

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica



TESIS DE MAESTRÍA

Codificación y recolección de datos correlacionados en redes de sensores.

Tesis presentada en opción al grado de
Máster en Telemática

Autor: Ing. Judiel Reyes Aguilar

Tutor: D.CT. Vitalio Alfonso Reguera

Santa Clara

2016

"Año 58 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica



TESIS DE MAESTRÍA

Codificación y recolección de datos correlacionados en redes de sensores

Autor: Ing. Judiel Reyes Aguilar

judielra@uclv.edu.cu

Tutor: D.CT. Vitalio Alfonso Reguera. Prof. Titular

Departamento de Telecomunicaciones y Electrónica

Facultad de Ingeniería Eléctrica

Universidad Central “Marta Abreu” de Las Villa, UCLV

E-mail: vitalio@uclv.edu.cu

Santa Clara

2016

"Año 58 de la Revolución"



Hago constar que la presente Tesis de Maestría fue realizada en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la Maestría en Telemática, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicado sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Autor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Coordinador de la
Maestría en Telemática

PENSAMIENTO

*“La mente es igual que un paracaídas, solo
funciona si se abre”*

Albert Einstein

AGRADECIMIENTOS

A mi familia por el cariño dado en tanto años.

A mi tutor, por los conocimientos y el ejemplo transmitido.

A mis compañeros de trabajo por su colaboración en este empeño.

Al claustro de profesores de la Facultad de Eléctrica por su abnegación en nuestra formación y desarrollo profesional.

A todos los que de una forma u otra han contribuido en esta investigación.

RESUMEN

Para conseguir un menor gasto energético en las redes de sensores inalámbricos los investigadores han propuesto el uso de la Codificación de Fuente Distribuida. Esta técnica se refiere a la compresión separada y la descompresión conjunta de fuentes mutuamente correlacionadas. Una de las problemáticas presentada en esta temática es desplegar una estrategia de recolección de los datos que implemente la codificación de forma eficiente.

El objetivo principal de esta investigación es evaluar el desempeño de diferentes estrategias de recolección de datos en redes de sensores que implementan la Codificación de Fuente Distribuida, atendiendo a parámetros como la probabilidad de error de bit, la energía consumida y el tiempo de vida de la red.

En este trabajo se abordan los elementos teóricos de la Codificación de Fuente Distribuida en redes de sensores y se implementan los algoritmos de codificación/decodificación para los modelos simétricos y asimétricos basados en códigos de bloques lineales. A partir de estas funciones se simulan las estrategias de recolección de datos secuencial y basada en clúster.

Los resultados de los experimentos demuestran que, si bien la estrategia basada en clúster tiene un mejor desempeño en cuanto a la probabilidad de error en la decodificación, la estrategia secuencial logra un mayor ahorro de energía total en la red. Por lo que la elección de estas estrategias implica un compromiso entre el consumo energético y el desempeño de la decodificación.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTOS DE LA CODIFICACIÓN DE FUENTE DISTRIBUIDA EN REDES DE SENSORES	6
1.1 Teorema de Slepian-Wolf.....	6
1.2 Esquema Slepian-Wolf	8
1.3 Esquema Wyner-Ziv	10
1.4 Recolección de datos en redes de sensores.....	12
1.5 Principios para la recolección de datos en redes de sensores que emplean Codificación de Fuente Distribuida.....	16
1.6 Estrategias de recolección de datos en redes de sensores que emplean Codificación de Fuente Distribuida.....	20
1.6.1 Estrategia de recolección de datos secuencial	20
1.6.2 Estrategia de recolección de datos basada en clúster.	22
CAPÍTULO 2. FUNCIONES Y ALGORITMOS	24
2.1 Algoritmos de Codificación de Fuente Distribuida.....	24
2.1.1 Modelo de Correlación	24
2.1.2 Codificador asimétrico.....	25
2.1.3 Decodificador asimétrico	25
2.1.4 Codificador simétrico.....	26
2.1.5 Decodificador simétrico.....	27
2.2 Estrategias de recolección de datos	28
2.2.1 Estrategia recolección de datos secuencial	28
2.2.2 Estrategia recolección de datos basada en clúster.....	31

CAPÍTULO 3. ANÁLISIS DE RESULTADOS	32
3.1 Descripción del modelo de simulación.....	32
3.2 Resultados de las simulaciones.....	35
3.2.1 Caso simple de dos fuentes y un terminal.....	35
3.2.2 Desempeño de las estrategias de recolección de datos en redes de sensores.....	37
3.2.2.1 Desempeño de la decodificación	38
3.2.2.2 Consumo de energía.....	41
3.2.2.3 Tiempo de vida de la red.....	44
CONCLUSIONES Y RECOMENDACIONES	47
Conclusiones	47
Recomendaciones	48
REFERENCIAS BIBLIOGRÁFICAS	49

INTRODUCCIÓN

Una de las aplicaciones de la ingeniería más investigada en los últimos años es las redes de sensores. Estas redes despliegan pequeños dispositivos llamados nodos, interconectados entre sí de forma inalámbrica, sobre un área geográfica para el monitoreo de fenómenos físicos. Los nodos sensores son capaces de captar, procesar y transmitir información extraída de áreas físicas de observación, al resto de los nodos de la red [1]. Actualmente estas redes se emplean en diversos campos de la ciencia, tales como: monitoreo de un hábitat; detección de incendios, terremotos o inundaciones; edificios “inteligentes”; control de tráfico y asistencia militar o civil; entre otros.

Típicamente un nodo sensor es un dispositivo diminuto que incluye tres componentes básicos: un subsistema de sensor para la adquisición de los datos del ambiente físico circundante; un subsistema de procesamiento y almacenamiento local de los datos; y un subsistema de comunicación inalámbrico para transmitir y recibir sus datos y los de otros nodos vecinos. Además de estos componentes, una fuente eléctrica abastece la energía necesaria para que los dispositivos realicen las tareas necesarias. Esta fuente a menudo consiste en una batería con limitado presupuesto de energía. En ocasiones puede ser imposible o inconveniente cambiar las baterías, porque los nodos pueden estar desplegados en un ambiente hostil o las funciones que realice la red no lo permiten. Es decir que, las redes de sensores deben tener un largo tiempo de vida, suficiente para cumplir las aplicaciones requeridas. En muchos casos este tiempo de vida se extiende por varios meses y años. Por lo que un requisito imprescindible en el diseño de los nodos y el modelo de red es la optimización del consumo de energía y la maximización de su tiempo de vida.

En muchos escenarios de las redes de sensores se logra una alta correlación espacial entre los datos captados por los nodos. Una consecuencia de esta correlación es que las lecturas entre nodos vecinos son altamente redundantes. Explotar esta correlación para remover los datos redundantes a través de una técnica de compresión, permite reducir los bits a transmitir y de esta forma reducir la energía consumida en el proceso de transmisión de los datos.

Las investigaciones han demostrado que aproximadamente el 80% del consumo de energía de cada nodo es usado en la transmisión de los datos. Lo que indica que el uso de técnicas de

compresión, que minimice el tamaño de los datos a transmitir, es imprescindible en la optimización energética en la red. Sin embargo, las restricciones de la energía limitan la capacidad de procesamiento y la potencia de transmisión de los nodos. Por otra parte, el impacto del ruido involucrado por el uso de canales inalámbricos dificulta la reconstrucción de los datos. Por lo que, para enviar los datos recolectados por los sensores a la estación base, se requiere buscar métodos robustos con alta compresión y baja complejidad, que permita una transmisión de los datos eficiente en el ahorro de energía [2]. Esta compresión puede alcanzarse convencionalmente con un método de codificación de fuente si se conoce la redundancia entre los datos. Sin embargo, esto requeriría comunicación entre los nodos previamente antes de iniciar el proceso de codificación lo cual no sería permisible por las restricciones energéticas dadas en las redes de sensores. Sorprendentemente en 1973, Slepian y Wolf demostraron en [3] que cuando las fuentes están correlacionadas la comunicación entre los codificadores no es necesaria para lograr la compresión de los datos, siempre que estos se decodifiquen en conjunto, dando las primeras nociones de la Codificación de Fuente Distribuida.

Según Stankovic, la Codificación de Fuente Distribuida se refiere a la compresión separada y descompresión conjunta de dos o más fuentes físicamente separadas (de aquí el término de distribuido) en los codificadores y decodificadas en conjunto en el decodificador. La Codificación de Fuente Distribuida es de esta forma un método de codificación que tiene como objetivo explotar las dependencias mutuas a través de diferentes fuentes que requieren no tener comunicación entre sí [4].

La Codificación de Fuente Distribuida aparece como un problema de la Teoría de la Información en [3], en este artículo Slepian y Wolf estudiaron el simple caso de la Codificación de Fuente Distribuida con dos fuentes discretas que son comprimidas independientemente y decodificadas sin pérdidas en el decodificador de forma conjunta, y determinaron una región para las razones de compresión alcanzable, demostrando que la codificación por separado es tan buena como la que se puede realizar en conjunto. Este resultado provocó una gran cantidad de investigaciones sobre la Teoría de Información, esfuerzos que resultaron en varias extensiones de este trabajo [5-11].

Desde finales de la década de los 70's, con los estudios de Wyner [5, 6] se demostró que era posible la realización de la Codificación de Fuente Distribuida usando códigos de canal, pero debido a la carencia de aplicaciones potenciales, el estudio del diseño de un código empezó solo a finales del pasado siglo. El lanzamiento de las redes de sensores retomó los estudios en este campo. El primer diseño práctico se reportó en el 2000 en [8, 9]. En los años posteriores se han desarrollado códigos más potentes.

Los estudios más recientes sobre la Codificación de Fuentes Distribuida investigan como desplegar de forma óptima esta técnica de compresión en las redes de sensores [10-12] y en otras aplicaciones como la codificación distribuida de video [13]. Actualmente se analiza como interactúa esta técnica de compresión con otras capas de red como el enrutamiento y el control de acceso al medio [14].

Uno de los principales problemas que ha planteado el uso de la Codificación de Fuentes Distribuida en las redes de sensores es el enrutamiento de los datos codificados. Los protocolos de enrutamiento deben garantizar que los datos mantengan la correlación y que sean recolectados de manera óptima. Desde el punto de vista teórico se han presentado diferentes estrategias de recolección que se clasifican en: estrategia secuencial y estrategia basada en clúster.

Por lo que una de las problemáticas científicas que tiene hoy esta temática es determinar cuál estrategia de recolección de datos es más eficiente para desplegar la Codificación de Fuente Distribuida en redes de sensores, atendiendo a parámetros como la probabilidad de error de bit, la energía consumida y el tiempo de vida de la red.

Para dar solución este problema se plantea como objetivo general de esta investigación:

Evaluar el desempeño de las estrategias de recolección de datos en redes de sensores que implementan la Codificación de Fuente Distribuida.

Para dar cumplimiento al objetivo general se derivan los siguientes objetivos específicos:

- Caracterizar la Codificación de Fuente Distribuida en las redes de sensores inalámbricos.
- Implementar en Matlab algoritmos de codificación de fuente distribuida y las estrategias de recolección de datos.

- Analizar mediante simulación de eventos discretos el comportamiento de los algoritmos de codificación y el desempeño de las estrategias de recolección de datos.

Los métodos científicos sobre los cuales se desarrolla la investigación son:

- El histórico lógico el cual permite contextualizar el problema de investigación, sus antecedentes y desarrollo.
- El analítico-sintético ya que es necesario estudiar las particularidades de las técnicas de codificación y de las estrategias de recolección de datos para lograr la integración de las partes constitutivas del objeto de investigación y determinar cuál estrategia de recolección de datos es más eficiente para desplegar la Codificación de Fuente Distribuida en redes de sensores.
- El inductivo-deductivo a través del cual se logra establecer generalidades en cuanto a las técnicas de codificación y las estrategias de recolección de datos, a partir de las experiencias de investigaciones previas de otros autores.
- La modelación mediante la cual se crean abstracciones con vistas a explicar la realidad. El modelo como sustituto del objeto de investigación. Opera en forma práctica o teórica con un objeto, no en forma directa, sino utilizando cierto sistema intermedio, auxiliar, natural o artificial. En este caso, la simulación se emplea como método empírico para evaluar el desempeño de las estrategias de recolección de datos.

Los principales resultados de la presente tesis se concentran en:

1. La obtención de estrategias viables para el ahorro de energía en las redes inalámbricas de sensores.
2. La caracterización de las diferentes técnicas de codificación de fuente distribuida y de las estrategias de recolección de datos.
3. Un material de estudio para la futura implementación de las redes inalámbricas de sensores.

Las redes de sensores son consideradas una potente línea de investigación debido a sus disímiles aplicaciones. Sin embargo, los métodos de codificación son una alternativa, para la eficiencia de estas redes aún en estudio. El impacto científico que se deriva de este trabajo

es la evaluación de las estrategias de recolección de datos propuestas en la literatura para la implementación eficiente de la Codificación de Fuente Distribuida en las redes de sensores. Sus resultados brindarán elementos para la elección de estas estrategias para futuros despliegues en las redes de sensores.

El informe de la investigación consta de introducción, capitulario, conclusiones, recomendaciones y referencias bibliográficas. En el Capítulo 1 se analizan los fundamentos teóricos de la Codificación de Fuente Distribuida y los principios dados por la literatura para la recolección de datos en redes de sensores que utilizan esta técnica de codificación. El Capítulo 2 se dedica a la descripción de las funciones y algoritmos de los esquemas de codificación/decodificación de fuente distribuida y de las estrategias de recolección de datos implementados. El Capítulo 3 resume la experimentación de las funciones creadas como parte del Capítulo 2 y el análisis de los resultados obtenidos en este proceso.

CAPÍTULO 1. FUNDAMENTOS DE LA CODIFICACIÓN DE FUENTE DISTRIBUIDA EN REDES DE SENSORES

En el presente capítulo se caracteriza la Codificación de Fuente Distribuida y su aplicación en redes de sensores inalámbricos. Primeramente, se analizan los fundamentos teóricos de este método de codificación. Se analiza el teorema de Slepian-Wolf y el esquema de Wyner-Ziv, explicando las estrategias simétricas y asimétricas. Finalmente se repasan los principales principios dados por la literatura para la recolección de datos en redes de sensores que implementan la Codificación de Fuente Distribuida.

1.1 Teorema de Slepian-Wolf

Según la teoría de la codificación de Shannon, para una fuente aleatoria X , una razón $R \geq H(x)$ es suficiente para transmitir X sobre un canal confiable hasta la estación base. Por lo que, si tenemos dos fuentes (X, Y) distribuidas idénticas e independientes, y son codificadas de forma separada, se requiere que la razón total $R = R_X + R_Y \geq H(X) + H(Y)$. Siendo R_X y R_Y la razón de codificación de las fuentes X y Y , respectivamente, y $H(X)$ y $H(Y)$ sus entropías.

Aplicando una codificación distribuida de manera conjunta las fuentes pueden ser codificadas a razones iguales que la entropía condicional de las fuentes. Esto requeriría de comunicación entre los nodos, lo cual implica una sobrecarga adicional en la red y el objetivo de la codificación en las redes de sensores es exactamente el contrario, reducir la cantidad de procesamiento de información y por consiguiente el consumo de energía.

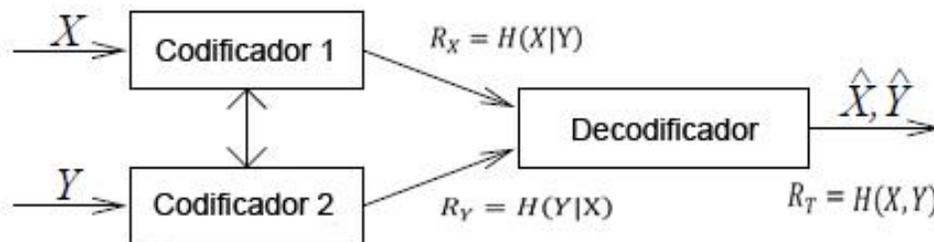


Figura 1.1: Codificación de Slepian-Wolf para dos fuentes correlacionadas.

Sin embargo Slepian y Wolf en [3] demuestran que la codificación distribuida se puede lograr sin comunicación entre las fuentes siempre que estas estén correlacionadas entre sí. De manera que, aplicando esta teoría, como se muestra en la Figura 1.1, una razón total de $R = R_X + R_Y \geq H(X, Y)$ es suficiente, incluso si las dos fuentes son codificadas de forma separada, siempre que los datos sean descodificados de forma conjunta. Donde $H(X, Y)$ hace referencia a la entropía conjunta de ambas fuentes.

De esta manera, Slepian-Wolf demostraron que dos fuentes X y Y pueden ser descodificadas sin pérdidas siempre que:

$$\begin{aligned} R_X &\geq H(X|Y) \\ R_Y &\geq H(Y|X) \\ R_T = R_X + R_Y &\geq H(X, Y) \end{aligned} \tag{1.1}$$

Donde R_X y R_Y son las razones usadas para la compresión de X y Y respectivamente. Estas ecuaciones, son conocidas como la región de la codificación de Slepian-Wolf, la cual es representada en la Figura 1.2, y muestran que la razón total puede ser tan pequeña como la entropía conjunta de las fuentes, lo cual es lo mismo que si las fuentes son codificadas en conjunto.

A partir de esta región se han planteado dos estrategias de codificación: la asimétrica y la simétrica.

El caso asimétrico, correspondiente a los puntos A y B de la Figura 1.2, es cuando una de las fuentes, por ejemplo Y , es conocida sin codificar por el decodificador. Entonces una razón no mayor que $H(X|Y)$ es suficiente para comprimir X . Aunque en el caso asimétrico es donde se logra una mayor compresión, este enfoque puede no ser aconsejable para algunas aplicaciones en las redes de sensores ya que requeriría tener disponible un nodo que brinde sus datos sin codificar, para poder decodificar la información del resto de los nodos, lo cual crearía un desbalance en la energía de la red pues algunos nodos tendrían un mayor consumo que otros. Dos soluciones se han planteado a este problema, la primera consiste en intercambiar periódicamente el rol de los nodos, de manera que la responsabilidad de actuar como información lateral no recaiga solo en un nodo sino que se remplace cada cierto tiempo. La segunda solución consiste en utilizar una estrategia simétrica. Este caso se refiere a

cualquier punto en la línea entre los puntos A y B de la Figura 1.2. En este enfoque ambas fuentes son codificadas a una razón $R_X > H(X|Y)$ y $R_Y > H(Y|X)$, siempre que se cumpla que $R_X + R_Y \geq H(X, Y)$, de manera que los datos codificados de ambas fuentes son decodificados de forma conjunta.

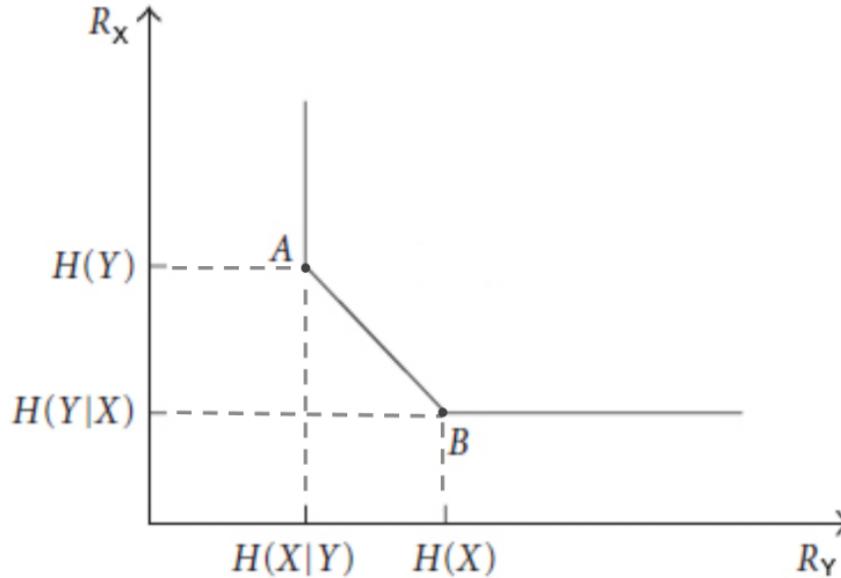


Figura 1.2: Región de razón para codificación.

1.2 Esquema Slepian-Wolf

Para una mayor comprensión del teorema de Slepian-Wolf se toma como referencia un caso asimétrico, donde Y actúa como información lateral en el decodificador. La codificación verdadera se realizará en X para distribuir 2^{nR_X} palabras de código entre la secuencia típica X distinguible si Y está adicionalmente disponible en el decodificador. El codificador va a descomponer la secuencia X en el par (X_1, X_2) y va a codificar solo X_1 con R_X bits, descartando X_2 . Luego el decodificador debe determinar X_2 , con una pequeña probabilidad de error, utilizando la correlación entre X_2 y Y.

Para lograr la compresión un principio fundamental es determinar un modelo de correlación común entre los datos. La correlación entre dos fuentes puede ser modelada como un canal de correlación virtual donde X es la entrada y Y es la salida del canal (como se muestra en la Figura 1.3). El canal es descrito como la probabilidad de error p, la cual se define como la probabilidad de que X sea diferente de Y. De esta manera una probabilidad p está dada por

una alta correlación, lo que hace posible la codificación con una razón baja. La correlación es por lo general modelada, en la literatura, como un canal virtual simétrico binario (BSC) o un canal con ruido gaussiano blanco (AWGN).

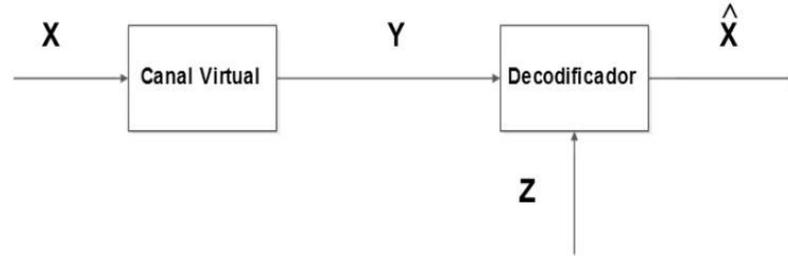


Figura 1.3: Canal de correlación virtual entre X y Y. Modificado de [4].

Debido a esto, en la literatura se sugiere usar los principios de codificación de canal para implementar la codificación de la fuente. Esto fue sugerido por primera vez por Wyner en [5, 6]. La idea general es que de la misma forma que la codificación del canal expande la razón para proteger la señal de los ruidos del canal, se puede usar de forma opuesta para reducir la razón de forma controlada. La técnica se basa en un proceso llamado *binning*¹ donde todas las posibles salidas de una señal de entrada son colocadas en *cosets*² separados, o *bins*. Esta teoría fue luego llamada “codificación de fuente distribuida usando síndromes” (DISCUS) en [8, 9]. El proceso de codificación utilizando síndromes es mostrado en la Figura 1.4.

Es fácil relacionar esta forma de codificación con un código de bloque binario lineal (k, n) , el cual tendría 2^{n-k} síndromes distintos, cada uno indicando un *bin* de 2^k palabras binarias de longitud n . Cada *bin* es un código *coset* del código de bloque binario lineal, preservando las propiedades de distancia Hamming del código lineal original en cada *bin*. De esta manera, una secuencia de n bits de entrada es mapeado dentro de sus correspondientes bits síndrome, logrando una razón de compresión de $n:(n-k)$. Esta aproximación es conocida como “esquema de Slepian-Wolf” [2].

¹ En inglés *bin* significa depósito o recipiente, por lo que el término *binning* se refiere a un proceso en el que la fuente es dividida (fragmentada) en diferentes muestras llamadas *cosets* o *bins*.

² En matemática un *coset* es un subconjunto que viene como resultado de la multiplicación de un elemento por todos los demás elementos de dicho conjunto.

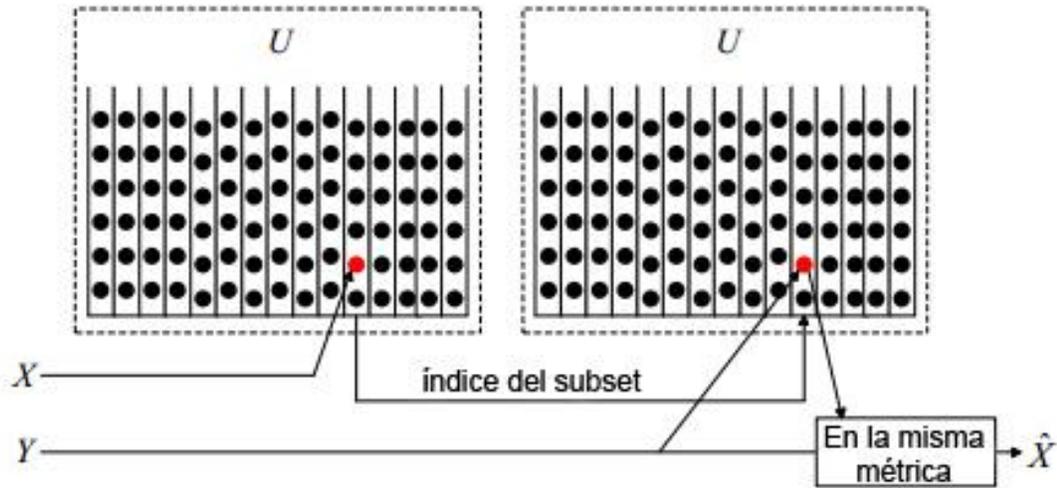


Figura 1.4. Proceso de codificación y decodificación usando *cosets*. Modificado de [15].

La codificación de fuente distribuida usando síndromes puede implementarse de varias formas dependiendo de la técnica de codificación de canal escogida. Las técnicas más comúnmente usadas en la codificación de fuente distribuida son los códigos de bloque lineales, los códigos convolucionales y los códigos concatenados. Sin embargo, como la correlación entre las fuentes es entendida como un “canal virtual”, un código de canal que tenga una mejor capacidad correctora, en la codificación de fuente distribuida, va a necesitar un menor grado de correlación entre las fuentes para obtener un mejor rendimiento en cuanto a la probabilidad de error. Por tal motivo las investigaciones para la implementación de la codificación distribuida en las redes de sensores se han enfocado en la codificación LDPC [16-21] y la codificación Turbo [22-24]. Ambos alcanzan rendimientos muy cercanos, aunque con una relativa complejidad en comparación con el resto de los códigos de canal. No obstante, los diseños LDPC son más flexibles y menos complejos, y por tanto más rápidos, fáciles y precisos. Esto ha hecho que los códigos LDPC sean los más investigados para la implementación de la codificación de fuente distribuida, aunque por su rendimiento también son sugeridos los códigos Turbo.

1.3 Esquema Wyner-Ziv

A pesar de que el teorema de Slepian-Wolf resultó un increíble aporte en la teoría de la codificación de fuente, tiene como limitante que solo puede aplicarse en fuentes discretas. Wyner y Ziv en 1976 en [5] ampliaron esta teoría para fuentes de valores continuos, e

introdujeron la distorsión para lograr entropía finitas. La idea principal es la introducción de un paso de cuantificación antes de la codificación de Slepian-Wolf, como se ilustra en el esquema representado en la Figura 1.5. El paso de la cuantificación forma parte de la teoría de la razón-distorsión y puede realizarse de varias formas dependiendo de la distorsión y la memoria de la entrada. De esta manera se enuncia la codificación de fuente distribuida como un problema de codificación de fuente (codificación Wyner-Ziv) y codificación de canal (codificación Slepian-Wolf).

La cuantificación es el proceso de mapear una secuencia vectorial o escalar de valores discretos o continuos, producidos por una fuente, a un set de símbolos digitales que pueden ser transmitidos o almacenados usando un número finito. En el caso de fuentes continuas (con valores en \mathbb{R} o \mathbb{R}^n) la cuantificación debe necesariamente usarse si la salida de la fuente es transmitida sobre un canal digital. En este caso en general, es imposible reproducir exactamente la salida de la fuente original, por lo que se habla entonces de la codificación/compresión con pérdida [25].

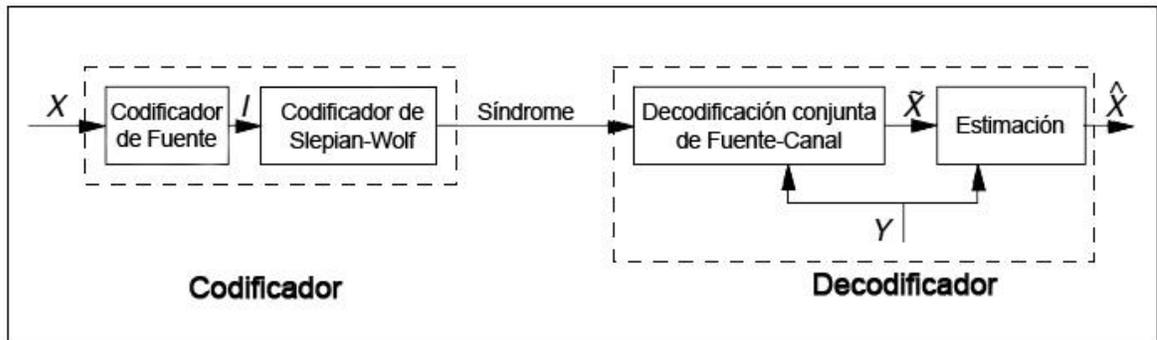


Figura 1.5: Diagrama de bloques de un codificador Wyner-Ziv genérico. Modificado de [2].

Básicamente existen dos tipos de cuantificación: la escalar y la vectorial. La cuantificación escalar codifica puntos de los datos de forma individual [26], mientras que la cuantificación vectorial agrupa los datos de entrada en vectores, cada uno codificados como un todo [25].

En [7, 27, 28] Pradhan y Ramchandran proponen un esquema Wyner-Ziv aplicable a redes de sensores mediante el uso de la cuantificación escalar (como códigos de fuente) y códigos trellis (como códigos de canal). Este esquema fue la base de estudios de diseños más complejos buscando mejores rendimientos. Sin embargo, las investigaciones posteriores

proponen esquemas basados en cuantificación vectorial Lattice, principalmente la cuantificación de códigos Trellis (TCQ) [29-31].

1.4 Recolección de datos en redes de sensores

El problema de recolección de los datos en las redes de sensores ha sido analizado fundamentalmente desde el punto de vista del enrutamiento. Se han presentado diversas estrategias en dependencia de la aplicación y las características propias de la red. Desde esta perspectiva, a la hora de seleccionar un modelo de recolección de datos y/o algoritmo de enrutamiento para las redes de sensores se deben tener en cuenta los siguientes principios [32]:

- Debido al relativamente gran número de nodos, no es posible construir un esquema de direccionamiento global, puesto que el costo de mantener la identidad de cada uno de ellos es alto en términos de cabeceras.
- Casi todas las aplicaciones de las redes de sensores requieren un flujo de datos desde los sensores hacia una estación base. Es por ello que se considera fundamentalmente un enrutamiento multipunto a punto, desde los nodos hacia el *terminal*³. Esto, sin embargo, no implica que haya flujos de datos de otra forma (por ejemplo: multicast, o punto a punto).
- Teniendo en cuenta las características de las aplicaciones se establece el comportamiento del flujo de los datos de los nodos a la estación base. De manera general este flujo en las redes de sensores se puede comportar de tres formas diferentes:
 - Trasmisión periódica, en donde los sensores periódicamente envían información a la estación base. Este comportamiento es utilizado principalmente en aplicaciones de monitorización.

³ El termino *terminal*, se utiliza para designar un nodo que tiene el rol de recolectar los datos en una red o una sección de la red. En la literatura también se utiliza el término *sink* o *sumidero* por su significado en español. En algunas topologías de red, el terminal puede ser la estación base.

- Trasmisiones bajo demanda, en donde los sensores solo envían información cuando es solicitada por la estación base. Estas transmisiones se emplean en aplicaciones de seguimiento a eventos.
- Trasmisiones ante la detección de eventos, en donde los sensores solo envían información ante la detección de un evento o cuando los datos medidos superan un umbral predefinido. Es utilizada en aplicaciones de vigilancia.

Atendiendo a estos desafíos se han propuesto varios algoritmos para el problema del enrutamiento de los datos en las redes de sensores. Estos mecanismos consideran las características y requisitos de la arquitectura de las redes y sus aplicaciones. La bibliografía clasifica estos esquemas de enrutamientos para redes de sensores, como: *enrutamientos planos, jerárquicos, y basados en ubicación* [33].

En el *enrutamiento plano*, el terminal envía “consultas” a las regiones correspondientes y espera por los datos de los sensores ubicados en las regiones seleccionadas. Como los datos son requeridos por consulta, nombramientos basados en atributos son necesarios para especificar las propiedades de los datos. SPIN, en inglés: *Sensor protocols for information via negotiation*, presentado en [34] fue el primer protocolo *de enrutamiento plano* desarrollado, en el cual se considera la negociación de los datos entre nodos con el fin de eliminar redundancias y ahorrar energía.

Luego fue diseñada la *Difusión Dirigida* [35] que se ha convertido en el principal exponente de este esquema. La idea principal es difundir los datos a través de los nodos usando un esquema de nombramiento de los datos. El objetivo es prescindir de operaciones innecesarias en la capa de red que permita un mejor ahorro de energía.

En la *Difusión Dirigida*, para solicitar una consulta, un “interés”⁴ es definido usando una lista de valores y atributos tales como: el nombre del objeto, intervalo, duración, área geográfica, etc. La solicitud de una información es difundida desde *terminal* al resto de la red a través de sus vecinos, de manera que se retransmite el “interés” o “solicitud” entre todos los nodos en la red.

⁴ En la *Difusión Dirigida* la solicitud de una información es definida como “interés”, el cual determina que datos son los que al *terminal* le interesa recolectar.

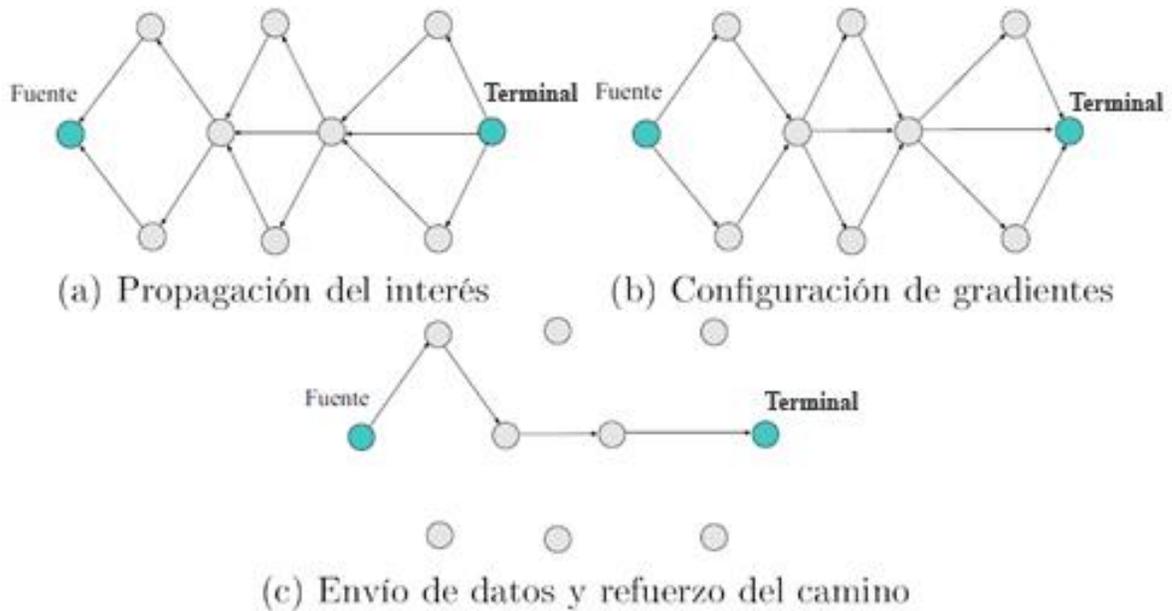


Figura 1.6: Fases del protocolo de Difusión Dirigida. (a) El terminal propaga en la red cuál es su interés. (b) Cada nodo retransmite la respuesta de la fuente con un gradiente adicional. (c) a partir de los gradientes recibidos el terminal determina cual es la mejor ruta y envía por esta nuevamente su interés reforzando el camino seleccionado; la fuente envía todos sus datos por este camino.

Cada nodo que recibe dicho “interés” lo puede almacenar para su uso posterior. El “interés” almacenado es usado para comparar los datos recibidos o generados con los valores solicitados.

Los datos involucrados en el “interés” transmitido tienen varios campos *gradientes*. Un gradiente es una respuesta de un vecino de que el interés fue recibido. El gradiente está caracterizado por la razón de datos, la duración y el tiempo de vencimiento derivado del campo de “interés” recibido. De manera, que utilizando intereses y gradientes las rutas son establecidas entre el *terminal* y las fuentes.

Varias rutas pueden establecerse a fin de que una sea elegida. El *terminal* reenvía el mensaje de interés original por la ruta seleccionada, para reforzar el camino a seguir, y el nodo fuente envía los datos por esta ruta, como se ilustra en la Figura 1.6.

Muchos otros protocolos se han propuesto basados en la *Difusión Dirigida* o conceptos similares a este, en donde las rutas son establecidas a partir de un gradiente que se basa en el costo del enlace.

Por su parte, el objetivo principal del *enrutamiento jerárquico* es mantener un eficiente consumo de energía de los nodos involucrados en comunicaciones de múltiples saltos [36]. Para ello utiliza técnicas como la formación de clústeres. La formación de clústeres, consiste en dividir la red en grupos de nodos llamados clúster. Para la formación de estos grupos o clústeres se tiene en cuenta fundamentalmente la conservación de la energía y la proximidad de los nodos a la cabeza del clúster.

La *jerarquía de clúster adaptativa de baja energía*, LEACH (en inglés: *Low Energy Adaptive Clustering Hierarchy*) propuesto en [37], es uno de los algoritmos de enrutamiento jerárquicos más populares para redes de sensores. LEACH elige aleatoriamente un número reducido de nodos como cabecera de clúster, este rol se intercambia entre los nodos cada cierto tiempo con el fin de balancear el consumo de energía entre los nodos. LEACH utiliza dos técnicas de acceso al medio: TDMA (acceso al medio por división de tiempo) y CDMA (acceso al medio por división de código) [38]. La combinación de estas técnicas permite la comunicación en dos dominios, uno entre los nodos dentro del clúster y otro entre los nodos cabecera y la estación base, respectivamente. La recolección de la información es centralizada y se ejecuta periódicamente. Por ello este protocolo es apropiado para aplicaciones de monitorización constante.

Como *Enrutamiento basados en localización* son clasificados varios protocolos que requieren la información de la localización de los nodos. En muchos casos la información de la ubicación es necesaria para calcular la distancia entre dos nodos particulares con el propósito de poder calcular el consumo de energía. Por ejemplo si la región muestreada por los sensores es conocida, usando la ubicación de los nodos, las consultas pueden ser difundidas solo para una región particular, eliminando un número significativo de transmisiones [39].

Sin embargo, cuando las redes de sensores utilizan Codificación de Fuente Distribuida, además del enrutamiento otros principios deben tenerse en cuenta a la hora de implementar una estrategia de recolección de los datos correlacionados y comprimidos por esta técnica de codificación.

1.5 Principios para la recolección de datos en redes de sensores que emplean Codificación de Fuente Distribuida.

Una de las principales problemáticas que se presenta en la recolección de datos en redes de sensores que implementan la Codificación de Fuente Distribuida es mantener un modelo de correlación global, que permita la decodificación sin errores de los datos en la red. Aunque se han estudiado diferentes modelos de correlación para aplicaciones de redes de sensores específicas [5, 40-42] en algunos casos se hace complejo llevar a la práctica algunas de estas aplicaciones y continuar utilizando una probabilidad conjunta para toda la red o una función de densidad en las redes de sensores especialmente si existen pocas áreas para probar o poca información acerca de la topología de la red.

Determinar el modelo de correlación se dificulta aún más cuando la red ocupa una gran dimensión ya que su comportamiento difiere mucho entre los dispositivos de la red. Una alternativa muy empleada en las redes de sensores para mantener la correlación entre los nodos es fragmentar la red en clústeres de nodos correlacionados, donde cada clúster tiene un nodo padre y muchos nodos hijos [2]. Los nodos hijos envían la información comprimida a los nodos padres el cual procesa y/o retransmite la información recolectada en el clúster a la estación base. Los clústeres se usan en redes de gran tamaño donde se hace difícil la correlación entre todos los nodos de la red y por tanto lo más conveniente es fragmentar la red en varios segmentos para facilitar la comunicación. Desde este enfoque es más viable implementar la estrategia de codificación asimétrica, de manera que los datos captados por el nodo padre son utilizados como la información lateral para decodificar la información de los nodos hijos en el clúster.

El primer acercamiento al despliegue de la Codificación de Fuentes Distribuida en una red lo dio Han en [43], donde se demuestra que las condiciones necesarias y suficientes para transmitir fuentes correlacionadas en un enlace son definidas en la ecuación (1.2). Esta expresión expone que para poder transmitir una fuente comprimida por la Codificación de Fuente Distribuida, en un enlace entre dos nodos i y j , se debe cumplir que:

- Para recuperar la fuente sin pérdidas, la capacidad del canal debe ser mayor que la razón R_i . Es decir que los datos deben ser codificados a una razón menor que la capacidad del canal por el cual se va a transmitir.

- De igual modo para que la compresión se realice bajo los principios de la Codificación de Fuente Distribuida la razón de compresión R_i tiene que estar dentro de los límites de Slepian-Wolf, es decir, siendo mayor o igual a la entropía condicional de las fuentes a codificar.

$$H(X_S|X_{S^c}) \leq \sum_{i \in S} R_i \leq \sum_{i \in S, j \in S^c} C_{ij} \quad (1.2)$$

A partir de la ecuación (1.2), Li y Ramamoorthy en [14] analizan la optimización de la energía en la transmisión a partir del criterio de que la capacidad de un enlace entre un nodo i y un terminal puede ser escrito como: $C_i(P_i) \equiv \log(1 + \gamma_i P_i)$, con γ_i como la ganancia del canal y la potencia de ruido normalizada a 1. Por lo que la asignación de la energía optima puede ser escrita como $P_i^* = (2^{R_i^*} - 1)/\gamma_i$ ⁵. Por lo que el problema de minimizar la suma de energía en la red puede también ser escrita como:

$$\min_{R_1, \dots, R_N} \sum_{i=1}^N P_i = \sum_{i=1}^N (2^{R_i} - 1)/\gamma_i \quad (1.3)$$

$$\text{Sujeto a } (2^{R_i} - 1)/\gamma_i \leq P_{max}, \forall i$$

$$(R_1, \dots, R_N) \in SW_N$$

En la ecuación (1.3) se declara que la minimización de la energía total de la red, usando la Codificación de Fuentes Distribuida está sujeta a las condiciones de que R_i esté dentro de los límites de la región de Slepian-Wolf y la energía usada en la transmisión en el nodo i tiene que ser menor que la energía residual de este nodo (P_{max}). Como la menor razón de codificación R_i se obtiene en la estrategia de codificación asimétrica, explotando esta estrategia se obtiene el menor consumo de energía en la red. Sin embargo, aunque se han realizado investigaciones para lograr códigos que implementen la Codificación de Fuente Distribuida para múltiples fuentes, en la práctica solo se han podido lograr códigos para dos fuentes. Por tal razón los algoritmos que implementan la codificación de Slepian-Wolf en redes proponen, para imponer las restricciones anteriores, un modelo que llaman *pairwise*

⁵ En este capítulo todas las variables señalizadas por * significan que es la versión óptima de la misma.

(en español significa: formar pares), para que dos fuentes puedan ser decodificadas de forma conjunta. Bajo este modelo, antes que la transmisión comience, se determina el par de fuentes que son decodificadas juntas en cada tiempo y se determinan las razones de las fuentes y los códigos correspondientes. Durante la transmisión, las fuentes codifican los mensajes por separado (sin comunicación entre sí) usando el código pre-asignado y el terminal realiza la decodificación conjunta a las combinaciones pre-asignadas. El problema de la asignación de recursos es determinar las combinaciones pares óptimas y las razones para los sensores de forma que la suma de energía es minimizada.

Según [14] una asignación de la razón tiene la propiedad *pairwise* si permite que el terminal decodifique las fuentes en un modo *pairwise*. Específicamente, la asignación de la razón se dice que se satisface la propiedad *pairwise* si para cada fuente X_i , $i = 1, \dots, N$ existe una secuencia ordenada de fuentes $(X_{i_1}, X_{i_2}, \dots, X_{i_k})$ tal que:

$$R_{i_1} \geq H(X_{i_1}), \quad (1.4)$$

$$R_{i_j} \geq H(X_{i_j} | X_{i_{j-1}}), \quad \text{para } 2 \leq j \leq k, y \quad (1.5)$$

$$R_i \geq H(X_i | X_{i_k}) \quad (1.6)$$

Tal asignación de la razón admite la posibilidad de que cada fuente pueda ser reconstruida en el decodificador para resolver una secuencia de operaciones en el caso asimétrico de la codificación Slepian-Wolf. Como la menor razón (menor gasto de energía) se consigue en el caso asimétrico, la mayoría de los modelos *pairwise* propuestos se basan en este concepto.

Un principio primordial para optimizar el consumo de energía en el enrutamiento de las redes de sensores es la cooperación de los nodos para tramitar la información hasta la estación base, de modo que los nodos intermedios tienen que retransmitir los datos de sus vecinos. Desde esta perspectiva el peso w_{ij} de cada enlace (i, j) entre nodo, desde el nodo fuente hasta el terminal, deben tenerse en cuenta para encontrar el camino óptimo. Según [44] el peso de un enlace entre los nodos es una función que depende de la distancia que separa los nodos. Por lo que el problema de la optimización puede ser escrito como la ecuación (1.7).

$$\min \sum_{(i,j) \in E} F(z_{ij}) w_{ij} \quad (1.7)$$

subjeto a $0 \leq z_{ij} \leq C_{ij}, \forall (i,j) \in E$ (*Restricción de la capacidad*)

$$\sum_{j \setminus (i,j) \in E} z_{ij} - \sum_{j \setminus (j,i) \in E} z_{ji} = \sigma_i, \forall i \in V \text{ (*Restricción de balance de flujo*)}$$

$$(R_1, R_2, \dots, R_N) \in SW_N \text{ (*Restricción de Slepian – Wolf*)}$$

Donde z_{ij} representa el flujo entre los nodos i y j ; y $\sigma_i = R_l$ si i es el nodo fuente l^{th} , $\sigma_i = -\sum_{i=1}^N R_i$ si i es el terminal, sino $\sigma_i = 0$.

Si no hay restricción de capacidad, la solución del problema tiene una interpretación y forma simple. La idea básica es que en la ausencia de restricción de capacidad no existe necesidad de separar el flujo en diferentes enlaces. Una vez que una ruta desde la fuente dada al terminal con costo mínimo es encontrada, la fuente simplemente enruta todos los datos a través del camino. Como el peso del enlace está estrechamente relacionado con la distancia entre los nodos de ese enlace, el camino óptimo es el que involucra las rutas más cortas (es decir la menor cantidad de saltos hasta el terminal, con los enlaces entre nodos de menor distancia). No obstante, la elección de estas rutas también está sujeta a la energía residual que tengan los nodos, la cual tiene que ser suficiente para transmitir los datos a la razón de codificación de Slepian-Wolf como establece la ecuación (1.3).

Si bien un esquema de enrutamiento que utilice los caminos más cortos desde la fuente al terminal y explote las propiedades de las estrategias de codificación asimétricas consigue el uso mínimo de la energía total de la red, no logra maximizar su tiempo de vida. Por una parte, las estrategias asimétricas explotadas en los métodos *pairwise* hacen que los nodos más cercanos a la estación base actúen en la mayoría de los escenarios como proveedores de la información lateral, enviando sus datos sin codificar, por lo que el consumo de energía por este concepto será mayor que la del resto de los nodos en la red. Por otra parte, incluso si no tuvieran la función de proveer la información lateral, la sobrecarga relacionada con el enrutamiento de la información del resto de los nodos en la red es suficientemente significativa para que en estos nodos el tiempo de vida sea menor. Para optimizar este parámetro en [11] se propone el uso de la codificación asimétrica conmutando

periódicamente la razón de codificación entre los nodos, cada T intervalo de tiempo. Es decir, intercambiando el rol de proveer la información lateral para la decodificación de los datos.

1.6 Estrategias de recolección de datos en redes de sensores que emplean Codificación de Fuente Distribuida.

Atendiendo los principios dados en el epígrafe 1.5, se puede resumir, la recolección de datos en redes de sensores que emplean la Codificación de Fuente Distribuida en dos estrategias fundamentales. La elección de estas estrategias depende de las características de la aplicación de la red, e involucra además la selección del protocolo de enrutamiento a utilizar en la red.

1.6.1 Estrategia de recolección de datos secuencial

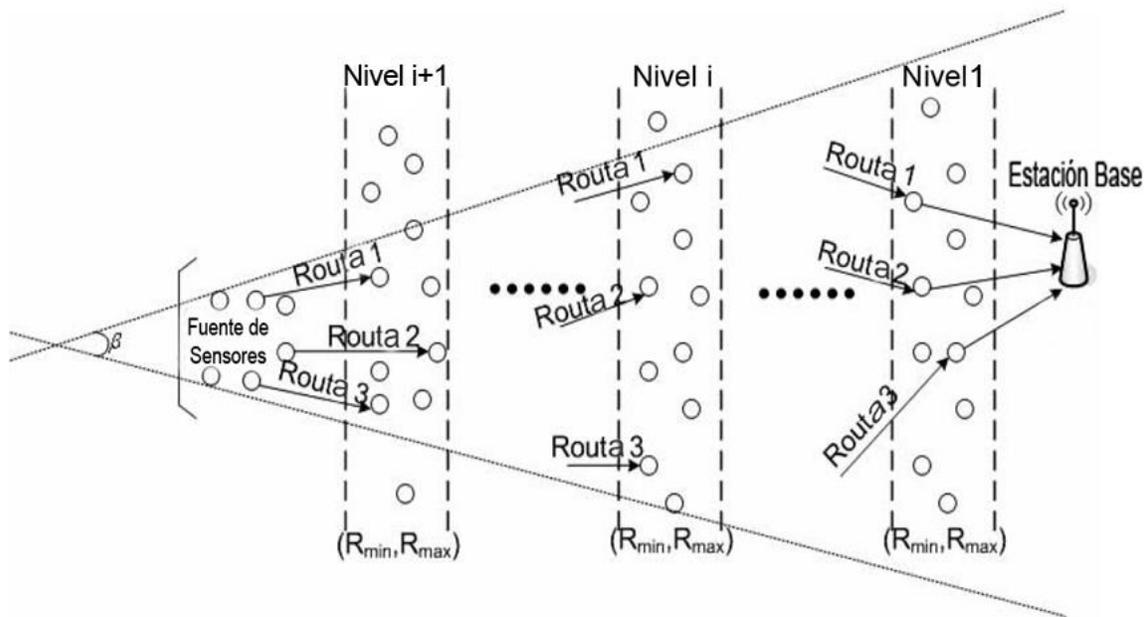


Figura 1.7: Estructura de una estrategia secuencial. Modificado de [45]

La estrategia secuencial se basa en el principio de *pairwise* usando una codificación asimétrica. Este método se basa en determinar el *árbol de ruta óptima* para cada nodo con respecto a la estación base. El *árbol de ruta óptima* especifica a través de qué nodos debe un nodo X enrutar sus datos hasta el terminal y cuál información se debe tomar como referencia para decodificar los datos de este nodo. De esta forma la red se segmenta en niveles, como se ilustra en la Figura 1.7, donde los nodos del nivel i deben enrutar los datos de los nodos

del nivel $i + 1$ hasta la estación base, según el camino establecido por el *árbol de ruta óptima*. De igual manera la estación base decodifica la información de los nodos del nivel $i + 1$ tomando como referencia la información enviada por los nodos del nivel i .

En esta estrategia diferentes protocolos de enrutamiento plano pueden utilizarse para determinar el *árbol de ruta óptima* de cada nodo a la estación base.

Un ejemplo claro de estrategia secuencial es el presentado en [46], en el cual se propone un esquema de enrutamiento para la optimización de una topología de red tipo cadena. En esta topología de red los nodos son desplegados de forma unidimensional. Para mantener la correlación entre los nodos, la red es segmentada en clústeres. La Figura 1.8 representa un clúster dentro de la red. En su trabajo Guogang Hua propone un modelo *pairwise* que explota la estrategia asimétrica de la codificación de Slepian-Wolf, dentro de cada clúster, siguiendo el siguiente procedimiento:

El nodo 1 codifica su información X_1 a una razón de $R_{X_1} = H(X_1|X_0)$ y lo envía al nodo 0 (que tiene el rol de cabecera del clúster). Una suma de chequeo del dato original X_1 es también enviado.

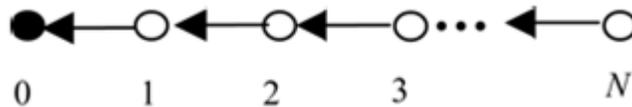


Figura 1.8: La forma de un clúster en la red de sensores tipo cadena de [46].

El nodo 0 usa su información X_0 como información lateral para con los bits del síndrome recibidos del nodo 1 decodificar X_1 . Después de la decodificación, el nodo 0 calcula la suma de chequeo de los datos decodificados. Si la suma de chequeo es la misma que la suma de chequeo de los datos decodificados, se asume que X_1 fue recibido exitosamente por el nodo cabecera, de otra forma el nodo cabecera solicita al nodo 1 que le reenvíe sus datos.

El nodo 2 codifica sus datos a una razón de $R_{X_2} = H(X_2|X_1)$ y lo envía al nodo 0 a través del nodo 1. En este tiempo ya el nodo 0 debe tener la información del nodo 1 la cual utiliza como información lateral para decodificar los datos del nodo 2. Si un error ocurre le solicita al nodo 2 le reenvíe sus datos.

El proceso continúa para todos los nodos dentro del clúster. Por tanto todos los nodos pueden codificar su información a una razón de $R_{X_n} = H(X_n|X_{n-1})$ y enviarla al nodo cabecera.

Modelos *pairwise* similares al propuesto por Guogang Hua son propuestos en la literatura para diferentes esquemas de enrutamiento. No obstante, un criterio importante que se puede visualizar desde el método propuesto por Guogang Hua es que en algunos escenarios la Codificación de Fuente Distribuida no puede ser usada sin una adecuada sincronización entre los nodos de una red de sensores. Si no existe una correcta sincronización de los datos se corre el riesgo de que se produzca una decodificación errónea (o incluso que la decodificación no se logre realizar), ya que el decodificador necesita recibir la información de las fuentes que están correlacionadas entre sí de manera simultánea para poder reconstruir los datos de manera correcta. Esta condición debe tenerse en cuenta a la hora de elegir los algoritmos de enrutamiento a implementar en la red [2].

El principal problema de esta estrategia es el tiempo de vida de la red, ya que los nodos que están en los niveles más cercanos al terminal deben retransmitir toda la información, junto a sus propios datos, lo que lo pone en desventaja con respecto a la energía residual del resto de los nodos de la red.

1.6.2 Estrategia de recolección de datos basada en clúster.

La estrategia de clúster se basa en fragmentar la red en secciones o clúster para de este modo mantener un modelo de correlación entre los datos recolectados en dicho segmento. Cada clúster cuenta con un nodo cabecera que cumple la función de recolectar los datos del resto de los nodos en el clúster y retransmitirlo al terminal (o estación base). Los datos del nodo cabecera suelen ser utilizados como referencia para decodificar los datos del resto de los nodos en la red. A diferencia de la estrategia secuencial, tanto la codificación simétrica como la asimétrica pueden ser utilizadas, aunque por su rendimiento se recomienda la codificación asimétrica. En los casos que la aplicación y la correlación de los datos de la red lo permitan, el nodo terminal puede usar sus datos como información lateral para decodificar los datos de los nodos cabecera y de esta forma decodificar los datos del resto de los nodos en cada clúster. Esto, aunque permite un ahorro mayor de energía provoca una mayor probabilidad de error

en la decodificación y en clúster demasiado alejados del terminal puede ser imposible de implantar.

En esta estrategia, protocolos de enrutamientos jerárquicos son implementados para establecer los clústeres y transmitir los datos. Aunque en la mayoría estos protocolos parámetros como la distancia o la ubicación de los nodos son utilizados a la hora de conformar los clústeres, se proponen modificaciones de estos protocolos donde se tenga en cuenta la correlación de los datos como principio fundamental a la hora de segmentar la red.

Para prolongar el tiempo de vida de la red en la estrategia de recolección basada en clúster, [11] propone intercambiar el rol de nodo cabecera de clúster cada un periodo de tiempo, de manera que se comparta la carga de tráfico por igual entre los nodos. Bajo este principio, [47] recomienda utilizar el protocolo de enrutamiento LEACH, el cual establece un método para intercambiar el rol de nodo cabecera de forma dinámica.

CAPÍTULO 2. FUNCIONES Y ALGORITMOS

En este capítulo se explican las funciones y algoritmos de los esquemas de codificación/decodificación de fuente distribuida y de las estrategias de recolección de datos implementados.

Los algoritmos de codificación presentados en este capítulo, son una modificación de los propuestos en [48]. En estos diseños solo se implementa el esquema de Slepian-Wolf, asumiendo que previamente se ha realizado un proceso de cuantificación en el cual se han convertido las fuentes continuas en discretas.

2.1 Algoritmos de Codificación de Fuente Distribuida

2.1.1 Modelo de Correlación

Lo primero a tener en cuenta en el despliegue de la Codificación de Fuente Distribuida es la correlación entre las fuentes. Como se ha explicado, esta correlación puede modelarse como un canal virtual. En un sistema de dos fuentes (X y Y) la fuente Y puede verse como una versión ruidosa de la fuente X (y viceversa), de manera que la diferencia de valores de X con respecto a Y está dada por los errores inducidos por el canal. Diferentes tipos de canales pueden usarse para modelar la correlación, en el presente trabajo se ha utilizado un Canal Simétrico Binario (BSC), como el que refleja la Figura 2.1, donde la probabilidad de los errores introducidos por el canal p va a estar dada por el grado de correlación ρ que tengan las fuentes, como refleja la ecuación (2.1). Este modelo de correlación es empleado en la simulación para generar fuentes correlacionadas que permitan comprobar los algoritmos de codificación/decodificación.

$$p = \frac{1-\rho}{2} \quad (2.1)$$

Donde ρ es el coeficiente de correlación entre las fuentes X y Y .

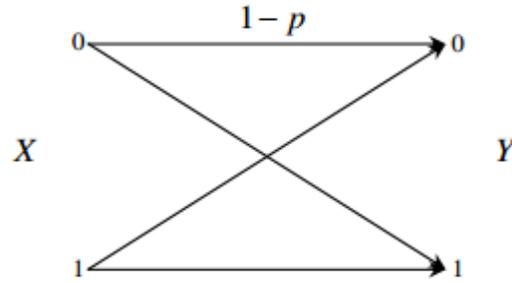


Figura 2.1: Modelado de correlación usando BSC.

2.1.2 Codificador asimétrico

Como se ha explicado el caso asimétrico para dos fuentes (X y Y) ocurre cuando una de las fuentes se transmite codificada y la otra es utilizada por el decodificador como información lateral para descifrar la fuente codificada.

La Codificación de Fuente Distribuida usando códigos de bloque se basa en el principio de síndromes. De manera que una fuente X de n bits es codificada en un síndrome de m bits usando la ecuación (2.2), donde H^T es la traspuesta de la matriz de chequeo de paridad del código con un tamaño $[m, n]$, cumpliéndose la condición de que la cantidad de bits de la palabra codificada debe ser menor que el número de bits contenido en la palabra de código original ($m < n$).

$$s = xH^T \quad (2.2)$$

2.1.3 Decodificador asimétrico

En este diseño, para el proceso de decodificación se propone el uso de una tabla, como la mostrada en la Figura 2.2 (que cumple la función del libro de código), con tantas columnas como posibles síndromes (2^m) y tantas filas como posibles valores de la fuente puedan existir en un síndrome ($2^n/2^m$).

Esta tabla es utilizada por el decodificador para almacenar todos los posibles valores de la fuente en cada uno de los síndromes. Cuando el síndrome de la fuente $X(S_X)$ llega al decodificador, este busca en la columna correspondiente cuál es el valor más correlacionado con la información de Y que recibe como información lateral y este debe ser el resultado correcto de la decodificación.

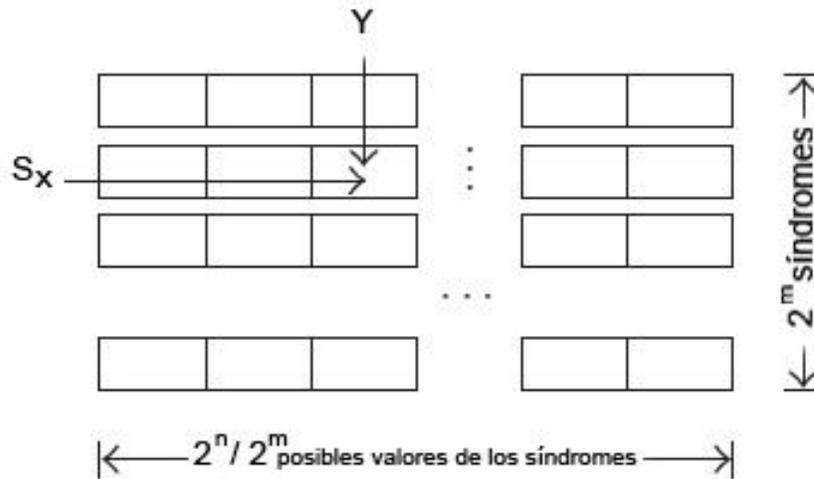


Figura 2.2: Decodificación en un caso asimétrico.

2.1.4 Codificador simétrico

El caso simétrico es una extensión del asimétrico explicado en el epígrafe anterior. En esta estrategia simétrica las dos fuentes (X y Y) son codificadas de manera que la razón total durante el proceso de decodificación siga siendo igual a la entropía conjunta de ambas fuentes ($R = H(X, Y)$). Por lo tanto, se necesitan dos matrices de chequeo de paridad que respondan a las razones de codificación que se requieren para cada fuente.

Por las propiedades de los códigos de bloque lineales es conocido que la matriz generadora (G) y la matriz de chequeo de paridad del código tienen la forma:

$$G = [I_k P_{k \times (n-k)}] \quad (2.3)$$

$$H = [P I_{n-k}]$$

Donde P es la matriz de paridad e I la matriz de identidad del código empleado. A partir de la matriz generadora se obtienen dos nuevas matrices de chequeo de paridad H_1 y H_2 con la siguiente forma:

$$H_1 = \begin{bmatrix} 0_{m_2, m_1} & I_{m_2} & 0_{m_2, n-k_1} \\ P_1 & 0_{n-k_1, m_2} & I_{n-k-1} \end{bmatrix} \quad (2.4)$$

$$H_2 = \begin{bmatrix} I_{m_1} & 0_{m_1, m_2} & 0_{m_1, n-k_1} \\ 0_{m_1, n-k_1} & P_2 & I_{n-k-1} \end{bmatrix}$$

En las matrices de chequeo de paridad H_1 y H_2 , se cumple que m_1 son las m_1 primeras filas de P y m_2 son las m_2 últimas filas de P. La forma de obtener los valores numéricos de m_1 , m_2 y k_1 es mostrada en la ecuación (2.5).

$$\begin{aligned}
 m_1 &= n - R_x \\
 m_2 &= n - R_y \\
 k_1 &= m_1 + m_2
 \end{aligned}
 \tag{2.5}$$

El proceso de codificación para ambas fuentes es similar al explicado en el caso asimétrico solo que en esta estrategia ambas fuentes son codificadas. Por lo que el proceso de codificación para esta estrategia está relacionado con la obtención de los síndromes S_X y S_Y a partir de una variación de la ecuación (2.2), donde cada fuente es multiplicada por la traspuesta de las matrices H_1 y H_2 , respectivamente.

2.1.5 Decodificador simétrico

De forma similar que en el caso asimétrico el decodificador para el caso simétrico se basa en el uso de dos tablas, una para cada fuente, que se obtienen de las matrices de chequeo de paridad H_1 y H_2 . Al igual que para una estrategia asimétrica las tablas obtenidas para el proceso de decodificación van a tener 2^{m_i} columnas que representan todos los síndromes posibles para cada fuente; y $2^n/2^{m_i}$ filas que representan los posibles valores de la fuente para cada síndrome dividido la cantidad de síndromes (el índice i identifica que se utiliza la tabla obtenida de la matriz H_i).

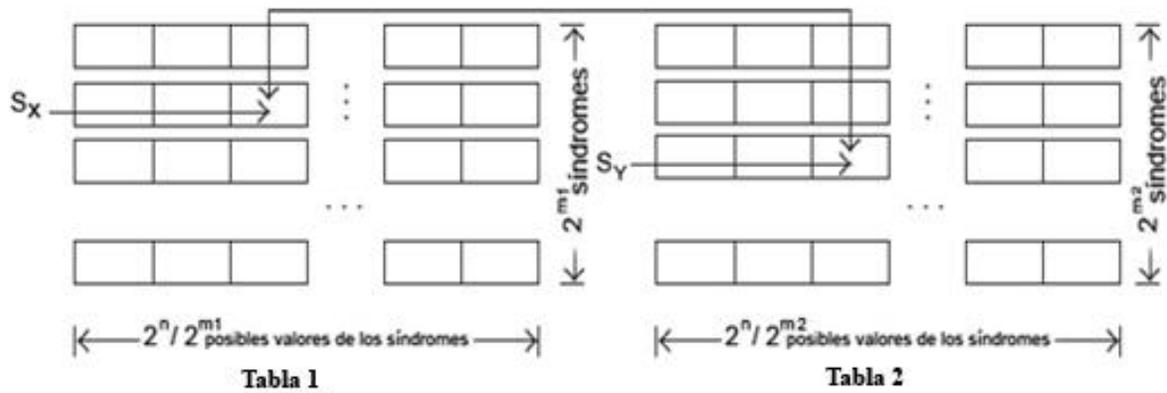


Figura 2.3: Decodificación en un caso simétrico.

Una vez ubicado en cada tabla el síndrome obtenido como resultado del proceso de codificación, el decodificador compara cada valor de la columna del síndrome S_X en la tabla 1 con los valores de la columna del síndrome S_Y en la tabla 2, dando como resultado los valores más correlacionados entre ambas tablas como ilustra la Figura 2.3.

2.2 Estrategias de recolección de datos

Teniendo en cuenta los elementos dados por la literatura para la recolección de datos en redes de sensores que emplean la Codificación de Fuente Distribuida, en el presente trabajo se implementan variantes de la estrategia de recolección de datos secuencial y de la estrategia de recolección de datos basada en clúster.

Las estrategias se analizan en condiciones similares. Ambas son diseñadas para actuar sobre un campo de nodos (en tres dimensiones) teniendo la estación base en el centro de la red. La estación base es un nodo terminal que tiene la función de recolectar/procesar toda la información de red. Por lo que el resto de los nodos deben enrutar estos datos hasta el terminal. Según la aplicación de la red de sensores este nodo puede también hacer mediciones del medio físico que estudia la red, por lo que en estas condiciones su información puede utilizarse como información lateral para decodificar los datos de otros nodos en la red, por lo tanto, este escenario también debe ser analizado en la estrategia de recolección a implementar. En ambas estrategias se implementan usando Codificación de Fuente Distribuida asimétrica.

Otras variantes de estas estrategias, en diferentes dimensiones y con uno o más terminales, pueden encontrarse en la bibliografía citada en los epígrafes 1.5 y 1.6. En ninguno de los casos se propone o implementa un protocolo de enrutamiento específico, sino estrategias generales de recolección de datos.

2.2.1 Estrategia recolección de datos secuencial

El primer paso en la conformación de la estrategia de recolección de datos secuencial es determinar el *árbol de ruta óptima* para cada nodo con respecto a la estación base.

Diferentes algoritmos pueden utilizarse para determinar el *árbol de ruta óptima* de cada nodo al terminal, basado en diferentes parámetros de enrutamiento. En el presente trabajo se utiliza el algoritmo de Dijkstra [49], tomando como peso de enrutamiento la distancias entre los

nodos, dado que este parámetro va ser determinante en el consumo de energía y el grado de correlación entre los nodos en las simulaciones que son realizadas en este proyecto.

El algoritmo de Dijkstra permite calcular caminos de menor longitud entre dos vértices cualesquiera de un grafo, con un peso en cada arista. Teniendo un grafo dirigido ponderado de N nodos no aislados, sea x el nodo inicial, un vector D de tamaño N guardará al final del algoritmo las distancias desde x al resto de los nodos. Los pasos del algoritmo de Dijkstra son:

1. Inicializar todas las distancias en D con un valor infinito relativo ya que son desconocidas al principio, exceptuando la de x que se debe colocar en 0 debido a que la distancia de x a x sería 0.
2. Sea $a = x$ (se toma a como nodo actual).
3. Se recorre todos los nodos adyacentes de a , excepto los nodos marcados los cuales son identificado como v_i .
4. Para el nodo actual, se calcula la distancia tentativa desde dicho nodo a sus vecinos con la siguiente fórmula: $dt(v_i) = D_a + d(a, v_i)$. Es decir, la distancia tentativa del nodo ' v_i ' es la distancia que actualmente tiene el nodo en el vector D más la distancia desde dicho nodo ' a ' (el actual) al nodo v_i . Si la distancia tentativa es menor que la distancia almacenada en el vector, se actualiza el vector con esta distancia tentativa. Es decir: Si $dt(v_i) = D_{v_i} \rightarrow D_{v_i} = dt(v_i)$.
5. Entonces se marca el nodo a como completo.
6. Se toma como próximo nodo actual el de menor valor en D (puede hacerse almacenando los valores en una cola de prioridad) y se vuelve a realizar este algoritmo desde el paso 3, mientras existan nodos no marcados.

Una vez terminado al algoritmo, D estará completamente lleno y se puede determinar cuál es el camino de menor longitud.

A partir de las rutas establecidas por el algoritmo de Dijkstra para cada nodo en la red, se determina como se transmiten los datos a la estación base y la forma en que esta "secuencialmente" debe decodificar estos datos a partir de los que establece como información lateral en la codificación asimétrica.

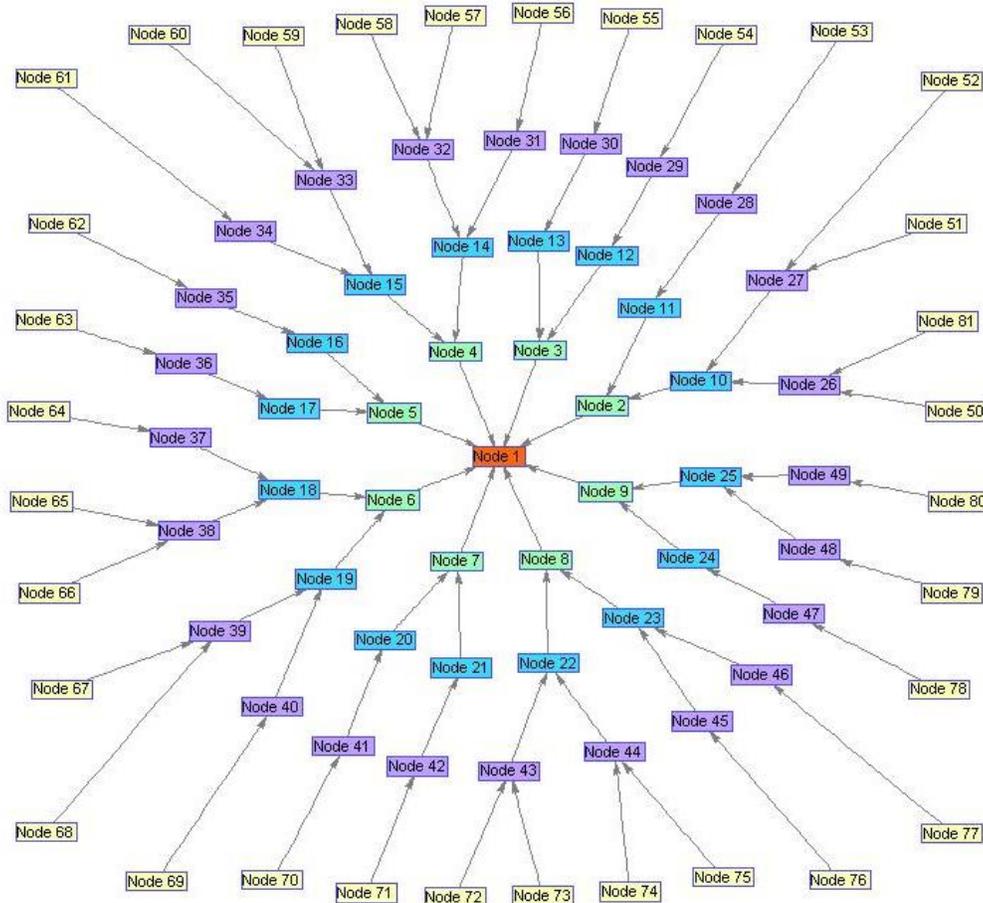


Figura 2.4: Topología de la estrategia secuencial

La Figura 2.4 muestra la estructura de enrutamiento/codificación establecida para una estrategia secuencial de una red de sensores simulada en Matlab. Cada nivel es representado con un color diferente, teniendo como estación base el Nodo 1. Tomando como ejemplo el Nodo 76, este debe enviar sus datos al Nodo 1 a través del Nodo 45. De igual modo el Nodo 45 debe enviar su información y la del Nodo 76 por medio del Nodo 23. El Nodo 23, envía su información y las del resto de los nodos subordinados a él (45, 46, 76, 77) al Nodo 8, el cual finalmente la reenvía al Nodo 1.

El Nodo 1 debe primero decodificar la información del Nodo 8 tomando como información lateral su propia información. Una vez decodificada la información del Nodo 8 esta se toma como referencia para decodificar los datos del Nodo 23, y así sucesivamente de forma secuencial hasta decodificar la información del Nodo 76.

2.2.2 Estrategia recolección de datos basada en clúster

El primer paso en esta estrategia es la conformación de los clústeres. En el presente trabajo, se asume una distribución de los nodos de la red de manera que la distancia entre los nodos cercanos (es decir la distancia de un salto) es siempre constante, y que el modelo de correlación de los datos está basado en esta distancia. De manera que para este caso específico los clústeres son establecidos a partir de la ubicación de los nodos. Sin embargo, esta estrategia de recolección de datos también puede implementarse basada en un protocolo de enrutamiento jerárquico que establezca los clústeres de forma dinámica a partir de diferentes parámetros.

En un segundo momento, en cada clúster se selecciona un nodo cabecera. Este nodo tendrá la función de recolectar los datos del resto de los nodos en el clúster y retransmitirlo al terminal (o estación base).

En este trabajo presentamos dos variantes de este método:

- La estrategia de clúster estática: En este esquema los nodos cabecera son elegidos de forma estática, de manera que, durante toda la vida útil de la red, los nodos seleccionados tendrán esta función.
- La estrategia de clúster conmutada: En este esquema el rol de nodo cabecera dentro de un clúster cambia de forma dinámica cada un periodo de tiempo determinado.

Para decodificar los datos se toman dos variantes de estos métodos:

- La estrategia de clúster con información lateral en el terminal: Donde el terminal (o estación base) decodifica la información del nodo cabecera de cada clúster a partir de su propia información.
- La estrategia de clúster con información lateral en el nodo cabecera: Donde los nodos cabecera de cada clúster envían sus datos sin codificar, los cuales son usados por el terminal (o estación base) para decodificar la información del resto de los nodos.

CAPÍTULO 3. ANÁLISIS DE RESULTADOS

En el desarrollo de este capítulo se resumen y analizan los resultados de los experimentos realizados empleando los algoritmos de codificación y de recolección de datos propuestos en el Capítulo 2.

Las estrategias de recolección de datos son evaluadas en función de la probabilidad de error de la decodificación, el consumo de energía y el tiempo de vida de los nodos en la red.

Los experimentos realizados fueron modelados y simulados utilizando la herramienta Matlab.

3.1 Descripción del modelo de simulación.

Para evaluar los algoritmos de codificación de fuente distribuida descrito en el epígrafe 2.1, primeramente, se plantea un caso simple de dos nodos que transmiten su información a un terminal, como el que se muestra en la Figura 1.1 del Capítulo 1. En este escenario los nodos codifican sus datos según el esquema de codificación planteado, mientras el terminal tiene la función de recolectar y decodificar los datos. Para su estudio los algoritmos de codificación/decodificación simétricos y asimétricos fueron implementados para diferentes códigos cíclicos.

Luego se analizan las estrategias de recolección de datos para redes de sensores que implementan la Codificación de Fuente Distribuida presentadas en el epígrafe 2.2 del Capítulo 2. Los experimentos fueron realizados en una red de sensores que abarca un área física de 1 km², con diferentes densidades de nodos, que van desde 25 a 120 nodos por km². A medida que aumenta el número de nodos en la red, disminuye la distancia entre nodos. En todas las estrategias se asume un escenario en el cual en el centro de la red se encuentra un nodo terminal el cual, al igual que en el caso simple de dos nodos, tiene la función de recolectar y decodificar todos los datos de la red.

Las estrategias de recolección de datos presentadas son evaluadas en cuanto al desempeño de la decodificación y el consumo de energía. El comportamiento de la decodificación en la red se analiza a partir de la probabilidad de error de bits y la probabilidad de error de paquetes

decodificados en la red, mientras que para la eficiencia energética se considera el consumo total de la energía y el tiempo de vida de la red.

Para evaluar el comportamiento del consumo de energía de las estrategias de recolección de datos se utilizó el modelo de consumo energético para redes de sensores propuesto en [50], donde se establece que:

La energía disipada en un nodo debido a la transmisión de un paquete de b bits, en una distancia d_{ij} esta dado por:

$$E_{Tx}(b, d_{ij}) = bE_{elec} + bd_{ij}^n E_{amp} \quad (3.1)$$

Donde:

- E_{elec} es la energía disipada por la electrónica de transmisión y recepción del sensor,
- E_{amp} es la energía disipada para amplificar la potencia,
- n es el exponente de pérdida de trayectoria basado en la distancia.

La energía disipada en la recepción de un paquete de b bits en un nodo sensor está dado por:

$$E_{Rx}(b) = bE_{elec} \quad (3.2)$$

El tiempo de vida de los nodos está dado por:

$$Tiempo\ de\ Vida = \frac{C_{bat}}{E_{cons} \times 365 \times 24 \times 60 \times Ite} \text{ [Años]} \quad (3.3)$$

Donde:

- C_{bat} es la capacidad de la batería del sensor.
- E_{cons} es la energía total consumida por el sensor en una iteración.
- Ite es la cantidad de iteraciones que realiza la red en un minuto.

En la ecuación (3.1) la energía de transmisión de un nodo se establece en función de la cantidad de bits a transmitir y la distancia entre el transmisor y el receptor. De manera que los nodos transmiten los datos con una potencia que garantiza que la relación señal a ruido sea constante en toda la red y que el ruido sea despreciable en la decodificación.

Para evaluar el desempeño energético no se tuvieron en cuenta otros gastos que son relevantes en estas redes (como es la energía disipada en procesos como el de realizar las mediciones al medio físico y el procesamiento de datos) fuera de lo que tiene que ver con la Codificación de Fuente Distribuida. Estas variables si bien son de interés para realizar una evaluación real del consumo energético en una red de sensores, pueden ser desestimadas para

analizar el desempeño de las estrategias de recolección de los datos, dado que su comportamiento será constante para todas las estrategias.

La energía disipada por un nodo en el proceso del procesamiento de la Codificación de Fuente Distribuida dependerá de las características del microprocesador que use el sensor y del código de canal que se utilice. No obstante en este trabajo se toma como referencia los valores dados en [51] para un esquema de codificación/decodificación similar al presentado en este trabajo.

Como se explica en el epígrafe 1.4, para ahorrar energía, la recolección de datos en las redes de sensores puede realizarse periódicamente, bajo demanda o ante la detección de un evento. Para evaluar el desempeño energético de las estrategias presentadas se analiza el peor de los casos que es cuando todos los nodos deben transmitir de forma constante con pequeños intervalos de tiempo entre las transmisiones. De manera que todos los nodos estarán transmitiendo a la estación base según esquema establecido para la estrategia de recolección.

La Tabla 3.1 resume los parámetros tomados como referencia en el desarrollo de este trabajo para todos los cálculos referidos con el consumo de energía.

Tabla 3.1. Valores de los parámetros usados para los cálculos energéticos.

Símbolo	Descripción	Valor
E_{elec}	Energía disipada por la electrónica	50 nJ/bit [50]
E_{amp}	Energía disipada para amplificar la potencia	100 pJ/bit/m ² [50]
n	Exponente de pérdida de trayectoria	2 [50]
E_{cod}	Energía disipada por la codificación	0.816μJ [51]
E_{dec}	Energía disipada por la decodificación	27.614 μJ [51]
C_{bat}	Capacidad de la batería	8.1 MJ*

* Se utiliza una batería AA con 1500 mAh y 1.5 V lo que equivale a $8.1 \cdot 10^6$ J

3.2 Resultados de las simulaciones.

3.2.1 Caso simple de dos fuentes y un terminal

Para analizar el desempeño de los esquemas de codificación/decodificación presentados en el epígrafe 2.1, estos algoritmos son evaluados en un escenario de dos nodos fuentes que codifican sus datos y los transmiten a un nodo terminal que recolecta y decodifica los datos de ambos nodos de forma conjunta.

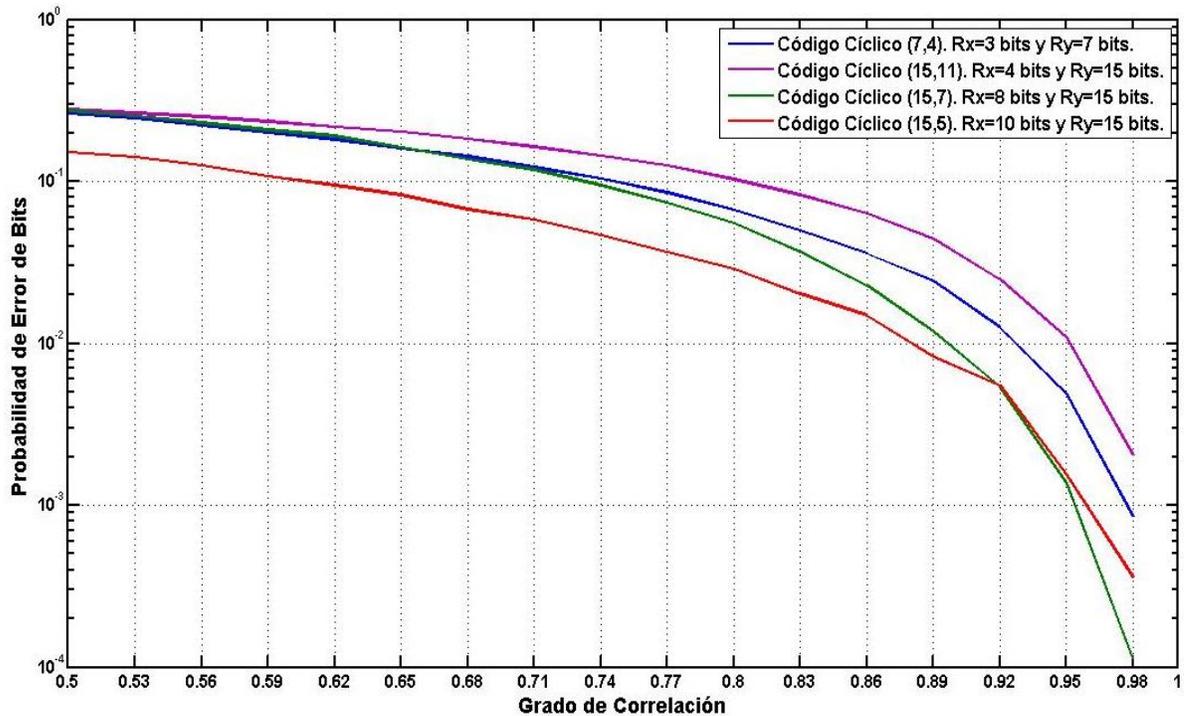


Figura 3.1. Probabilidad de error de bits contra correlación utilizando códigos cíclicos para el caso asimétrico.

La Figura 3.1 ilustra el desempeño de la probabilidad de error de bits obtenido en este escenario utilizando un esquema de codificación asimétrico con diferentes códigos cíclicos variando el grado de correlación entre los nodos. La gráfica demuestra que ante una mejor correlación entre las fuentes se obtiene un mejor desempeño en cuanto a la probabilidad de error.

Según la figura, dentro de los códigos cíclicos implementados el que presenta la mayor razón de codificación es el código (15,11), sin embargo, el comportamiento de este código en cuanto a la probabilidad de error de bits es el menos apropiado de todos. Mientras que, el

código cíclico (15,5) es que el presenta el mejor comportamiento en cuanto a la relación probabilidad de error de bits contra grado de correlación, pero es el de menor razón de codificación.

La Figura 3.2 muestra los resultados de este experimento comparando el comportamiento de un código cíclico (7,4) para los diferentes casos simétricos y asimétricos que se pueden obtener con este código. Esta gráfica confirma la teoría de que el caso asimétrico es el más apropiado para su implementación en la recolección de los datos. La figura muestra que en cuanto a la relación probabilidad de error de bits contra correlación, el caso asimétrico tiene un mejor comportamiento que el simétrico.

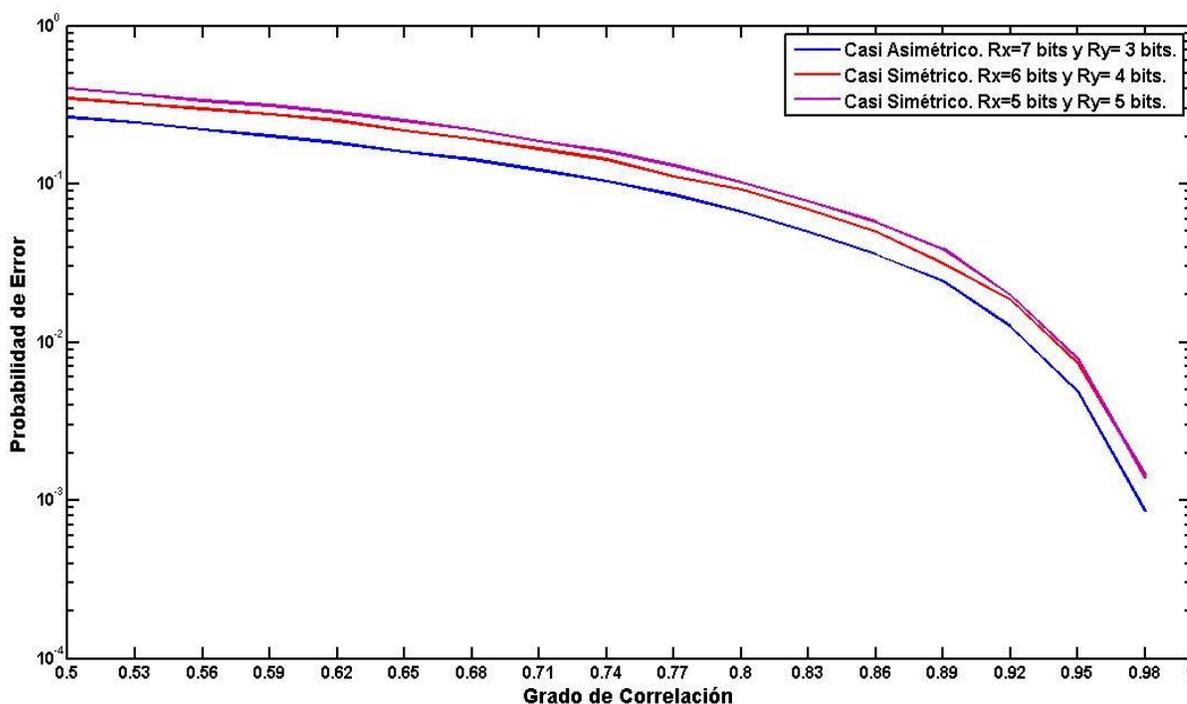


Figura 3.2. Probabilidad de error de bits contra correlación utilizando códigos cíclicos (7,4).

La Figura 3.3 muestra la relación que existe entre la probabilidad de error de bits y E_0/N_0 en un caso asimétrico utilizando códigos cíclicos con una correlación entre los nodos del 95%. Según estos resultados se puede afirmar que a medida que aumenta la relación señal a ruido se produce una disminución de la probabilidad de error de bits hasta aproximadamente los 6 dB, valor a partir del cual las gráficas comienzan a tener un comportamiento aproximadamente lineal. Este comportamiento se debe a que para un valor de E_0/N_0 mayor a los 6 dB se obtiene la menor probabilidad de error esperado para el grado de correlación

evaluado. Por lo que garantizando un valor mayor igual a este se puede obtener una decodificación óptima.

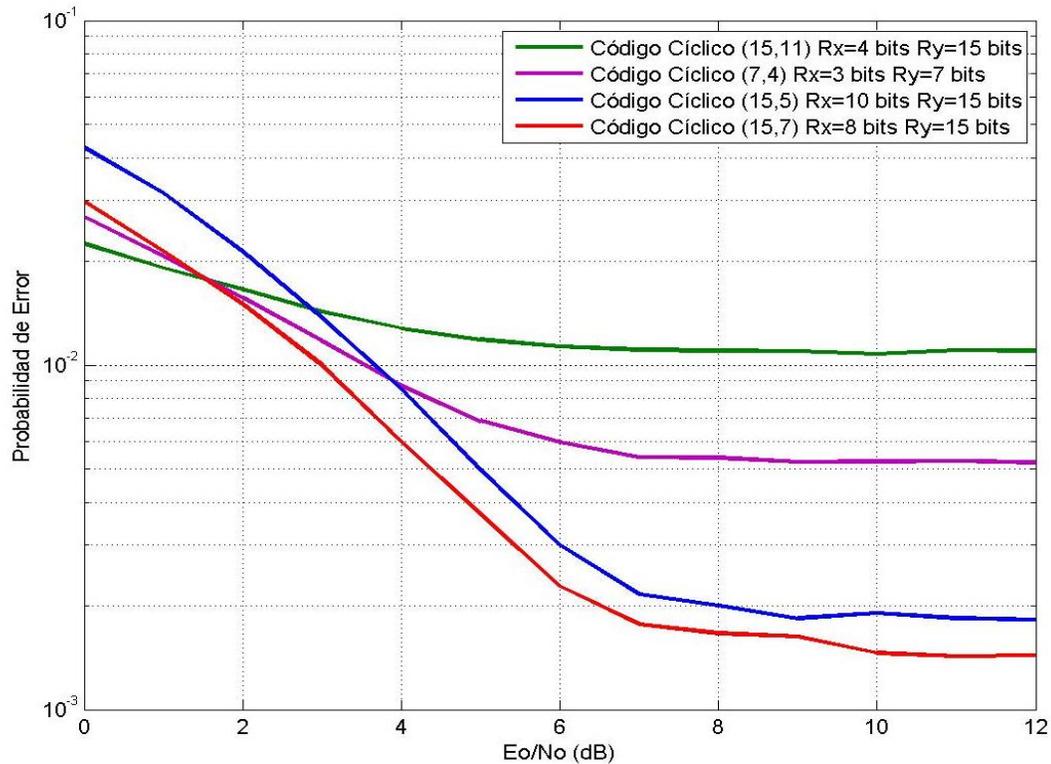


Figura 3.3. Comportamiento de la probabilidad de error de bits contra E_o/N_o en el caso asimétrico para una correlación de 0,95.

3.2.2 Desempeño de las estrategias de recolección de datos en redes de sensores.

En esta sección se analizan los experimentos realizados a las estrategias de recolección de datos para redes de sensores que implementan la Codificación de Fuente Distribuida presentadas en el epígrafe 2.2 del Capítulo 2.

Para la experimentación se plantea el escenario de una red de sensores con un área de 1 km² con un nodo terminal en el centro de la red que realiza la función de recolectar y decodificar todos los datos de la red según el modelo de recolección establecido.

A la hora de simular una red de sensores que implementa la codificación de fuente distribuida, diversos modelos de correlación pueden plantearse los cuales van a estar determinados por el comportamiento de las fuentes de datos y de la aplicación de la red. Los

modelos de correlación pueden ser variables o constantes y su comportamiento es esencial para evaluar el desempeño de la decodificación cuando se utiliza la codificación de fuente distribuida. Para simplificar los experimentos se asume un modelo de correlación en la red de forma tal que mantiene un grado de correlación entre los nodos cercanos de 0.98. Este valor se toma a partir de los resultados presentados en el epígrafe 3.2.1 que muestran que con este grado de correlación se garantiza una probabilidad de error de bits menor a 10^{-3} en un caso asimétrico de dos fuentes y un terminal.

Para el caso de las estrategias de clúster, los nodos cabecera son identificados dentro de las gráficas como CHN por las siglas en inglés (*Cluster Head Node*) utilizada en la literatura.

3.2.2.1 Desempeño de la decodificación

Las Figuras 3.4 y 3.5 muestran el desempeño de la probabilidad de error de bits y de la probabilidad de error de paquetes cuando se utiliza un algoritmo de Codificación de Fuente Distribuida asimétrico con los códigos cíclicos (7,4), (15,5), (15,7) y (15,11) en las estrategias de recolección de datos propuestas en el epígrafe 2.2 del Capítulo 2. Las gráficas muestran que la estrategia basada en clúster, en sus dos variantes, tiene un mejor desempeño que la estrategia secuencial en cuanto a la probabilidad de error en la decodificación.

En este parámetro el peor desempeño lo tiene la estrategia secuencial, esto se debe a que en esta técnica los errores se heredan de un nodo a otro en la decodificación.

Por su parte en la estrategia de clúster con el terminal como información lateral, aunque con un desempeño mejor, la probabilidad de error toma valores similares a los de la estrategia secuencial. En esta estrategia el terminal toma sus datos como referencia para decodificar la información de los nodos cabeceras de clúster, entrando a jugar un papel importante la distancia entre los nodos. A medida que los nodos se van alejando del terminal disminuye la correlación de estos y aumenta la probabilidad de error en la decodificación.

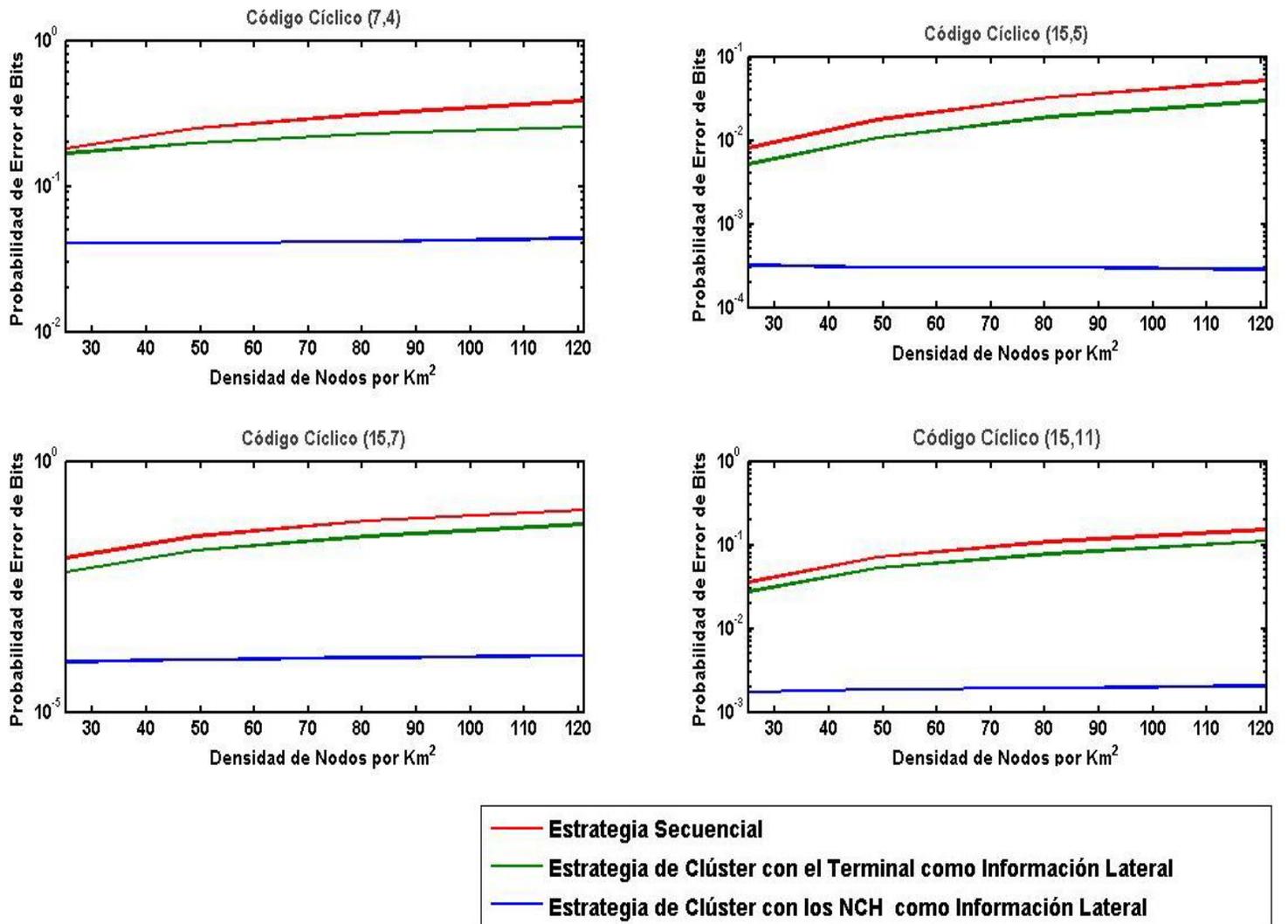


Figura 3.4. Comportamiento de la probabilidad de error de bits para las diferentes estrategias de recolección de datos utilizando para la Codificación de Fuente Distribuida los códigos cíclicos (7,4), (15,5), (15,7) y (15,11).

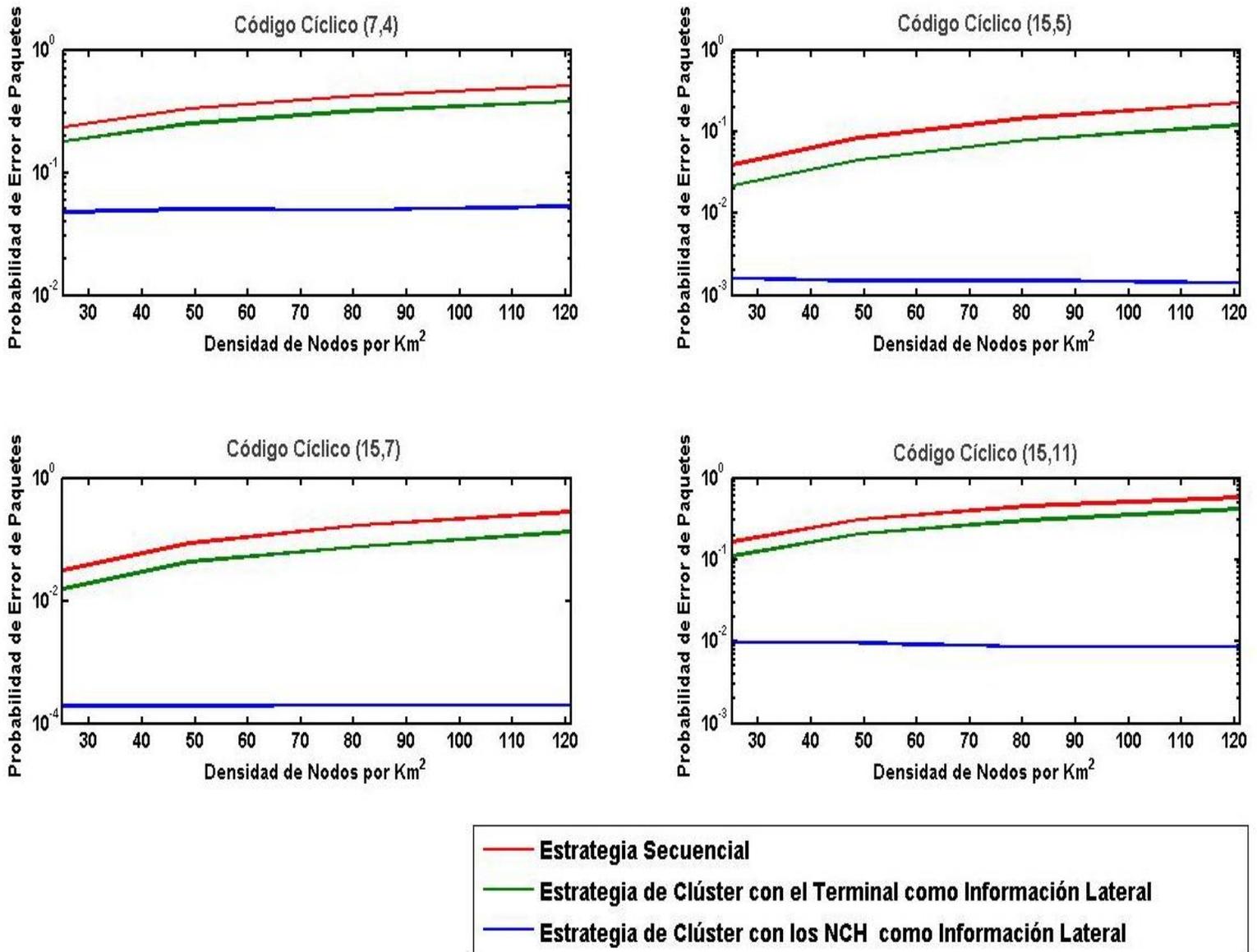


Figura 3.5. Comportamiento de la probabilidad de error de paquetes para las diferentes estrategias de recolección de datos utilizando para la Codificación de Fuente Distribuida los códigos cíclicos (7,4), (15,5), (15,7) y (15,11).

Los resultados dados en este experimento muestran que con respecto a la probabilidad de error de bit la estrategia de clúster con los nodos cabecera como información lateral logra un desempeño un 65% mejor que la estrategia secuencial y un 60% con respecto a la estrategia de clúster con el terminal como información lateral. En este caso la red es fragmentada en clúster para mantener un grado de correlación de 0.98 para todos los nodos dentro del clúster. El nodo cabecera envía su información sin codificar al terminal para que este utilice estos datos para decodificar la información del resto de los nodos en el clúster logrando una muy baja probabilidad de error en la recolección y decodificación de todos los datos de la red.

3.2.2.2 Consumo de energía

La Figura 3.6 compara el desempeño de la energía total consumida en las diferentes estrategias de recolección de datos durante un minuto.

Los resultados demuestran que, en correspondencia con el código cíclico que se utilice y la estrategia de recolección de datos a implementar, el uso de la Codificación de Fuente Distribuida logra una eficiencia energética entre un 20% a 55%. A medida que el código de bloque implementado en la codificación es más potente mayor, es la compresión de los datos y mayor el ahorro energético, aunque como se demuestra en las Figuras 3.1, 3.4 y 3.5 esto significa un peor desempeño en la probabilidad de error en la decodificación. La estrategia secuencial y la estrategia de clúster con el terminal como información lateral tienen desempeños energéticos muy similares. En el primer caso, aunque las distancias de transmisión son pequeñas, comparado con el resto de las estrategias, el consumo de energía está condicionado por la cantidad de bit que transmiten los nodos, ya que los nodos intermedios tienen que retransmitir la información de los nodos en un nivel superior. Por su parte, en las estrategias de clúster el consumo está determinado por la distancia de los nodos cabeceras con respecto al terminal y la cantidad de nodos en cada clúster. La estrategia de clúster con los nodos cabeceras como información lateral es la de mayor consumo de energía ya que en este caso las cabeceras de clúster envían sus datos al terminal sin comprimir.

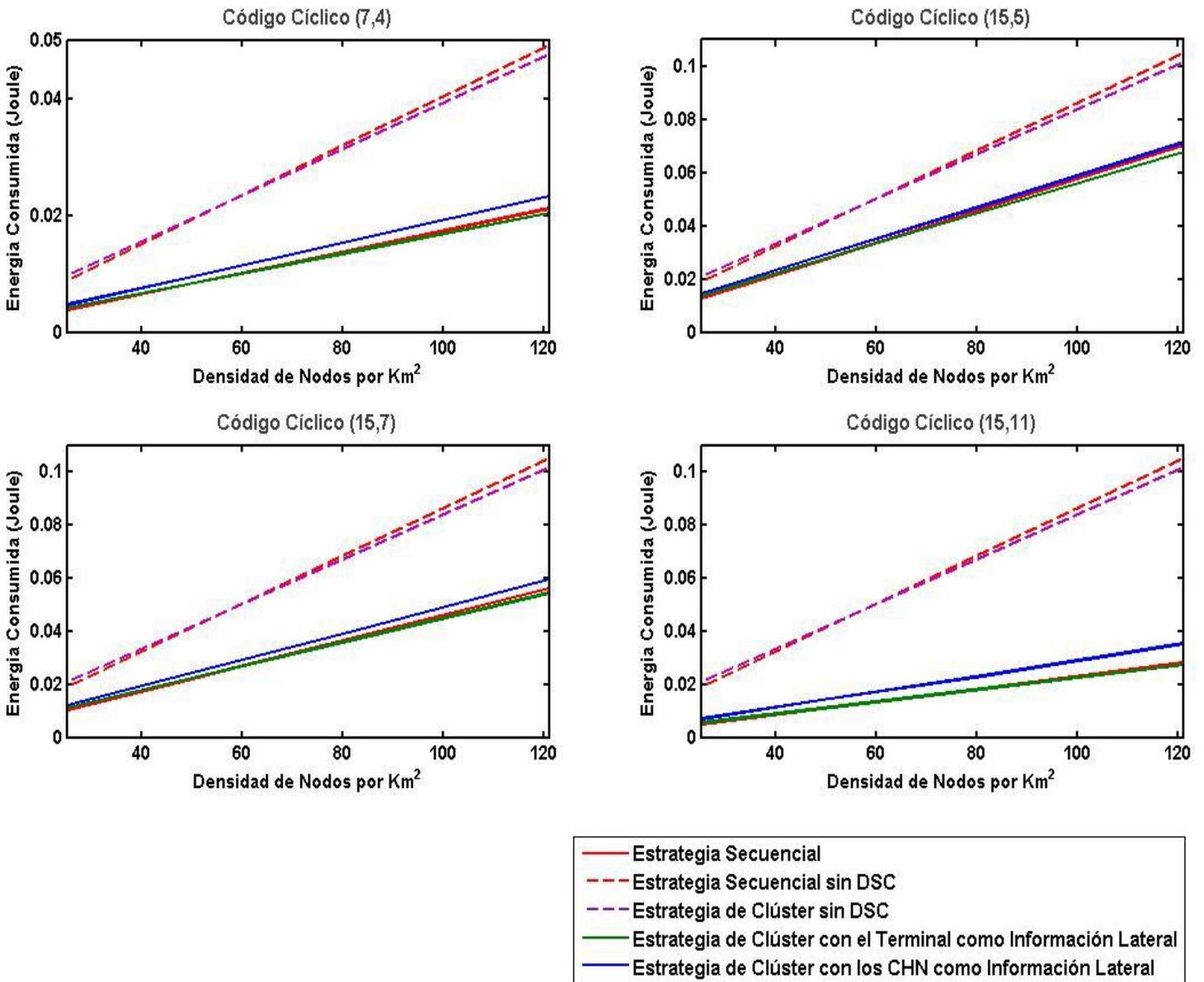


Figura 3.6. Energía total consumida por la red en las estrategias de recolección de datos en un minuto; utilizando para la Codificación de Fuente Distribuida los códigos cíclicos (7,4), (15,5), (15,7) y (15,11).

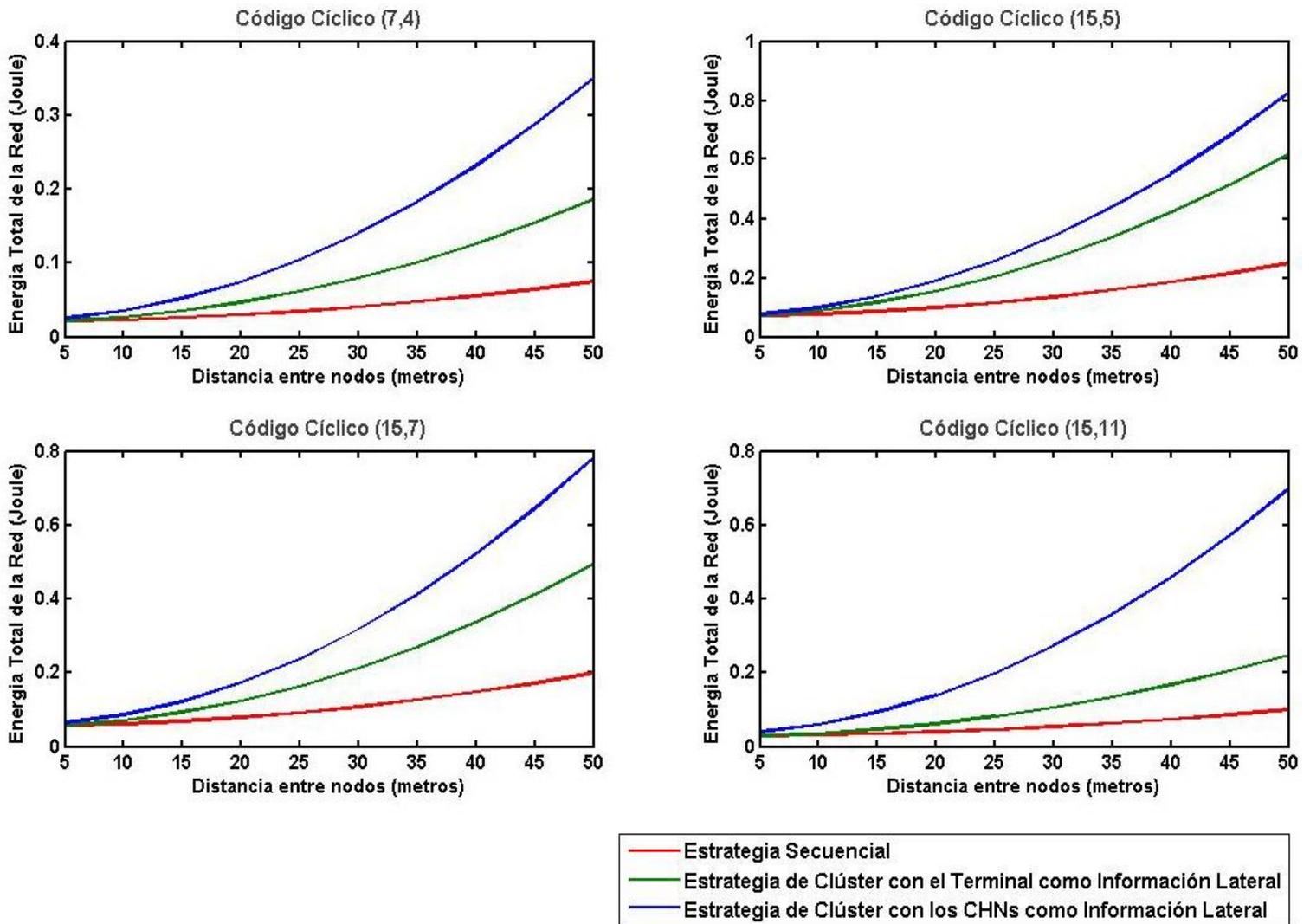


Figura 3.7. Energía total consumida en un minuto por una red de 120 nodos para diferentes valores de la distancia entre los nodos cercanos.

La Figura 3.7 muestra los resultados de la experimentación de una red con 120 nodos en la cual se analiza el consumo de energía total consumida en un minuto en las estrategias de recolección de datos, para diferentes valores de la distancia entre los nodos cercanos. Este experimento muestra cuánto influye la distancia en el consumo de energía de las diferentes estrategias de recolección de datos y cómo a medida que aumenta la distancia entre los nodos va empeorando el consumo de energía de la red. Como se esperaba de los experimentos anteriores, el peor desempeño en esta prueba se obtiene en la estrategia de clúster con los nodos cabecera como información lateral.

3.2.2.3 Tiempo de vida de la red

La Figura 3.8 representa el comportamiento del tiempo de vida de la red para diferentes estrategias de recolección de datos implementando una Codificación de Fuente Distribuida con un código cíclico (7,4). Como en los experimentos de la eficiencia energética, en esta prueba, se demuestra que el uso de la Codificación de Fuente Distribuida logra una mejora en el desempeño del tiempo de vida de hasta un 50%.

En este experimento las estrategias de clúster son probadas en dos condiciones: de forma estática o de forma conmutada. En el primer caso los nodos cabecera son elegidos de forma estática, de manera que, durante toda la vida útil de la red, los nodos seleccionados tendrán esta función. Por su parte en el esquema conmutado el rol de nodo cabecera dentro de un clúster cambia de forma dinámica cada un periodo de tiempo determinado.

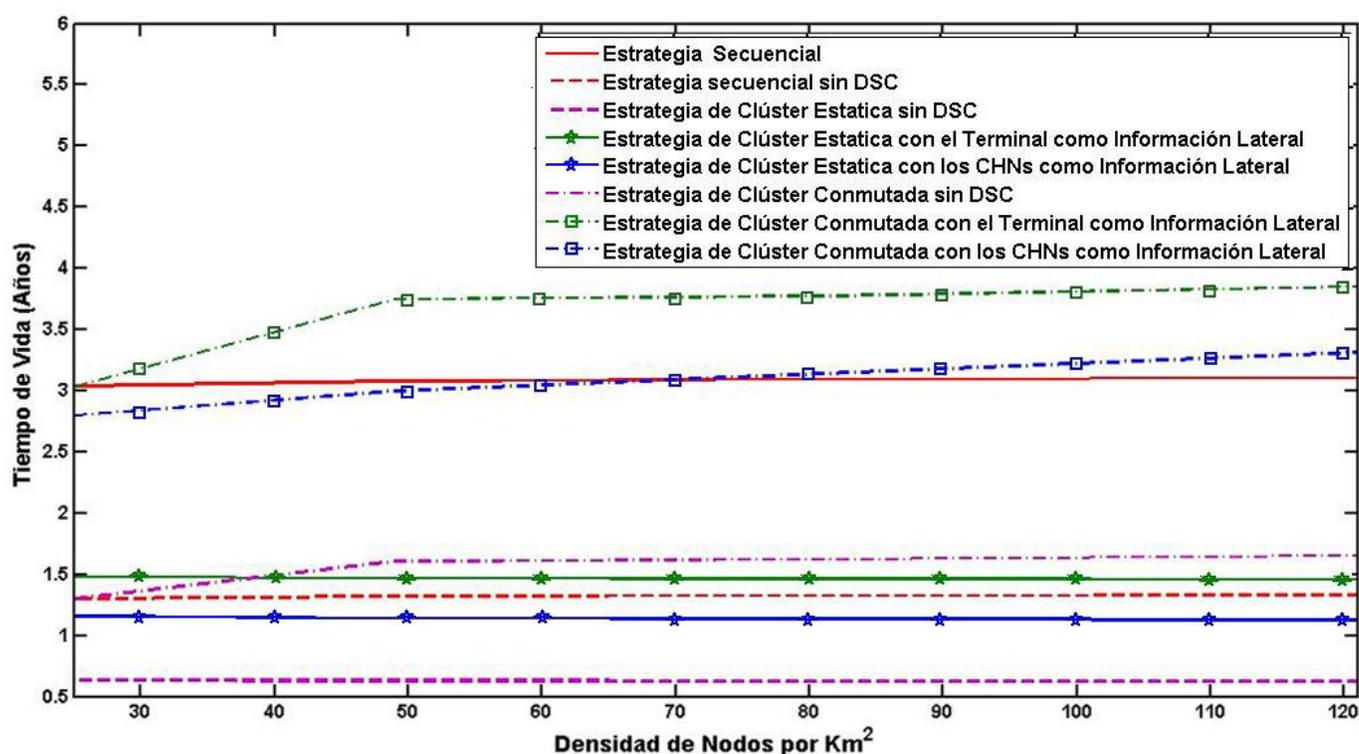


Figura 3.8. Tiempo de vida de la red para diferentes estrategias de recolección de datos implementando una Codificación de Fuente Distribuida con un código cíclico (7,4).

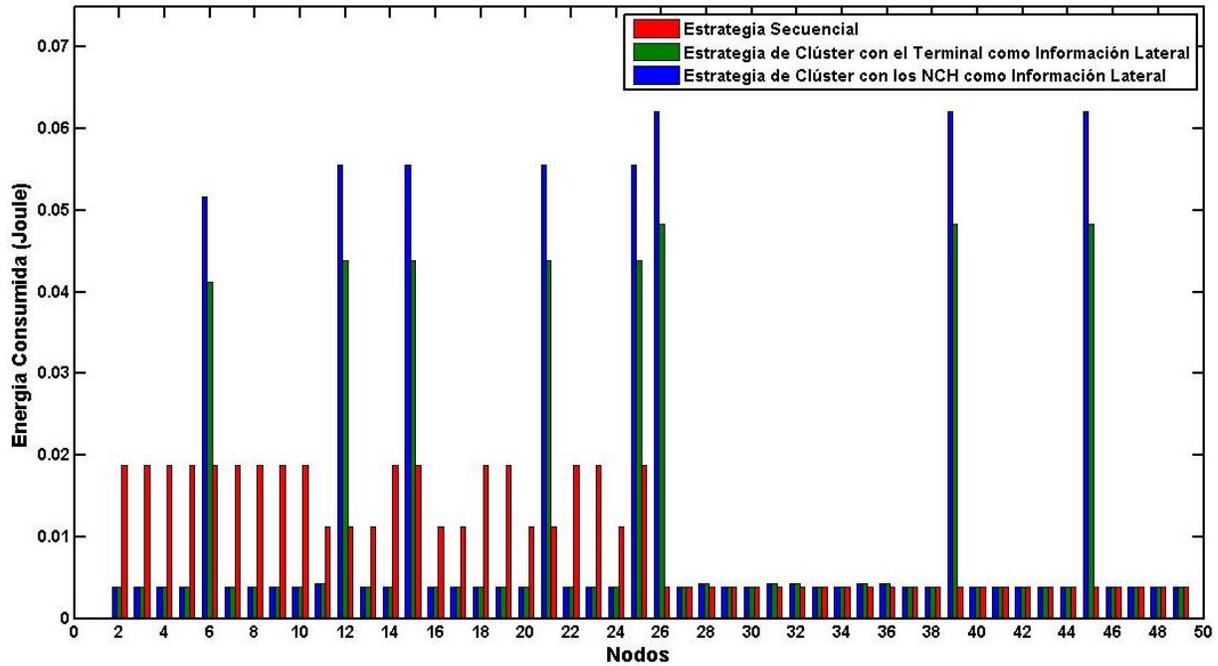


Figura 3.9. Distribución de energía en una red de 50 nodos. Se representa la estrategia secuencial y las variantes de la estrategia de clúster estática.

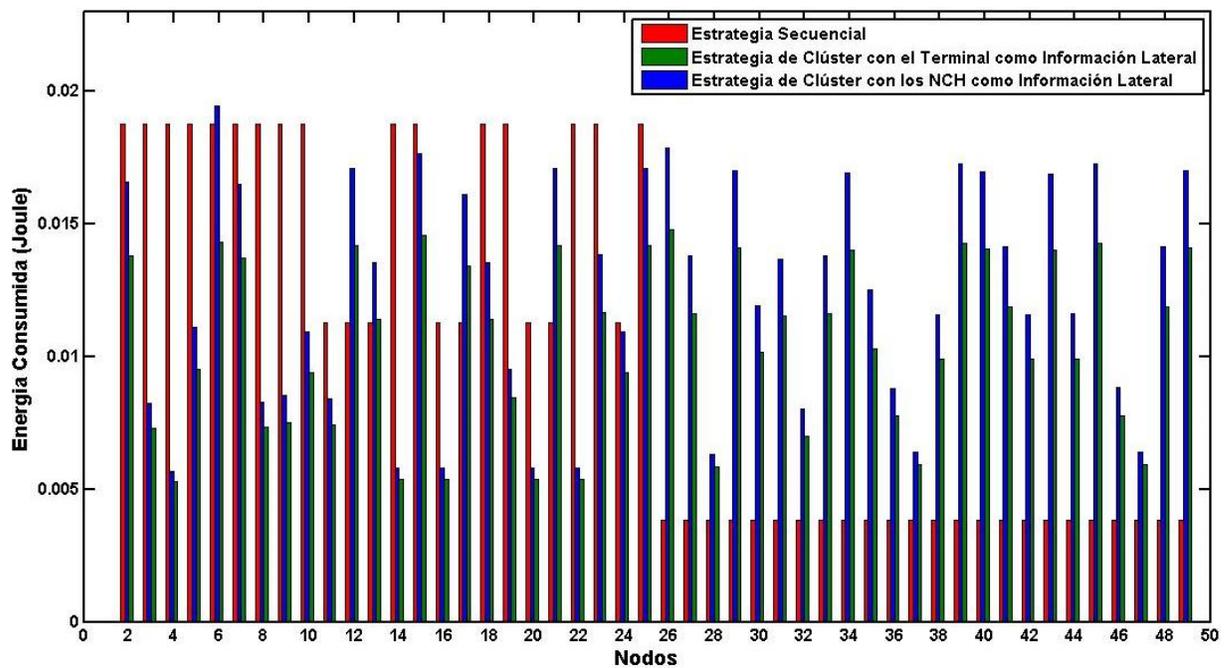


Figura 3.10. Distribución de energía en una red de 50 nodos. Se representa la estrategia secuencial y las variantes de la estrategia de clúster conmutada.

La Figura 3.8 muestra como las estrategias de clúster estática, como se espera de los experimentos de eficiencia energética, tiene un peor desempeño del tiempo de vida de la red que la estrategia secuencial. Sin embargo, cuando se aplica un esquema conmutado las estrategias de clúster logran un mejor tiempo de vida que la estrategia secuencial. La explicación de este fenómeno se demuestra en las Figuras 3.9 y 3.10.

La Figura 3.9 compara la distribución de energía en una red de 50 nodos en la estrategia secuencial y las variantes de estrategia de clúster estática. En este escenario, en la estrategia secuencial a medida que los nodos se acercan al terminal aumenta la energía a consumir, por lo que el tiempo de vida de la red va estar determinado por la vida útil de los nodos más cercanos al terminal. Por su parte en las estrategias de clúster estáticas la carga de tráfico y el consumo de energía se concentran en los nodos cabeceras de los clústeres, como estos son permanentes, el tiempo de vida en estas estrategias depende de la vida útil de las cabeceras de clúster.

La Figura 3.10 representa la distribución de energía en una red de 50 nodos en la estrategia secuencial y las variantes de estrategia de clúster conmutada. En este escenario la estrategia secuencial tiene el mismo comportamiento que en la figura 3.9. Sin embargo, como para este escenario en las estrategias de clúster, el rol de nodo cabecera se conmuta cada un determinado periodo de tiempo, la carga de tráfico para estas estrategias de recolección está mejor distribuida consiguiendo mejor desempeño del tiempo de vida que la estrategia secuencial.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Como resultado del desarrollo de la presente investigación se ha logrado un estudio teórico de la Codificación de Fuentes Distribuida en redes de sensores inalámbricos, enfocado especialmente a las estrategias de recolección de los datos correlacionados cuando se implementa esta técnica de codificación. Las estrategias analizadas fueron clasificadas en dos grupos: secuencial y basada en clústeres.

Utilizando la herramienta Matlab se implementaron los algoritmos de codificación/decodificación para los modelos simétricos y asimétricos de la Codificación de Fuente Distribuida basada en códigos de bloques lineales. Los algoritmos de codificación propuestos si bien no tienen rendimientos apropiados para su aplicación en una red de sensores real, por su sencillez facilitan el estudio de esta técnica. A partir de estas funciones se logró simular las estrategias de recolección de datos secuencial y basada en clúster en una red de sensores.

La simulación de la Codificación de Fuente Distribuida de dos nodos fuentes cuyos datos son enviados a un terminal que tiene la función de recolectarlos y decodificarlos, demuestra que un aumento de la correlación entre los nodos en las redes de sensores, tanto en el caso simétrico como asimétrico, se traduce en una mejora de la probabilidad de error de bits. Por lo que para un buen despliegue de la Codificación de Fuente Distribuida en estas redes se deben buscar mecanismos que permitan explotar la mejor correlación en todo momento. La comparación de estos casos, demostró que el enfoque asimétrico es más viable que el simétrico, en su aplicación para redes de sensores. Esto se debe a que el caso asimétrico tiene un mejor desempeño en la corrección de errores, logra una mayor reducción de la información a transmitir y una menor degradación de la señal.

Respecto a las estrategias de recolección de datos codificados, los experimentos demuestran que la estrategia basada en clúster tiene un mejor desempeño en cuanto a la probabilidad de error en la decodificación, especialmente si se utilizan los nodos cabecera de clúster como información lateral para decodificar los datos del resto de los nodos en sus respectivos

clústeres. Sin embargo, la estrategia secuencial y la de clúster con el terminal como información lateral logran un mayor ahorro de energía, debido a que cuando se utilizan los nodos cabecera como información lateral, estos deben enviar mayor cantidad de bit para que los datos sean decodificados. Por lo que la elección de la estrategia de recolección dependerá de un compromiso entre el consumo energético y el desempeño de la decodificación.

Con relación al tiempo de vida de la red, los experimentos demuestran que en este aspecto la estrategia secuencial logra un mejor comportamiento que las estrategias de clúster estáticas. Sin embargo, si el rol de nodo cabecera se conmuta dentro del clúster cada un determinado periodo de tiempo, las estrategias de clúster pueden alcanzar y mejorar el desempeño de la estrategia secuencial.

Recomendaciones

Con el objetivo de profundizar en el tema de la Codificación de Fuente Distribuida y su implementación en redes de sensores, se proponen las siguientes recomendaciones para investigaciones futuras:

- Extender la presente investigación implementando las estrategias de recolección de datos aquí analizadas en diferentes protocolos de enrutamiento desarrollados para las redes de sensores.
- Ampliar los experimentos propuestos, a partir de la utilización de algoritmos de codificación más potentes como los basados en códigos LDPC o Turbo.
- Continuar el estudio de la Codificación de Fuente Distribuida desde otras aristas, tales como: la interacción de esta con otras capas de red, el estudio de esquemas de codificación Wyner-Ziv, y la utilización de estas técnicas de codificación en aplicaciones específicas de las redes de sensores.

REFERENCIAS BIBLIOGRÁFICAS

- [1] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks*: John Wiley & Sons Ltd, 2010.
- [2] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *Signal Processing Magazine, IEEE*, vol. 21, pp. 80-94, 2004.
- [3] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *Information theory, IEEE Transactions on*, vol. 19, pp. 471-480, 1973.
- [4] V. Stankovic, L. Stankovic, and S. Cheng, "Distributed source coding: Theory and applications," in *Signal Processing Conference, 2010 18th European*, 2010, pp. 1879-1883.
- [5] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *Information Theory, IEEE Transactions on*, vol. 22, pp. 1-10, 1976.
- [6] A. D. Wyner, "The rate-distortion function for source coding with side information at the decoder\ 3-II: General sources," *Information and control*, vol. 38, pp. 60-80, 1978.
- [7] S. S. Pradhan and K. Ramchandran. (1999) Distributed Source Coding Using Syndromes (DISCUS): Design and Construction.
- [8] S. S. Pradhan and K. Ramchandran, "Distributed source coding: Symmetric rates and applications to sensor networks," in *Data Compression Conference, 2000. Proceedings. DCC 2000*, 2000, pp. 363-372.
- [9] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *Information Theory, IEEE Transactions on*, vol. 49, pp. 626-643, 2003.
- [10] H. Arjmandi, M. Taki, and F. Lahouti, "Lifetime maximized data gathering in wireless sensor networks using limited-order distributed source coding," *Signal Processing*, vol. 91, no. 11, pp. 2661–2666, 2011.
- [11] N. Abughalieh, K. Steenhaut, B. Lemmens, and A. Nowé, "A Mutual Algorithm for Optimizing Distributed Source Coding in Wireless Sensor Networks " *International Journal of Distributed Sensor Networks*, 2012.
- [12] H. Arjmandi and F. Lahouti, "Resource optimized distributed source coding for complexity constrained data gathering wireless sensor networks," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 2094–2101, 2011.
- [13] S. Tubaro and A. Vetro, "Distributed Video Coding," *Hindawi*, 2013.
- [14] S. Li and A. Ramamoorthy, "Networked distributed source coding," in *Theoretical Aspects of Distributed Computing in Sensor Networks*, ed: Springer, 2011, pp. 191-224.

- [15] Z. Tang. (2007) Distributed Source Coding Schemes for Wireless Sensor Networks.
- [16] R. K. Bhattar, K. Ramakrishnan, and K. Dasgupta, "Density Evolution Technique for LDPC Codes in Slepian-Wolf Coding of Nonuniform Sources," *International Journal of Computer Applications IJCA*, vol. 7, pp. 1-7, 2010.
- [17] J. J. Micallef, R. A. Farrugia, and C. J. Debono, "Low-density parity-check codes for asymmetric distributed source coding," in *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on*, 2010, pp. 985-988.
- [18] I. Shahid and P. Yahampath, "Distributed joint source-channel coding of correlated binary sources in wireless sensor networks," in *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*, 2011, pp. 236-240.
- [19] S.-Y. Shin, M. Jang, J. W. Kang, and S.-H. Kim, "New distributed source coding scheme based on LDPC codes with source revealing rate-adaptation," in *Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on*, 2011, pp. 917-920.
- [20] A. D. G. Biroli, M. Martina, and G. Masera, "An ldpc decoder architecture for wireless sensor network applications," *Sensors*, vol. 12, pp. 1529-1543, 2012.
- [21] X. Pan, R. Liu, and X. Lv, "Low-complexity compression method for hyperspectral images based on distributed source coding," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, pp. 224-227, 2012.
- [22] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, 2001, pp. 1400-1404.
- [23] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Data Compression Conference, 2002. Proceedings. DCC 2002*, 2002, pp. 252-261.
- [24] J. Chen and A. Abedi, "Distributed turbo coding and decoding for wireless sensor networks," *IEEE communications letters*, vol. 15, pp. 166-168, 2011.
- [25] A. Gersho and R. M. Gray, *Vector quantization and signal compression* vol. 159: Springer Science & Business Media, 2012.
- [26] J. Z. Sun and V. K. Goyal, "Scalar quantization for relative error," in *2011 Data Compression Conference*, 2011, pp. 293-302.
- [27] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed Compression in a Dense Microsensor Network," *IEEE Signal Processing Magazine*, 2002.
- [28] R. Zamir, S. Shamai, and U. Erez, "Nested linear lattice codes for structured multiterminal binning," 2002.
- [29] X. Chen and E. Tuncel, "Low-delay prediction-and transform-based Wyner-Ziv coding," *IEEE Transactions on Signal Processing*, vol. 59, pp. 653-666, 2011.
- [30] C. Ling, S. Gao, and J.-C. Belfiore, "Wyner-Ziv coding based on multidimensional nested lattices," *IEEE Transactions on Communications*, vol. 60, pp. 1328-1335, 2012.

- [31] S. Smirani, M. Kamoun, M. Sarkiss, A. Zaidi, and P. Duhamel, "Wyner-Ziv lattice coding for two-way relay channel," in *Advanced Technologies for Communications (ATC), 2012 International Conference on*, 2012, pp. 207-212.
- [32] J. Querol Morata, "Estudio y evaluación de prestaciones de redes inalámbricas de sensores," 2011.
- [33] V. Kumar, S. Jain, and S. Tiwari, "Energy efficient clustering algorithms in wireless sensor networks: A survey," *IJCSI International Journal of Computer Science Issues*, vol. 8, 2011.
- [34] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 174-185.
- [35] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 56-67.
- [36] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, pp. 11113-11153, 2012.
- [37] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*, 2000, p. 10 pp. vol. 2.
- [38] Y.-L. Chen, N.-C. Wang, Y.-N. Shih, and J.-S. Lin, "Improving low-energy adaptive clustering hierarchy architectures with sleep mode for wireless sensor networks," *Wireless personal communications*, vol. 75, pp. 349-368, 2014.
- [39] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," *Telecommunication Systems*, vol. 52, pp. 2419-2436, 2013.
- [40] P. Wang and I. F. Akyildiz, "Spatial correlation and mobility-aware traffic modeling for wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, pp. 1860-1873, 2011.
- [41] S. He, J. Chen, D. K. Yau, and Y. Sun, "Cross-layer optimization of correlated data gathering in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 1678-1691, 2012.
- [42] L. A. Villas, A. Boukerche, H. A. De Oliveira, R. B. De Araujo, and A. A. Loureiro, "A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks," *Ad Hoc Networks*, vol. 12, pp. 69-85, 2014.
- [43] T. Han, "Slepian-Wolf-Cover theorem for network of channels," *INFO. AND CONTR.*, vol. 47, pp. 67-83, 1980.

-
- [44] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian–Wolf: Theory, Algorithms, and Scaling Laws (2005)," *IEEE Transactions on Information Theory*, vol. 51. No. 12, 2005.
- [45] H. Wang, D. Peng, W. Wang, H. Sharif, and H.-H. Chen, "Interplay Between Routing and Distributed Source Coding in Wireless Sensor Network," *IEEE. ICC 2007 proceedings*, 2007.
- [46] C. W. C. Guogang Hua, "Correlated data gathering in wireless sensor networks based on distributed source coding " *Int. J. Sensor Networks*, vol. 4 Nos. 1/2, 2008.
- [47] G. Zhang, H. Xu, and H. Deng, "Distributed Source Coding Algorithm Suitable for Clustering Wireless Sensor Network," *Sensors & Transducers*, vol. 23, pp. 157-161, 2013.
- [48] K. J. R. Liu and S. Haykin, "Handbook on Array Processing and Sensor Networks," ed: Wiley-IEEE Press, 2010, cap.19.
- [49] Y. Huang, Q. Yi, and M. Shi, "An improved Dijkstra shortest path algorithm," in *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2013, pp. 0226-0229.
- [50] m. Z. M. N. Halgamuge, and K. Ramamohanarao, "An estimation of sensor energy consumption," *Progress In Electromagnetics Research*, vol. 12, pp. 259-295, 2009.
- [51] F. Oldewurtel, M. Foks, and P. Mahonen, "On a practical distributed source coding scheme for wireless sensor networks," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, 2008, pp. 228-232.