



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**TEMA: ANÁLISIS DE DESEMPEÑO DE CÓDIGOS FOUNTAIN
PARA CANALES MÓVILES CON DESVANECIMIENTO**

AUTOR: GARZÓN NAVARRETE, BILLY ANDRÉS

DIRECTOR: ING. OLMEDO CIFUENTES, GONZALO FERNADO

SANGOLQUÍ

2018



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que el trabajo de titulación, ***“ANÁLISIS DE DESEMPEÑO DE CÓDIGOS FOUNTAIN PARA CANALES MÓVILES CON DESVANECIMIENTO”*** fue realizado por el señor ***Garzón Navarrete, Billy Andrés*** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 27 de julio del 2018.

Ing. Gonzalo Olmedo PhD.
DIRECTOR

C. C.: 1711696342



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

AUTORÍA DE RESPONSABILIDAD

Yo, *Garzón Navarrete, Billy Andrés*, declaro que el contenido, ideas y criterios del trabajo de titulación: “Análisis de desempeño de códigos Fountain para canales móviles con desvanecimiento” es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 27 de julio del 2018.

Billy Andrés Garzón Navarrete

C. C: 1720737103



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, *Garzón Navarrete, Billy Andrés* autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “*Análisis de desempeño de códigos Fountain para canales móviles con desvanecimiento*” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 27 de julio del 2018.

Billy Andrés Garzón Navarrete

C. C: 1720737103

DEDICATORIA

*“Si tuviera la suerte de alcanzar alguno de mis ideales,
sería en nombre de toda la humanidad.”*

Níkola Tesla

A mi madre Susy y a mi padre Wilson, por el apoyo, esfuerzo y amistad que me supieron brindar,
desde niño hasta hoy.

A mis hermanos Michelle y Fernando, cuyo ejemplo de generosidad hacia la familia, me enseñó
a dar lo mejor de uno para los demás, sin esperar nada a cambio.

A mi tutor y a cada uno de los maestros que supieron brindarme todos sus conocimientos, pero
sobre todo su amistad y camaradería.

Billy Andrés Garzón Navarrete

AGRADECIMIENTO

A mis padres Wilson y Susy, cuyo ejemplo de responsabilidad y esfuerzo, me llevó a culminar esta etapa de mi vida, siendo mi profesión la mejor herencia que les puedo agradecer.

A mis hermanos Fernando y Michelle, quienes fueron y seguirán siendo mis primeros amigos, siendo un ejemplo de muy gratas virtudes y cuidados hacia mí.

A una gran mujer que me acompañó en esta etapa tan importante en mi vida, sin dejarme desmayar, estando en los altos y bajos momentos, caminando de mi mano con; amor, amistad y respeto. Por esto y muchas cosas más; gracias Rosalía.

A cada uno de los miembros de mi familia, tíos y tías, primos y primas, y por supuesto a mis abuelitos que han sido un peldaño para seguir adelante, en cada etapa de mi vida.

A mis maestros, los cuales fueron la base fundamental, para haber llegado hasta esta etapa de mi vida profesional, sobre todo a mi tutor: Gonzalo Olmedo, el cual supo brindarme su apoyo y compartir sus conocimientos al cien por ciento, al igual que su profesionalismo. A cada uno de los docentes del CIRAD, cuya amistad, apoyo y conocimiento, marcó una gran camaradería y aprecio hacia ellos; en especial al ingeniero Paúl Bernal por toda la ayuda y confianza brindada, desde el momento en que llegué a la universidad, hasta mi etapa de egreso. Un agradecimiento cordial también a muchos de los docentes que ahora los considero verdaderos amigos como: Rodolfo Gordillo, Cristian Vega, Darwin Aguilar, Freddy Acosta, Fabián Sáenz, Julio Larco. Y por último, pero no menos importante me parece adecuado agradecer a los docentes cuyas enseñanzas marcaron mi paso por la universidad, y por su puesto marcaron mi forma de hacer las cosas a nivel profesional, quiero brindar un saludo con una alta gratitud para los ingenieros: Patricio Vizcaíno, Román Lara, Víctor Proaño y Alberto Albuja.

A todos los grandes amigos que me acompañaron en cada etapa de la vida, desde el colegio y sé que hasta siempre, Aldo y Byron. A los que se encargaron de hacerme ver que la universidad es la mejor etapa de la vida, Bryan, Eduardo, Alex, Juan, Víctor, Ron, Suco, Tito, Ched. Y por supuesto aquellos que fueron más que mis amigos; mis hermanos, Marco, Carlos, Fernando, Jorge, Alex, Kevin y Johan. Un agradecimiento especial a todo el Team Tonsupa, los cuales con su amistad y apoyo me hicieron vivir mi último semestre lleno de buenos momentos. A todo el equipo de Tecnologías del Hospital Eugenio Espejo, gracias por su compañerismo amistad y profesionalismo, todo este grupo de unión laboral me permite seguir adelante en mi carrera profesional y ocupacional.

Billy Andrés Garzón Navarrete

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
ÍNDICE DE ECUACIONES	xii
RESUMEN	xiii
ABSTRACT	xiv
CAPÍTULO I	
GENERALIDADES	
1.1 Justificación	1
1.2 Objetivos	2
1.2.1 General	2
1.2.2 Objetivos Específicos	2
1.3 Metodología	2
1.4 Resultados esperados	3
CAPÍTULO II	
FUNDAMENTOS TEÓRICOS	
2.1 Principios de codificación	4
2.1.1. Códigos LDPC	4
2.1.2. Códigos Digital Fountain	6
2.1.3. Transformada de Luby	8
2.1.3.1 Terminología	8

2.1.3.2 Proceso de Codificación.....	9
2.1.3.3 Proceso de Decodificación.....	10
2.1.4 Codificación raptor.....	12
2.2 Descripción del modelo del canal.....	13
2.2.1 Canal AWGN (Ruido Gaussiano Blanco Aditivo).....	13
2.2.2 Relación Señal a Ruido (SNR).....	14
2.2.3 Canal con Desvanecimiento.....	16
2.2.4 Canal de tipo Rayleigh.....	16

CAPÍTULO III

DEFINICIÓN DEL ESCENARIO

3.1 Estructura del sistema de comunicaciones.....	19
3.1.1 Diagrama de bloques.....	19
3.1.1.1 Diagrama de bloques en un canal de tipo AWGN.....	20
3.1.1.2 Diagrama de bloques en un canal con desvanecimiento de tipo Rayleigh.....	21
3.1.1.3 Diagrama de bloques-pruebas principales.....	21
3.2 Diseño de codificación y decodificación.....	23
3.2.1 Fountain-LT.....	23
3.2.1.1 Codificación.....	23
3.2.1.1 Decodificación.....	25
3.2.2 Proceso de codificación y decodificación Fountain-Raptor.....	31
3.2.2.1 Codificación.....	32
3.2.2.2 Decodificación.....	32

CAPÍTULO IV

ANÁLISIS DE RESULTADOS

4.1 Canal de tipo AWGN.....	35
4.1.1. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de mil bits transmitidos).....	35
4.1.2. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 10 mil bits transmitidos).....	36

4.1.3. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 100 mil bits transmitidos).....	38
4.1.4. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 1 millón de bits transmitidos).....	39
4.1.5. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, comparación de cada variación relacionada al tamaño en bits de los paquetes transmitidos.....	40
4.1.6. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, con variación en sus iteraciones.....	41
4.2 Canal con desvanecimiento de tipo Rayleigh.....	43
4.2.1. Desempeño del codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de mil bits transmitidos).....	43
4.2.2. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 5 mil bits transmitidos).....	44
4.2.3. Desempeño del codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 10 mil bits transmitidos).....	45
4.2.4. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 100 mil bits transmitidos).....	46
4.2.5. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, comparación de cada variación, relacionada al tamaño en bits de los paquetes transmitidos.....	47
4.2.6. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, con variación en sus iteraciones.....	48

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.....	50
5.2. RECOMENDACIONES.....	52
REFERENCIAS.....	53

ÍNDICE DE FIGURAS

Figura 1. Gráfico de Tanner correspondiente a la matriz de paridad H.....	5
Figura 2. Representación de Códigos Fountain	6
Figura 3. Canal de borrado de tipo Binario (BEC)	7
Figura 4. Terminología.....	9
Figura 5. Ejemplo-Proceso de Codificación LT.....	9
Figura 6. Grafo Bipartido	10
Figura 7. Ejemplo de Decodificación (4 Iteración).....	11
Figura 8. Ejemplo Codificación (Completa)	12
Figura 9. Grafo-Código Raptor	12
Figura 10. PDF de una variable aleatoria Gaussiana	14
Figura 11. Escenario de comunicaciones con desvanecimiento	16
Figura 12. PDF de una v.a. Rayleigh para distintos valores de σ	17
Figura 13. Comparación de probabilidad de error de un canal AWGN, con otro con desvanecimiento de tipo Rayleigh, utilizando modulación BPSK.....	18
Figura 14. Diagrama de Bloques-Primer escenario de comunicaciones (Canal AWGN)	20
Figura 15. Diagrama de Bloques-Segundo escenario de comunicaciones (Canal RAYLEIGH) .	21
Figura 16. Diagrama de Bloques-Codificación Raptor vs LDPC (Canal AWGN).....	22
Figura 17. Diagrama de Bloques-Codificación LT vs LDPC (Canal RAYLEIGH).....	23
Figura 18. Representación de la matriz generadora de código (G), en un grafo de Tanner.....	24
Figura 19. Grafo de Tanner inicial para Decodificación Fountain	25
Figura 20. Primer paso del proceso de decodificación	27
Figura 21. Segundo paso del proceso de decodificación	27
Figura 22. Tercer paso del proceso de decodificación	28
Figura 23. Cuarto paso del proceso de decodificación	28
Figura 24. Quinto paso del proceso de Decodificación	29
Figura 25. Final de la Decodificación (Mensaje Recuperado).....	29
Figura 26. Mosaico de Decodificación Fountain-LT Fallida	30
Figura 27. Mosaico de Decodificación Fountain-LT Exitosa	31
Figura 28. Grafo de Tanner (Codificación Raptor).....	32
Figura 29. Inicio de Decodificación Raptor (Decodificador LT).....	33
Figura 30. Culminación de Decodificación Raptor (Decodificador LDPC).....	33
Figura 31. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 1000 bits transmitidos.....	35
Figura 32. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 10 mil bits transmitidos.	36

Figura 33. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 100 mil bits transmitidos.38

Figura 34. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 1 millón de bits transmitidos.39

Figura 35. Comparación del desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC de cada variación relacionada al tamaño en bits de los paquetes transmitidos.40

Figura 36. Comparación del desempeño del código Raptor, vs un código de tipo LDPC con variación en sus iteraciones para decodificación.41

Figura 37. Comparación del desempeño del código Raptor, vs un código de tipo LDPC con variación en sus iteraciones para decodificación (paquetes transmitidos con tamaño 1 millón de bits).42

Figura 38. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 1000 bits transmitidos.43

Figura 39. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 5000 bits transmitidos.44

Figura 40. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 10 mil bits transmitidos.45

Figura 41. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 100 mil bits transmitidos.46

Figura 42. Comparación del desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC de cada variación relacionada al tamaño en bits de los paquetes transmitidos.47

Figura 43. Comparación del desempeño del código Raptor, vs un código de tipo LDPC con variación en sus iteraciones para decodificación.48

ÍNDICE DE TABLAS

Tabla 1 Resumen representativo de valores relacionado a los elementos detallados del grafo de Tanner Resumen representativo de valores relacionado a los elementos detallados del grafo de Tanner	26
Tabla 2 Resumen representativo de valores relacionado a los elementos detallados del grafo de Tanner de la Figura 25.	30

ÍNDICE DE ECUACIONES

Ecuación 1	13
Ecuación 2	14
Ecuación 3	14
Ecuación 4	15
Ecuación 5	15
Ecuación 6	15
Ecuación 7	15
Ecuación 8	16
Ecuación 9	17
Ecuación 10	24
Ecuación 11	25

RESUMEN

Este trabajo presenta, un análisis de desempeño de los códigos correctores de error, *Digital Fountain-Raptor*, considerados “códigos de nueva generación”, debido a que se utilizan en aplicaciones multimedia de pago con calidad *Full HD* y *4K*, aplicaciones de tipo *multicast* donde existe un número grande de usuarios conectados a un servidor, enviando tráfico al mismo al solicitar “ser escuchados” (video juegos *on-line*). Los primeros códigos de tipo *Digital Fountain*, son los *Fountain-LT (Luby Transform)*, estos dan origen a una característica esencial de los códigos *Fountain*, la cual es, generar una cantidad potencialmente grande de paquetes y enviarlos al receptor, el mismo que tan sólo necesitan conocer cierta cantidad de dichos paquetes para reconstruir la información. El término viene dado por su analogía con una “fuente” de agua, de la cual emanan continuamente gotas, donde cualquier persona que tenga sed, no tiene más que llenar un vaso de esa fuente, sin importar la cantidad, ya que gracias a dichas gotas la fuente se ha llenado y saciará su sed. Los códigos *LT* se utilizaron como base para la codificación y decodificación *Raptor*, ya que la estructura de estos códigos, es una combinación entre un código de tipo *LT*, con un pre-codificador de tipo *FEC*, en este caso se utilizó al código *LDPC*. Para el análisis realizado se planteó dos tipos de escenarios de comunicaciones; en un canal de tipo *Gaussiano (AWGN)* y el otro, modelado mediante un desvanecimiento de tipo *Rayleigh*, utilizando una modulación *BPSK* para ambos casos.

PALABRAS CLAVE:

- **CÓDIGO DIGITAL FOUNTAIN-RAPTOR**
- **COMBINACIÓN DE CÓDIGOS CORRECTORES**
- **CÓDIGOS LT BASE DE CODIFICACIÓN Y DECODIFICACIÓN RAPTOR**
- **PRE CODIFICADOR DE TIPO FEC-LDPC**
- **CÓDIGOS LT BASE DE CODIFICACIÓN Y DECODIFICACIÓN RAPTOR**

ABSTRACT

This paper presents, a performance analysis of the error correction codes, Digital Fountain-Raptor, considered "new generation codes", because they are used in multimedia payment applications with Full HD and 4K quality, multicast applications where There is a large number of users connected to a server, sending traffic to it by requesting "be heard" (video games online). The first Digital Fountain type codes are the Fountain-LT (Luby Transform), which gives rise to an essential characteristic of the Fountain codes, which is to generate a potentially large number of packets and send them to the receiver, the same as they only need to know a certain amount of said packages to reconstruct the information. The term is given by its analogy with a "source" of water, from which drops emanate continuously, where any person who is thirsty, has only to fill a glass from that source, regardless of the quantity, since thanks to said drops the fountain has been filled and it will quench your thirst. The LT codes were used as the basis for the Raptor encoding and decoding, since the structure of these codes is a combination of an LT type code with a FEC type pre-encoder, in this case the LDPC code was used. For the analysis carried out, two types of communication scenarios were proposed; in a Gaussian type channel (AWGN) and the other, modeled by a Rayleigh type fading, using a BPSK modulation for both cases.

KEYWORDS:

- **FOUNTAIN-RAPTOR DIGITAL CODE**
- **COMBINATION OF CORRECTOR CODES**
- **LT CODES RAPTOR CODING AND DECODING BASE**
- **PRE FEC-LDPC TYPE ENCODER**
- **LT CODES RAPTOR CODING AND DECODING BASE**

CAPÍTULO I

1. GENERALIDADES

1.1 Justificación

Nuevos sistemas de transmisión que ofrecen contenido multimedia en calidad $4K^1$ y en un futuro pretenden garantizar visualización en $8K$, así como el avance tecnológico en las redes de nueva generación de telefonía móvil, denominada $5G$. Han llevado a la necesidad de manejar recursos tecnológicos que hagan que los sistemas de comunicaciones sean más robustos; por lo cual, la importancia del criterio de reutilización de frecuencias con el afán de aprovechar al máximo el ancho de banda de un sistema de comunicaciones. Así como lograr que el desempeño del sistema en cuanto a su capacidad se refiere, sea lo más cercana posible al límite de *Shannon*; y por supuesto el manejo de códigos de detección y corrección de errores de canal más sofisticados, creados como complemento y soporte de cada una de estas nuevas tecnologías.

Tomando en cuenta estas consideraciones, se analizó la factibilidad de plantear escenarios típicos de Telecomunicaciones, haciendo énfasis en codificadores de nueva generación. Para este caso se utilizaron los códigos *Digital Fountain*, los cuales según sus características (MacKay, 2005), son un tipo de codificación de “fuente” muy útil en canales de comunicación unidireccionales y en comunicaciones *multicas*.

Los códigos *LT* (Luby, 2002) fueron los primeros en definirse como códigos de tipo *Fountain*, mientras que los códigos Raptor (Shokrollahi, 2006), son una evolución de los códigos *LT* que consiguen reducir su costo computacional y mejorar su desempeño.

La propuesta presentada a continuación se centra en el análisis de desempeño de los códigos de tipo *Fountain-Raptor*, en dos tipos de escenarios de comunicaciones, el primero en un canal de tipo *AWGN* y el segundo para un canal con desvanecimiento de tipo *RAYLEIGH*.

Cada análisis se realizará mediante la interpretación de probabilidad de error de bit (*BER*), en función de la relación señal a ruido (*SNR*).

¹ 4K.- Supone una mejora en la resolución y calidad de la imagen que es 4 veces más que la tecnología actual de alta definición (Full HD), alcanzado los 3840×2160 píxeles. Por lo cual se le ha otorgado el seudónimo de Ultra HD

1.2 Objetivos

1.2.1 General

Analizar el desempeño del código de tipo *Fountain-Raptor*, tanto para un canal de comunicaciones con desvanecimiento, así como en uno que posee solamente ruido gaussiano blanco aditivo, mediante el uso de fuentes de codificación y de decodificación, empleando el software pertinente de simulación.

1.2.2 Objetivos Específicos

- Realizar un estudio del estado del arte relacionado a métodos y aplicaciones de los códigos *Fountain*, en especial los códigos correctores de canal de tipo *Raptor*.
- Plantear un escenario de comunicaciones, simulando el envío de información a través de un canal de tipo *AWGN*, con variaciones en la cantidad de paquetes de bits transmitidos, y determinar su desempeño.
- Plantear un escenario de comunicaciones, simulando el envío de información a través de un canal de tipo *RAYLEIGH*, con variaciones en la cantidad de paquetes de bits de transmitidos, y determinar su desempeño.
- Comparar la tasa de error de bit (*BER*) con todas las variaciones de envío de paquetes de bits en ambos escenarios y determinar cuán deteriorada realmente, resulta la señal original en base al valor inicial y final de paquetes de bits de información transmitidos.

1.3 Metodología

Para realizar el análisis, se utilizaron fuentes de codificación y de decodificación mediante una herramienta de simulación matemática, desarrollo de algoritmos y visualización de datos en forma gráfica (*Matlab*®), a fin de tener una interpretación correcta de los datos receptados, en base a los siguientes criterios:

- Análisis de *BER* en función de la relación señal a ruido (*SNR*) en un canal *AWGN*.

- Comparación entre una codificación *LDPC*, vs una codificación de tipo *Fountain-Raptor*.
 - Realizar variaciones en el número de iteraciones del código *LDPC*, vs una codificación tipo *Fountain-Raptor*, con una iteración para la decodificación de su pre-codificador.
- Análisis de BER vs la relación señal a ruido (SNR) en un canal con desvanecimiento, de tipo *RAYLEIGH*.
 - Comparación entre una codificación *LDPC*, que es la que se empleó como el pre-codificador de la codificación de tipo *Raptor*, vs una codificación de tipo *Raptor*.
 - Realizar variaciones en el número de iteraciones del código *LDPC*, vs una codificación de tipo *Fountain-Raptor*, con una iteración para la decodificación de su pre-codificador.

Cada uno de los escenarios mencionados, brindó información valiosa e importante para determinar el desempeño de los códigos *Fountain-Raptor*.

Adicionalmente, cada escenario planteado se comparó entre sí, con la finalidad de determinar cómo se desempeña el codificador frente a las variaciones en el envío de paquetes de bits de información, tanto en el primer escenario, así como también comprobar como interviene el fenómeno del desvanecimiento, presente en el segundo escenario, en el rendimiento de un sistema de comunicaciones.

1.4 Resultados esperados

El análisis de probabilidad de error de bit en función de la relación señal a ruido en cada uno de los escenarios planteados, permitirá realizar una evaluación de desempeño de este tipo de codificadores, los mismos que son aplicados en sistemas de comunicaciones con tendencias a nuevos métodos de transmisión de información, sobre todo aquellos que se realizan a través de *Broadcasting*, o *multicast*. Por lo cual se espera que dicha codificación proporcione un rendimiento eficiente, frente al fenómeno de desvanecimiento en sistemas de comunicaciones móviles.

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

2.1 Principios de codificación

2.1.1. Códigos LDPC

Los códigos de Baja Densidad (*LDPC -Low-Density Parity Check*), también conocidos como códigos *Gallager* en honor a Robert G. Gallager, se presentaron a principios de los años sesenta (Gallager, 1960); debido a las limitaciones existentes en la capacidad de cómputo y de almacenamiento de los equipos disponibles en la época, y a la aparición de los códigos *RS*, fueron abandonados hasta que en la década de los noventa se “redescubrieron” sus bondades.

Su denominación emana del hecho de que la matriz de chequeo de paridad **H**, contiene pocos unos en comparación con el número de ceros (matriz dispersa).

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Este tipo de códigos se incluyen entre los códigos de bloque lineales²; básicamente, pueden representarse de dos modos: mediante su matriz de paridad (como todos los códigos bloque lineales), o bien mediante una representación gráfica, conocida como Grafo de *Tanner* (Tanner, 1981); el paso de un modo a otro es directo (Figura 1).

El Grafo de *Tanner* es un grafo bipartito, es decir, los nodos se separan en dos conjuntos diferentes y se conectan mediante arcos. Los dos tipos de nodos se conocen como nodos variables (*v-nodes*) y nodos de chequeo (*check-nodes*).

²Un Código de Bloque es Lineal cuando la combinación lineal de cualquier par de palabras de código dada, C_i y C_j , genera otra palabra código.

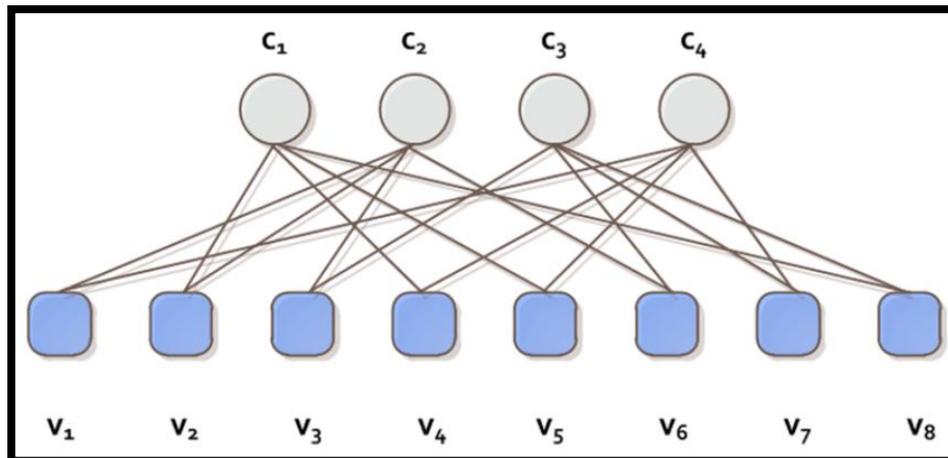


Figura 1. Gráfico de Tanner correspondiente a la matriz de paridad H

Si se define w_r como el número de unos por fila y w_c como el número de unos por columna, que posee la matriz H ; la misma que será considerada de baja densidad si $w_c \ll k$ y $w_r \ll n$, donde n y k son los valores que definen el código bloque (n, k) . De acuerdo a esto, la matriz de paridad debe ser grande para considerarse de baja densidad.

A partir de estos parámetros se pueden clasificar los *LDPC* como:

1. **LDPC regulares:** Si las filas y columnas de H tienen un peso uniforme, es decir, si w_c y $w_r = (n/k)$, son constantes para cada fila y columna (en la figura 5, $w_c=2$ y $w_r=4$).
2. **LDPC irregulares:** Si las filas y columnas de H no tienen un peso uniforme, es decir, si w_r y w_c no son constantes.

Los códigos *Digital Fountain* (DF) son un ejemplo de códigos *LDPC* irregulares. Fueron resultado de un largo proceso de estudio de los *LDPC* retomado por D. McKay, M. Luby, M. Mitzenmacher, A. Shokrollahi, J. Byers y D. Spielman, los cuales desarrollaron diferentes tipos de *LDPC* (Gallager, 1960) basándose en el esquema de *Digital Fountain*; *Luby Transform* (LT) y los códigos *Raptor*.

En (Mackay, 2003), se presentan algunos métodos para conseguir que los códigos *LDPC* trabajen de forma más eficiente. Con la intención de mejorar el rendimiento de estos códigos *LDPC* regulares, surge la idea de generar códigos *LDPC* (Luby, Mitzenmacher, Shokrollahi, & A, 2001). La irregularidad de los grafos es la razón vital por la que los códigos *Digital Fountain* (DF) son tan eficientes en la corrección de errores.

2.1.2. Códigos Digital Fountain

Los códigos *DF* son aquellos que permiten implementar un método óptimo de distribución de contenidos, en el que una fuente genera una cantidad potencialmente infinita de paquetes y los envía por la red. Los receptores tan sólo necesitan recoger cierta cantidad de esos paquetes para reconstruir la información. El término viene por su analogía con una fuente. Se puede imaginar una fuente de la que emanan continuamente gotas de agua, donde cualquier persona que tenga sed, no tiene más que tomar un vaso de agua de esta fuente, debido a que las gotas la han llenado y podrá saciar su sed.

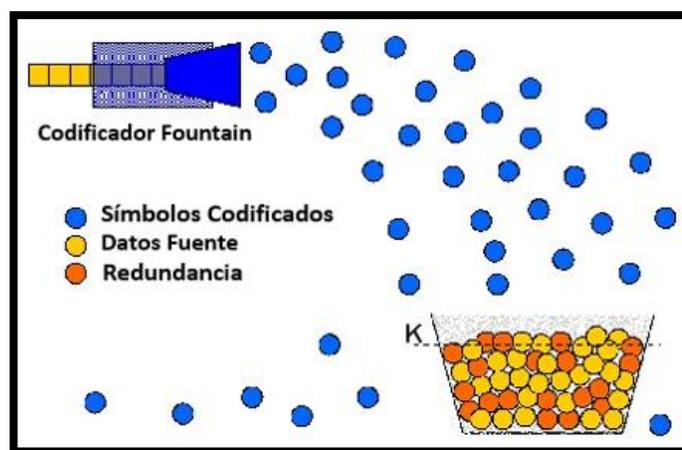


Figura 2. Representación de Códigos Fountain

Fuente: (Wesabi, Al-Arashi, & Alshadadi, 2018)

La Figura 2, representa la analogía de la codificación de tipo Fountain, con una fuente de agua, siendo la fuente (*Codificador Fountain*) el servidor, las gotas (*Símbolos Codificados*) los paquetes que se envían, y el balde el receptor.

Como se puede apreciar en dicha ilustración, el receptor posee elementos de redundancia para otorgar robustez a la decodificación (puntos de color naranja).

Finalmente, los Datos Fuente son los símbolos de información recuperada (puntos de color amarillo).

En una situación ideal, si el mensaje original está formado por k símbolos, sólo se necesitarán k paquetes para recuperar la información.

Codificador *Fountain*

- Produce una cantidad potencialmente infinita de gotas de agua, que representan paquetes codificados.

Decodificador:

- Recoge las gotas que emanan de la fuente hasta que el cubo esté lleno.
- El decodificador puede recuperar la información original, independientemente de las gotas que haya recogido.

Son clasificados como, códigos de corrección de borrado (*erasure-correcting codes*). Lo cual es útil para la transmisión de datos a través de redes sujetas a eliminación de paquetes de retransmisión que limitan el ancho de banda, por ejemplo aquellos sistemas donde se utiliza el protocolo *UDP (User Datagram Protocol)*. Normalmente esto se considera en aplicaciones de tiempo real como el *streaming* de video.

Debido a esta eliminación de paquetes, se clasifican a estos, como canales de borrado (*erasure channel*).

La metáfora, expresada en la Figura 2. Ayuda a comprender que los códigos *DF*, se tratan de un tipo de códigos *rateless*, en el sentido de que el número de paquetes que puede generar es potencialmente grande (al igual que la fuente gotas de agua), además se pueden enviar tantos paquetes como sean necesarios para recuperar los datos, independientemente de la estadística de probabilidad de borrado que tenga el canal.

Por ejemplo, en un canal de borrado de tipo binario (*BEC*), el cual posee un modelo de canal (Figura 3), donde se muestra un mapeo desde la entrada de canal *X* a la salida de canal *Y* (con el símbolo de borrado establecido por "?").

La probabilidad de borrado es p_e .

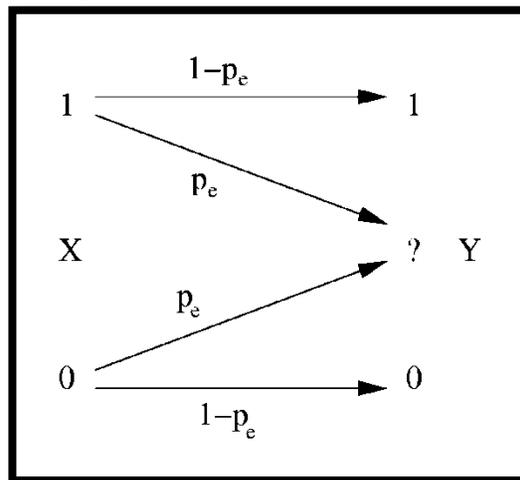


Figura 3. Canal de borrado de tipo Binario (BEC)

Fuente: (Mackay, 2003)

Los códigos de corrección de borrado, puede producir una cantidad potencialmente infinita de símbolos codificados, además solo necesitan recibir suficientes símbolos para decodificar con alta probabilidad la información de origen.

Los datos pueden reconstruirse a partir de un subconjunto cualquiera de K paquetes, siendo K algo mayor que k . Estos códigos asimismo, tienen una complejidad de codificación y decodificación muy pequeñas.

2.1.3. Transformada de Luby

Los códigos Transformada de Luby, o códigos *LT* fueron publicados por Luby en 2002 (Luby M. , 2002). Los códigos *LT*, son *rateless* y no se ha de fijar tasa de código de antemano.

Fueron los primeros códigos que cumplían completamente el concepto de *Digital Fountain* presentados en (Byers, Luby, Mitzenmacher, & Rege, 56-67).

Los códigos *LT* permiten generar símbolos codificados dinámicamente de forma ilimitada. Es una de las características que hace que el rendimiento de los *LT* sea independiente de la probabilidad de borrón del canal.

Como es bien sabido, los protocolos habituales para la transferencia de ficheros se basan en dividir la información en K paquetes que son enviados a su destino. Debe garantizarse la correcta recepción de todos los paquetes para dar por concluida la transmisión de forma satisfactoria. No obstante, pueden producirse errores en la transmisión, por lo que se emplean diferentes técnicas para tratar de superarlos. Las más sencillas implican el uso de un canal de retorno a través del cual el cliente, en caso de recepción errónea, informa al servidor, qué paquetes deben ser retransmitidos. Mediante la utilización de estos códigos se evita el empleo del canal de retorno.

2.1.3.1 Terminología

Antes de proceder a la descripción de los códigos *LT*, conviene fijar la notación y nomenclatura a utilizar. Tomando como base la Figura 4, los datos a enviar se consideran como una matriz de K x L bits que se dividen en K símbolos de información de L bits. Tras la codificación, a los símbolos creados se les denominará símbolos codificados (equivalentemente a paquetes codificados).

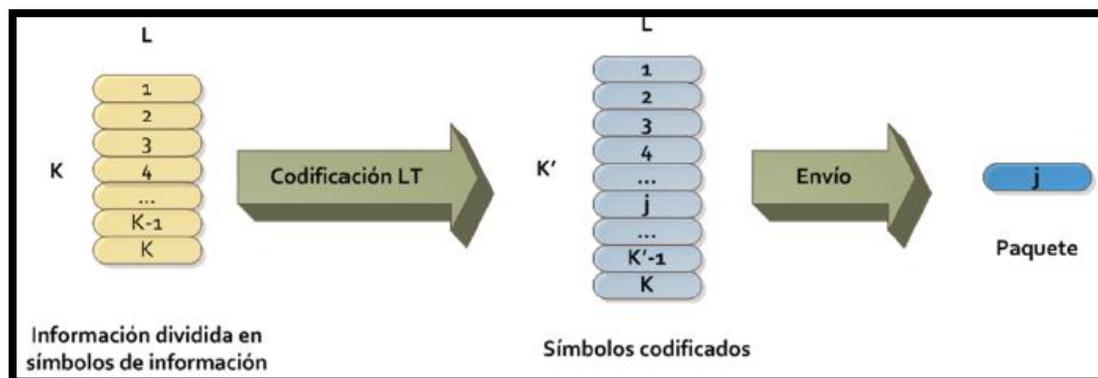


Figura 4. Terminología

2.1.3.2 Proceso de Codificación

El proceso de codificación de los códigos *LT* es simple. Al igual que los códigos *LDPC*, estos códigos también se pueden definir mediante un grafo bipartito o gráfico de *Tanner*.

Se considera un fichero de $K \times L$ bits, y se divide en K partes de la misma longitud, L bits; cada una de estas partes es un símbolo de información. A partir de esos K símbolos (de L bits cada uno), se crean los símbolos codificados sumando varios de ellos en base a una distribución de grado determinada (Figura 5). Se denomina grado al número de símbolos de información que se suman para obtener el símbolo codificado.

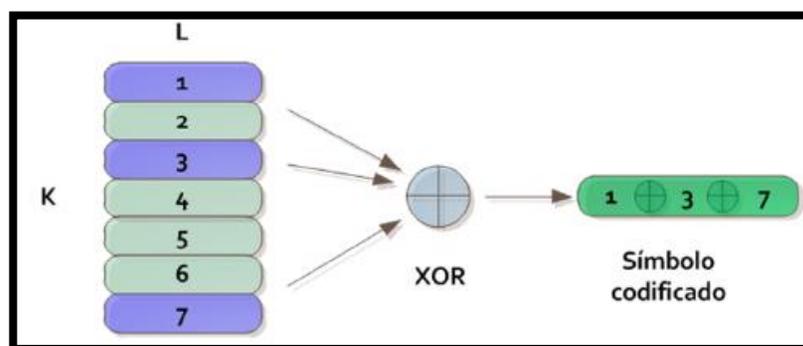


Figura 5. Ejemplo-Proceso de Codificación LT

La codificación puede representarse mediante un *grafo bipartito*, según se muestra en la Figura 6, en el que se pueden encontrar dos tipos de nodos: los símbolos de información o bloques, representados mediante círculos; y los símbolos codificados, representados mediante cuadrados. Los nodos de los diferentes grupos pueden estar conectados, pero no los del mismo grupo. Las líneas que unen los nodos se conocen como *arcos*, y dos nodos que están conectados por un arco, *vecinos*. El número de vecinos por los que está formado un determinado símbolo codificado se conoce como *grado*.

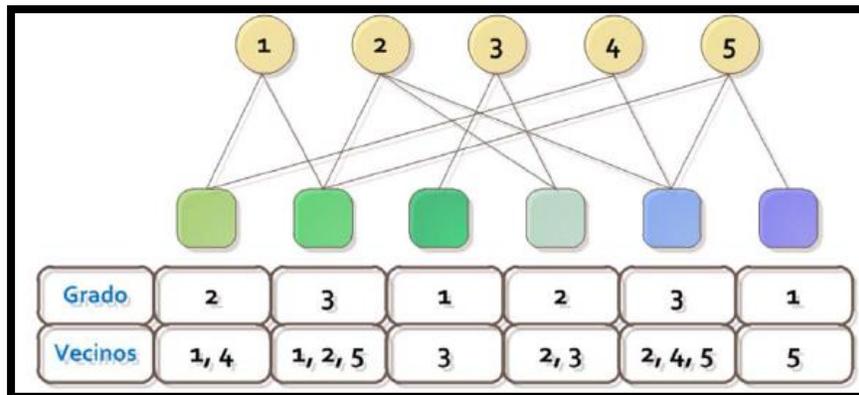


Figura 6. Grafo Bipartido

Esta información se puede enviar al receptor explícitamente, formando tanto el grado como los vecinos de cada símbolo parte de la cabecera *DF* (*Digital Fountain*), o bien utilizando un generador de números pseudo-aleatorio de forma que el receptor, a partir de la misma semilla que en el emisor, sea capaz de replicar tanto el grado como los vecinos de cada símbolo de información.

Los códigos *DF* definen una fuente continua de datos. Sin embargo, previo al inicio de la transmisión se podrá negociar entre el emisor y el receptor un valor máximo de símbolos codificados a recibir, superado el cual se interrumpirá la transmisión.

Igualmente se podrán intercambiar información que describa el mensaje original, de tal forma que la transmisión se dará por concluida cuando el receptor haya reconstruido el mensaje original.

2.1.3.3 Proceso de Decodificación

Cuando se transmite un símbolo codificado, se recibe correctamente o bien se pierde. El decodificador *LT* trata de recuperar los símbolos de información a partir de los símbolos codificados que recibe. Para decodificar, se asume que se conocen tanto el grado como los vecinos de cada símbolo codificado.

El proceso de decodificación se lleva a cabo de la siguiente manera:

1. Se recibe un paquete (palabra de código).
2. Se busca un símbolo codificado, e_m , que esté conectado a tan sólo un símbolo de información i_j . Si no es posible, volver al paso 1.
 - 2.1 Se asigna $i_j = e_m$
 - 2.2 Se calcula $e_l = e_l \oplus i_j \forall l \neq m$, tal que el nodo e_l esté conectado al nodo i_j .
 - 2.3 Se quitan los arcos que unen i_j con e_l .
3. Volver al paso 1.

El proceso de decodificación se detiene si no se recuperan símbolos de información porque no hay nodos de grado 1 en el grafo bipartito. Si la decodificación finaliza sin recuperar todos los símbolos de información, se dice que el decodificador ha fallado. De otro modo, la decodificación ha tenido éxito.

En las Figuras 7 y 8. Se desarrolla un ejemplo de la decodificación suponiendo los símbolos de un bit. Las flechas rojas representan la situación **2.1** del algoritmo mencionado, es decir, se ha encontrado un paquete de grado uno y se asigna su valor en el símbolo de información indicado. Las flechas azules representan la situación **2.2**, cuando se hace la XOR entre el símbolo de información recién obtenido y los símbolos codificados que le tienen como *vecino*.

Las flechas que desaparecen en cada iteración corresponden a la situación **2.3**, los símbolos de información se denotarán por $\{i_1, i_2, i_3, i_4, i_5\}$ y los codificados $\{e_1, e_2, e_3, e_4, e_5\}$.

En la iteración 1, se ve el grafo inicial. Seguido, se encuentra un paquete de grado 1, por lo que se copia al símbolo de información al que está conectado, es decir, se realiza la operación $i_3 = e_3$.

A continuación dado que i_3 es a su vez vecino de un símbolo codificado, e_4 , y con arreglo del algoritmo de decodificación se tiene $e_4 = e_4 \oplus i_3 = 1$ y el símbolo codificado no altera su valor; este mismo símbolo codificado disminuye entonces, su grado y queda conectado a tan sólo un símbolo de información, i_2 , por lo que se copia su valor a éste, es decir $i_2 = e_4$ (iteración 3).

En la cuarta iteración, habiendo conseguido el segundo símbolo de información, i_2 , y siendo éste vecino de dos símbolos codificados, e_2 y e_5 , siguiendo el proceso de decodificación definido se realizan dos operaciones: $e_2 = e_2 \oplus i_2 = 0$ y $e_5 = e_5 \oplus i_2 = 0$; ambos símbolos disminuyen su grado. El proceso de decodificación continúa hasta que se recupera el mensaje transmitido.

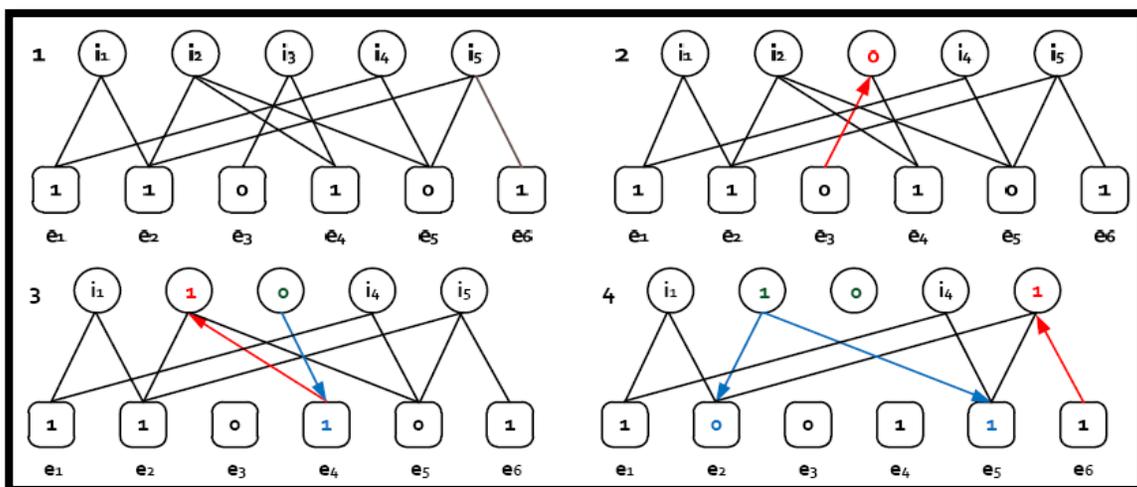


Figura 7. Ejemplo de Decodificación (4 Iteración)

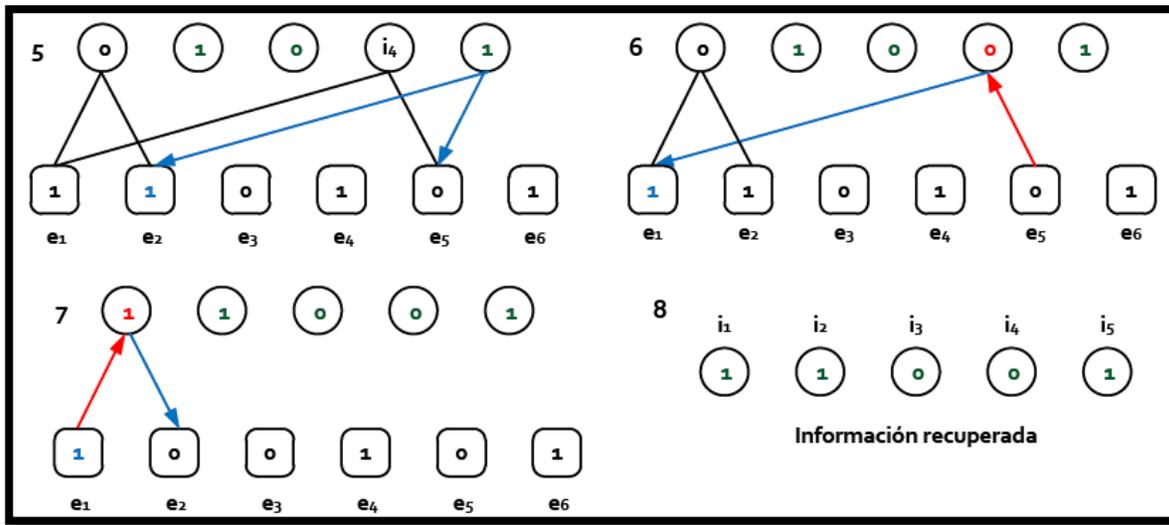


Figura 8. Ejemplo Codificación (Completa)

Para que todo el proceso de decodificación resulte efectivo, es necesario elegir una distribución de grado adecuada ya que su elección resulta *crítica* en el diseño de los códigos *LT*.

A continuación se analizan diversas distribuciones de grado y qué características deben incorporar para que su implementación real resulte eficiente.

2.1.4 Codificación raptor

Estos códigos se consideran una extensión de los *LT*, ya que el mensaje original se pre codifica, dando lugar a unos nodos intermedios en el gráfico de *Tanner*; estos nodos intermedios son a los que se les aplica entonces la codificación *LT*.

El proceso se puede observar en la Figura 9.

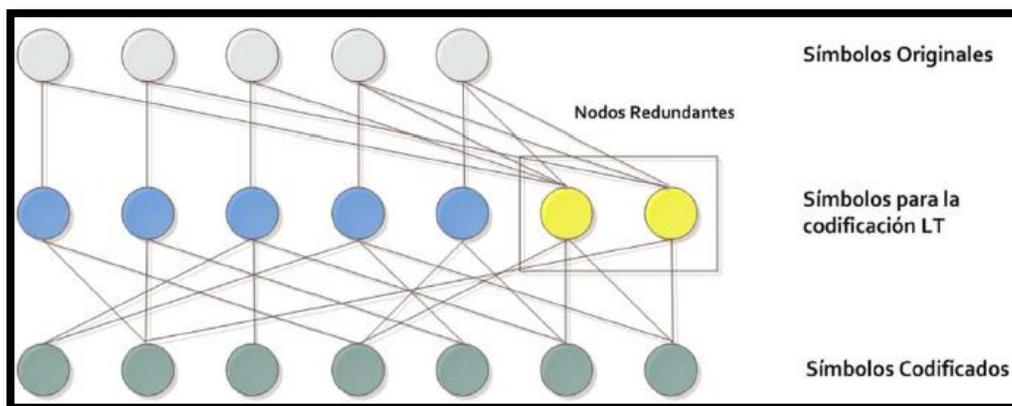


Figura 9. Grafo-Código Raptor

Así definidos, se pueden considerar los *LT* como un tipo de código *Raptor* sin pre codificación, la pre codificación se puede realizar también en varios pasos, por ejemplo, aplicando primero un código *Hamming*, a continuación, un *LDPC* y entonces codificar los símbolos resultantes mediante *LT*.

También existen una subclase de códigos *Raptor* sin codificación *LT*, denominados *Pre-Coding-Only*, o códigos *Raptor PCO* (Shokrollahi A. , 2006).

Usando un código de tipo *FEC* como pre código, se evita la necesidad de cubrir todos los símbolos de *LT*, sólo se necesita recuperar una fracción constante, ya que el mensaje original se puede recuperar gracias a la capacidad de corrección de borrados del pre código.

La decodificación se realiza usando primero el decodificador *LT* para obtener los nodos intermedios, y a continuación se obtienen los símbolos originales aplicando el algoritmo de decodificación del pre-código.

2.2 Descripción del modelo del canal

2.2.1 Canal AWGN (Ruido Gaussiano Blanco Aditivo)

Consiste en un proceso aleatorio en el que cada muestra, es una variable aleatoria Gaussiana de media cero y varianza σ^2 . De esta forma la ecuación (1), describe la salida $y(t)$ del canal AWGN.

$$y(t) = x(t) + n$$

Ecuación 1.

Donde $x(t)$ es la señal de entrada y n es la densidad espectral uniforme. Este modelo de canal es muy usado en análisis de sistemas de comunicaciones.

Desde el punto de vista estadístico, se considera al ruido como una variable aleatoria y por tanto se describe a partir de su función de densidad de probabilidad.

De forma general, la variable aleatoria *Gaussiana* X es definida en función de dos parámetros: media m , y desviación estándar σ^2 . Se la presenta de la forma $X = (m, \sigma^2)$.

La *PDF* (*Probability Density Function*) de esta variable aleatoria está dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Ecuación 2.

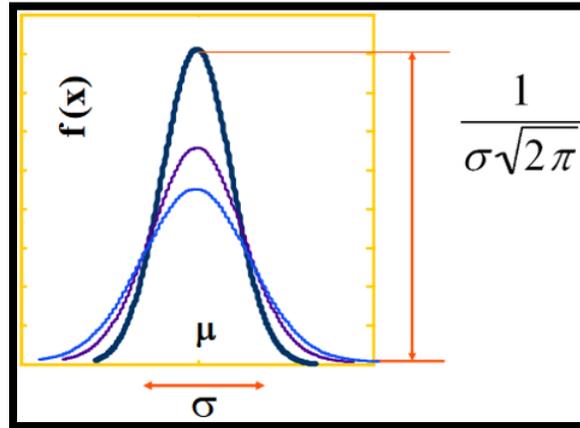


Figura 10. PDF de una variable aleatoria Gaussiana

2.2.2 Relación Señal a Ruido (SNR)

La Relación Señal a Ruido (SNR – *Signal-to-Noise Ratio*) es una proporción entre potencia de la señal de interés y la potencia del ruido interferente.

Este coeficiente es expresado en decibeles (dB) y se lo usa principalmente para graficar la probabilidad de error en función de esta.

La SNR está dada por la siguiente ecuación:

$$SNR(dB) = 10 \log\left(\frac{P_s}{P_{No}}\right)$$

Ecuación 3.

donde:

P_s Potencia de la Señal (W)
 P_{No} Potencia del Ruido (W)

La relación señal a ruido, es un parámetro de entrada que se puede expresar en términos de $\frac{E_b}{N_0}$, como se muestra en la ecuación (4).

$$\frac{E_b}{N_0} = 10^{\frac{SNR(dB)}{10}}$$

Ecuación 4.

La energía por bit E_b , es la energía total de la señal, dividida para el número de bits contenidos en la señal. Por lo que se puede expresar a E_b , a través de la ecuación (5) como la potencia media de la señal multiplicada por la duración de un bit.

$$E_b = \frac{1}{N_{muestras} \cdot f_{bit}} \sum_{n=1}^N x^2(n)$$

Ecuación 5.

Donde $N_{muestras}$ es el número total de muestras en la señal, y f_{bit} es la frecuencia de bit. Además, se puede calcular a partir de la ecuación (6), la densidad espectral de potencia del ruido N_0 , conociendo previamente los valores de $\frac{E_b}{N_0}$ y E_b

$$N_0 = \frac{E_b}{E_b/N_0}$$

Ecuación 6.

La densidad de potencia espectral del ruido unilateral muestra la cantidad de potencia de ruido presente en un 1 [Hz] del ancho de banda de la señal. Ahora para encontrar la varianza o potencia media del ruido se debe conocer el ancho de banda del ruido. Para una señal real x , muestreada a f_s [Hz], el ancho de banda del ruido será la mitad de la tasa de muestreo, por lo tanto, la varianza se encuentra multiplicando la densidad de potencia espectral del ruido por el ancho de banda del ruido, como en la ecuación (7).

$$\sigma^2 = \frac{N_0 \cdot f_s}{2}$$

Ecuación 7.

Donde σ^2 es la varianza de ruido en [W]. Dado que el ruido tiene una media de cero, su potencia y su varianza son idénticas.

2.2.3 Canal con Desvanecimiento

Para propósitos de procesamiento de la señal recibida y para su análisis, los canales móviles con desvanecimiento se establecen a través de diferentes modelos de propagación de pequeña escala (Ramos, 2018), como se puede observar en la Figura 11.

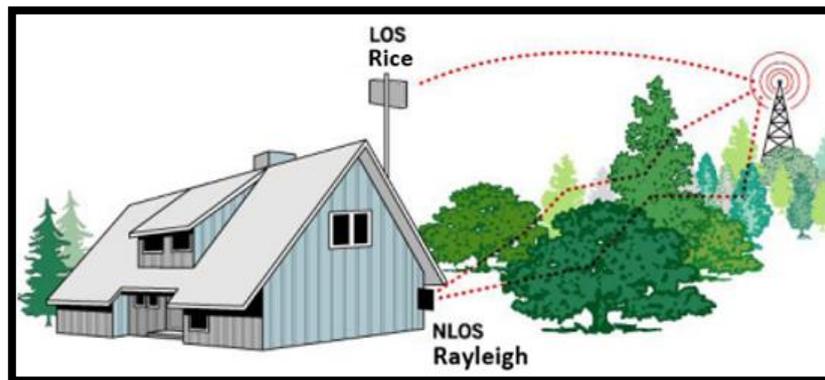


Figura 11. Escenario de comunicaciones con desvanecimiento

2.2.4 Canal de tipo Rayleigh

Este canal sigue el comportamiento de la **Distribución Rayleigh**, la cual se produce cuando la línea de visión directa entre emisor y receptor se ve interrumpida por obstáculos o el nivel de potencia de la señal, es inferior al nivel de ruido.

La función de distribución de probabilidad *Rayleigh* proviene de dos variables aleatorias gaussianas X_1 y X_2 , con media cero ($m = 0$) y varianza σ^2 . En la ecuación (8), se considera una variable aleatoria *Rayleigh* X , tal que:

$$X = \sqrt{X_1^2 + X_2^2}$$

Ecuación 8.

Entonces la función de distribución de probabilidad (*PDF*) de X es una distribución de tipo *Rayleigh* y viene dado por la ecuación (9):

$$p(x) = \begin{cases} \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, & x > 0 \\ 0, & \text{para el resto} \end{cases}$$

Ecuación 9.

Donde x es la envolvente de la señal recibida, σ es la desviación estándar de las variables aleatorias gaussianas y σ^2 representa la potencia promedio de la señal recibida antes de la detección de la envolvente.

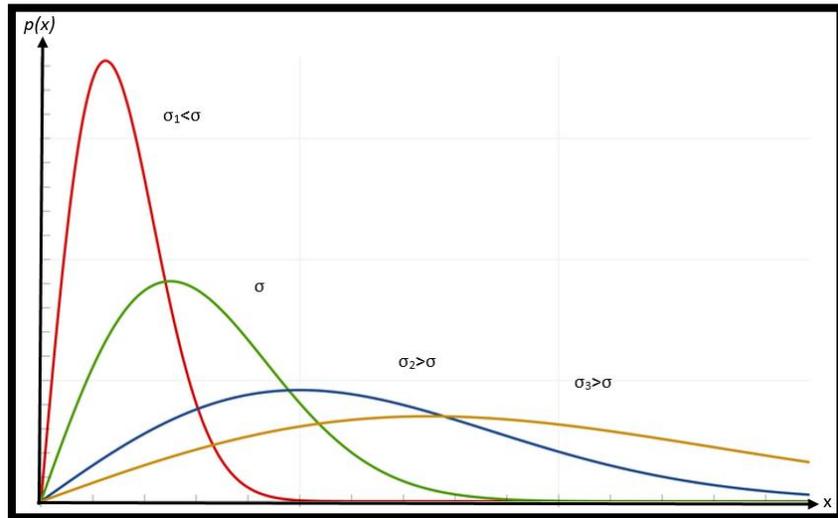


Figura 12. PDF de una v.a. Rayleigh para distintos valores de σ

En la figura 13, se puede observar la comparación de probabilidad de error de un canal AWGN, con otro con desvanecimiento de tipo Rayleigh, utilizando modulación BPSK.

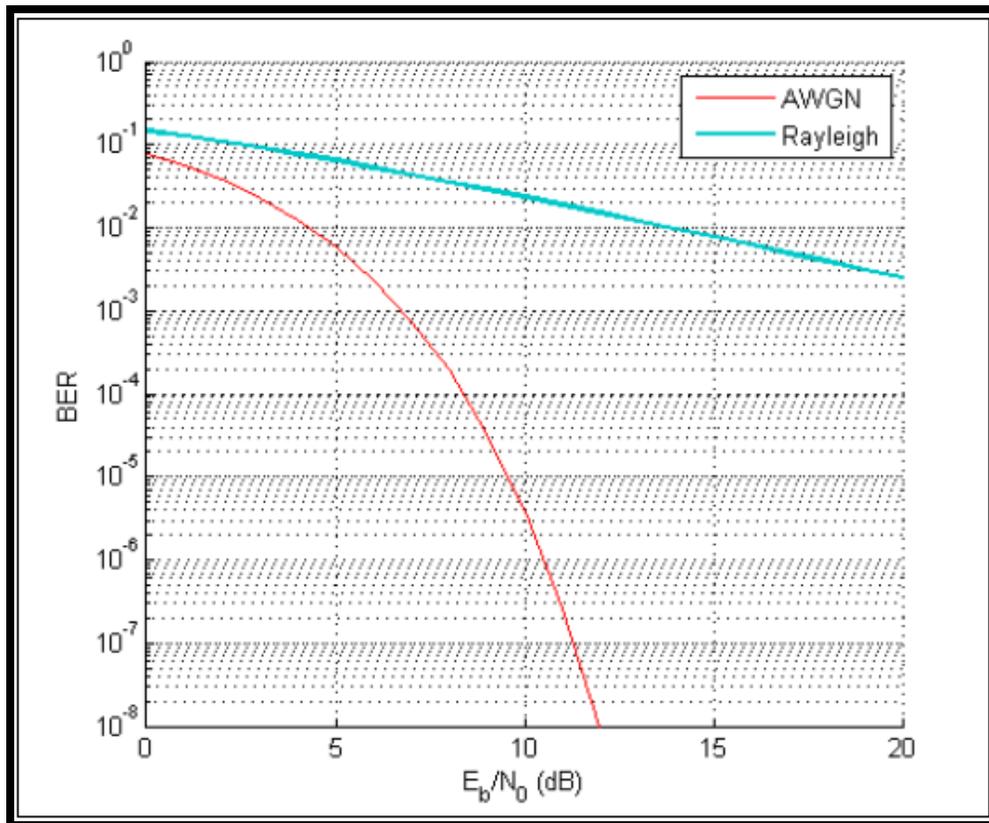


Figura 13. Comparación de probabilidad de error de un canal AWGN, con otro con desvanecimiento de tipo Rayleigh, utilizando modulación BPSK

Con los antecedentes contribuidos en esta sección, en base a la descripción del modelamiento del canal de transmisión. Se realizó el diseño y simulación del canal AWGN y del canal con desvanecimiento de tipo *Rayleigh* mediante la utilización de funciones proporcionadas por el entorno Matlab®.

Fueron estos dos tipos de canales seleccionados como los escenarios de comunicaciones, para la transmisión de un conjunto aleatorio de bits, que representan datos de información.

CAPÍTULO III

3. DEFINICIÓN DEL ESCENARIO

3.1 Estructura del sistema de comunicaciones

3.1.1 Diagrama de bloques

La comunicación es la transferencia de información a través de un canal considerado como medio de transmisión, desde un punto denominado como elemento emisor, hasta otro punto llamado receptor.

Un sistema de comunicaciones es un conjunto de subsistemas y mecanismos que permiten la comunicación entre el emisor y el receptor, ya sea utilizando como canal, un medio guiado (cables de cobre) o no guiado (aire), para establecer dicha comunicación. El objetivo principal en un sistema de comunicaciones, es que exista legibilidad en la información que se transmite en el mismo, por ejemplo en una llamada telefónica, los aspectos donde se debe ser más minucioso es, tanto en la claridad del mensaje de voz, así como los tiempos máximos de retardo (considerados en el orden de los mili segundos), para que el mensaje se establezca entre emisor y receptor y de esta forma garantizar calidad en la comunicación.

Es importante recalcar que detrás de todo este proceso existen mecanismos complementarios que ayudan a garantizar calidad en los sistemas de comunicaciones, sobre todo en sistemas de tipo digital. Uno de los mecanismos más importantes es la codificación de canal, ya que esta permite codificar el mensaje de una forma específica (dependiendo la codificación utilizada), permitiéndole al receptor recuperar el mensaje, tras corregir cierta cantidad de errores (depende de la robustez del codificador).

Cabe mencionar que se debe considerar el canal de comunicaciones, por donde se envía la información (canales fijos, móviles), ya que existen características propias en los mismos.

Por último pero no menos importante, está el tipo de modulación utilizada para la transmisión, ya que este proceso es el que “prepara” a la señal para viajar a través del canal de comunicaciones. De igual manera elegir el tipo de modulación depende del canal de comunicaciones.

Para cada uno de los escenarios de prueba se consideró los siguientes aspectos:

- **Fuente de Información:** Se consideró una secuencia aleatoria de bits.
- **Codificación de Canal:** Codificación de tipo *Raptor*.
- **Pre codificación Raptor:** Código *LDPC*.
- **Modulación:** *BPSK*

Tras cada una de estos aspectos, se presenta un diagrama de bloques de la estructura del sistema de comunicaciones analizado para cada uno de los escenarios de pruebas.

3.1.1.1 Diagrama de bloques en un canal de tipo AWGN

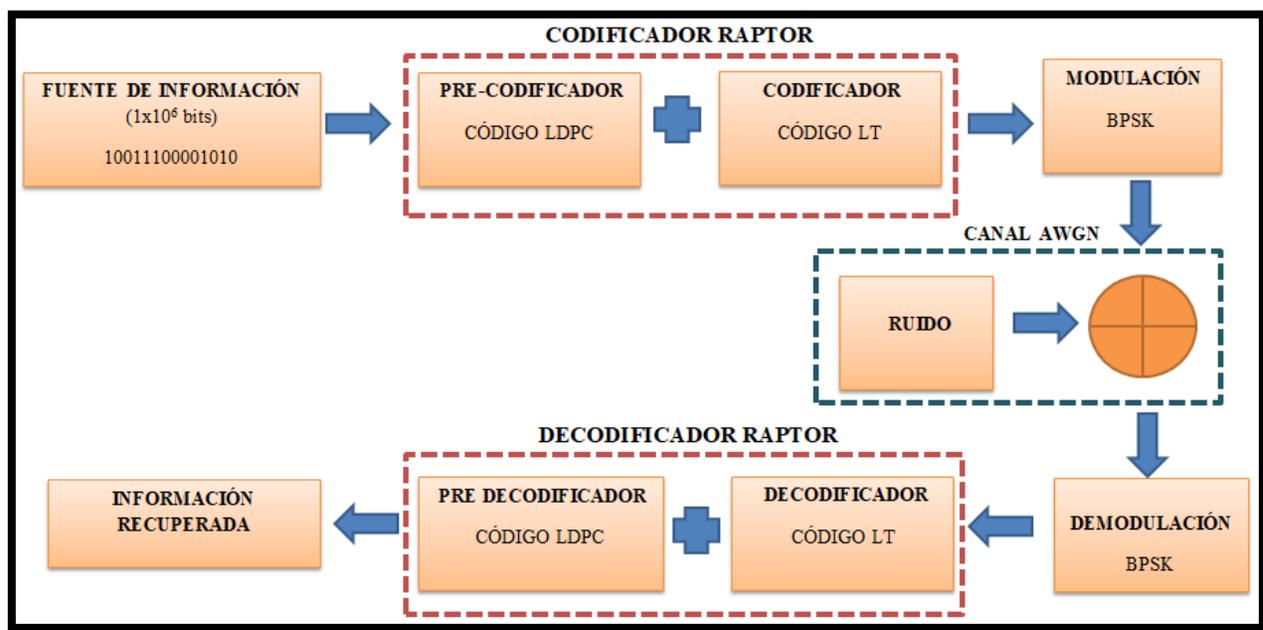


Figura 14. Diagrama de Bloques-Primer escenario de comunicaciones (Canal AWGN)

Como se aprecia en la Figura número 14, la codificación de canal de tipo *Raptor* se la realiza en dos partes, inicialmente se toma los bits de información y se los codifica mediante un código *LDPC*, estos nuevos bits codificados se los ingresa a una segunda codificación, la cual se realiza mediante el uso de los códigos *Fountain-LT*; el proceso de la “doble” codificación se lo conoce como codificación de tipo *Raptor*.

El Ruido *AWGN* agregado al canal se lo realizó mediante la función “*randn*”, propia del entorno de Matlab®, tomando en cuenta que el tamaño de dicho ruido, debía ser el mismo que el de la señal modulada, para proceder a la suma de ambos.

3.1.1.2 Diagrama de bloques en un canal con desvanecimiento de tipo Rayleigh.

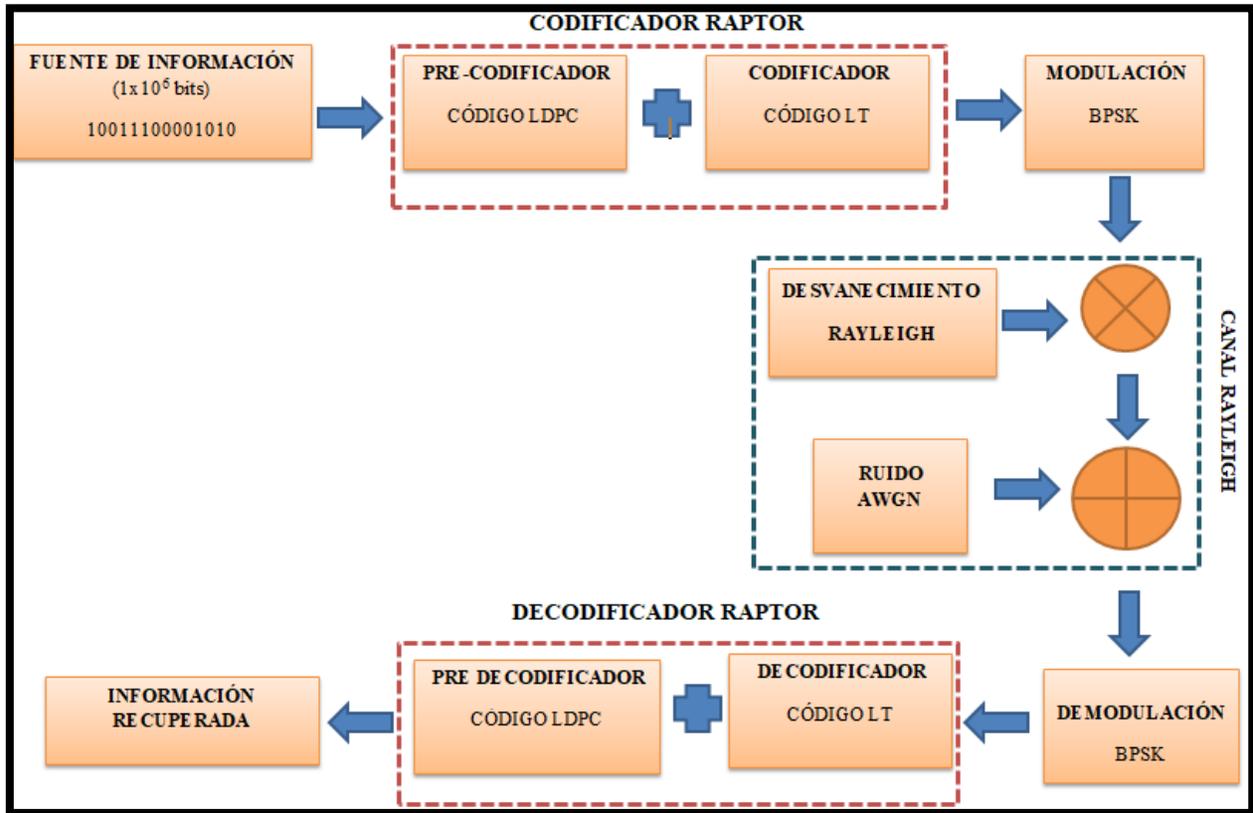


Figura 15. Diagrama de Bloques-Segundo escenario de comunicaciones (Canal RAYLEIGH)

Al igual que en el primer escenario, se establece la codificación de tipo *Raptor*, mediante el uso del pre-codificador de tipo *LDPC*, seguido de la codificación *LT*, para la transmisión de bits de información y la respectiva decodificación para la recepción de los mismos.

El modelamiento de canal de tipo *RAYLEIGH*, se lo estableció en dos partes, inicialmente se realiza la multiplicación de la señal modulada por el desvanecimiento de tipo *Rayleigh*, seguido se añade Ruido *Gaussiano* para simular el fenómeno de dicho canal.

3.1.1.3 Diagrama de bloques-pruebas principales

Tras cada uno de los escenarios planteados de forma general, se consideró realizar variaciones en las pruebas principales que sirvan como complemento, a fin de tener una gama más amplia de

resultados, donde se pueda evidenciar el contraste de dichas variaciones en cada escenario, por lo cual se decidió establecer las siguientes pruebas, descritas con sus respectivos diagramas de bloques.

3.1.1.3.1 Diagrama de bloques-codificación Raptor vs LDPC (Canal AWGN)

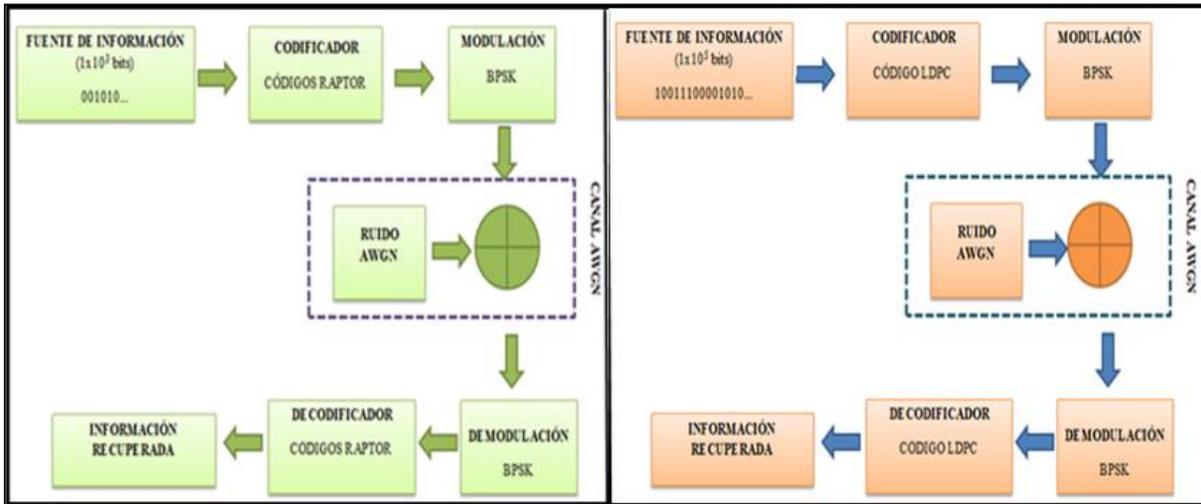


Figura 16. Diagrama de Bloques-Codificación Raptor vs LDPC (Canal AWGN).

La Figura número 16 muestra el diagrama de bloques del primer escenario, en el cuál se realizó variaciones del número de bits transmitidos, con mil, diez mil, cien mil y un millón de bits. Dicha prueba fue empleada con el afán de mostrar que el código de tipo Raptor es una evolución de la codificación LDPC, debido a la arquitectura que tiene el mismo. Por lo cual se estimó que el desempeño del código Raptor sea superior.

De esta manera se pudo evidenciar gráficamente, como este tipo de codificación tiene cierta dependencia en cuanto se refiere a la cantidad de datos de información, para evaluación de su desempeño.

3.1.1.3.2 Diagrama de Bloques-Codificación Raptor vs LDPC (Canal Rayleigh)

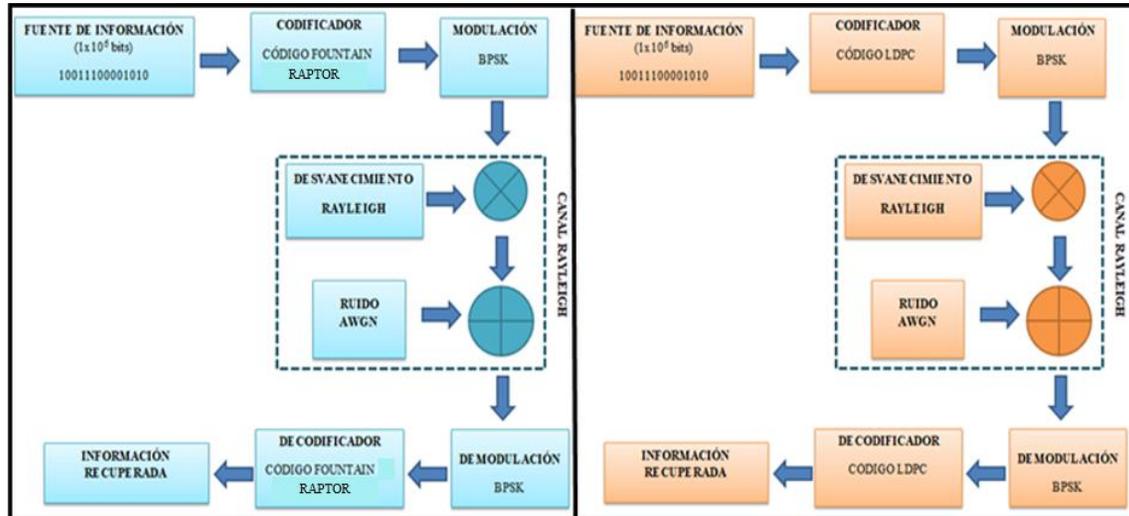


Figura 17. Diagrama de Bloques-Codificación LT vs LDPC (Canal RAYLEIGH).

La Figura número 17 muestra el diagrama de bloques del segundo escenario establecido. Dicha prueba se realizó de igual manera con la variación en la cantidad de bits transmitidos. En este caso la finalidad no fue solamente observar el desempeño del codificador Raptor ante el LDPC, sino que también permitió evidenciar, como el factor del desvanecimiento afecta al desempeño de dichas codificaciones.

Es importante mencionar que cada una de las afirmaciones presentadas en esta sección, se corroboraron en el análisis de resultados de este trabajo de investigación.

3.2 Diseño de codificación y decodificación

3.2.1 Fountain-LT

3.2.1.1 Codificación

El proceso utilizado tanto para la codificación, así como la decodificación es expresado mediante las ecuaciones, algoritmos y ejemplos mostrados a continuación.

Inicialmente para detallar dichos procesos, se definen las siguientes variables:

- Para símbolos de información s_1, s_2, \dots, s_k
- Para símbolos codificados t_1, t_2, \dots, t_n

- Para la matriz generadora aleatoria G

La ecuación número 10 describe matemáticamente la codificación *Fountain-LT*.

$$t_n = \sum_{k=1}^K s_k G_{kn}$$

Ecuación 10.

Ejemplo:

$$s = [1 \ 0 \ 1]$$

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots \\ 0 & 1 & 1 & 1 & \dots \\ 0 & 1 & 0 & 0 & \dots \end{bmatrix} \Rightarrow t = s \cdot G \Rightarrow t = [1 \ 1 \ 1 \ 0 \ \dots]$$

La palabra de código es expresada mediante un grafo de *Tanner*, para mayor entendimiento del proceso de decodificación.

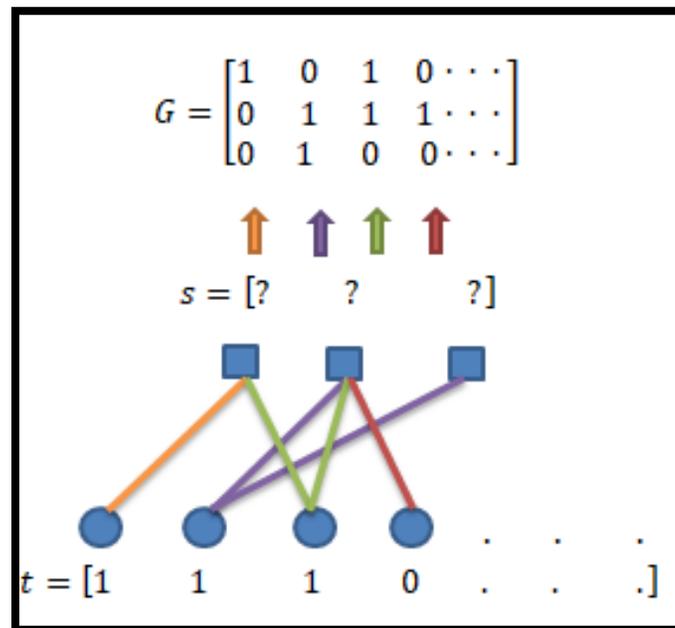


Figura 18. Representación de la matriz generadora de código (G), en un grafo de Tanner.

Cada color de flecha (↗), representa la descomposición de la matriz generadora de código, en el grafo de *Tanner*, donde los círculos (●), representan los nodos variables, es decir los símbolos codificados, que cambiará la cantidad de los mismos según las dimensiones de la matriz generadora (*G*).

3.2.1.1 Decodificación

La ecuación número 11 describe matemáticamente la decodificación *Fountain-LT*.

$$s_k = \sum_{n=1}^K t_n G^{-1}_{nk}$$

Ecuación 11.

$$t = [1 \ 1 \ 1 \ 0]$$

$$G^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots \\ 0 & 1 & 1 & 1 & \dots \\ 0 & 1 & 0 & 0 & \dots \end{bmatrix}^{-1}$$

El grafo de *Tanner* Representado en la Figura 19, es el establecido ya, en la Figura 18, con la diferencia de que para este caso, dicha ilustración permitirá evidenciar de manera gráfica la **Decodificación *Fountain***, paso a paso.

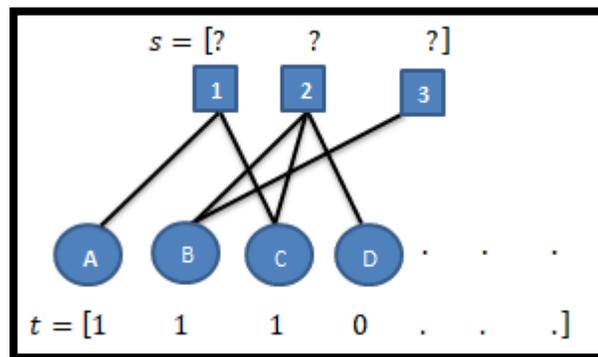


Figura 19. Grafo de Tanner inicial para Decodificación Fountain

De igual manera que en la codificación, para describir el proceso de decodificación se definen los elementos del grafo de *Tanner* y que es lo que representa cada uno de ellos:

-  Símbolos Codificados-palabra de código (t_n).
- | Líneas de unión (arcos)
-  Símbolos de información o bloques.

Es importante mencionar elementos como los *vecinos* y el *grado* de los elementos, definiéndose:

- *vecinos* (v): 2 o más nodos conectados por medio de 2 o más arcos.
- *grado* (g): El número de vecinos por los que está formado un determinado símbolo codificado.

La tabla Número 1, muestra el resumen representativo de valores relacionado a los elementos detallados del grafo de Tanner de la Figura número 19.

Tabla 1

Resumen representativo de valores relacionado a los elementos detallados del grafo de Tanner

t	A	B	C	D
g	1	2	2	1
v	1	(2,3)	(1,2)	2

Tras establecer cada uno de estos elementos, se detalla el proceso de decodificación:

- En primer lugar el algoritmo elige un nodo variable que posea grado igual a 1, es decir ($g=1$). Cabe recalcar que esta elección se lo hace mediante un barrido de izquierda a derecha, de acuerdo al grafo de *Tanner* y con respecto a la matriz **G**, se lo hace desde la primera columna de la izquierda hacia la derecha de igual manera, para esta ocasión se inicia con el nodo **A** (Figura 20).

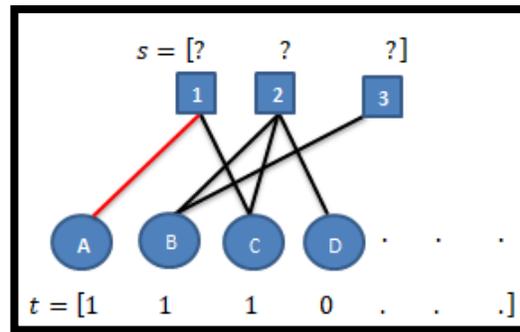


Figura 20. Primer paso del proceso de decodificación

- Dicho valor se asigna al primer bit de información, en este caso el nodo variable **A** tiene un valor codificado de **1**, este pasa directamente al primer elemento de s (Figura 21).

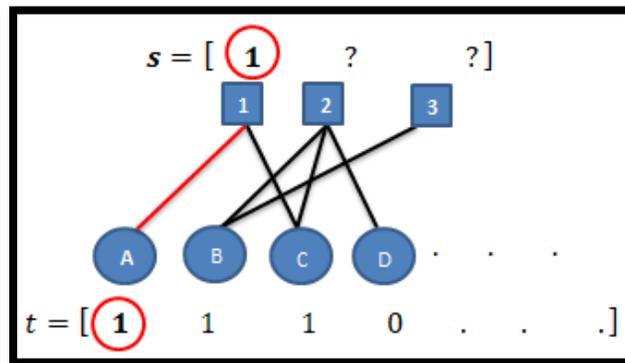


Figura 21. Segundo paso del proceso de decodificación

Seguido se realiza la verificación de los vecinos relacionados al nodo variable escogido inicialmente, en este caso el “nodo-vecino” involucrado es el número uno. Tras ubicarse en este punto se procede a realizar una operación XOR entre el valor actual del “nodo-vecino” (para este ejemplo es **1**), con el otro extremo al que se conecta dicho nodo vecino, en este caso, se conecta al nodo variable **C**, el cual tiene un valor inicial de **1**; por lo tanto dicha operación dará como resultado **0** ($1 \oplus 1 = 0$), una vez realizada esta operación, el resultado obtenido será reemplazado en el mismo nodo variable, que para el ejemplo es el nodo variable **C**; tomando **C**, ahora el valor de **0** (Figura 22).

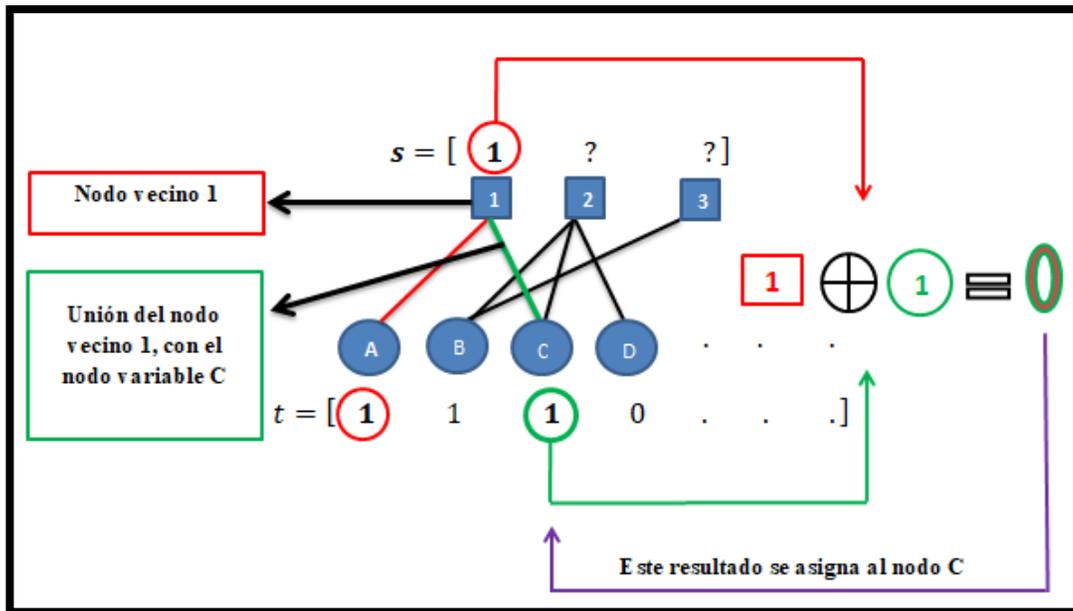


Figura 22. Tercer paso del proceso de decodificación

Tras cada dicho proceso, se eliminan las líneas de unión (grado) relacionadas al tercer paso de decodificación, obteniéndose el grafo de *Tanner* ilustrado en la Figura 23.

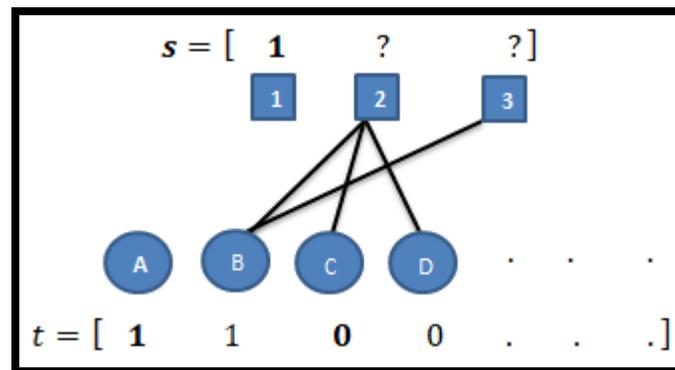


Figura 23. Cuarto paso del proceso de decodificación

El proceso de decodificación continúa, repitiendo el paso número 1, por lo cual se procede con el barrido en el gráfico de *Tanner* para encontrar “otro nodo variable” con grado 1. La Figura 23 permite identificar este elemento mencionado, siendo el mismo nodo variable **C**.

Entonces el valor de **C**, pasará directamente al “nodo vecino” número 2, y las operaciones XOR, se las realizará entre dicho “nodo vecino” y los nodos variables tanto **B**, así como **D**. Finalmente dichos resultados de las sumas exclusivas, se los asigna a los mismos nodos variables, que intervienen en la operación lógica. Es decir:

- Nodo vecino 2= 0
- Nodo vecino 2 (0) \oplus Nodo variable B (1) =1
- Nodo vecino 2 (0) \oplus Nodo variable D (0) =0
- Nodo variable B= 1
- Nodo variable D= 0

La Figura 24, muestra el resultado del proceso descrito.

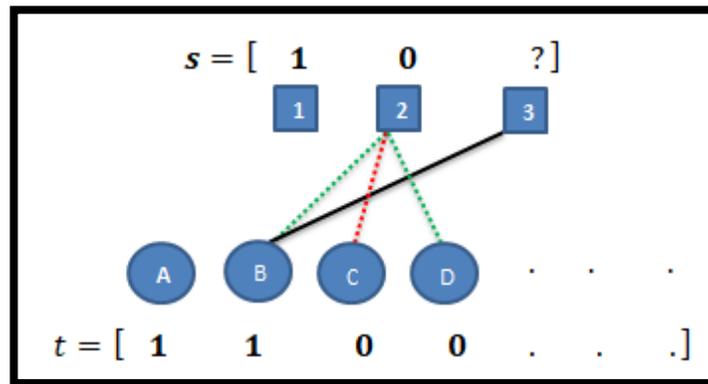


Figura 24. Quinto paso del proceso de Decodificación

Para culminar con la decodificación del mensaje, se realiza nuevamente el paso número 1. Obteniéndose satisfactoriamente el mensaje enviado s (Figura 25).

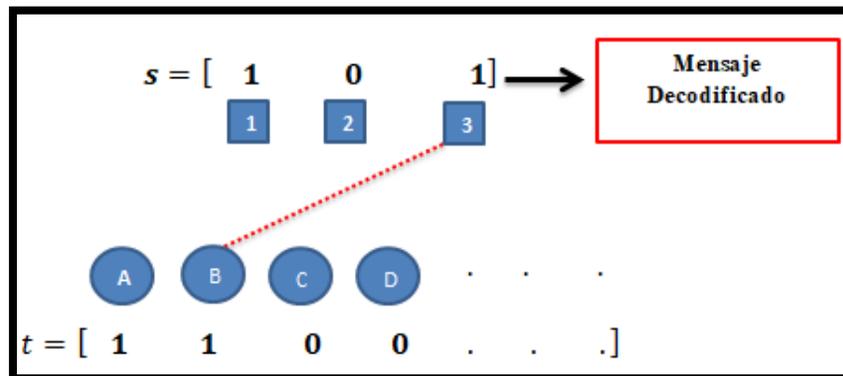


Figura 25. Final de la Decodificación (Mensaje Recuperado)

Es importante mencionar que este tipo de decodificación funciona siempre y cuando existan nodos variables de grado 1 inicialmente, y en cada paso de decodificación parcial. Caso contrario falla la decodificación, debido a esto es comprensible que con una mayor cantidad de bits codificados, es decir cuanto mayor es el mensaje, mayor es la probabilidad de tener nodos variables de grado 1.

Por ejemplo:

La Figura 26 se compone de un mosaico, en donde se visualiza una codificación *Fountain-LT* en un grafo de *Tanner*, dicho grafo resalta las siguientes propiedades

Tabla 2

Resumen representativo de valores relacionado a los elementos detallados del grafo de Tanner de la Figura 25.

g	2	2	1	2	1	3
v	(1,3)	(1,2)	(3)	(4,6)	(5)	(1,4,6)

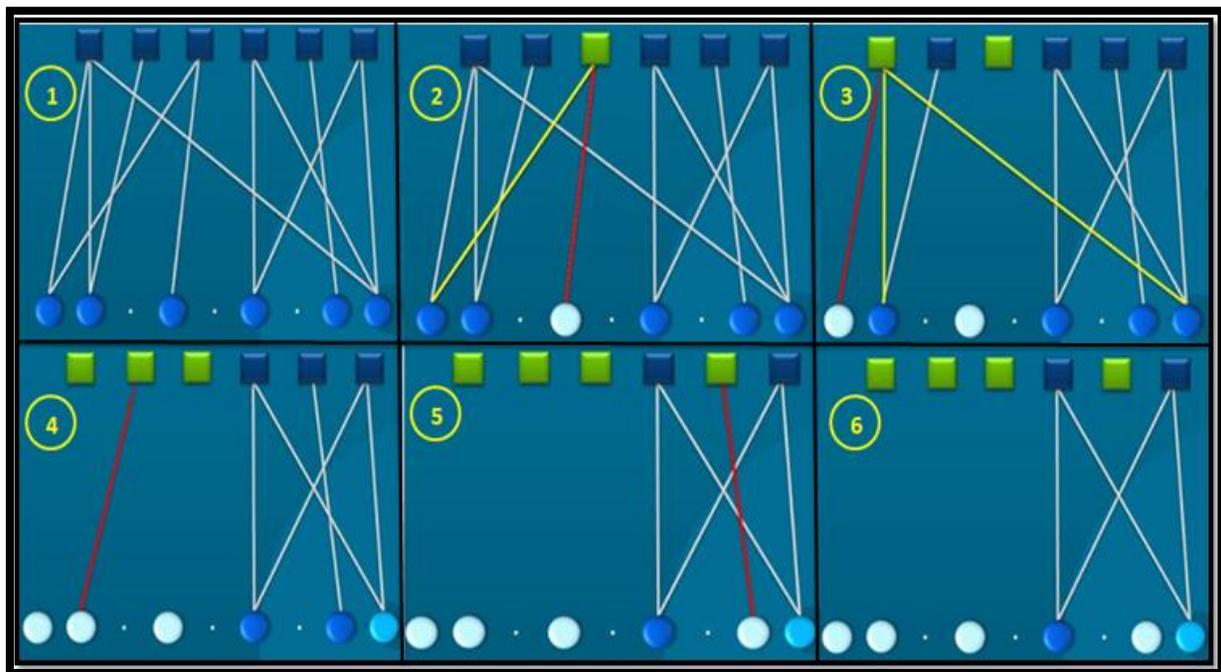


Figura 26. Mosaico de Decodificación Fountain-LT Fallida.

Como se aprecia en el recuadro número **6** del mosaico de la Figura 26, la decodificación *Fountain-LT*, presenta una situación donde no existen más elementos (nodos variables) de grado 1, por lo cual se concluye que la decodificación ha fallado.

Por otro lado la Figura 27, muestra un mosaico similar, con la diferencia que en este se ha aumentado un nodo variable (**recuadro 1, mosaico**) más al gráfico de *Tanner*, en otras palabras hay más símbolos codificados, debido a esto, se “incrementó” la probabilidad de que existan mayor número de nodos variables con grado 1 ($g=1$).

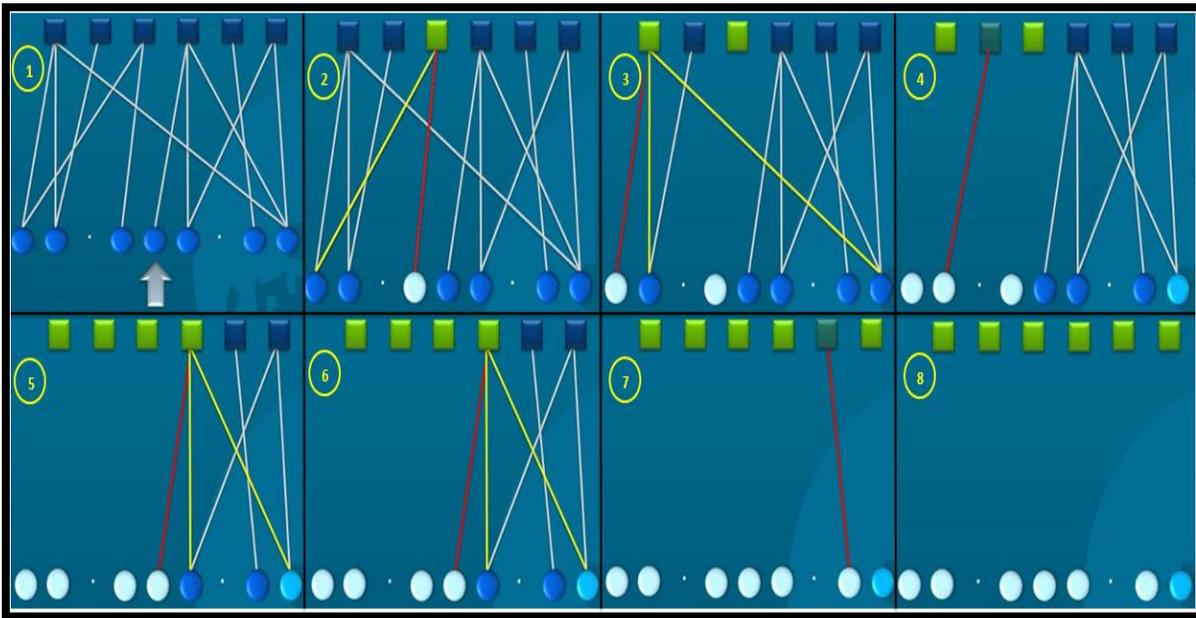


Figura 27. Mosaico de Decodificación Fountain-LT Exitosa

Como se aprecia en el recuadro número **8** del mosaico de la Figura 27, la decodificación *Fountain-LT*, presenta una situación donde la codificación fue exitosa.

3.2.2 Proceso de codificación y decodificación Fountain-Raptor

Debido al inconveniente y limitación que tiene el proceso de decodificación de tipo *Fountain-LT*, se establece la codificación de tipo Fountain-Raptor, la misma que solventa este problema de un modo sencillo, el cual es utilizar un pre-codificador para los bits de información, antes de emplear la codificación de tipo *Fountain-LT*. En este caso se utiliza como pre-codificador, la codificación *LDPC*.

3.2.2.1 Codificación

La Figura 28, muestra un grafo de *Tanner* de la codificación de tipo *Raptor*.

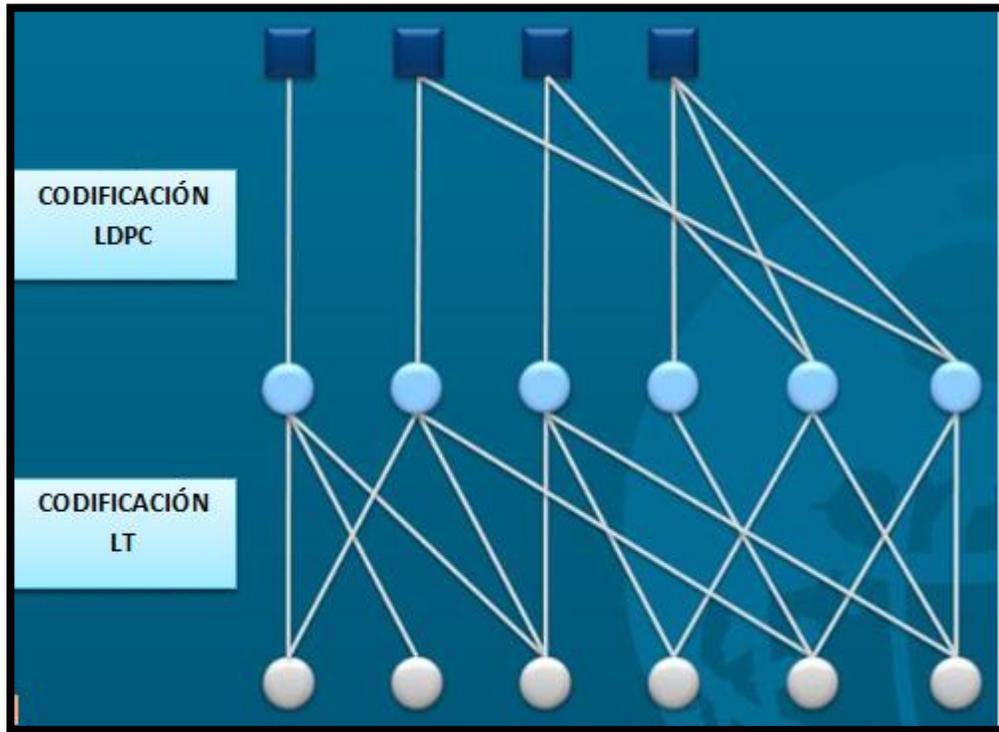


Figura 28. Grafo de Tanner (Codificación Raptor)

3.2.2.2 Decodificación

En este proceso se inicia con, la decodificación del código *LT*, siguiendo el algoritmo antes mencionado.

Para este ejemplo se considera un error en el símbolo de color **rojo**, el mismo que representa un símbolo borrado o dañado, por algún efecto del canal de comunicaciones. Esto se realiza con el afán de observar cómo actúa el pre-codificador ante una aparente falla de decodificación *LT*.

Como se aprecia en la Figura 29, dicho error en un símbolo, provoca falla en la decodificación de tipo *Fountain-LT*. Los símbolos en color negro, representan una palabra de código que no se pudo decodificar (recuadro número **6** del mosaico).

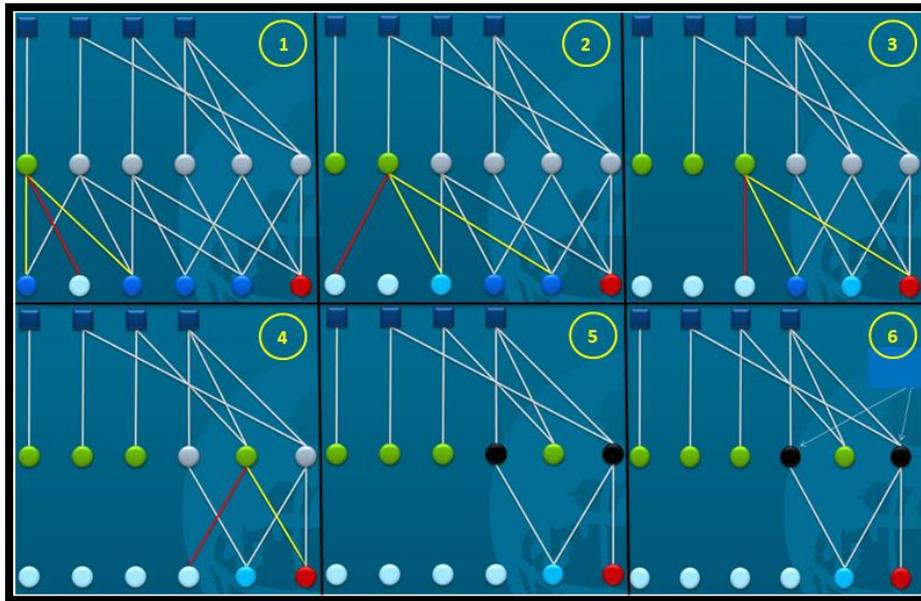


Figura 29. Inicio de Decodificación Raptor (Decodificador LT).

Seguidamente se establece la decodificación de tipo *LDPC* como se muestra en la Figura 30.

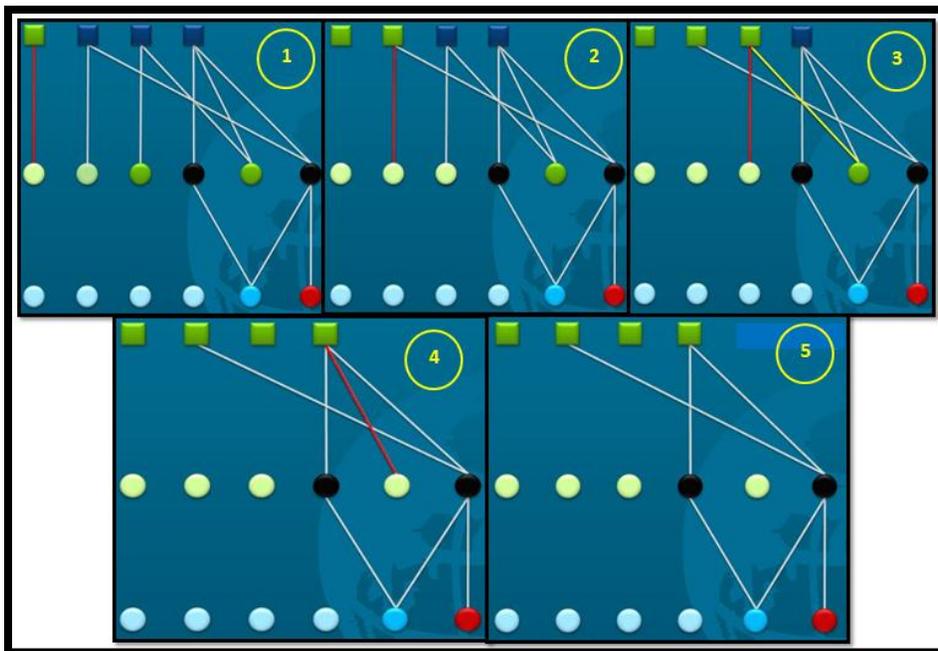


Figura 30. Culminación de Decodificación Raptor (Decodificador LDPC).

Dicho proceso de decodificación permite evidenciar que el pre-codificador actúa como una serie de nodos redundantes, los mismos que permiten recuperar completamente el mensaje original, así exista errores de recepción de datos, o fallas en la decodificación inicial (Decodificación *LT*)

CAPÍTULO IV

4. ANÁLISIS DE RESULTADOS

En relación a que la arquitectura del codificador de tipo *Fountain-Raptor*, posee un bloque de codificación de tipo *LT*, el cual depende de una cantidad potencialmente grande de elementos de grado uno, representados por un solo arco de unión a los símbolos codificados en el grafo bipartido de *Tanner*. Se estableció enfocarse en pruebas donde se permita evidenciar como afecta la variación de bits de información a dicho codificador, ya que esto se traduciría en; que la probabilidad de poseer más elementos de grado uno para la decodificación, aumentaría si la cantidad de información a ser codificada tiende a ser mayor.

Con este análisis previo, se utilizarán bloques de bits de información los cuales serán codificados y posteriormente decodificados, para finalmente verificar de manera gráfica el desempeño de la codificación de tipo *Raptor*, ante una de tipo *LDPC* en dos tipos de escenarios de comunicaciones; tanto para un canal de tipo *AWGN*, así como para un canal con desvanecimiento de tipo *Rayleigh*, utilizando una modulación de tipo *BPSK*, para ambos casos.

Es importante mencionar que la transmisión se la realizará utilizando bloques de 100, 1000, 10 mil, 100 mil y 1 millón de bits de información.

4.1 Canal de tipo AWGN

4.1.1. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de mil bits transmitidos).

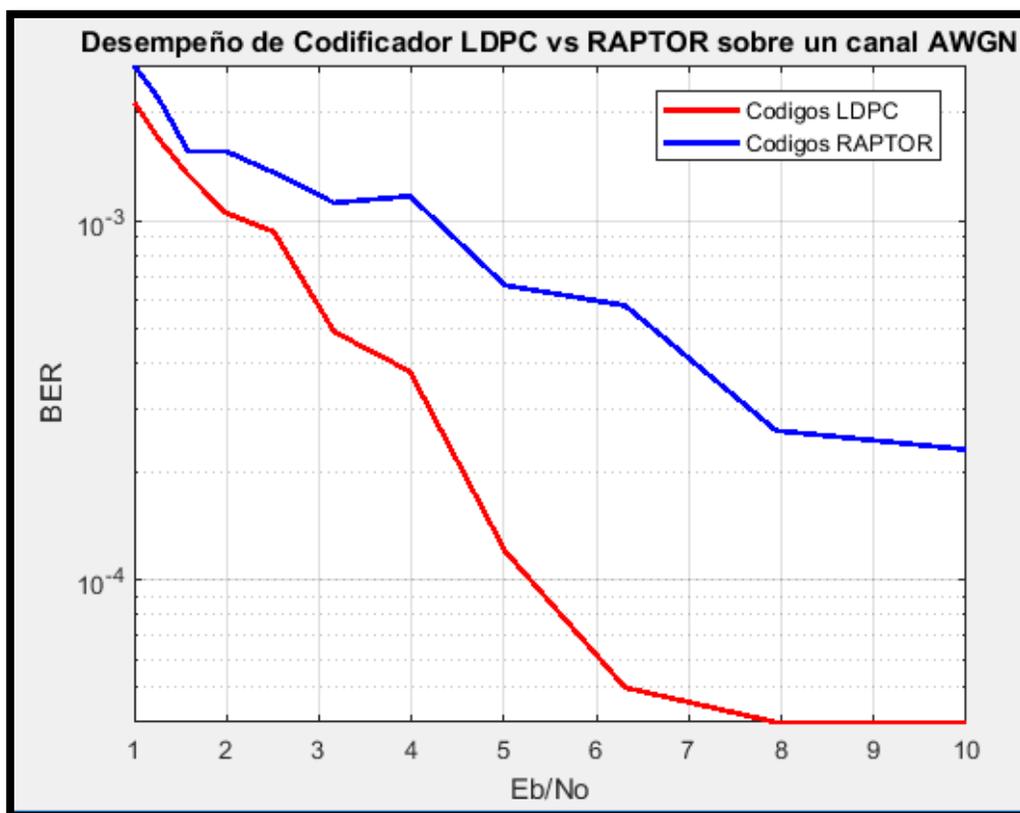


Figura 31. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 1000 bits transmitidos.

El primer escenario de prueba empleado en un canal de tipo Gaussiano, permitió verificar el desempeño del codificador de tipo *Fountain-Raptor*, ante un codificador de tipo LDPC, en donde se especificó una cantidad de paquetes de mil bits, para la transmisión.

La Figura 31, muestra el comportamiento del código *Raptor*, al transmitir dichos paquetes con esta cantidad de bits, reflejando un desempeño inferior al código LDPC.

Esto se debe a la arquitectura de codificación que poseen los códigos *Raptor*, ya que la misma se basa en la unión de un código LDPC y un código *LT*, como se lo explicó en el capítulo III. Un factor importante para realizar la decodificación *LT* con éxito, es que existan nodos intermedios de grado 1, para cumplir con el algoritmo de decodificación.

Aunque la pre-codificación *LDPC* ayude a solventar este inconveniente como se verifica en la Figura 30. Hay que recalcar que dicha solución solo permite generar cierto número de nodos redundantes, así que también depende de la codificación *LT* para una decodificación exitosa.

Entonces es lógico que, cuando se transmite paquetes compuestos por una mayor cantidad de bits de información, exista una probabilidad más alta de que haya, mayor cantidad de elementos de grado 1, para realizar la respectiva decodificación *LT*.

Por el contrario si los paquetes compuestos por una cantidad de bits de información, es baja, en este caso mil bits; el desempeño del codificador decae, ya que la mayoría de estos no pudieron ser recuperados, haciendo que la pre-codificación *LDPC* simplemente obtenga ciertos bits de información.

4.1.2. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 10 mil bits transmitidos).

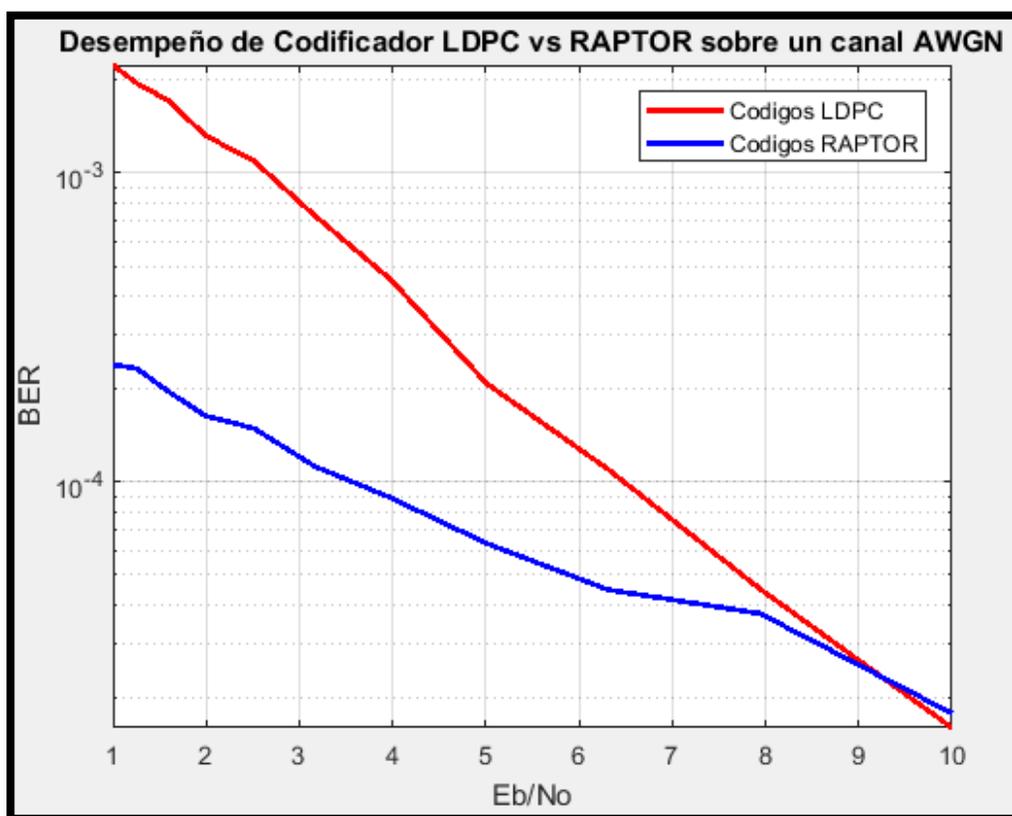


Figura 32. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 10 mil bits transmitidos.

La Figura 32, muestra como el desempeño del codificador de tipo *Raptor* es superior al *LDPC*, tras la variación en la cantidad de bits que componen los paquetes de información transmitidos, en

este caso al aumentar diez veces más el número de bits en un paquete, el codificador *Fountain-Raptor* obtiene un *BER* de $10e-4$ con una relación de Energía de bit a ruido de 3,5 dB aproximadamente, versus una *Eb/No* en la codificación *LDPC* mayor a 6 dB.

Cabe recalcar que esta diferencia de 3 dB aproximadamente de *Eb/No* entre ambos codificadores, es muy representativa, ya que esto indica que al utilizar una codificación *Fountain-Raptor*, la potencia de la señal transmitida será aproximadamente la mitad, frente a la de la codificación de tipo *LDPC*.

Es importante mencionar que al realizar esta variación de aumento en la cantidad de bits que componen los paquetes de información enviados, se pudo corroborar lo que se explica en la Figura 31, en relación a la arquitectura del codificador de tipo *Raptor*. “cuando se transmite paquetes compuestos por una mayor cantidad de bits de información, existe una probabilidad más alta de que haya, mayor cantidad de elementos de grado 1, para realizar la respectiva decodificación *LT*.”.

4.1.3. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 100 mil bits transmitidos).

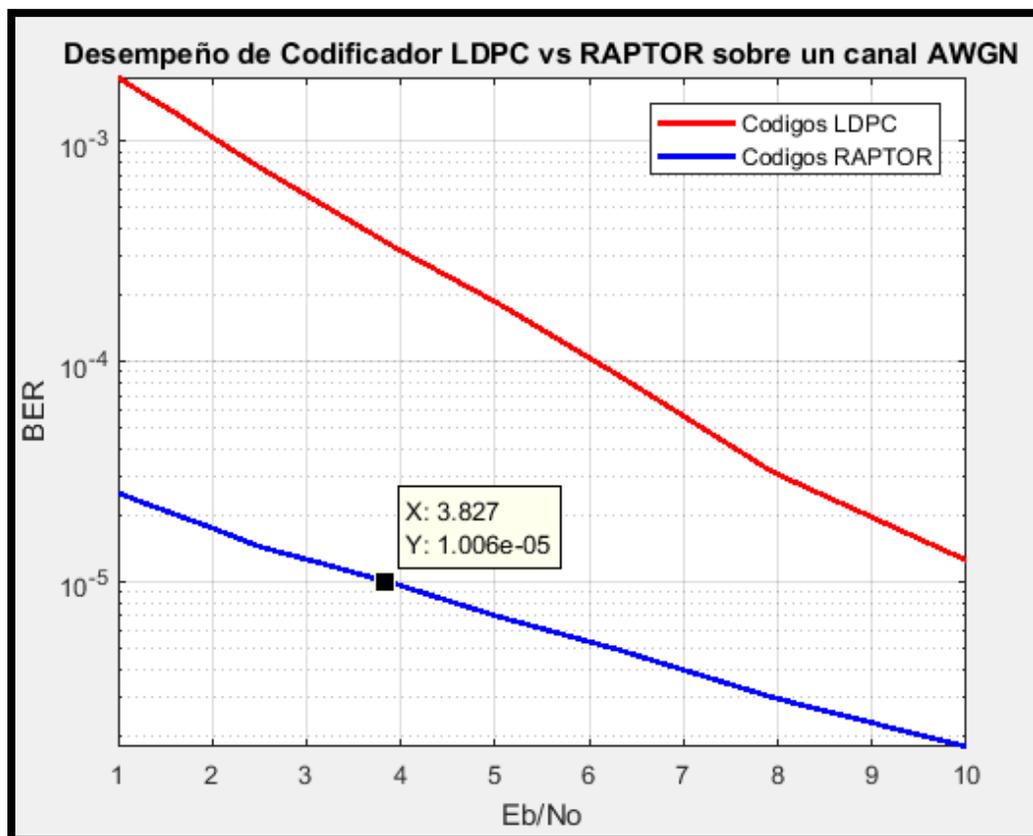


Figura 33. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 100 mil bits transmitidos.

Al igual que en el caso anterior, la variación en la cantidad de bits por paquete transmitidos, permitió evidenciar una vez más, la superioridad de la codificación de tipo *Raptor* ante la de tipo *LDPC*, mostrando resultados de *BER* de $10e-5$, con una relación de Energía de bit a ruido de 3,8 dB aproximadamente, versus un *BER* de $10e-4$, con una *Eb/No* de 6 dB, en la codificación *LDPC*, como se logra apreciar en la Figura 33.

Es importante mencionar que la codificación *LDPC*, con la cual se realiza la comparación, es la misma utilizada para el pre-codificador de tipo *Raptor*. Sin embargo se nota una gran diferencia en cuanto al desempeño entre ambas, por lo cual se puede afirmar que el código de tipo *Fountain-Raptor* es una evolución de la codificación *LDPC*.

4.1.4. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 1 millón de bits transmitidos).

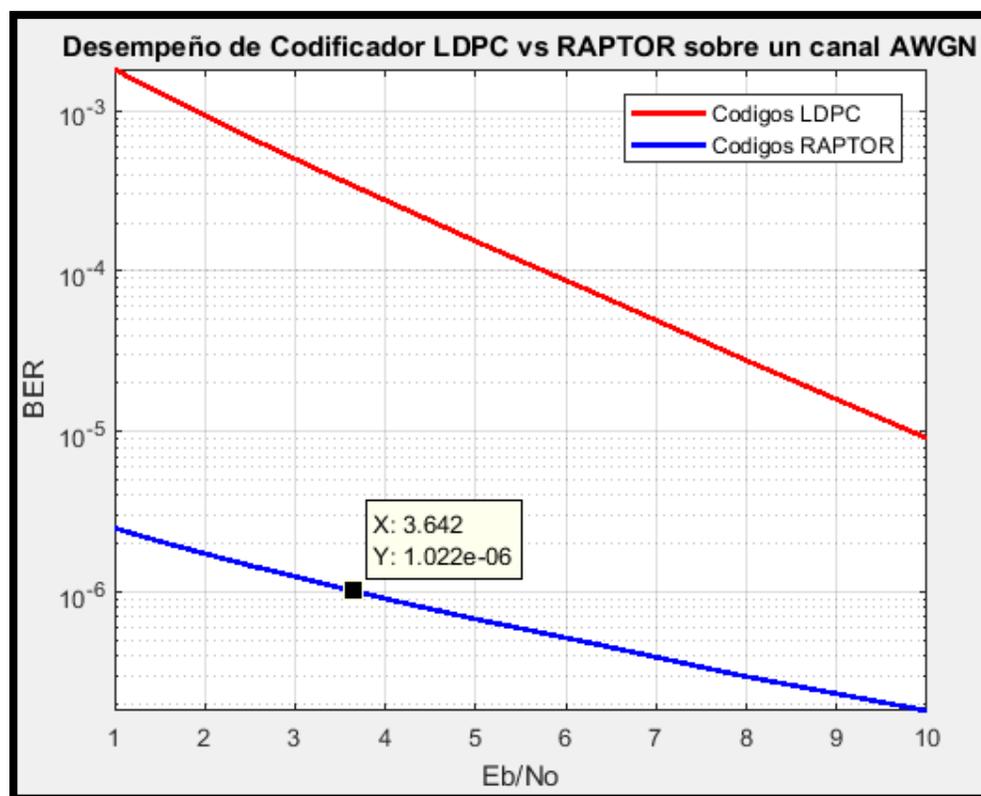


Figura 34. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 1 millón de bits transmitidos.

Finalmente se realiza una prueba más para este tipo de canal (*AWGN*), variando la cantidad de bits que componen los paquetes de información enviados, (1 millón para este caso).

Se puede apreciar en la Figura 34, que el desempeño del codificador tiende a ser proporcional al *BER*, ya que al aumentar la cantidad de bits de información contenidos en un paquete, a 1 millón, el desempeño también mejora a $10e-6$, conservando un promedio de E_b/N_0 de 3,6 dB. Incluso la curva tiende a tomar una forma lineal en el rango de E_b/N_0 analizado.

Cabe recalcar que hay que tomar en cuenta el tamaño del paquete con la cantidad de bits que se envía, debido a que también procesar una gran cantidad de datos de información implica, un mayor gasto computacional para el procesador del equipo receptor, ya que tendría que trabajar mucho más, debido a la cantidad de operaciones lógicas que debe realizar para cumplir con el algoritmo de decodificación. Lo que equivaldría para un ambiente real de Telecomunicaciones como, la reducción del tiempo de vida útil de dicho dispositivo, ocasionando que a largo plazo se produzcan gastos económicos para mantenimiento correctivo del mismo.

4.1.5. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, comparación de cada variación relacionada al tamaño en bits de los paquetes transmitidos

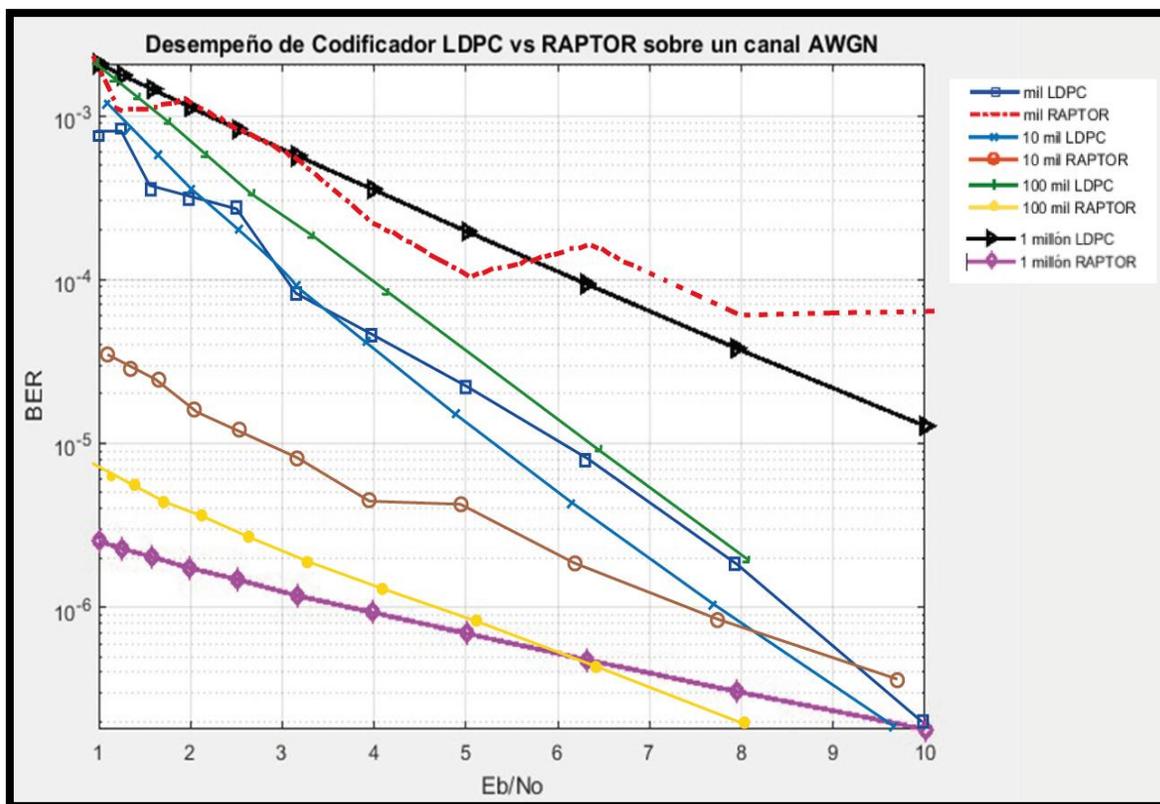


Figura 35. Comparación del desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC de cada variación relacionada al tamaño en bits de los paquetes transmitidos.

La Figura 35, ilustra la comparación de desempeño de los codificadores empleados para las pruebas en el primer escenario de comunicaciones, en donde se evidencia cada una de las variaciones realizadas en cuanto se refiere al tamaño de paquetes en bits de información utilizados, denotando el desempeño más alto del codificador de tipo *Raptor* al transmitir paquetes con tamaño de 1 millón de bits, obteniendo un BER de $10e-6$ con una Eb/No de 4 dB. Mientras que la el desempeño más bajo, incluso menor al de *LDPC* se estableció al transmitir únicamente paquetes con tamaño de mil bits de información, obteniéndose un desempeño del codificador de $10e-4$ con una Eb/No de 5 dB.

4.1.6. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, con variación en sus iteraciones.

Como prueba complementaria se realizó una variación en la codificación *LDPC*, en base al número de iteraciones con las que realiza el algoritmo de decodificación, con el afán de compararla con el codificador de tipo *Raptor* en su máximo valor desempeño, tomando en cuenta que su pre-codificador trabajará con una sola iteración y de esta forma se pueda identificar la ganancia de este tipo de código *Fountain*, al utilizar menos recursos computacionales.

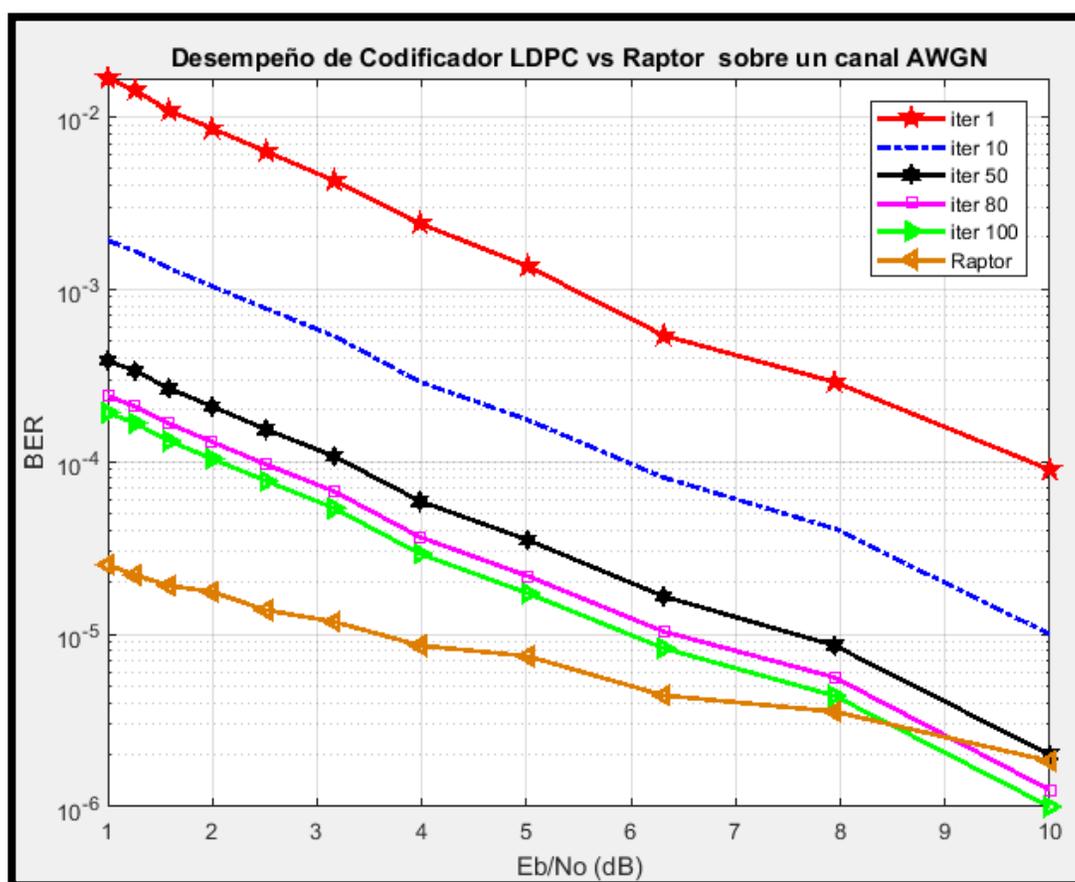


Figura 36. Comparación del desempeño del código Raptor, vs un código de tipo LDPC con variación en sus iteraciones para decodificación.

La Figura 36 permite evidenciar como al aumentar el número de iteraciones en la codificación *LDPC*, el algoritmo mejora su desempeño, incluso llega a igualarse al del codificador de tipo *Raptor* en el caso de una E_b/N_0 de 8 dB. Es importante mencionar que la cantidad de bits de información enviados por paquete, es de 100 mil.

Para realizar un comparativo con el valor máximo de desempeño de la codificación *Raptor*, se utilizó paquetes de información con tamaño de 1 millón de bits, como se aprecia en la Figura 37,

evidenciando como para esta cantidad de bits de información sumamente grande, el número de iteraciones del código *LDPC* de 100; no bastó para igualar el desempeño de la codificación *Raptor*.

Si se llegase a aumentar el número de iteraciones aún más, se traduciría en un costo computacional realmente alto para el micro procesador, debido a la cantidad de operaciones lógicas que debería realizar.

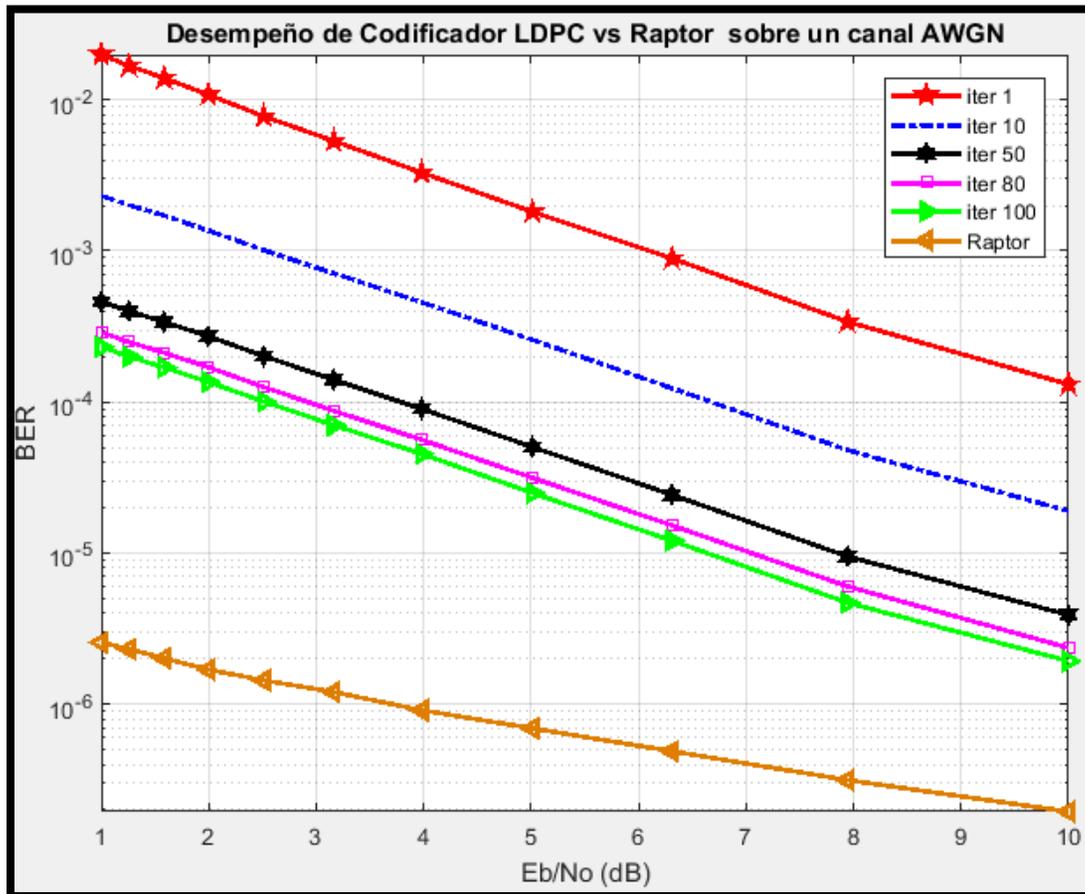


Figura 37. Comparación del desempeño del código Raptor, vs un código de tipo LDPC con variación en sus iteraciones para decodificación (paquetes transmitidos con tamaño 1 millón de bits).

4.2 Canal con desvanecimiento de tipo Rayleigh

4.2.1. Desempeño del codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de mil bits transmitidos).

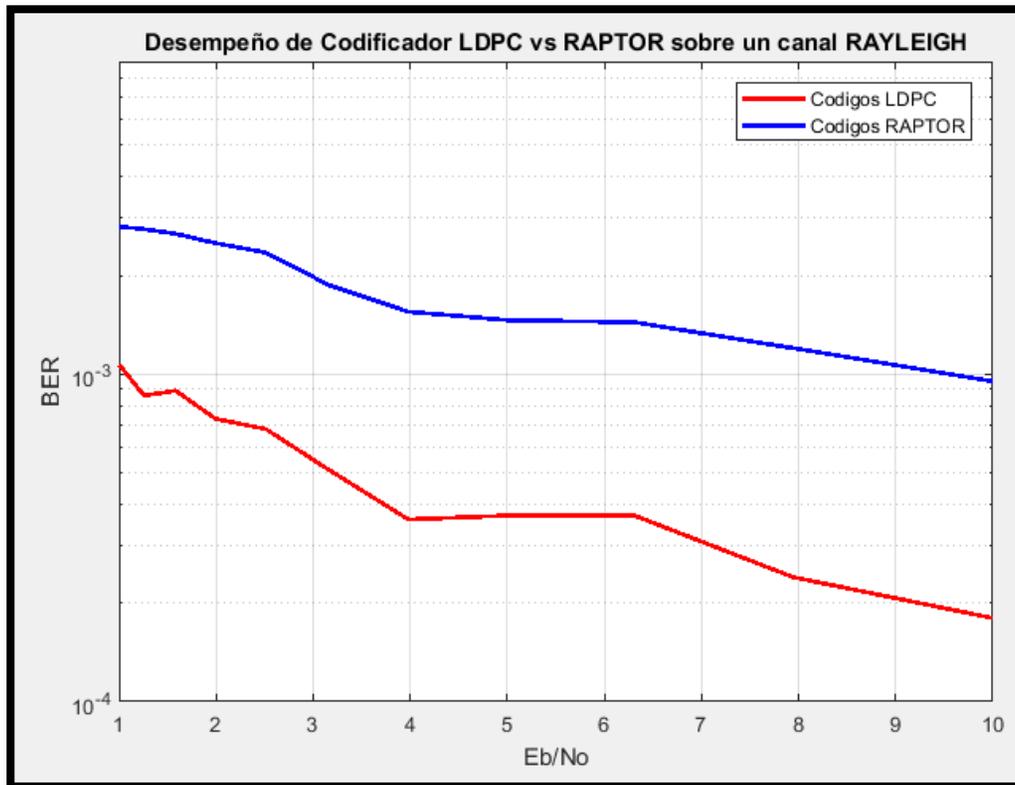


Figura 38. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 1000 bits transmitidos.

Se realiza una comparación entre la codificación de tipo *Raptor* y los códigos de tipo *LDPC* similar al escenario de la primera prueba, en este caso a través de un canal de tipo *Rayleigh*, con la finalidad de observar cómo afecta el fenómeno del desvanecimiento al desempeño de cada uno de los codificadores analizados.

Es importante mencionar que el desvanecimiento presente en el canal, se encarga de atenuar la señal de información y por ende la amplitud del ruido es mayor, lo que ocasiona que el desempeño del codificador tienda a disminuir, en relación al resultado visto en *AWGN*.

Esto se puede apreciar en la Figura 38, ya que las curvas de desempeño tienden a graficarse en un rango más amplio de E_b/N_0 .

Además también es importante notar que de igual manera, como en la primera prueba, la transmisión de paquetes con tamaño de 1000 bits de información hace que el desempeño del codificador *LDPC*, sea superior.

4.2.2. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 5 mil bits transmitidos).

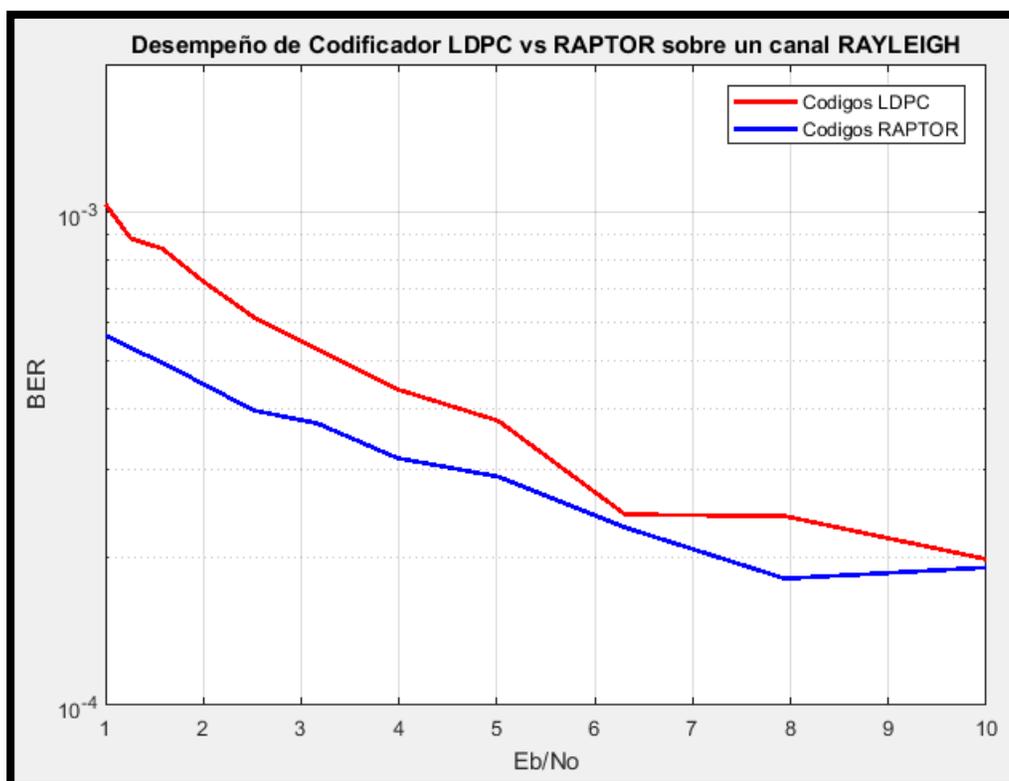


Figura 39. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 5000 bits transmitidos.

De igual manera, se realizó una variación en la prueba establecida anteriormente, aumentando el número de bits que componen cada paquete de información, a cinco mil.

Esto se lo realizó con el afán de comprobar aproximadamente con qué cantidad de bits por paquete de información se logra obtener un desempeño similar entre los dos codificadores.

La Figura 39, denota un desempeño mayor por parte del codificador de tipo *Raptor*, aunque se puede apreciar también que es muy parejo al del codificador de tipo *LDPC*.

4.2.3. Desempeño del codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 10 mil bits transmitidos).

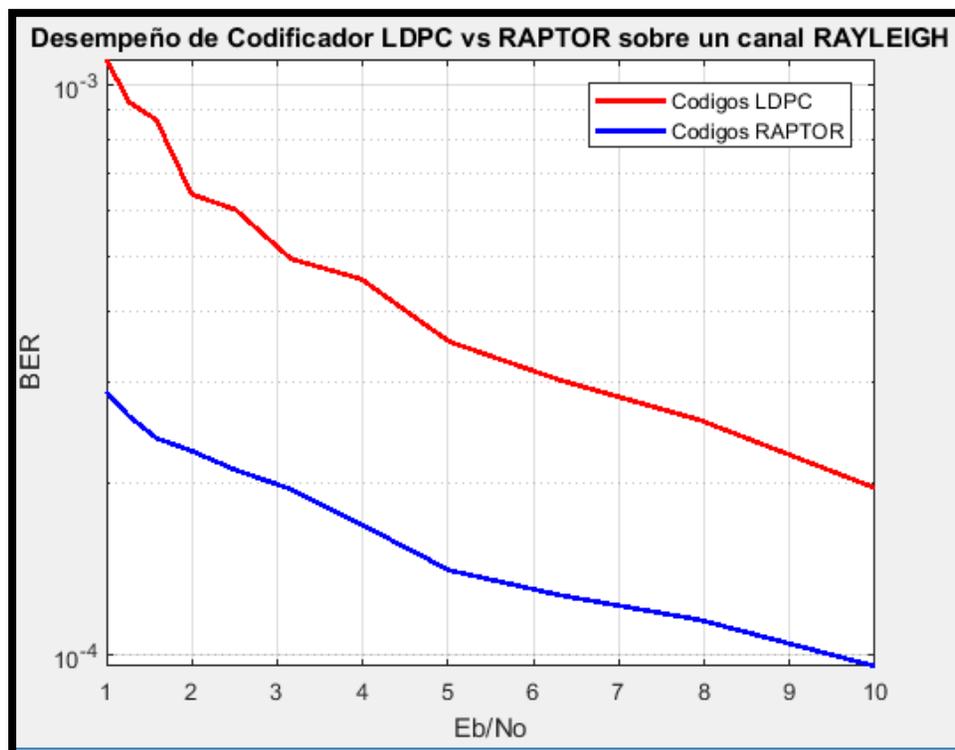


Figura 40. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 10 mil bits transmitidos.

Al variar nuevamente el número de bits transmitidos, en este caso 10 mil por paquete; se puede observar que el codificador *Fountain-Raptor* obtiene un *BER* de $10e-4$ con una relación de Energía de bit a ruido de 10 dB aproximadamente.

Cabe recalcar que esta diferencia de 7 dB aproximadamente de *Eb/No* entre ambos codificadores, es muy representativa, ya que esto indica que el efecto del desvanecimiento en el canal afecta drásticamente el desempeño del codificador, pudiéndose de esta manera corroborar lo que se explicó en la Figura 38, con respecto al modelamiento del desvanecimiento en un canal de tipo *Rayleigh*. “El desvanecimiento presente en el canal, modelado por una distribución de tipo *Rayleigh*, se encarga de atenuar la señal de información y por ende la amplitud del ruido es mayor, lo que ocasiona que el desempeño del codificador tienda a disminuir, en relación al resultado visto en *AWGN*”.

4.2.4. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC (paquetes de 100 mil bits transmitidos).

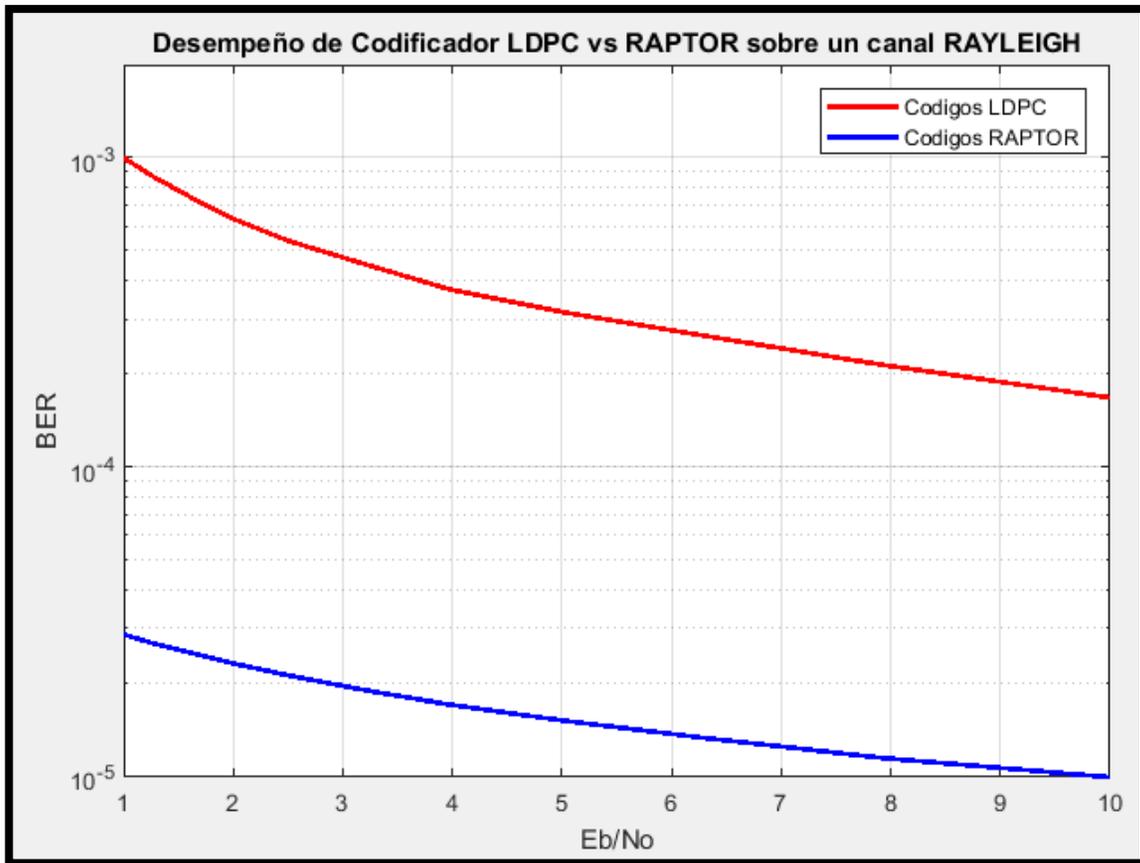


Figura 41. Desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC con paquetes de 100 mil bits transmitidos

Finalmente se realiza una última variación en la cantidad de bits transmitidos por paquete, siendo 100 mil bits, en donde se obtiene un comportamiento similar del desempeño del codificador al de las pruebas realizadas en el canal de *AWGN*, ya que de igual manera la *Eb/No*, se mantiene con el mismo valor, en este caso aproximadamente de 10 dB. Lo que varía de igual manera en relación a la cantidad de bits que componen los paquetes de transmisión, es el *BER*; mostrando un resultado de 10e-5.

Cabe recalcar que al igual que en el caso del primer escenario de prueba, la curva tiende a tomar una forma lineal en el rango de *Eb/No* analizado.

4.2.5. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, comparación de cada variación, relacionada al tamaño en bits de los paquetes transmitidos.

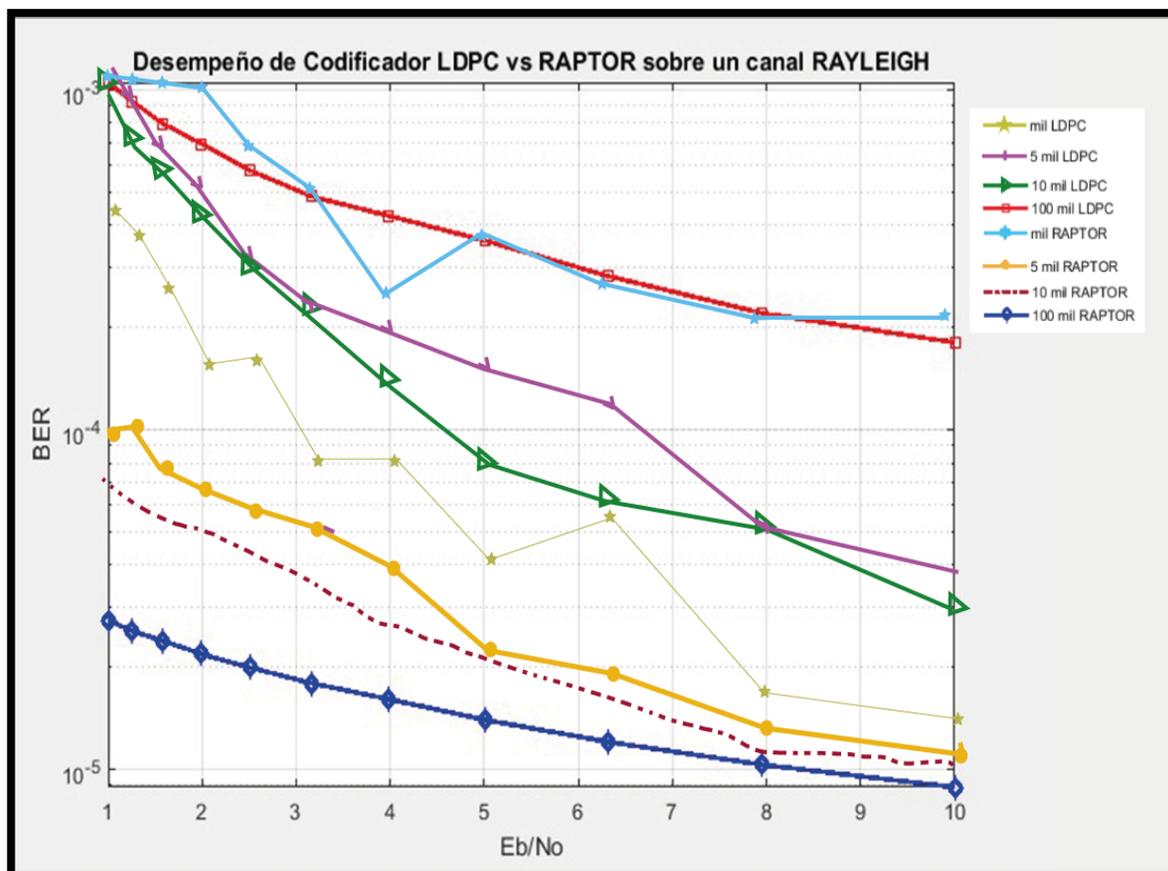


Figura 42. Comparación del desempeño del código corrector de errores de tipo Raptor, vs un código de tipo LDPC de cada variación relacionada al tamaño en bits de los paquetes transmitidos.

La Figura 42, ilustra la comparación de desempeño de los codificadores empleados para las pruebas en el segundo escenario de comunicaciones, en donde se evidencia cada una de las variaciones realizadas en cuanto se refiere al tamaño de paquetes en bits de información utilizados, denotando el mayor desempeño del codificador de tipo Raptor al transmitir paquetes con tamaño de 1 millón de bits, obteniendo un BER de $10e-5$ con una Eb/No de 8 dB. Mientras que la el desempeño más bajo, incluso menor al de $LDPC$ se estableció al transmitir únicamente mil bits de información, obteniéndose un BER de $10e-3$.

4.2.6. Desempeño del Codificador de tipo Fountain Raptor vs un codificador de tipo LDPC, con variación en sus iteraciones.

De igual manera que en el primer escenario, se realizó como prueba complementaria una variación en la codificación *LDPC*, en base al número de iteraciones con las que se realiza el algoritmo de decodificación del mismo, con el afán de compararla con el codificador de tipo *Raptor* en su máximo valor desempeño y de esta forma identificar la ganancia de este tipo de codificación, tomando en cuenta que el pre-codificador de *Raptor*, utiliza para la decodificación una sola iteración.

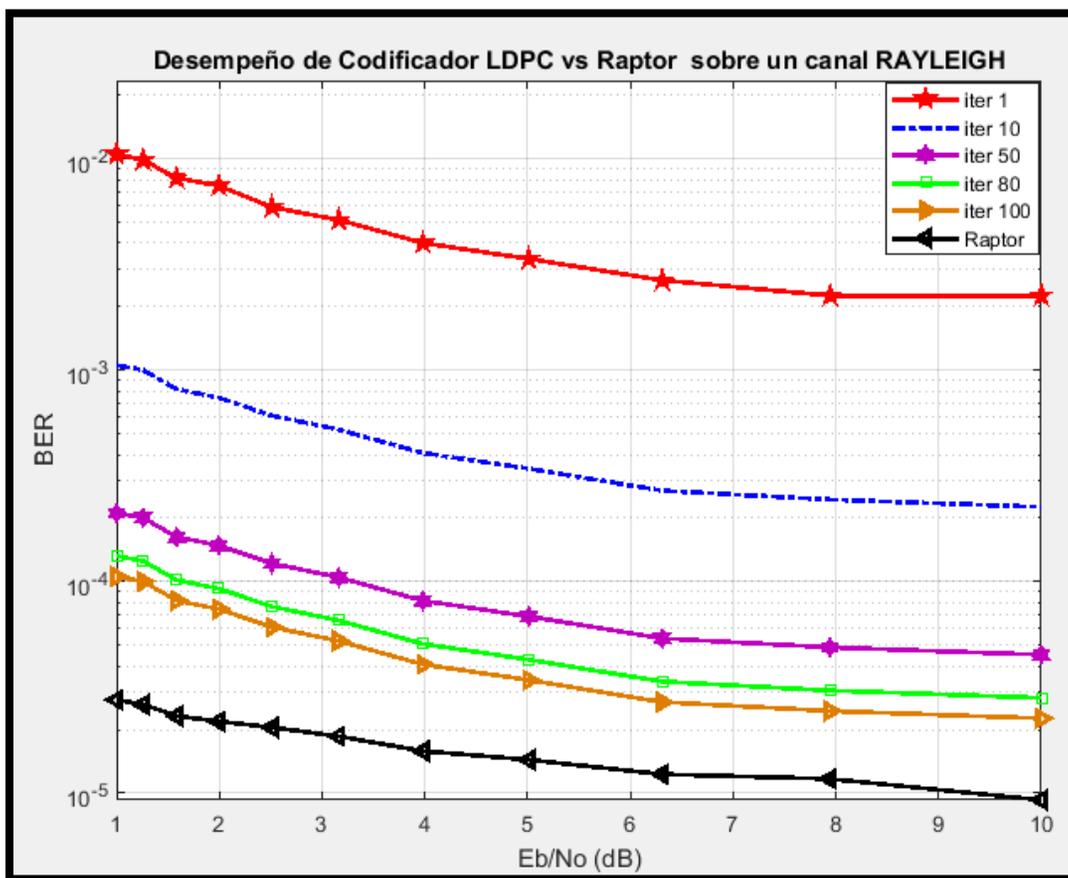


Figura 43. Comparación del desempeño del código Raptor, vs un código de tipo LDPC con variación en sus iteraciones para decodificación.

La Figura 43 permite evidenciar como al aumentar el número de iteraciones en la codificación *LDPC*, el algoritmo mejora su desempeño, siendo cercano al desempeño de la codificación *Raptor*.

A diferencia del primer escenario el desempeño del codificador de tipo *LDPC* con un número total de 100 iteraciones, no llega a ser igual al desempeño del código Raptor en ningún valor de

E_b/N_0 . Este comportamiento está relacionado con el tipo de canal de comunicaciones, ya este al poseer un desvanecimiento modelado por una distribución de tipo *Rayleigh* influye aún más en la corrección de errores en el decodificador.

De igual manera, si se llegase a aumentar el número de iteraciones aún más, con el afán de mejorar el desempeño del codificador de tipo *LDPC*, se traduciría en un costo computacional realmente alto para el micro procesador, debido a la cantidad de operaciones lógicas que debería realizar.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Al transmitir una mayor cantidad de bits por paquetes de información, existe una probabilidad más alta de que haya mayor cantidad de elementos de grado 1, para realizar la respectiva decodificación de tipo *LT*.
- La pre-codificación *LDPC* ayuda a generar cierto número de nodos redundantes, en la codificación de tipo *LT*, para una decodificación exitosa.
- La variación en la cantidad de bits que componen cada uno de los paquetes de información transmitidos, permitió evidenciar una vez más, la superioridad de la codificación de tipo *Raptor* ante la de tipo *LDPC*, mostrando resultados de *BER* de $10e-5$, con una relación de Energía de bit a ruido de 3,8 dB aproximadamente, versus un *BER* de $10e-4$, con una *Eb/No* de 6 dB, en la codificación *LDPC*.
- La codificación *LDPC*, con la cual se realiza la comparación, es la misma utilizada para el pre-codificador de tipo *Raptor*. Sin embargo se nota una gran diferencia en cuanto al desempeño entre ambas, por lo cual se puede afirmar que el código de tipo *Fountain-Raptor* es una evolución de la codificación *LDPC*.
- El desempeño del codificador *Fountain-Raptor* tiende a ser proporcional al *BER*, ya que al aumentar la cantidad de bits de información, el desempeño también mejora, aunque conserva un valor promedio de *Eb/No*.
- El procesar una gran cantidad de datos de información implica, un mayor gasto computacional para el procesador del equipo receptor, ya que trabaja mucho más, debido a la cantidad de operaciones lógicas que debe realizar para cumplir con el algoritmo de decodificación.
- El desvanecimiento presente en el canal, modelado por una distribución de tipo *Rayleigh*, se encarga de atenuar la señal de información y por ende la amplitud del ruido es mayor, lo que ocasiona que el desempeño del codificador tienda a disminuir, en relación al resultado obtenido en *AWGN*.

- Al variar el número de bits en 10 mil, por paquete transmitido, para un canal de tipo *Rayleigh*; se puede observar que el codificador *Fountain-Raptor* obtiene un *BER* de $10e-4$ con una relación de Energía de bit a ruido de 10 dB aproximadamente. Existe una diferencia con la prueba en el canal de *AWGN* de 7 dB aproximadamente de *Eb/No*, lo cual es muy representativa, ya que esto indica que el efecto del desvanecimiento en el canal afecta drásticamente el desempeño del codificador.
- El aumento en el número de iteraciones en la codificación *LDPC*, hace que el codificador mejore su desempeño, incluso llega a igualarse al del codificador de tipo *Raptor* en el caso de una *Eb/No* de 8 dB. Al utilizar la cantidad de 100 mil bits por paquete de información en un canal *AWGN*.
- Al realizar un comparativo en un canal *AWGN* con el valor máximo de desempeño de la codificación *Raptor*, se utilizó un tamaño de paquete de datos de 1 millón, evidenciando como para esta cantidad de bits de información sumamente grande el número de iteraciones del código *LDPC* de 100; no bastó para igualar el desempeño de la codificación *Raptor*, tomando en cuenta que esta no utiliza más que una iteración en para la decodificación del pre-código.
- El desempeño del codificador de tipo *LDPC* con un número total de 100 iteraciones, no llega a ser igual al desempeño del código *Raptor*, en ningún valor de *Eb/No*, pese a que este no utiliza más que una iteración para la decodificación del pre-código. Este comportamiento está relacionado con el tipo de canal de comunicaciones, ya que este al poseer un desvanecimiento modelado por una distribución de tipo *Rayleigh* influye aún más en la corrección de errores en el decodificador.

5.2. RECOMENDACIONES

- Realizar un mayor número de pruebas, en donde existan muchas más variaciones en cuanto a fenómenos físicos de canales de comunicaciones.
- Emplear variaciones de modulaciones de nueva generación, con el afán de verificar cuanto se complementa a esta codificación y logra hacerla más eficiente.
- Realizar variaciones en la codificación *Raptor*, específicamente en el pre-código, utilizando otro tipo de códigos *FEC*, en el mismo; o combinaciones de códigos, con el afán de verificar si existe mayor desempeño del código.
- Continuar con el siguiente paso en la nueva generación de códigos correctores de errores, los cuales son denominados, códigos de dos dimensiones.

REFERENCIAS

- Byers, J. W., Luby, M., Mitzenmacher, M., & Rege, A. (56-67). A digital fountain approach to reliable distribution of bulk data. *SIGCOMM*, 1998.
- Gallager, R. G. (1960). Low-density parity-check codes. *IEEE Transactions on Information Theory* 8, 21-28.
- Luby, M. (2002). LT Codes, Foundations of Computer Science. *The 43 Annual IEEE Symposium*, 271-280.
- Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A., & A, D. (2001). Spielman Improved low-density parity-check codes using irregular graphs. . *IEEE Transactions on Information Theory*, 47(2), 585-598.
- Mackay, D. J. (2003). Information Theory, Interference and Learning Algorithms. Cambridge University press.
- Ramos, F. (15 de junio de 2018). *Radioenlaces*. Obtenido de tecnologías inalámbricas y diseño de radio enlaces: <http://www.radioenlaces.es/articulos/perdidas-en-obstaculos/>
- Shokrollahi, A. (2006). Raptor codes. *IEEE/ACM Trans. Netw.* 14 (SI), 2551-2567.
- Shokrollahi, A. (2006). Raptor codes, Application Programming Interface, Advance Card System Limited. *IEEE Transactions on Information Theory*, Vol. 52 No. 6, 2561-2567.
- Shokrollahi, A., & Luby, M. (2006). Foundations and Trends . *Communications and Information Theory* Vol. 6, 256-258.
- Tanner, M. (1981). A recursive approach to low complexity codes. *IEEE Transactions on information theory* Vol IT-27, No 5, 120-125.
- Wesabi, M. A., Al-Arashi, W. H., & Alshadadi, M. M. (12 de abril de 2018). *ResearchGate*. Obtenido de ResearchGate: https://www.researchgate.net/figure/Figure-1-Fountain-Code-Concept_fig1_320465565