{i,j}^2(n) P_j(n, b_n)} \quad (11)$$

Multiplico la ecuación para $\frac{1}{H_{i,i}^2}$ por conveniencia.

$$\gamma_i(b_i(n)) = \frac{H_{i,i}^2(n)P_i(n, b_n)}{N_i(n) + \sum_{j=1, j \neq i}^M H_{i,j}^2(n)P_j(n, b_n)} * \frac{1}{H_{i,i}^2(n)} * \frac{1}{H_{i,i}^2(n)} \quad (12)$$

$$\gamma_i(b_i(n)) = \frac{P_i(n, b_n)}{\frac{N_i(n)}{H_{i,i}^2(n)} + \frac{\sum_{j=1, j \neq i}^M H_{i,j}^2(n)P_j(n, b_n)}{H_{i,i}^2(n)}} \quad (13)$$

Pasamos el denominador del lado derecho de la ecuación al lado izquierdo,

$$\gamma_i(b_i(n)) * \left(\frac{N_i(n)}{H_{i,i}^2(n)} + \frac{\sum_{j=1, j \neq i}^M H_{i,j}^2(n)P_j(n, b_n)}{H_{i,i}^2(n)} \right) = P_i(n, b_n) \quad (14)$$

$$\frac{\gamma_i(b_i(n))N_i(n)}{H_{i,i}^2(n)} + \gamma_i(b_i(n)) \frac{\sum_{j=1, j \neq i}^M H_{i,j}^2(n)P_j(n, b_n)}{H_{i,i}^2(n)} = P_i(n, b_n) \quad (15)$$

$$\gamma_i(b_i(n)) \frac{\sum_{j=1, j \neq i}^M H_{i,j}^2(n)P_j(n, b_n)}{H_{i,i}^2(n)} - P_i(n, b_n) = -\frac{\gamma_i(b_i(n))N_i(n)}{H_{i,i}^2(n)} \quad (16)$$

Finalmente multiplicamos la ecuación por -1 para acomodarla:

$$P_i(n, b_n) - \gamma_i(b_i(n)) \sum_{j=1, j \neq i}^M P_j(n, b_n) \frac{H_{i,j}^2(n)}{H_{i,i}^2(n)} = \gamma_i(b_i(n)) \frac{N_i(n)}{H_{i,i}^2(n)} \quad (17)$$

La ecuación anterior (17) se puede expresar en matriz de la forma:

$$(I - A_n)x_n(b_n) \geq y_n$$

Donde,

$$(I - A_n) = \begin{bmatrix} 1 & -\gamma_i(b_1(n)) \frac{H_{i,1}^2(n)}{H_{11}^2(n)} & -\gamma_i(b_1(n)) \frac{H_{M,1}^2(n)}{H_{11}^2(n)} \\ -\gamma_i(b_i(n)) \frac{H_{1i}^2(n)}{H_{ii}^2(n)} & 1 & -\gamma_i(b_i(n)) \frac{H_{M,i}^2(n)}{H_{ii}^2(n)} \\ -\gamma_i(b_M(n)) \frac{H_{1M}^2(n)}{H_{MM}^2(n)} & -\gamma_i(b_M(n)) \frac{H_{iM}^2(n)}{H_{MM}^2(n)} & 1 \end{bmatrix}$$

$$x_n(b_n) = [P_1(n, b_n) \dots P_i(n, b_n) \dots P_M(n, b_n)]^T$$

$$y_n = \left[\gamma_i(b_1(n)) \frac{N_1(n)}{H_{11}^2(n)} \dots \gamma_i(b_i(n)) \frac{N_i(n)}{H_{ii}^2(n)} \dots \gamma_i(b_M(n)) \frac{N_M(n)}{H_{MM}^2(n)} \right]^T$$

Resolviendo el sistema anterior se obtiene el vector de potencia para transmitir un bit mas $b_i(n)$ en la subportadora n para todos los usuarios M . Algo que hay que tomar en cuenta es que el vector $P_n(n, b_n)$ puede contener elementos negativ

os, lo cual me indica que ese usuario no puede transmitir el bit que se le asigne en la subportadora n .

Suponga que A_n es cualquier matriz no-negativo, mediante la misma aplicación de la permutación de ambas filas y columnas de A_n , se puede escribir esta matriz como una matriz superior triangular [5].

$$PA_nP^{-1} = C = \begin{bmatrix} C_1 & D_{1,2} & \dots & D_{1,k} \\ 0 & C_2 & \dots & D_{2,k} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & C_k \end{bmatrix} \quad (18)$$

Donde O es una matriz cuyos elementos son todos 0, C_k una matriz cuadrada irreducible y P es una matriz de permutación. Reemplazando tenemos:

$$(I - A_n)x_n(b_n) \geq y_n$$

$$\Leftrightarrow P(I - A_n)P^{-1}Px_n \geq Py_n$$

$$\Leftrightarrow (I - PA_nP^{-1})Px_n \geq Py_n$$

$$\Leftrightarrow (I - C)p \geq q \quad (19)$$

Donde $p = [p_1 \dots p_k]^T = Px_n$ y $q = [q_1 \dots q_k]^T = Py_n$. En consecuencia $x_n \geq 0$ es equivalente a $p \geq 0$, y la existencia y no-negatividad de $(I - A_n)^{-1}$ es equivalente a $(I - C)^{-1}$.

Suponga que un vector no-negativo p satisface la ecuación (19).

Entonces p_n satisface la siguiente desigualdad [1]:

$$(I - C_m)^{-1}p_m \geq q_m, \text{ para todo } m = 1 \dots k \quad (21)$$

Como C_m es irreducible $\lambda(C_m) < 1$, por lo tanto $\lambda(C) < 1$, además $\sum_{k=0}^{\infty} C^k$ existe y es igual a $(I - C)^{-1}$.

Ahora suponga que existe un vector no-negativo x_n que satisfaga la ecuación (7), entonces $(I - A_n)^{-1} = \sum_{k=0}^{\infty} (A_n)^k$ existe por lo anterior explicado. Teniendo en cuenta que $x_n^* = (I - A_n)^{-1}y_n$ y sabiendo que x_n es cualquier vector que satisfaga (7). Teniendo $(I - A_n)^{-1} \geq 0$ y $y_n \geq 0$, multiplicando ambos lados de la ecuación (7) por $(I - A_n)^{-1}$ no cambia la desigualdad. Por lo tanto $x_n \geq (I - A_n)^{-1}y_n = x_n^*$.

CODIGO DE MATLAB DEL ALGORITMO BIT-LOADING

```
clc
%=====
% Constantes y Variables Utilizadas
%=====

f=8.5:0.0043125:12; %expresado en MHz
f1=4.5:0.0043125:8; %expresado en MHz para la primera banda
f=f.*1e6;
f1=f1.*1e6;
N=size(f,2); %N es el numero de subportadoras
N1=size(f1,2); %N1 es el numero de subportadoras para la primera
banda
BW=0.0043125; %Ancho de banda de cada subportadora en MHz
M=3;%M es el numero de usuarios
M1=3;%M es el numero de usuarios para la primera banda

d=[0.2 0.5 0.7]; %distancia medida en kilometros
kfext=8e-20; %constante de la formula del FEXT
Gamma = 10^(12.75/10); %SNR Gap
b = zeros(N,M); %Bits en cada subportadora
b1 = zeros(N1,M1); %Bits en cada subportadora para la primera
banda

fi = zeros(N,M); %SINR
fi1 = zeros(N1,M1); %SINR para la primera banda

F = zeros(N,M); %Banderas
F1 = zeros(N1,M1); %Banderas para la primera banda

A= ones (M,M);
B= zeros (M,1);
x= zeros (M,1);
A1= ones (M1,M1);
B1= zeros (M1,1);
x1= zeros (M1,1);

No = 10^((-140-30)/10); % PSD del Ruido en dBm
e_i= zeros (1,M); % Vecor de incrementos
H = zeros (M,M,N); % Canal
num =zeros(M,N);
den =zeros(M,N);
SNR =zeros(M,N);
Cost= zeros (N,M);
```

```

% Matriz de costos
Pb = (10^((14.5-30)/10))/N; % Límite de potencia por cada
subportadora

e_i1= zeros (1,M1); % Vecor de incrementos para la primera
banda
H1 = zeros (M1,M1,N1); % Canal para la primera banda
num1 =zeros (M1,N1);
den1 =zeros (M1,N1);
SNR1 =zeros (M1,N1);
Cost1= zeros (N1,M1); % Matriz de costos para la primera banda
Pb1 = (10^((14.5-30)/10))/N1; % Límite de potencia por cada
subportadora para la primera banda

b_cap=11; % Límite de bits a transmitir
Ts=1/4000; % Periodo de Símbolo
B_targ= 40*1e6*Ts; % Número total de bits transmitidos
Pot_Trans=zeros (M,N); %Potencia Transmitida
Pot_Trans1=zeros (M1,N1); %Potencia Transmitida para la primera
banda

fi_ant=0;

%%

%MODELO DEL CANAL Y DEL FEXT DEPENDIENDO DE LA DISTANCIA
H(1,1,:)=Cable(f,d(1));
H(2,2,:)=Cable(f,d(2));
H(3,3,:)=Cable(f,d(3));

H(1,2,:) = HtFexD(f,kfext,d(1),d(2),2);
H(1,3,:) = HtFexD(f,kfext,d(1),d(3),2);

H(2,1,:) = HtFexD(f,kfext,d(2),d(1),2);
H(2,3,:) = HtFexD(f,kfext,d(2),d(3),2);

H(3,1,:) = HtFexD(f,kfext,d(3),d(1),2);
H(3,2,:) = HtFexD(f,kfext,d(3),d(2),2);

H=abs(H);

H1(1,1,:)=Cable(f1,d(1));
H1(2,2,:)=Cable(f1,d(2));
H1(3,3,:)=Cable(f1,d(3));

H1(1,2,:) = HtFexD(f1,kfext,d(1),d(2),2);
H1(1,3,:) = HtFexD(f1,kfext,d(1),d(3),2);

H1(2,1,:) = HtFexD(f1,kfext,d(2),d(1),2);
H1(2,3,:) = HtFexD(f1,kfext,d(2),d(3),2);

```

```

H1(3,1,:) = HtFexD(f1,kfext,d(3),d(1),2);
H1(3,2,:) = HtFexD(f1,kfext,d(3),d(2),2);

H1=abs(H1);

No= No*BW*1e6; % Ruido
%%

for i=1:N
    for j=1:M
        e_i(j)=1;
        fi(i,j)= Ganma*(2^(e_i(j))-1);           % Se calcula el SINR
        for ii= 1:M
            for jj= 1:M
                A(ii,jj)= -fi(i,jj)*(H(ii,jj,i)/H(ii,ii,i))^2;
                A(jj,jj)= 1;
            end
            B(ii)= fi(i,ii)*No/((H(ii,ii,i))^2);
        end
        x = inv(A)*B;
        Cost(i,j)= sum(x);
        e_i(j)= 0;
        fi(i,j)=0;
        if (x(j)> Pb | x(j)<0)
            F(i,j)=1; % Se establece como saturada la subportadora
        end
    end
end

Cost_aux= Cost; % Se busca el usuario con el menor
costo
Cost_aux(find(F==1))= inf; % en cada subcanal
[C,K]=min(Cost_aux');

for i=1:N1
    for j=1:M1
        e_i1(j)=1;
        fil(i,j)= Ganma*(2^(e_i1(j))-1);           % Se calcula el
SINR
        for ii= 1:M1
            for jj= 1:M1
                A1(ii,jj)= -fil(i,jj)*(H1(ii,jj,i)/H1(ii,ii,i))^2;
                A1(jj,jj)= 1;
            end
            B1(ii)= fil(i,ii)*No/((H1(ii,ii,i))^2);
        end
        x1 = inv(A1)*B1;
        Cost1(i,j)= sum(x1);
        e_i1(j)= 0;
        fil(i,j)=0;
        if (x1(j)> Pb1 | x1(j)

```

```

<0)
            F1(i,j)=1; % Se establece como saturada la subportadora
i
                    % para el ususario j
            end
        end
    end

Cost_aux1= Cost1; % Se busca el usuario con el
menor costo
Cost_aux1(find(F1==1))= inf; % en cada subcanal
[C1,K1]=min(Cost_aux1');

while((sum(sum(b))< B_targ) & (sum(sum(F))< M*N))
    [CC,m]= min(C);
    b(m,K(m))= b(m,K(m))+ 1; % Se añande un bit al usuario K(m)
en el subcanal m
    fi(m,K(m))= Ganma*(2^(b(m,K(m))))-1); % Se calcula el SINR
        for ii= 1:M
            for jj= 1:M
                A(ii,jj)= -fi(m,jj)*(H(ii,jj,m)/H(ii,ii,m))^2;
                A(jj,jj)= 1;

            end
            B(ii)= fi(m,ii)*No/((H(ii,ii,m))^2);
        end
    Pot_Trans(:,m) = inv(A)*B; % Se calcula la potencia

    for j=1:M
        e_i(j)=1;
        fi_ant=fi(m,j);
        fi(m,j)= Ganma*(2^(b(m,j)+e_i(j)))-1); % Se calcula el SINR

        for ii= 1:M
            for jj= 1:M
                A(ii,jj)= -fi(m,jj)*(H(ii,jj,m)/H(ii,ii,m))^2;
                A(jj,jj)= 1;

            end
            B(ii)= fi(m,ii)*No/((H(ii,ii,m))^2);
        end
        x =inv(A)*B;
        if (min(x)>0)
            Cost(m,j)= sum(x)-sum(Pot_Trans(:,m)); % Se calcula el
costo de añadir
        end
        e_i(j)= 0; % un bit mas al
subcanal m
        fi(m,j)=fi_ant;
    end
end

```



```

        if (b(m,j)==b_cap)      % F(m,j)=0 significa que se puede añadir
mas bits                      % para el usuario j y subcanal m
            F(m,j)=1;          %F(m,j)=1 significa que el subcanal m de
        end                    % esta saturado
    el usuario j
        if (max(x)>Pb | min(x)<0)
            F(m,j)=1;
        end

    end

    Cost_aux= Cost;
    Cost_aux(find(F==1))= inf;
    [C,K]=min(Cost_aux');

end

Pot_Total=sum(sum(Pot_Trans));

while((sum(sum(b1))< B_targ) & (sum(sum(F1))< M1*N1))
    [CC1,m1]= min(C1);
    b1(m1,K1(m1))= b1(m1,K1(m1))+ 1;    % Se añande un bit al
usuario K(m) en el subcanal m
    fil(m1,K1(m1))= Ganma*(2^(b1(m1,K1(m1)))-1);    % Se calcula el
SINR
        for ii= 1:M1
            for jj= 1:M1
                A1(ii,jj)= -fil(m1,jj)*(H1(ii,jj,m1)/H1(ii,ii,m1))^2;
                A1(jj,jj)= 1;

            end

            B1(ii)= fil(m1,ii)*No/((H1(ii,ii,m1))^2);
        end
    Pot_Trans1(:,m1) = inv(A1)*B1;    % Se calcula la potencia

    for j=1:M1
        e_il(j)=1;
        fi_ant=fil(m1,j);
        fil(m1,j)= Ganma*(2^(b1(m1,j)+e_il(j))-1);    % Se calcula el
SINR

        for ii= 1:M1
            for jj= 1:M1
                A1(ii,jj)= -fil(m1,jj)*(H1(ii,jj,m1)/H1(ii,ii,m1))^2;
                A1(jj,jj)= 1;

            end

            B1(ii)= fil(m1,ii)*No/((H1(ii,ii,m1))^2);
        end
    x1 =inv(A1)*B1;

```

```

        if (min(x1)>0)
            Cost1(m1,j)= sum(x1)-sum(Pot_Trans1(:,m1)); % Se calcula el
costo de añadir
        end
        e_i1(j)= 0; % un bit mas al
subcanal m
        fil(m1,j)=fi_ant;

        if(b1(m1,j)==b_cap) % F(m,j)=0 significa que se puede
añadir mas bits
            F1(m1,j)=1; % para el ususario j y subcanal m
        end %F(m,j)=1 significa que el subcanal m de
el ususario j
        if (max(x1)>Pb1 | min(x1)<0) % esta saturado
            F1(m1,j)=1;
        end

    end

    Cost_aux1= Cost1;
    Cost_aux1(find(F1==1))= inf;
    [C1,K1]=min(Cost_aux1');

end

Pot_Total1=sum(sum(Pot_Trans1));

% Calculo de la capacidad del sistema
for i=1:M
    num(i,:)= reshape(((abs(H(i,i,:)))^2),1,N).*Pot_Trans(i,:);
end

    den(1,:)=
((reshape((abs(H(1,2,:)))^2,1,N).*Pot_Trans(2,:))^1/0.6) +
((reshape((abs(H(1,3,:)))^2,1,N).*Pot_Trans(3,:))^1/0.6))^0.6 +
No;
    den(2,:)=
((reshape((abs(H(2,1,:)))^2,1,N).*Pot_Trans(1,:))^1/0.6) +
((reshape((abs(H(2,3,:)))^2,1,N).*Pot_Trans(3,:))^1/0.6))^0.6 +
No;
    den(3,:)=
((reshape((abs(H(3,1,:)))^2,1,N).*Pot_Trans(1,:))^1/0.6) +
((reshape((abs(H(3,2,:)))^2,1,N).*Pot_Trans(2,:))^1/0.6))^0.6 +
No;

    SNR_Gap=10^(12.75/10);

    for i=1:M
        SNR(i,:)=num(i,:)/(den(i,:).*SNR_Gap);
    end

```

```

for i=1:M1
    num1(i,:)= reshape(((abs(H1(i,i,:))).^2),1,N1).*Pot_Trans1(i,:);
end

den1(1,:)=
((reshape((abs(H1(1,2,:))).^2,1,N1).*Pot_Trans1(2,:)).^(1/0.6) +
((reshape((abs(H1(1,3,:))).^2,1,N1).*Pot_Trans1(3,:)).^(1/0.6))).^0.
6 + No;
den1(2,:)=
((reshape((abs(H1(2,1,:))).^2,1,N1).*Pot_Trans1(1,:)).^(1/0.6) +
((reshape((abs(H1(2,3,:))).^2,1,N1).*Pot_Trans1(3,:)).^(1/0.6))).^0.
6 + No;
den1(3,:)=
((reshape((abs(H1(3,1,:))).^2,1,N1).*Pot_Trans1(1,:)).^(1/0.6) +
((reshape((abs(H1(3,2,:))).^2,1,N1).*Pot_Trans1(2,:)).^(1/0.6))).^0.
6 + No;

SNR_Gap1=10^(12.75/10);

for i=1:M1
    SNR1(i,:)=num1(i,:)./(den1(i,:).*SNR_Gap1);
end

fs=1/Ts;

Cap= fs.*log2(1+SNR)+ fs.*log2(1+SNR1); % Calculo de la capacidad

fprintf(1,'Potencia Total = %f %s\n',10*log10(Pot_Total)+30,'dbm');
fprintf(1,'Número total de bits = %d \n',B_targ);

% GRAFICA DEL NUMERO DE BITS ASIGNADOS POR USUARIO
figure(1)

i=1:1:M;
stem(i,sum(b(:,i)));
ylabel('Bits'), xlabel('Usuario');
title('Numero de bits asignados por cada Usuario');
hold on

% GRAFICA DE LA CAPACIDAD DE CADA USUARIO
figure(2)
j=1:1:M;
Cp=Cap';
stem(j,sum(Cp(:,j)));
ylabel('Bits/s'), xlabel('Usuario');
title('Capacidad de cada usuario');
hold on

```

```

% GRAFICA DEL NUMERO DE BITS POR CADA SUBPORTADORA
figure(3)

i=1:1:N;
subplot(2,2,1:2); stem(i,b(i,1));
ylabel('Bits'), xlabel('Subportadora');
title('Numero de bits asignados por cada subportadora (Usuario
1)');
hold on
subplot(2,2,3); stem(i,b(i,2));
ylabel('Bits'), xlabel('Subportadora');
title('Numero de bits asignados por cada subportadora (Usuario
2)');
hold on
subplot(2,2,4); stem(i,b(i,3));
ylabel('Bits'), xlabel('Subportadora');
title('Numero de bits asignados por cada subportadora (Usuario
3)');
hold on

% GRAFICA DE LA POTENCIA TOTAL POR USUARIO
figure(4)

j=1:1:M;
PT=Pot_Trans';
stem(j,10*log10(sum(PT(:,j)))+30);
ylabel('Potencia (dbm)'), xlabel('Usuario');
title('Potencia total por cada Usuario');
hold on

```

SIMULACION SIN EL ALGORITMO PROPUESTO

```

clc
%=====
% Constantes y Variables Utilizadas
%=====

f=8.5:0.0043125:12; %expresado en MHz
f1=4.5:0.0043125:8; %expresado en MHz para la primera banda
f=f.*1e6;
f1=f1.*1e6;
N=size(f,2); %N es el numero de subportadoras
N1=size(f,2); %N es el numero de subportadoras para la primera
banda

BW=0.0043125; %Ancho de banda de cada subportadora en MHz
M=3; %M es el numero de usuarios
M1=3; %M es el numero de usuarios para la primera banda

d=[0.2 0.5

```

```

0.7];          %distancia medida en kilometros
kfect=8e-20;   %constante de la formula del FEXT
Ganma = 10^(12.75/10); %SNR Gap
b = zeros(N,M); %Bits en cada subportadora
fi = zeros(N,M); %SINR
F = zeros(N,M); %Banderas
A= ones (M,M);
B= zeros (M,1);
x= zeros (M,1);

b1 = zeros(N1,M1); %Bits en cada subportadora para la primera
banda
fil = zeros(N1,M1); %SINR para la primera banda
F1 = zeros(N1,M1); %Banderas para la primera banda
A1= ones (M1,M1);
B1= zeros (M1,1);
x1= zeros (M1,1);

No = 10^((-140-30)/10); % PSD del Ruido en dBm
e_i= zeros (1,M); % Vecor de incrementos
H = zeros (M,M,N); % Canal
num =zeros(M,N);
den =zeros(M,N);
SNR =zeros(M,N);
Cost= zeros (N,M); % Matriz de costos
Pb = (10^((14.5-30)/10))/N; % Limite de potencia por cada
subportadora

e_i1= zeros (1,M1); % Vecor de incrementos para la primera
banda
H1 = zeros (M1,M1,N1); % Canal para la primera banda
num1 =zeros(M1,N1);
den1 =zeros(M1,N1);
SNR1 =zeros(M1,N1);
Cos1t= zeros (N1,M1); % Matriz de costos para la primera banda
Pb1 = (10^((14.5-30)/10))/N1; % Limite de potencia por cada
subportadora para la primera banda

b_cap=11; % Limite de bits a transmitir
Ts=1/4000; % Periodo de Símbolo
B_targ= 40*1e6*Ts; % Número total de bits transmitidos
Pot_Trans=zeros(M,N); %Potencia Transmitida
Pot_Trans1=zeros(M1,N1); %Potencia Transmitida para la primera
banda
fi_ant=0;
fg=0;
fg1=0;
%%

%MODELO DEL CANAL Y DEL FEXT DEPENDIENDO DE LA DISTANCIA
H(1,1,:)=Cable(f,d(1));
H(2,2,:)=Cable(f,d(2));
H(3,3,:)=Cable(f,d(3));

```

```

H(1,2,:) = HtFexD(f,kfext,d(1),d(2),2);
H(1,3,:) = HtFexD(f,kfext,d(1),d(3),2);

H(2,1,:) = HtFexD(f,kfext,d(2),d(1),2);
H(2,3,:) = HtFexD(f,kfext,d(2),d(3),2);

H(3,1,:) = HtFexD(f,kfext,d(3),d(1),2);
H(3,2,:) = HtFexD(f,kfext,d(3),d(2),2);

H=abs(H);

```

**%MODELO DEL CANAL Y DEL FEXT DEPENDIENDO DE LA DISTANCIA PARA EL 1
CANAL**

```

H1(1,1,:)=Cable(f1,d(1));
H1(2,2,:)=Cable(f1,d(2));
H1(3,3,:)=Cable(f1,d(3));

H1(1,2,:) = HtFexD(f1,kfext,d(1),d(2),2);
H1(1,3,:) = HtFexD(f1,kfext,d(1),d(3),2);

H1(2,1,:) = HtFexD(f1,kfext,d(2),d(1),2);
H1(2,3,:) = HtFexD(f1,kfext,d(2),d(3),2);

H1(3,1,:) = HtFexD(f1,kfext,d(3),d(1),2);
H1(3,2,:) = HtFexD(f1,kfext,d(3),d(2),2);

H1=abs(H1);

```

No= No*BW*1e6; **% Ruido**
%%

```

while (1)
    for i=1:N
        for j=1:M
            if (sum(sum(b))< B_targ)
                b(i,j)=b(i,j)+1;
            else
                fg=1;
                break;
            end
        end
    end
    if (fg==1) break; end
end

while (1)
    for i=1:N1
        for j=1:M1

```

```

        if (sum(sum(b1)) < B_targ)
            b1(i,j)=b1(i,j)+1;
        else
            fg1=1;
            break;
        end
    end
end
if (fg1==1) break; end
end

for i=1:N
    for j= 1:M

        fi(i,j)= Gamma*(2^(b(i,j))-1);    % Se calcula el SINR
        for ii= 1:M
            for jj= 1:M
                A(ii,jj)= -fi(i,jj)*(H(ii,jj,i)/H(ii,ii,i))^2;
                A(jj,jj)= 1;

            end

            B(ii)= fi(i,ii)*No/((H(ii,ii,i))^2);
        end
        Pot_Trans(:,i) = inv(A)*B;    % Se calcula la potencia

    end
end

Pot_Total=sum(sum(Pot_Trans));

for i=1:N1
    for j= 1:M1

        fil(i,j)= Gamma*(2^(b1(i,j))-1);    % Se calcula el SINR
        for ii= 1:M1
            for jj= 1:M1
                A1(ii,jj)= -fil(i,jj)*(H1(ii,jj,i)/H1(ii,ii,i))^2;
                A1(jj,jj)= 1;

            end

            B1(ii)= fil(i,ii)*No/((H1(ii,ii,i))^2);
        end
        Pot_Trans1(:,i) = inv(A1)*B1;    % Se calcula la potencia

    end
end

Pot_Total1=sum(sum(Pot_Trans1));

% Calculo de la capacidad del sistema

```

```

for i=1:M
    num(i,:)= reshape(((abs(H(i,i,:))).^2),1,N).*Pot_Trans(i,:);
end

    den(1,:)=
((reshape((abs(H(1,2,:))).^2,1,N).*Pot_Trans(2,:)).^(1/0.6) +
((reshape((abs(H(1,3,:))).^2,1,N).*Pot_Trans(3,:)).^(1/0.6))).^0.6 +
No;
    den(2,:)=
((reshape((abs(H(2,1,:))).^2,1,N).*Pot_Trans(1,:)).^(1/0.6) +
((reshape((abs(H(2,3,:))).^2,1,N).*Pot_Trans(3,:)).^(1/0.6))).^0.6 +
No;
    den(3,:)=
((reshape((abs(H(3,1,:))).^2,1,N).*Pot_Trans(1,:)).^(1/0.6) +
((reshape((abs(H(3,2,:))).^2,1,N).*Pot_Trans(2,:)).^(1/0.6))).^0.6 +
No;

    SNR_Gap=10^(12.75/10);

for i=1:M
    SNR(i,:)=num(i,:)/(den(i,:).*SNR_Gap);
end

for i=1:M1
    num1(i,:)= reshape(((abs(H1(i,i,:))).^2),1,N1).*Pot_Trans1(i,:);
end

    den1(1,:)=
((reshape((abs(H1(1,2,:))).^2,1,N1).*Pot_Trans1(2,:)).^(1/0.6) +
((reshape((abs(H1(1,3,:))).^2,1,N1).*Pot_Trans1(3,:)).^(1/0.6))).^0.6 +
No;
    den1(2,:)=
((reshape((abs(H1(2,1,:))).^2,1,N1).*Pot_Trans1(1,:)).^(1/0.6) +
((reshape((abs(H1(2,3,:))).^2,1,N1).*Pot_Trans1(3,:)).^(1/0.6))).^0.6 +
No;
    den1(3,:)=
((reshape((abs(H1(3,1,:))).^2,1,N1).*Pot_Trans1(1,:)).^(1/0.6) +
((reshape((abs(H1(3,2,:))).^2,1,N1).*Pot_Trans1(2,:)).^(1/0.6))).^0.6 +
No;

    SNR_Gap1=10^(12.75/10);

for i=1:M1
    SNR1(i,:)=num1(i,:)/(den1(i,:).*SNR_Gap1);
end

fs=1/Ts;

Cap= fs.*log2(1+SNR)+ fs.*log2(1+SNR1); % Calculo de la capacidad

fprintf(1,'Potencia Total = %f %s\n',10*log10(Pot_Total)+30,'dbm');
fprintf(1,'Número total de bits = %d \n',B_targ);

```



```

% GRAFICA DEL NUMERO DE BITS ASIGNADOS POR USUARIO
figure(1)

i=1:1:M;
stem(i, sum(b(:, i)));
ylabel('Bits'), xlabel('Usuario');
title('Numero de bits asignados por cada Usuario');
hold on

% GRAFICA DE LA CAPACIDAD DE CADA USUARIO
figure(2)
j=1:1:M;
Cp=Cap';
stem(j, sum(Cp(:, j)));
ylabel('Bits/s'), xlabel('Usuario');
title('Capacidad de cada usuario');
hold on

% GRAFICA DEL NUMERO DE BITS POR CADA SUBPORTADORA
figure(3)

i=1:1:N;
subplot(2,2,1:2); stem(i, b(i, 1));
ylabel('Bits'), xlabel('Subportadora');
title('Numero de bits asignados por cada subportadora (Usuario
1)');
hold on
subplot(2,2,3); stem(i, b(i, 2));
ylabel('Bits'), xlabel('Subportadora');
title('Numero de bits asignados por cada subportadora (Usuario
2)');
hold on
subplot(2,2,4); stem(i, b(i, 3));
ylabel('Bits'), xlabel('Subportadora');
title('Numero de bits asignados por cada subportadora (Usuario
3)');
hold on

% GRAFICA DE LA POTENCIA TOTAL POR USUARIO
figure(4)

j=1:1:M;
PT=Pot_Trans';
stem(j, 10*log10(sum(PT(:, j)))+30);
ylabel('Potencia (dbm)'), xlabel('Usuario');
title('Potencia total por cada Usuario');
hold on

```

FUNCION DE TRANSFERENCIA DEL FEXT

```
function [FEXT] = HtFexD(f,kfext,d_origen,d_destino,n)
%
%nombre archivo: SPsdFexD.m
%Argumentos:
% d_origen: distancia del FEXT origen (Km)
% d_destino: distancia FEXT destino (Km)
%Retorno:
% Funcion de transferencia del FEXT del cable

%***** Funcion de transferencia FEXT
%kf=7.16e-19;
%kf=0.0056
%usando factor de conversion: 1Km = 3280.83pies
FEXT=
(n^(0.6))*(kfext*(d_origen*3280.83).*(f.^2)).*(Cable(f,d_destino));

end
```

FUNCION DEL MODELO DEL CABLE

```
function [H]=Cable(f,d)

%fmin          = 0.00138*1e6;          % Frecuencia minima (Hz)
%fmax          = 12e6;                % Frecuencia maxima (Hz)
%fgranularity  = 4312.5;              % granularidad en el dominio de la
frecuencia (Hz) igual al ancho de banda por tono

%f = (fmin:fgranularity:fmax);

%MODELADO DEL CABLE
%
=====
====
%MODELO EMPIRICO BRITISH TELECOM
%
=====
=====
% TP2 0.5 mm 24-AWG
%
=====
=====
```

```
cable.param.r0c=174.55888;
cable.param.r0s=0;
cable.param.ac=0.053073481;
cable.param.as=0.0;
cable.param.l0=617.29539e-6;
cable.param.loo=478.97099e-6;
cable.param.b=1.1529766;
cable.param.fm=553.760e3;
cable.param.coo=50e-9;
cable.param.c0=0;
cable.param.ce=0;
cable.param.g0=234.87476e-15;
cable.param.ge=1.38;
```

```
%
```

```
=====
=====
```

```
% distancia para cable
```

```
%
```

```
=====
=====
```

```
%d= 0.5; % en kms
```

```
%
```

```
=====
=====
```

```
% CAMBIO DE VARIABLE PARA UTILIZAR EN LA SIMULACION
```

```
r0c=cable.param.r0c;
r0s=cable.param.r0s;
ac=cable.param.ac;
as=cable.param.as;
l0=cable.param.l0;
loo=cable.param.loo;
b=cable.param.b;
fm=cable.param.fm;
coo=cable.param.coo;
c0=cable.param.c0;
ce=cable.param.ce;
g0=cable.param.g0;
ge=cable.param.ge;
```

```
%
```

```
=====
=====
```

```
if (as==0) && (r0s==0)
```

```
    r0s = inf; % si el valor de ros y as son cero se cambia
ros=0 para evitar indeterminacion
```

```

end;
%
=====
=====

%
=====
=====
%HALLAR PARAMETROS PRIMARIOS SEGUN EL MODELO
%
=====
=====

    f2=f.^2;
    fmb=(f./fm).^b;

    R=1./((1./sqrt(sqrt(r0c.^4+ac.*f2))) + (1./sqrt(sqrt(r0s.^4+
as.*f2)))));
    L=(l0+l00*fmb)/(1+fmb);
    C=c0+c0.*f.^(-ce);
    G=g0.*f.^(ge);

%PARAMETROS SECUNDARIOS SEGUN EL MODELO

    Z=R+j.*2*pi.*f.*L;
    Y=G+j.*2*pi.*f.*C;

    gamma=sqrt(Z.*Y);
    Z0=sqrt(Z./Y);

%HALLAMOS LOS PARAMETROS ABCD

Aparam=cosh(gamma*d);
Bparam=Z0.*sinh(gamma*d);
Cparam=(1./Z0).*sinh(gamma*d);
Dparam=cosh(gamma*d);

Zl= 100;    % SELECCIONAMOS ZL=100 PARA ACOUPLE DE LA CARGA
Zs=135;

% CALCULAMOS LA FUNCION DE TRANSFERENCIA H O PERDIDA DE INSERCIÓN

H=(Zl)/(Aparam.*Zl+Bparam+Cparam.*Zs.*Zl+Dparam.*Zs);

```

BIBLIOGRAFIA

[1] Jungwon Lee, Ranjan V. Sonalkar and Jhon M. Cioff. "Multi-user Discrete Bit-Loading for DMT-based DSL Systems". IEEE GLOBECOM, Nov 2002.

[2] T. Starr, J. M. Cioffi and P.J. Silverman. "Understanding Digital Subscriber Line Technology". Upper Saddle River, NJ: Prentice Hall 1999.

[3] J. Campello. "Practical bit-loading for DMT". IEEE International Conference on Communications , June 1999.

[4] D. Hughes-Hartogs. "Ensemble Modem Structure for Imperfect Transmission Media" U.S Patents 4 679 227 (July 1987) and 4 883 706 (May 1989).

[5] B. Marcus, P. Siegel and R. Roth. "Constrained systems and coding for recording channels". Chapter 20 of HandBook of Coding Theory, ed. W.C Huffman and V. Pless Elsevier Press, 1998.

[6] D. Mitra. "An asynchronous distributed algorithm for power control in celular radio system". Proc. 4th WinLab Wksp. Third generation Wireless Info. Network, Rutgers Univ., 1993.

[7] Cajetan M. Akujuobi and Jie Shen. "Efficient Multi-user Parallel Greedy Bit-Loading Algorithm with Fairness Control For DMT Systems". Prairie View AEM University, Texas USA, November 1997.

[8] Sigurd Schelstraete and Ben Runyan. "Designing for VDSL2". Ikanos Communications, May 2005.

