



UNIVERSIDAD NACIONAL
AVENIDA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**ESTUDIO DE LAS OPORTUNIDADES DE CODIFICACIÓN
EN REDES INALÁMBRICAS TIPO *AD HOC***

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA

P R E S E N T A:

CARLOS GUILLERMO GUADARRAMA GONZÁLEZ

DIRECCIÓN DE LA TESIS: DR. JAVIER GÓMEZ CASTELLANOS

CO-DIRECCIÓN DE LA TESIS: DR. VÍCTOR MANUEL RAMOS RAMOS

MÉXICO, D.F.

2011



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice de contenidos

Capítulo 1

| | |
|---------------------------------------|---|
| Introducción..... | 1 |
| 1.1. Especificación del problema..... | 2 |
| 1.2. Objetivo..... | 2 |
| 1.3. Desarrollo..... | 3 |
| 1.4. Estructura de la tesis..... | 3 |

Capítulo 2

| | |
|---|----|
| Fundamentos de redes inalámbricas..... | 5 |
| 2.1. IEEE 802.11..... | 6 |
| 2.2. Arquitectura IEEE 802.11..... | 6 |
| 2.2.1. BSS independiente..... | 6 |
| 2.2.2. Modo infraestructura..... | 7 |
| 2.3. Control de Acceso al Medio en IEEE 802.11..... | 7 |
| 2.3.1. Distributed Coordination Function (DCF)..... | 8 |
| 2.3.2. Point Coordination Function (PCF)..... | 10 |

Capítulo 3

| | |
|---|----|
| Codificación en red (Network Coding)..... | 11 |
| 3.1. Introducción..... | 12 |
| 3.2. Codificación en redes en condiciones reales..... | 14 |

Capítulo 4

| | |
|------------------------------------|----|
| Oportunidades de codificación..... | 13 |
|------------------------------------|----|

| | |
|---|----|
| 4.1. Bob y Alice..... | 18 |
| 4.2. Malla de nodos..... | 19 |
| 4.2.1. Tiempo entre arribos..... | 22 |
| 4.2.2. Oportunidades de codificación..... | 22 |
| 4.2.3. Tiempo de codificación..... | 24 |
| 4.2.4. Caudal de datos (throughput)..... | 36 |
| 4.2.5. Retardo..... | 27 |
| Capítulo 5 | |
| Resultados de simulación..... | 29 |
| 5.1. Bob y Alice..... | 29 |
| 5.2. Malla de nodos..... | 32 |
| 5.2.1. UDP..... | 32 |
| 5.2.2. TCP..... | 38 |
| Capítulo 6 | |
| Conclusiones..... | 43 |
| Referencias..... | 46 |

Índice de figuras

| | |
|---|----|
| Figura 2.1. Red extendida. | 4 |
| Figura 2.3. Problema de las terminales ocultas..... | 5 |
| Figura 2.4. Transmisión de una trama usando el esquema <i>CTS/RTS</i> | 6 |
| Figura 2.5. Diagrama de flujo de <i>CSMA/CA</i> con <i>RTS/CTS</i> | 7 |
| Figura 3.1. Escenario de <i>Bob</i> y <i>Alice</i> | 11 |
| Figura 3.2. <i>Red Mariposa</i> | 12 |
| Figura 3.3. Oportunidad de codificación en el nodo <i>B</i> | 14 |
| Figura 4.1. Red de nodos en forma de malla simulada en <i>NS-2</i> | 16 |
| Figura 4.3. Oportunidad de codificación en el nodo <i>i</i> | 19 |
| Figura 5.1. Escenario de <i>Bob</i> y <i>Alice</i> en <i>NS-2</i> | 27 |
| Figura 5.2. Ganancia de codificación sobre <i>TCP</i> | 28 |
| Figura 5.3. Ganancia en caudal de datos sobre <i>UDP</i> | 28 |
| Figura 5.4. <i>CDF</i> de la ganancia en codificación para <i>TCP</i> y <i>UDP</i> a 15kbps..... | 29 |
| Figura 5.4. Gráfica que muestra los tiempos entre arribos contra los tiempos de codificación..... | 31 |
| Figura 5.5. Gráfica que muestra el porcentaje de paquetes codificados | 32 |
| Figura 5.6. Gráfica del porcentaje de paquetes codificados variando el <i>timeout</i> | 33 |
| Figura 5.7. Gráfica de la tasa de transmisión contra el caudal de datos real a distintos <i>timeouts</i> | 34 |
| Figura 5.8. Caudal de datos real variando el <i>timeout</i> , sobre distintas tasas de transmisión..... | 34 |
| Figura 5.9. Valores del retardo adicional variando el <i>timeout</i> , sobre distintas tasa de transmisión..... | 35 |
| Figura 5.10. Gráfica de las oportunidades de codificación sobre <i>TCP</i> | 36 |
| Figura 5.11. Retardo adicional sobre el protocolo <i>TCP</i> | 38 |
| Figura 5.12. Comparativa de los valores del caudal de datos promedio y el caudal de datos promedio | |

Índice de tablas

| | |
|---|----|
| Tabla 3.1. Operación de adición en el campo de los números binarios..... | 10 |
| Tabla 4.1. Definiciones de los términos usados en este capítulo..... | 20 |
| Tabla 4.2. Pseudocódigo para calcular el tiempo de codificación promedio..... | 25 |

Lista de acrónimos

| Acrónimo | Significado |
|-----------------|--|
| <i>WLAN</i> | Red Inalámbrica de Área Local |
| <i>UDP</i> | Protocolo de Datagramas de Usuario |
| <i>TCP</i> | Protocolo de Control de Transmisión |
| <i>LAN</i> | Red de Área Local |
| <i>IEEE</i> | Instituto de Ingenieros Eléctricos y Electrónicos |
| <i>FTP</i> | Protocolo de Transferencia de Archivos |
| <i>CBR</i> | Tasa Constante de Bits |
| <i>PSTN</i> | Red Telefónica Pública Conmutada |
| <i>CSMA/CA</i> | Acceso Múltiple con Detección de Portadora con Evasión de Colisiones |
| <i>CSMA/CD</i> | Acceso Múltiple con Detección de Portadora con Detección de Colisiones |
| <i>NAV</i> | Network Allocation Vector |
| <i>NS-2</i> | Network Simulator 2 |
| <i>AODV</i> | Encaminamiento Vector Distancia Ad Hoc en Demanda |
| <i>DSDV</i> | Encaminamiento de Destino Secuencial Vector-Distancia |
| <i>RTS</i> | Solicitud para enviar |
| <i>CTS</i> | Listo para enviar |
| <i>CDF</i> | Cumulative Distribution Function |
| <i>RTT</i> | Tiempo de Retorno |

Resumen

Las redes inalámbricas se hacen cada vez más presentes en nuestra vida cotidiana. La codificación en red, mejor conocida como *network coding*, ha sido un gran aporte a las redes inalámbricas encontrando como su principal beneficio el incremento en el caudal de datos (*throughput*). Existen resultados que muestran que para ciertos escenarios, se pueden ahorrar cantidades considerables de energía en los nodos [19] e incrementar el caudal de datos de la red gracias a la reducción en cuanto al número de transmisiones de paquetes [17]. Es de esperar que en escenarios de redes inalámbricas, donde intuitivamente existirían grandes posibilidades de incrementar el caudal de datos, se obtuvieran los beneficios que ofrece la codificación en red. En este trabajo de tesis se muestran algunos factores que pueden ayudar a la toma de decisiones al momento de implementar un sistema que soporte la codificación en red. Específicamente, este trabajo se enfoca en observar el comportamiento del tráfico de paquetes sobre una red de malla tipo *ad hoc*. De esta manera podemos determinar la ganancia de codificación y en caudal de datos sobre este tipo de escenarios.

Nuestro trabajo está enfocado al estudio de oportunidades de codificación sobre redes inalámbricas, particularmente en redes *ad hoc* que usen tecnología *WLAN*. Presentamos algunas medidas de desempeño que ayudan a determinar qué tan buena es la codificación en red para escenarios como la malla de nodos mencionada anteriormente. La evaluación de las oportunidades de codificación muestra que para una malla de nodos, la codificación en red puede ayudar a incrementar el caudal de datos pero bajo condiciones de poco tráfico. Los resultados muestran que la codificación en red con altas tasas de tráfico no benefician al incremento del caudal de datos de la red.

El principal aporte de este trabajo son los resultados obtenidos de las medidas de desempeño de oportunidades de codificación sobre la malla de nodos. Sobre tráfico *UDP* se concluyó que los factores principales para incrementar el caudal de datos son equilibrar la tasa de transmisión y el *timeout*, este equilibrio puede alcanzarse sin afectar demasiado el retardo de los paquetes. El *timeout* es una variable que definimos como el tiempo máximo que puede esperar un paquete para ser codificado en un nodo. Encontrar un *timeout* adecuado puede beneficiar la ganancia del caudal de datos, pero incrementarlo demasiado podría generar altos retardos que hagan inoperable la red para ciertas aplicaciones. De la misma forma, tener un *timeout* muy pequeño podría no ofrecer ningún beneficio en el desempeño general de la red.

Sobre tráfico *TCP* se determinó que generar retardos en los nodos, por más cortos que éstos sean, tiene un impacto negativo en el valor del caudal de datos y en el retardo. El desempeño global de *TCP* se ve afectado debido a los *timeouts* colocados en los nodos. Sobre *TCP* no existe ganancia en caudal de datos más que en el escenario canónico de la codificación en red, *Bob* y *Alice*. En el escenario de la malla es preferible no usar estas técnicas de codificación con tráfico *TCP*.

Agradecimientos

Quiero agradecer al Dr. Javier Gómez y al Dr. Víctor Ramos por su apoyo en la dirección de este trabajo, gracias por sus observaciones, aportaciones y su tiempo. En lo personal le doy las gracias al Dr. Víctor Ramos por las enseñanzas y la gran motivación que me ha dado. Mi más grande admiración y respeto.

Agradezco a mi madre por todo lo que me ha dado. Sin ella no hubiera logrado hacer lo poco o mucho que he hecho. Gracias a ustedes hermanas por su comprensión y apoyo en todos los momentos de mi vida. No tengo palabras para agradecerle familia por todo lo que me has dado.

Gracias a mis amigos, cada uno de ustedes me ha dejado algo. Gracias por acompañarme durante este tiempo escuchándome, dándome ánimos.

Te agradezco Esmeralda, por tu gran ejemplo, por tu empuje y motivarme a seguir avanzando, por mostrarme que un camino seguro a la superación es la constancia y la disciplina.

De igual manera, quiero agradecer a los proyectos PAPIIT y IN106609, así como al proyecto CONACyT 105117.

Capítulo 1

Introducción

Hemos sido partícipes del gran avance que ha tenido la tecnología durante estos últimos años. Las comunicaciones como las conocíamos hasta hace un par de décadas no se parecen en nada a lo que vivimos en la actualidad. Un gran ejemplo de ello es *Internet* que se ha convertido en el medio de comunicación más importante a nivel mundial. Cada vez más usuarios están conectados a la red de redes, como se le suele llamar. Esta gran demanda de usuarios requiere constantemente de satisfacer nuevas necesidades de comunicación como mayor ancho de banda y disponibilidad de la información, por ejemplo. Asimismo, con el paso del tiempo la movilidad ha jugado un papel muy importante en las comunicaciones.

De utilizar a la comunicación inalámbrica con fines bélicos durante la primera y segunda guerra mundial, hoy en día ha pasado a ser algo muy común. Podemos ver a muchas personas con teléfonos celulares, laptops o puntos de acceso con salida a *Internet* en centros comerciales y universidades. La comodidad de la movilidad ha sido el propósito de muchos avances científicos, los cuales buscan nuevas ideas para contrarrestar las limitantes que tienen las redes inalámbricas. Muchos de estos avances se han enfocado en gran parte en crear métodos más eficientes para enviar información por el medio de comunicación inalámbrico, el aire.

La codificación en red (*network coding*) es un conjunto de técnicas que rompen con el paradigma de flujos de información a través de un canal de comunicación inalámbrico. Estas técnicas realizan mezclas de flujos de información ajenos entre sí. Existen trabajos de gran importancia que muestran las bondades que estas técnicas ofrecen. Preguntas como en qué tipos de escenarios pueden aplicarse estas técnicas son parte de esta tesis así como qué papel tiene el retardo de codificación sobre el comportamiento de *TCP*.

La codificación en red fue creada con el fin principal de incrementar el caudal de datos en las redes inalámbricas, sin embargo se le han encontrado otro tipo de aplicaciones. En este trabajo se trata de dar una respuesta a qué pasa con la codificación en red en escenarios donde intuitivamente podrían aprovecharse estas técnicas para hacer más eficiente la comunicación en la red.

1.1 Especificación del problema

En trabajos como [17] se ha comprobado lo que la codificación en red ayuda a incrementar el caudal de datos de la red a muy bajo costo de procesamiento. En otras investigaciones como en [19] se han visto sus aplicaciones en el ahorro de energía en redes tipo *ad hoc*. En trabajos muy citados como [16] y [18], las técnicas de codificación son vistas desde un enfoque más apegado a la teoría de gráficas sobre tráfico *multicast*. Estos trabajos muestran las capacidades máximas teóricas de la codificación en red, todo esto bajo condiciones ideales de tráfico. Estudios más recientes como [20] buscan determinar las mejores rutas que deberían tomar los paquetes para generar más oportunidades de codificación y así aprovechar mejor estas técnicas.

Sin embargo, existen muchos detalles técnicos que no se han tratado en estos trabajos. Por ejemplo, no se conoce el impacto en el desempeño de la red de factores como el tiempo que se guarda un paquete antes de ser transmitido sin codificarse con otro, o cuál es el desempeño de las técnicas de codificación en protocolos como *TCP*. En este trabajo se trata de dar una respuesta a estos cuestionamientos.

1.2 Objetivo

El objetivo general de este trabajo es dar una respuesta a cuáles son realmente las capacidades que tiene la codificación en red en escenarios particulares de redes inalámbricas.

Como objetivos particulares está el experimentar en un escenario de una malla de nodos inalámbricos. También los son el estudiar valores como el caudal de datos y el retardo, así como el porcentaje de

oportunidades de codificación existentes en las mallas inalámbricas. Otro objetivo es la implementación de la codificación en red en el simulador *NS-2* que utilizamos para el escenario de *Bob y Alice* que mencionamos a detalle más adelante.

1.3 Desarrollo

El desarrollo de este trabajo se divide en dos partes. La primera parte es una implementación de la codificación en red en el escenario mencionado en el capítulo 3 de *Bob y Alice*, y replicar parte de los experimentos realizados en [17].

La segunda parte es la realización de simulaciones y la evaluación de resultados arrojados por el simulador *NS-2* (que no soporta las técnicas de codificación) en una red de malla de cien nodos inalámbricos. En particular, el trabajo principal de la segunda parte de la tesis consiste en realizar mediciones de variables de interés como el caudal de datos, retardo y oportunidades de codificación. Estas mediciones se consiguieron modificando ciertos protocolos del simulador *NS-2*. Aunque estas modificaciones no consideran por completo las técnicas que abarca la codificación en red, se realizaron ajustes al simulador para mejorar la precisión de los resultados.

1.4 Estructura de la tesis

Este documento está dividido en cuatro partes, la primera se refiere a los antecedentes teóricos relacionados con este trabajo de investigación. En el capítulo 2 se presentan los fundamentos de redes inalámbricas, el cual está enfocado hacia las *WLAN*. En este capítulo se describen algunas de las problemáticas que se tienen en los medios inalámbricos y cómo es que se han podido contrarrestar. En el capítulo 3 se estudia a la codificación en red desde un punto de vista práctico, se muestran ejemplos de cómo operan y se enuncian los puntos necesarios a contemplar al implementar estas técnicas.

En la segunda parte se describe la propuesta de nuestro estudio de oportunidades de codificación, la idea principal de lo que deseamos calcular y cómo es que lo realizamos. Esto se presenta en el capítulo 4, donde se explica también de manera formal cómo se realizaron varios cálculos de algunas de las variables buscadas.

La tercera parte del trabajo se presenta en el capítulo 5, donde se muestran los resultados que obtuvimos de las oportunidades de codificación en redes inalámbricas tipo *ad hoc*. Se hace un análisis por todas las variables mencionadas en el capítulo 4 donde se muestran los resultados de nuestras

simulaciones.

Finalmente, la última parte de este trabajo se presenta en el capítulo 6, donde se hace una reflexión sobre los resultados obtenidos en el capítulo 5, concluyendo sobre las variables a tomar en cuenta al implementar la codificación en red en un sistema inalámbrico. Así también se habla de trabajo a futuro de esta investigación y las mejoras que se pueden hacer.

Capítulo 2

Fundamentos de redes inalámbricas

En este capítulo revisamos los fundamentos básicos de redes inalámbricas. Daremos las bases para comprender los mecanismos de comunicación utilizados particularmente en las redes *IEEE 802.11* también llamado *Wi-Fi*. Dado que este estándar es el más usado en redes inalámbricas tipo *ad hoc*, como las que se estudian en este trabajo de tesis, es importante conocer su funcionamiento. En el resto del capítulo se menciona el modo de operación de *Wi-Fi*, así como la manera en que este estándar utiliza y comparte el canal inalámbrico para comunicarse con otros dispositivos que utilizan esta misma tecnología.

Una red inalámbrica es un conjunto de dispositivos (computadoras, teléfonos celulares, aparatos electrodomésticos, etc.) los cuales comparten información y componentes de hardware y de software, usando al aire como canal de comunicación. En lo que se refiere a comunicación digital, existen principalmente dos tipos de redes inalámbricas: las redes celulares y las redes de área local inalámbricas (*WLAN*). Las redes celulares fueron creadas con el propósito de realizar llamadas de voz como se hace en la telefonía convencional o *PSTN*, pero actualmente se pueden transferir datos y multimedia usando tecnologías más recientes. Por otro lado las *WLAN* son las redes que actualmente más se usan en los hogares, comercios, universidades, etc., de las cuales entraremos en detalle a lo

largo de este capítulo.

2.1 *IEEE 802.11*

IEEE 802.11 es un conjunto de estándares, los cuales fueron aprobados por el comité *IEEE 802* en el año de 1997 y que llevan a cabo la comunicación inalámbrica de área local sobre las frecuencias 2.4, 3.6 y 5GHz. Idealmente se esperaba que las redes inalámbricas ofrecieran los mismos servicios que ofrecían en su momento las redes cableadas, sin embargo se enfrentaron retos que en las redes cableadas no se presentaban [1]. Algunos de estos problemas son: uso de frecuencia, interferencia, seguridad, energía, movilidad y caudal de datos.

2.2 *Arquitectura IEEE 802.11*

La arquitectura del *IEEE 802.11* es un conjunto de componentes que interactúan entre sí para ofrecer una red que soporte movilidad en sus nodos.

El componente principal de esta arquitectura es el Conjunto Básico de Servicio (*BSS*), que es el conjunto de nodos inalámbricos que se comunican en la *WLAN*. Al área de cobertura del *BSS* se le denomina Área Básica de Servicio (*BSA*). Si un nodo se mueve fuera del *BSA* no podrá comunicarse directamente con los otros nodos del *BSS*.

Existen principalmente dos formas en que los nodos del *BSS* pueden comunicarse: de manera independiente y en modo infraestructura.

2.2.1 *BSS independiente*

Este modo de operación se conoce como red tipo *ad hoc*. En esta modalidad del *IEEE 802.11* los nodos de la *BSA* son capaces de comunicarse directamente entre ellos. Además de poder enviar y recibir paquetes en su capa de aplicación, los nodos participan de manera colaborativa en el encaminamiento. Se coordinan para encontrar las mejores rutas para encaminar a los paquetes, siguiendo reglas de un protocolo de encaminamiento. Es necesario que las tablas de rutas se actualicen continuamente y de manera eficiente, ya que la posición de los nodos puede cambiar constantemente ocasionando que la comunicación en la red sea ineficiente.

2.2.2 Modo infraestructura

Las limitantes físicas que se tienen con las redes inalámbricas determinan el área de cobertura de los nodos. Para algunas redes, las distancias cubiertas son suficientes, en otros casos es necesario incrementar el área de cobertura de los nodos.

Además de existir de manera independiente, los *BSS* pueden formar parte de redes extendidas. Este componente de la arquitectura del *IEEE 802.11* se le llama Sistema de Distribución (*DS*). El Conjunto de Servicio Extendido (*ESS*) es una red que tiene como componentes a varios *BSS* unidos a través de un *DS*. Como se muestra en la fig. 2.1, la unión de los distintos *BSS* hacia el *DS* en el *ESS* es a través un dispositivo llamado Punto de Acceso (*AP*).

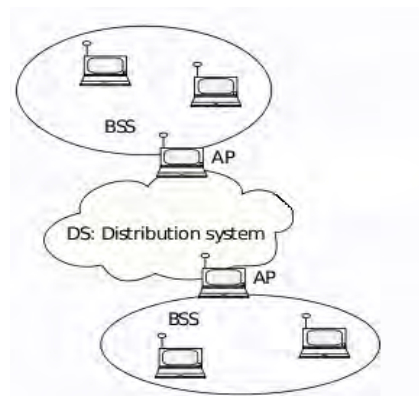


Figura 2.1. Red extendida. Unión de BSSs con un DS. La unión de cada BSS con el DS es a través de un AP.

El *AP* es un elemento de un *BSS* que otorga acceso al *DS*. La información de un *BSS* hacia el *DS* viaja vía el *AP*, es decir, si los elementos del *BSS* desean comunicarse con el *DS* deben hacerlo a través del *AP*. Actualmente este tipo de comunicación la podemos encontrar comúnmente en hogares, oficinas y escuelas. Donde el *AP* actúa como pasarela para conectar a los elementos del *BSS* a otras redes como *Internet*.

2.3 Control de Acceso al Medio en IEEE 802.11

El acceso al medio en las *WLAN* es distinto al que se utiliza en las *LAN*. En *IEEE 802.11* se cuentan principalmente dos tipos de métodos de acceso al medio, éstos están en función de las necesidades de prioridad de acceso a los nodos. Estos métodos son llamados funciones de coordinación.

2.3.1 Distributed Coordination Function (DCF)

DCF es el método de acceso base en *IEEE 802.11*, está implementado en todos los nodos de una *WLAN*, tanto en modo *ad hoc* como en modo infraestructura. Esta función también es conocida como *CSMA/CA* [1].

Las *WLAN* se basan en *CSMA/CA* a diferencia de las redes cableadas que usan el protocolo *Ethernet (IEEE 802.3)* y que utilizan *CSMA/CD*. No es conveniente usar *CSMA/CD* en las *WLAN* ya que, a pesar que el aire es un medio compartido, el área de cobertura no es la misma para todos los nodos de la red. No es posible que un nodo siempre detecte cuando otros nodos estén transmitiendo, si éstos están fuera de su área de cobertura. A este problema se le conoce como el problema de las terminales ocultas [1].

El problema de las terminales ocultas se aminora mediante el envío de mensajes de control previos a la transmisión del paquete de datos. En *CSMA/CA* un nodo que desea transmitir avisa al nodo destino que le enviará una trama de datos. Esto lo hace enviando un mensaje de control llamado *Request To Send (RTS)* en el cual se coloca información como el *NAV*, que es el tiempo que necesita el nodo fuente para poder transmitir una trama de datos. El *NAV* es importante para otros nodos que deseen transmitir, ya que este tiempo será el que tendrán que esperar para contender por el uso del canal, si es que desean transmitir. Una vez que el nodo destino recibe exitosamente el *RTS*, contesta enviando un mensaje *Clear To Send (CTS)* en el cual avisa al nodo fuente que está listo para recibir su trama de datos. Cuando el nodo fuente recibe el *CTS* ya puede enviar su trama de datos al nodo destino. Si la trama de datos es recibida correctamente, el nodo destino enviará un mensaje *ack* al nodo fuente para indicar que su trama de datos fue recibida satisfactoriamente. Una vez que se termine este intercambio de mensajes, los otros nodos podrán competir por el canal. Mencionemos que los mensajes *CTS* y *RTS* son capturados por los nodos dentro del área de cobertura del nodo que lo emite. De esta manera se trata de informar a los nodos cercanos que se desea realizar una transmisión, y así, evitar colisiones.

La propagación de las tramas en el medio inalámbrico hace necesario que los nodos tengan que esperar tiempos específicos antes de enviar cualquier tipo de trama. La fig. 2.2 muestra los tiempos que existen entre los envíos de cada trama. Observamos que antes de iniciar cualquier contención en el canal, los nodos esperan un tiempo *DIFS* antes de intentar transmitir su *RTS*. Cuando el *RTS* llega al nodo destino, éste espera un tiempo *SIFS* para transmitir su *CTS* al nodo fuente. Al recibir el *CTS*, el nodo fuente espera también un *SIFS* para poder transmitir su trama de datos. Finalmente el nodo destino, antes de enviar su mensaje de *ack*, también espera un tiempo *SIFS*. Antes de realizar otra contienda por

el canal, todos los nodos que deseen transmitir deberán esperar un *DIFS*, para después esperar un intervalo adicional obtenido con el algoritmo de *exponential backoff* (*EB*). Como nota observemos que el valor del *NAV* se actualiza con el envío del *CTS* y también de la trama de datos.

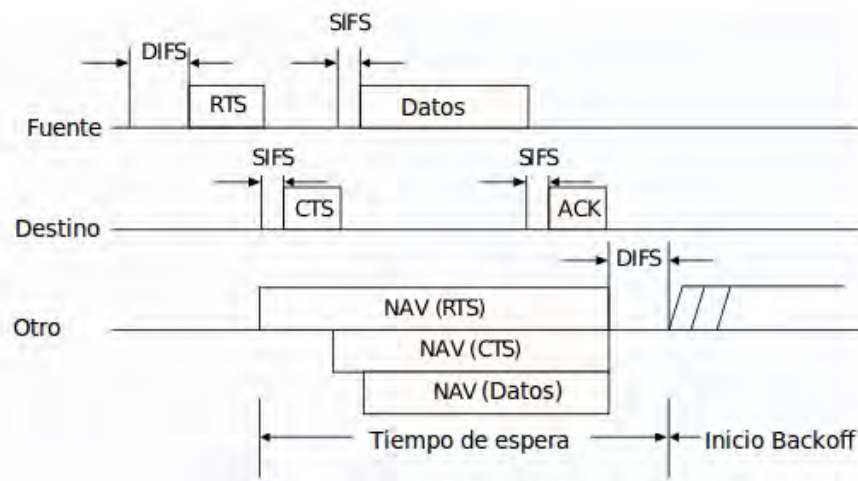


Figura 2.2. Transmisión de una trama usando el esquema CTS/RTS.

El algoritmo de *EB* [4] ayuda a que el canal se reparta de manera justa entre los nodos de la red. Antes de que un nodo use el canal, éste debe detectar si está ocupado. Si el canal está libre entonces puede intentar transmitir enviando una solicitud con un *RTS* hacia el nodo destino. En caso que el canal no esté libre para transmitir, el nodo fuente debe esperar un tiempo aleatorio que le devuelva el algoritmo de *EB*. En el primer intento después de colisionar un paquete, el algoritmo de *EB* toma un número aleatorio en el intervalo $[0, 2^3 - 1]$ [1] y lo multiplica por el valor de $51.2 \mu s^{-1}$. Este valor será el que esperará el nodo fuente antes de volver a detectar el canal e intentar transmitir nuevamente. Si ocurre otra colisión, el intervalo de números que elija el algoritmo se incrementará usando la fórmula $n = rand(2^{cw} - 1)^2$, donde el valor inicial de *cw* (ventana de contención) es 3 y puede llegar incrementalmente hasta 8. Si existe una nueva colisión cuando la ventana de contención es 8, su valor no se incrementará más. Cuando finalmente se logre una transmisión exitosa, el valor de *cw* del nodo que transmitió se reducirá al valor del *cw* mínimo.

1 Tiempo de propagación de un paquete de 512 bits sobre un enlace de 10Mbps.
 2 $rand(n)$ es una función que regresa un número aleatorio del conjunto $\{0, 1, \dots, n\}$

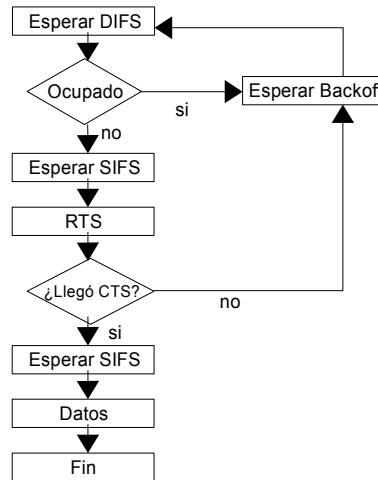


Figura 2.3. Diagrama de flujo de CSMA/CA con RTS/CTS.

El uso de los mensajes de control *RTS/CTS* es opcional en los nodos. Se pueden personalizar los elementos del *BSS* para que los mensajes *RTS/CTS* estén activos por defecto, que nunca se envíen o que sólo se envíen si las tramas de datos tienen un cierto tamaño. Este control de mensajes puede ser ajustado modificando la variable *RTSthreshold* de los radios Wi-Fi.

2.3.2 Point Coordination Function (PCF)

PCF es un método opcional de acceso al medio que ofrece el *IEEE 802.11* en su modo infraestructura. Es usado por los *AP* del *BSS* para coordinar el acceso al medio de los nodos. Como mencionamos anteriormente, el modo infraestructura ofrece comunicación vía el *AP*. El *AP* funge como orquestador dentro del *BSS* determinando qué nodo tiene derecho a transmitir.

Se definen un par de periodos para que los nodos puedan contender por el canal: Periodo de Contención (*CP*) y Periodo Libre de Contención (*FCP*). El primero es usado en modo *DCF*, donde los nodos que componen el *BSS* contenden por el canal de comunicación de manera equitativa, sin jerarquía entre los nodos. El *FCP* se usa en el modo infraestructura, aquí el *AP* decide qué nodos y en qué momento transmiten. Este par de periodos pueden alternarse y es posible ajustar sus duraciones, todo esto en función de las necesidades de comunicación que se deseen en la red. *PCF* además de ser un método de acceso libre de contenciones, permite transmisiones síncronas de las tramas de datos. Esta funcionalidad del *IEEE 802.11* no se utiliza comercialmente.

Capítulo 3

Codificación en red (*Network Coding*)

Sean u y v vectores longitud n con entradas en F_2 ¹. Se define a la operación *xor* (\oplus) entre u y v como $u \oplus v = \{u_1 + v_1, \dots, u_n + v_n\}$, donde la operación binaria $+$ se realiza sobre el campo finito F_2 como se muestra en la tabla 3.1.

| | | |
|-----|---|---|
| $+$ | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Tabla 3.1. Operación de adición en el campo F_2

Por notación, a lo largo de este documento utilizaremos a los símbolos $+$, $-$ y \oplus para referirnos a la operación *xor*. Las propiedades como campo de $GF(2^n)$ son de interés para explicar la teoría de codificación en red. En la práctica $GF(2^n)$ se convierte en la estructura sobre la cual realizamos cálculos y mediciones cuando nos referimos a codificación y a decodificación de datos.

¹ Campo finito de los números binarios

3.1 Introducción

La codificación en red es un conjunto de técnicas que aprovechan la naturaleza de difusión (*broadcast*) de los ambientes inalámbricos para incrementar el caudal de datos de la red. Los principios de la codificación en red surgen en un trabajo de *Ahlswede et al.* [13], en el cual se busca optimizar el envío de tráfico *multicast*. Desde entonces, se ha visto su enorme potencial en redes tipo *ad hoc* y de malla, de las cuales se han realizado múltiples investigaciones con diferentes fines, como el aumento en el caudal de datos, variantes en la codificación de paquetes, ahorro de energía, entre otros.

La codificación en red rompe con el enfoque que se tenía sobre flujos de información sobre una red. Enfocándonos en tráfico *unicast* es común ver conexiones entre nodos, en las cuales la información viaja de manera independiente respecto a otros flujos de datos. Usualmente los flujos de tráfico sobre una red son ajenos entre sí. La codificación en red busca mezclar información entre flujos de datos y así aprovechar la naturaleza de difusión inherente a los ambientes inalámbricos.

Podemos observar el escenario de la fig. 3.1 donde los nodos *A* y *B* desean intercambiar información y usan al nodo *R* como pasarela. Con el método tradicional se esperaría que por ejemplo, *A* enviara su mensaje a *R* y después *R* reenviara a *B* el mensaje que recibió de *A*. De esta forma *B* recibiría el mensaje que *A* quiso enviarle. Análogamente, *B* enviaría su mensaje para *A* usando a *R*. Observemos que, para lograr que tanto *A* y *B* reciban sus mensajes correspondientes, se utilizaron cuatro transmisiones. Usando la codificación en red, el número de transmisiones podría disminuirse a solamente tres. Digamos *A* enviaría su mensaje a *R* para *B*, pero *R* en lugar de reenviar ese mensaje inmediatamente, tomaría un tiempo de espera para que llegue el mensaje de *B* para *A*. Cuando llegue dicho mensaje, *R* realizaría una operación *xor* del mensaje de *A* y *B*, y lo reenviaría en difusión para *A* y *B*. Para realizar la decodificación del mensaje codificado ($A \oplus B$), debemos suponer que tanto *A* y *B*, retienen el mensaje que enviaron inicialmente a *R* hasta que les llegue el paquete de $A \oplus B$ enviado por *R*. A este escenario se le denomina comúnmente como el escenario de *Bob* y *Alice* [17], el cual estudiamos más a adelante en este trabajo.

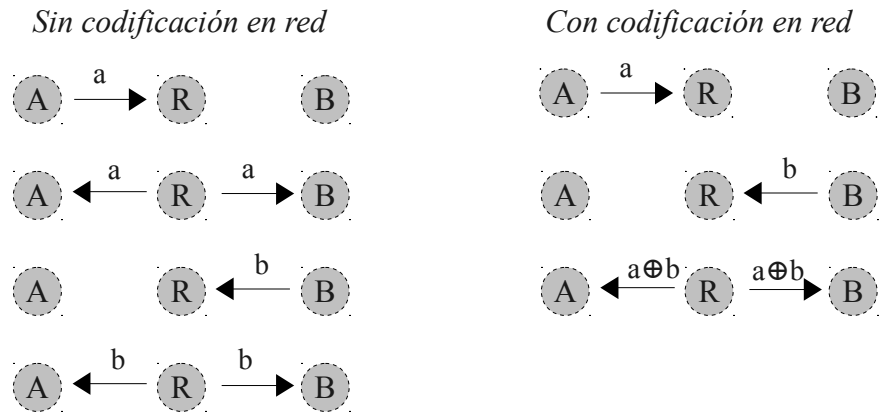


Figura 3.1. Escenario de Bob y Alice. Nodos A y B intercambian información vía el nodo R. Usando la codificación en red se puede disminuir el número de transmisiones a tres en lugar de cuatro.

Aunque uno de los principales objetivos de la codificación en red es el incrementar el caudal de datos en redes inalámbricas, a lo largo de su estudio se han podido observar otro tipo de beneficios. En seguida se muestra el potencial que existe al utilizar la codificación en red en topologías con varios nodos. En el ejemplo de la fig. 3.2 se tienen dos nodos fuentes $F1$ y $F2$, los cuales desean enviar su mensajes al nodo $D1$ y al nodo $D2$. Dado que los nodos destino $D1$ y $D2$ no se encuentran en el área de cobertura de los nodos $F1$ y $F2$, entonces éstos usan a $R1$ como pasarela. En este escenario existe una restricción extra, la capacidad de los enlaces está limitada a un mensaje por unidad de tiempo. Con el método tradicional se esperaría que tanto $F1$ como $F2$ mandaran sus mensajes de manera independiente a través de los nodos $R1$ y $R2$. Este proceso les tomaría tres transmisiones por mensaje, un total de seis transmisiones y más competencia por el canal. Usando la codificación en red se podría realizar lo mismo en sólo cuatro transmisiones. Para realizar lo anterior, una secuencia de eventos podría ser la siguiente: digamos $F1$ y $F2$ envían su mensaje a $R1$, quien realizaría la operación xor de estos dos mensajes para así enviarlos hacia $R2$ en una sola transmisión. $R2$ entregaría a $D1$ y $D2$ de igual forma enviando en difusión el paquete codificado que le llegó de $R1$. Es claro que tanto $D1$ como $D2$ pueden decodificar suponiendo que guardaron el mensaje que les llegó inicialmente de $F1$ y $F2$, respectivamente. En este ejemplo podemos ver que la codificación en red también es útil en escenarios donde se tienen restricciones en cuanto a la capacidad del canal, optimizando su uso e inclusive disminuyendo el tiempo de retardo de los mensajes.

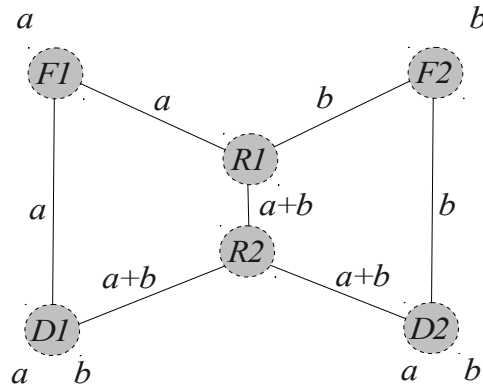


Figura 3.2. Red Mariposa. Los nodos $F1$ y $F2$ desean enviar información a los nodos $D1$ y $D2$.

Escenarios como el de *Bob y Alice*, y el de la *red mariposa* (red de la fig. 3.2) sirven para ejemplificar cómo es que opera la codificación en red, mostrando su beneficio e impacto. Existen otro tipo de topologías de red donde puede verse todo su potencial, usualmente éstas presentan a un nodo central que realiza la codificación de mensajes.

3.2. Codificación en redes en condiciones reales

En términos prácticos, para poder lograr que la codificación en red sea una realidad, en [17] se proponen un conjunto de funcionalidades que deben ofrecer los nodos de la red.

- **Escuchar de manera oportunista (*opportunistic listening*).** Se refiere al aprovechamiento de la naturaleza de difusión de las redes inalámbricas. Todos los nodos inalámbricos que participen en la red deben contar, en primer lugar, con una antena omnidireccional y configurarla en modo promiscuo. Es decir, los nodos estarán escuchando cada mensaje que pase por sus áreas de cobertura, aunque éstos no estén destinados hacia ellos. Los mensajes escuchados serán guardados temporalmente en algún *buffer*. Estos mensajes serán útiles cuando los receptores requieran decodificar mensajes y también tendrán importancia para los nodos que realicen codificación de mensajes. Codificar varios mensajes en uno solo no garantiza la correcta decodificación en los receptores, es por esto que se necesita el uso de otras técnicas para realizar buenas codificaciones.

No en todos los tipos de redes inalámbricas es posible usar esta técnica. En sistemas donde sea importante el ahorro de energía no es conveniente tener en modo promiscuo los radios de los nodos. De la misma forma, si se tienen limitantes en cuanto al uso de memoria, el almacenamiento en exceso de mensajes tendría un impacto negativo en el espacio de los *buffers* de los nodos. Para sistemas con este tipo de restricciones es necesario buscar otro tipo de soluciones.

- **Codificación oportunista (*opportunistic coding*).** Los nodos que realicen la codificación de mensajes deben garantizar, con una cierta probabilidad, que los nodos receptores podrán decodificar los mensajes a los que se les realizó la operación *xor*. Para esto, debe existir sincronización entre los nodos que conforman la red. Un nodo pasarela debe saber con cierta probabilidad cuáles mensajes tienen almacenados los nodos receptores y así, realizar la codificación de varios mensajes con más certeza.
- **Conocimiento de la red.** Tener un buen conocimiento de la ubicación de los nodos receptores, es un punto clave para tomar una buena decisión de codificación, y así garantizar la correcta decodificación. Para ello es necesario apoyarse en protocolos de enrutamiento, mensajes de notificación y también analizar los tiempos de retardo de los paquetes. Un error de codificación puede impactar en retardos, retransmisiones y pérdidas, lo cual significa ineficiencia en el uso del canal.

En la fig. 3.3 se observa un posible escenario donde se muestra la necesidad de considerar los puntos mencionados anteriormente. Por ejemplo, supongamos que en algún instante de tiempo los nodos *A*, *B*, *C*, *D* y *E* tienen almacenados paquetes en sus *buffers* (fig. 3.3a). Los nodos *A*, *C*, *D* y *E* guardan paquetes que fueron escuchados en modo promiscuo, mientras que el nodo *B* (*pasarela*) almacena los paquetes que le fueron enviados por los otros nodos (fig. 3.3b). La tarea del nodo *B* es despachar sus paquetes usando la menor cantidad de transmisiones posibles garantizando la decodificación en los receptores. En la fig. 3.3c se muestran algunas de las opciones que tendría *B* para codificar. Una opción podría ser realizar la operación *xor* de los paquetes *P1* y *P3* destinados para *C* y *A* respectivamente. De esta manera el nodo *C* podría decodificar a *P1* ya que entre los paquetes que tiene almacenados en su buffer está *P3*. El nodo *C* simplemente realizaría $(P1 + P3) - (P3)$ para obtener a *P1*. Esta sin duda sería una buena codificación, ya que en una sola transmisión tanto *A* como *C* podrían tener su paquete correspondiente. Una mejor opción para *B* sería codificar a *P1*, *P2* y *P3*, de esta manera *A*, *C* y *D* podrían decodificar y obtener sus respectivos paquetes. El nodo *A* realizaría la resta

$(P1+P2+P3)-(P1+P2)$ para obtener a $P3$, C realizaría $(P1+P2+P3)-(P2+P3)$ para obtener a $P1$ y finalmente D restaría $(P1+P2+P3)-(P1+P3)$ obteniendo a $P2$. Sin embargo existe una mejor opción para B en la cual se pueden codificar los cuatro paquetes de su cola de espera y así, enviarse en una sola transmisión. Esta opción es que B codifique a $P1, P2, P3$ y $P4$, lo cual hace posible que tanto A, C, D y E puedan decodificar sus correspondientes paquetes. Con la resta $(P1+P2+P3+P4)-(P1+P2+P4)$ el nodo A podría obtener a $P3$, C realizaría $(P1+P2+P3+P4)-(P2+P3+P4)$ para obtener a $P1$, el nodo D calcularía $(P1+P2+P3+P4)-(P1+P3+P4)$ para obtener $P2$ y, finalmente, E realizaría la resta $(P1+P2+P3+P4)-(P1+P2+P3)$ para obtener su paquete $P4$.

El ejemplo de la fig. 3.3 es un escenario donde existe ganancia en el número de transmisiones y donde la mejor codificación ahorraría tres transmisiones. Para lograr esto, el nodo B debe determinar con alta probabilidad qué paquetes están almacenados en los *buffers* de los nodos A, C y D . Teniendo un buen conocimiento del estado de la red, el nodo B podría saber qué nodos están más cercanos a él y también qué tan cerca están sus vecinos entre sí. Aquí suponemos que el nodo B almacena los paquetes que llegan a su cola de espera un tiempo antes de transmitirlos. También suponemos que los nodos A, C, D y E almacenan los paquetes que escuchan en su área de cobertura que no son destinados para ellos.

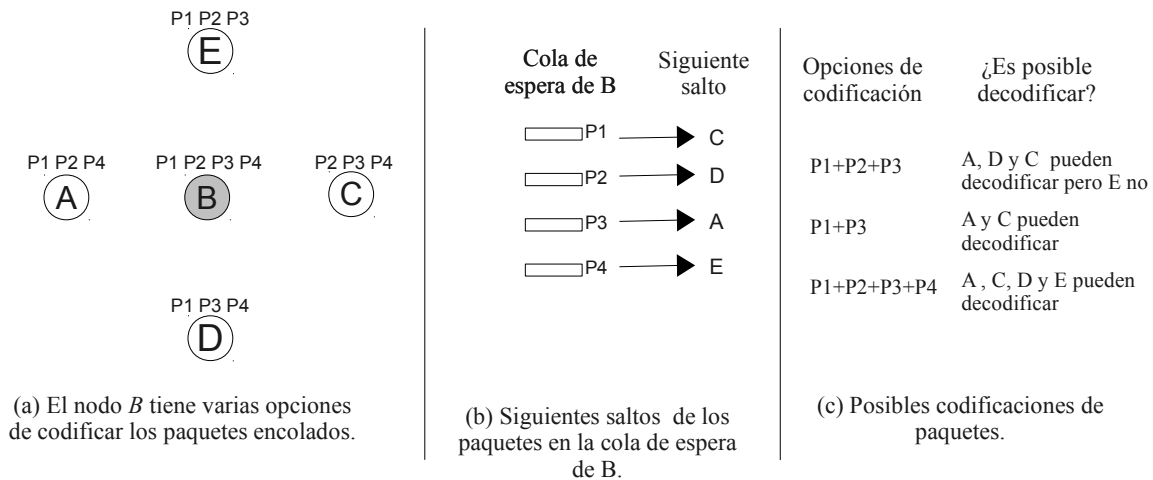


Fig. 3.3. En un instante en la red, B tiene un conjunto de paquetes en su cola de espera. Usando la codificación en red, B debe decidir la manera óptima de entregar los paquetes a sus destinos.

Capítulo 4

Oportunidades de codificación

En este capítulo estudiamos experimentalmente el comportamiento de las oportunidades de codificación que tiene una red inalámbrica en dos distintos escenarios: *Bob y Alice*, y la malla de nodos. Mostramos la manera en que realizamos los cálculos de valores importantes como el caudal de datos, retardo, oportunidades de codificación, ganancia de codificación, entre otros.

La herramienta que utilizamos para realizar nuestras simulaciones es *NS-2*, el simulador más utilizado para la investigación en redes. *NS-2* simula eventos discretos, e implementa una gran cantidad de protocolos sobre las distintas capas del modelo *TCP/IP*.

En nuestras simulaciones, cada uno de los nodos implementa un protocolo de encaminamiento estático, eliminando de esta manera los paquetes de señalización generados por protocolos de encaminamiento. Una razón importante para el uso de este protocolo de encaminamiento es que los nodos participantes permanecen estáticos durante todo el tiempo de la simulación. Con el encaminamiento estático no es necesario que los nodos actualicen sus tablas de rutas, como lo hacen otros protocolos de encaminamiento en redes inalámbricas como *AODV* o *DSDV*. Si los nodos de la red estuvieran en movimiento sería necesario utilizar algún otro protocolo de encaminamiento donde se pudieran actualizar periódicamente las tablas de rutas de los nodos. Al existir movimiento, los nodos deben conocer la topología de la red para así evitar pérdida de mensajes que afecten negativamente en el uso del canal, ocasionando mayor retardo de los paquetes.

4.1 Bob y Alice

El escenario de *Bob y Alice* (fig. 4.1) se utiliza frecuentemente para explicar los beneficios de la codificación en red de una manera sencilla. Una variable importante es la ganancia en cuanto al número de transmisiones. Con el conocimiento de esta variable se puede deducir el número de transmisiones que se ahorran durante una serie de transmisiones y se puede determinar qué tan adecuado es un escenario para implementar la codificación en red en él.

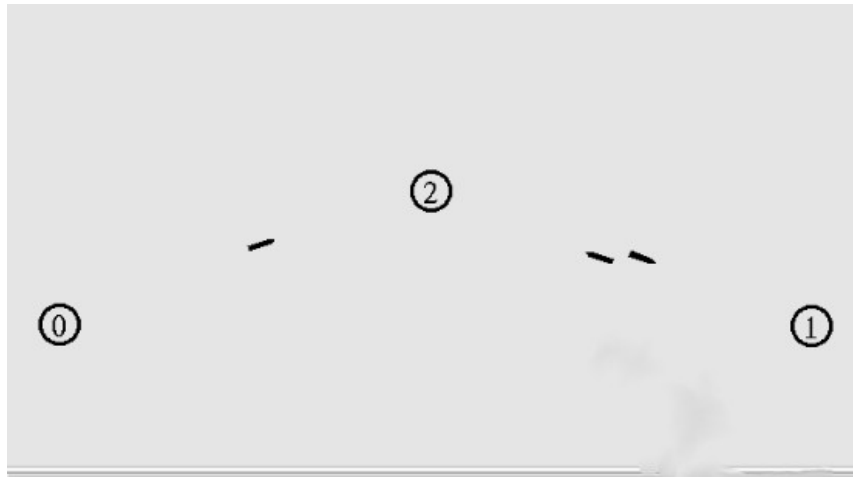


Figura 4.1. Escenario de Bob y Alice en NS-2. Nodo Bob (nodo 0) y Alice (nodo 1) comparten información a través del nodo pasarela (nodo 2).

4.1.1 Ganancia de codificación

La medición de la ganancia de codificación se define como

$$g = \frac{T_x}{T_{x_c}} \quad (4.1)$$

donde T_{x_c} representa el número de transmisiones realizadas para poder compartir paquetes entre los nodos *Bob(0)* y *Alice(1)* usando la codificación en red, en este caso implementadas en NS-2. El valor de T_x representa el número de transmisiones que se realizaron para que los nodos *Bob* y *Alice* intercambiaran información sin el uso de estas técnicas.

En la práctica, el cálculo de esta medida de ganancia se hace contando el número de transmisiones realizadas por los tres nodos en un segundo. Por ejemplo, supongamos que en la simulación donde no se emplea codificación en red se realizan m transmisiones en el segundo i y en la simulación donde sí

se emplean estas técnicas, se realizan n transmisiones en el segundo i . Esto claro, para compartir paquetes desde el nodo *Bob* hacia el nodo *Alice* y viceversa. Por lo tanto la ganancia de codificación en el segundo i será m/n .

4.2 Malla de nodos

Es importante conocer el comportamiento de otros escenarios intuitivamente propicios para aprovechar la codificación en red. Para este propósito, se utiliza la topología mostrada en la fig. 4.2, la cual está formada de cien nodos inalámbricos. Los nodos son separados a una distancia de 200 metros entre sí, con el fin de aumentar más oportunidades de codificación y evitar un mayor número de contiendas por el canal. Cada nodo utiliza el estándar *IEEE 802.11* como capa *MAC*, una antena omnidireccional y emplea encaminamiento estático. Así también, los nodos utilizan a *TCP* y *UDP* como protocolos en la capa de transporte.

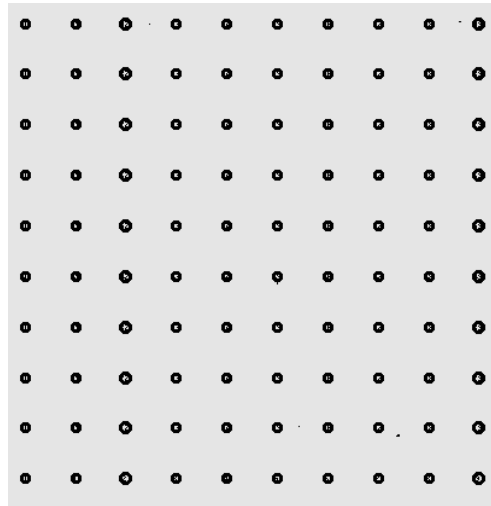


Figura 4.2. Red de nodos en forma de malla simulada en NS-2.

En lo que se refiere a las aplicaciones que generan los datos, se usa tráfico *CBR* para el protocolo *UDP* y una aplicación *FTP* para *TCP*. Las simulaciones se realizan por separado para *UDP* y *TCP*.

Tanto para *TCP* como para *UDP* se simula tráfico cruzado, es decir, se colocaron aplicaciones *FTP/CBR* (según el protocolo de transporte) en los extremos de la malla creando conexiones entre ellas. Usando el ejemplo en la fig. 4.2, si numeramos a los nodos en orden descendente y de izquierda a derecha desde el cero hasta el noventa y nueve entonces, existirían conexiones del nodo cero al nodo

nueve, del nodo diez al nodo diecinueve, etc., y viceversa; conexiones entre los nodos en los extremos de la malla. Usando la misma numeración, existirían conexiones de los nodos del primer renglón de la malla con los del último y viceversa (fig. 4.3). Como nota debemos mencionar que los nodos que no pertenecen a los extremos de la malla, no participan como agentes de generación de información, su principal papel durante las simulaciones es como nodos de encaminamiento (nodos pasarela). Es decir, sólo reenvían paquetes en función de los destinos que obtienen de su tabla de rutas. La razón por la cual no se utilizan a los nodos pasarela como agentes de datos es para evitar la saturación del medio inalámbrico y también para obtener el efecto de tráfico cruzado deseado.

Los resultados que mostramos en este trabajo se obtuvieron de los datos generados por el simulador *NS-2*. No se implementaron todas las técnicas de codificación en red para el escenario de la malla de nodos, sin embargo se realizaron modificaciones necesarias al código del simulador, adecuándolo a las técnicas de codificación sugeridas en [17]. Las omisiones de implementación y consideraciones usadas para la elaboración de este trabajo se enuncian al final del capítulo.

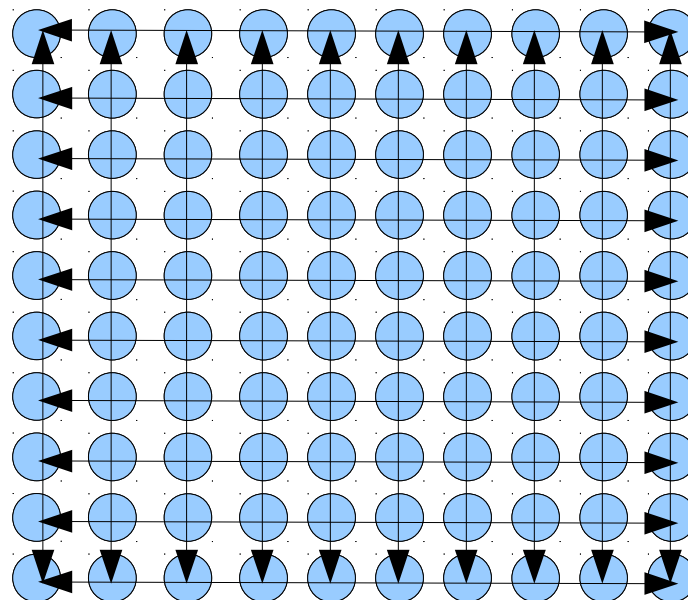


Figura 4.3. Conexiones de datos en la red de malla

El estudio de las oportunidades de codificación se divide en distintas mediciones, estas son: tiempo entre arribos de paquetes, tiempo de codificación, oportunidades de codificación, retardo adicional y caudal de datos.

Para el tiempo entre arribos de paquetes medimos la frecuencia con que llegan paquetes a los nodos, este valor es útil como referencia para compararlo con el tiempo de codificación. La comparación es importante para observar con qué frecuencia llega un paquete a un nodo y también con qué frecuencia se puede codificar un paquete.

Las oportunidades de codificación son valores que nos dan un valor aproximado del número de paquetes que pueden ser codificados en la malla. Este valor está relacionado con el tiempo de codificación y el tiempo entre arribos mencionados anteriormente, esta relación se describirá más adelante.

Para los valores de retardo y caudal de datos se describe de manera formal cómo se realiza su medición, estos valores nos ayudaron a sacar conclusiones sobre el uso de la codificación en red sobre el escenario de la malla de nodos.

Para facilitar la referencia hacia los términos usados en este capítulo haremos uso de la tabla 4.1 en la cual se muestran la nomenclatura y definiciones de estos conceptos.

Tabla 4.1. Definiciones de los términos usados en este capítulo.

| Término | Nomenclatura | Definición |
|--|---------------------|--|
| <i>Tiempo entre arribos</i> | TA | Inverso de la frecuencia promedio de llegadas de paquetes a un nodo. |
| <i>Oportunidad de codificación</i> | OC | Evento en el que se puede realizar una combinación lineal de dos paquetes que llegan hacia un mismo nodo dado un conjunto de condiciones. Esta combinación lineal suele ser una operación <i>xor</i> . |
| <i>Tiempo de simulación de paquete</i> | T | Tiempo asociado a un paquete dentro del archivo de trazas en <i>NS-2</i> , ya sea que el paquete llegue, salga o sea roto por un nodo. |
| <i>Tiempo promedio de codificación</i> | TC | Tiempo promedio que debe esperar un nodo para poder codificar dos paquetes. |
| <i>Caudal de datos real</i> | τ | Valor del <i>caudal de datos</i> contando la ganancia obtenida de codificaciones de paquetes. |
| <i>Retardo adicional</i> | $\bar{\Delta}$ | Tiempo que pierde un paquete esperando a ser codificado en los nodos a lo largo de su ruta. |
| <i>Timeout</i> | t | Tiempo máximo que puede esperar un paquete al ser codificado en un nodo. |

4.2.1. Tiempo entre arribos

Definimos al tiempo entre arribos como la diferencia entre la llegada de dos paquetes provenientes de nodos distintos hacia un mismo nodo. Este cálculo se realiza, como en todas las demás mediciones, con los resultados arrojados por el simulador *NS-2*, que están representados como una serie de eventos discretos llamados archivos de trazas.

Del archivo de trazas utilizamos un valor que denominamos tiempo de simulación de paquete, este tiempo es asociado a un paquete en su arribo, salida o ruptura en un nodo. Este tiempo de simulación lo usamos para calcular valores de importancia en la tesis como el caudal de datos, tiempo entre arribos, tiempo de codificación, etc.

Sea T_j^X el tiempo de llegada del paquete X al nodo j y sea TA_j el tiempo entre arribos promedio al nodo j , entonces;

$$TA_j = \frac{\sum_{X=1}^{n_j-1} (T_j^{X+1} - T_j^X)}{n_j - 1} \quad (4.2)$$

donde n_j es el número de paquetes que llegan al nodo j . En este cálculo suponemos que los paquetes llegan en orden ascendente, es decir $T_j^{X+1} \geq T_j^X \forall X = 1..n_j$. En otras palabras, TA_j es la suma de las diferencias de tiempos de simulación de los paquetes que llegan al nodo j , dividido por el número de muestras que llegan a este nodo.

Finalmente, sea \bar{TA} el tiempo entre arribos promedio en toda la malla, entonces;

$$\bar{TA} = \frac{\sum_{j=0}^{N-1} TA_j}{N} \quad (4.3)$$

con $N=100$, que es el número de nodos en la malla.

4.2.2 Oportunidades de codificación

Llamamos oportunidad de codificación al evento en el cual es posible realizar una combinación lineal de dos paquetes en un nodo. Esta oportunidad se da bajo las siguientes condiciones (ver fig. 4.3):

1. Llega un paquete P_{ki}^X proveniente del nodo k hacia el nodo i , este paquete tiene como próximo salto el nodo j , vecino del nodo i .
2. El nodo i espera un tiempo (*timeout*) t sin enviar el paquete P_{ki}^X a su vecino, el nodo j .

3. Llega un paquete P_{ji}^Y proveniente del nodo j hacia al nodo i , este paquete tiene como próximo salto el nodo k , vecino del nodo i . Para esto, P_{ji}^Y llega dentro del *timeout* t , es decir $T_i^X - T_i^Y < t$, donde T_i^X y T_i^Y son los tiempos de simulación de arribo al nodo i de los paquetes P_{ki}^X y P_{ji}^Y respectivamente.

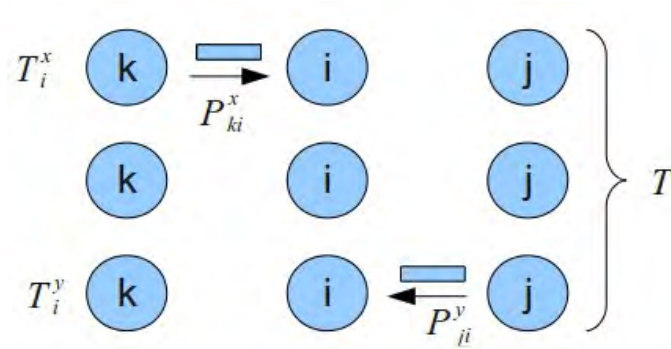


Figura 4.3. Oportunidad de codificación en el nodo i .

Para calcular la fracción de paquetes codificados, se realiza un promedio de las oportunidades de codificación individual por nodo y finalmente se calcula el promedio de paquetes codificados sobre todos los nodos de la malla. En otras palabras, si llamamos a OC_i el número de oportunidades de codificación del nodo i y a \bar{C}_i como la fracción de paquetes codificados en el nodo i entonces,

$$\bar{C}_i = \frac{2(OC)_i}{n_i} \quad (4.4)$$

con n_i como el número de paquetes que llegan al nodo i . Dado que en las modificaciones realizadas al simulador sólo se contempló la codificación de dos paquetes en una sola transmisión, es importante señalar que en el mejor de los casos pueden existir $n_i/2$ oportunidades de codificación, es decir $OC_i \leq n_i/2$. Intuitivamente el valor de \bar{C}_i nos dice cuántos paquetes de los n_i que llegaron al nodo i pudieron codificarse junto con algún otro, es decir cuántas parejas de paquetes, en promedio, pudieron enviarse en una sola transmisión; la idea básica de la codificación en red.

Finalmente para el cálculo total del porcentaje de oportunidades de codificación, sea \bar{C} la fracción de paquetes codificados en toda la red que definimos como

$$\bar{C} = \frac{\sum_{i=0}^{N-1} \bar{C}_i}{N} \quad (4.5)$$

donde N es el número de nodos de la malla.

4.2.3 Tiempo de codificación

Cuando se realizó el cálculo del porcentaje de oportunidades de codificación \bar{OC} , se desconocía el valor esperado de \bar{C} de nuestros experimentos. El tiempo de codificación sirvió para este propósito, conocer el valor esperado de \bar{C} .

Para ilustrar un valor de tiempo de codificación, veamos un ejemplo. Observemos la fig. 4.5 donde se muestran una serie de eventos consecutivos en los cuales el nodo i recibe paquetes de nodos vecinos. En el primer evento, el nodo k envía su paquete P_{ki}^X al nodo i cuyo siguiente salto es el nodo j . A partir de este momento se medirá el tiempo de codificación hasta que llegue un paquete al nodo i , cuyo siguiente salto sea el nodo k y sea enviado por j . Este paquete llega hasta el cuarto evento, que es el paquete P_{ji}^Y enviado por el nodo j cuyo siguiente salto es el nodo k . El tiempo de codificación de los paquetes P_{ki}^X y P_{ji}^Y es $T_i^Y - T_i^X$.

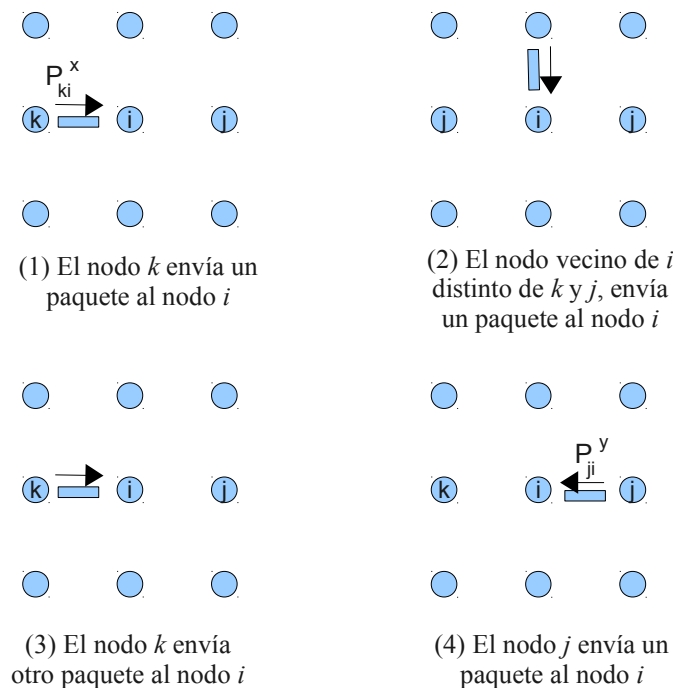


Figura 4.5. Una posible sucesión de eventos en una sección de la malla.

Formalmente llamamos al tiempo promedio de codificación en el nodo i , TC_i , como el tiempo promedio que debe esperar el nodo i para poder realizar una combinación lineal de dos paquetes (la

operación *xor* en nuestro caso) sin esperar un *timeout*. Es decir, no existe alguna restricción de tiempo para poder codificar dos paquetes. La única condición que debe cumplirse es respecto a las rutas que siguen los paquetes que van a ser codificados. Al igual que en una oportunidad de codificación, sean P_{ki}^X y P_{ji}^Y paquetes no necesariamente consecutivos que llegan al nodo i provenientes de los nodos k y j , respectivamente, y con siguientes saltos, j para P_{ki}^X y k para P_{ji}^Y . Sean dos funciones f_s y f_d las cuales, para un determinado paquete, devuelven el salto siguiente y el salto anterior respectivamente, entonces

$$TC_i = \frac{\sum_{P_{ki}^X, P_{ji}^Y \in S_i} |T_i^Y - T_i^X|}{|S_i|} \quad (4.6)$$

donde T_i^X y T_i^Y son los tiempos de arribo de P_{ki}^X y P_{ji}^Y respectivamente, y

$$S_i = \{P_{ki}^X, P_{ji}^Y \in TOT_i \mid T_i^X < T_i^Y, f_s(P_{ki}^X) = f_d(P_{ji}^Y), f_d(P_{ki}^X) = f_s(P_{ji}^Y)\} \quad (4.7)$$

Es decir, S_i es el conjunto de paquetes que puede codificarse sin restricción de *timeout* en el nodo i , con TOT_i como el conjunto total de paquetes del nodo i .

En la práctica este cálculo se realiza procesando los archivos de trazas de *NS-2*. Estos archivos son líneas de texto que representan los eventos que ocurrieron durante la simulación ejecutada. Cada uno de los eventos se ordenan en forma cronológica ascendente, es decir, conforme sucedieron en la simulación. Los eventos representan las acciones que ocurrieron a los paquetes de la red durante la simulación. Algunos de los más importantes atributos con los que cuentan los eventos son:

- tipo de evento (paquete enviado, recibido, roto o reenviado),
- tiempo del evento,
- nodo en el que ocurrió el evento,
- identificador del paquete,
- tipo de paquete (*tcp*, *udp*, *ack*, de señalización),
- dirección *IP* destino y fuente del paquete,
- dirección *MAC* destino y fuente del paquete, y
- TTL^1

1 Time To Live

de los cuales, para el cálculo del TC_i , sólo son relevantes el uso del tiempo del evento y las direcciones MAC destino y fuente de los paquetes recibidos en el nodo i .

Para el cálculo del TC_i se utilizó el algoritmo mostrado en el pseudocódigo de la Tabla 4.2.

```

calcula  $TC_i$ 
  inicia
     $TC_i \leftarrow 0$ 
     $t \leftarrow 0$ 
     $j \leftarrow 0$ 
     $cont \leftarrow 0$ 
  haz
     $P_{ki}^X \leftarrow TOT_i[j]$ 
    haz
       $j \leftarrow j + 1$ 
       $X_b \leftarrow TOT_i[j]$ 
      mientras ( $f_s(P_{ki}^X) = f_d(P_{ji}^Y) \wedge f_d(P_{ki}^X) = f_s(P_{ji}^Y)$ )
         $t \leftarrow t + (P_{ji}^Y - P_{ki}^X)$ 
         $j \leftarrow j + 1$ 
         $cont \leftarrow cont + 1$ 
      mientras  $j < |TOT_i|$ 
     $TC_i \leftarrow \frac{t}{cont}$ 
  regresa  $TC_i$ 
termina

```

Tabla 4.2. Pseudocódigo para calcular el tiempo de codificación promedio.

Una vez calculados todos los TC_i para cada uno de los nodos de la malla, se realiza el cálculo para toda la red que, al igual que en los anteriores cálculos, es el promedio de la suma de cada tiempo de codificación individual sobre el número de nodos involucrados, es decir

$$\bar{TC} = \frac{\sum_{i=0}^{N-1} TC_i}{N} \quad (4.8)$$

donde \bar{TC} es el valor promedio de tiempo de codificación en toda la red.

4.2.4 Caudal de datos (throughput)

El caudal de datos lo medimos como el promedio de los paquetes que llegan a los agentes de datos por segundo (FTP o CBR). Del conjunto de paquetes recibidos en la capa de aplicación se realiza, en

primer lugar, un promedio del caudal de datos por nodo. Después de esta medición, se calcula el caudal de datos global de la red.

Al igual que en los cálculos anteriores, el caudal de datos se mide usando los archivos de trazas del simulador. Estos archivos, como mencionamos anteriormente, están compuestos de eventos de los cuales tomamos sólo algunos atributos y los organizamos en dos conjuntos.

Sea R_i el conjunto de paquetes recibidos por los agentes de datos (las aplicaciones *FTP* o *CBR*) en el nodo i . Definimos a τ_i como el caudal de datos en el nodo i , el cual lo definimos como

$$\tau_i = \frac{|R_i|}{TS} \quad (4.9)$$

donde TS es el tiempo de duración de la simulación. La medición del caudal de datos sólo se realiza en los nodos destino que suponemos numerados del nodo 0 al nodo $M-1$.

Finalmente sea $\bar{\tau}_R$ el caudal de datos promedio de la red, entonces

$$\bar{\tau}_R = \frac{\sum_{i=0}^{M-1} \tau_i}{M} \quad (4.10)$$

donde M es el número de nodos destino en la red. Como mencionamos anteriormente, se mide el caudal de datos por nodo de manera individual y finalmente se realiza un promedio entre los nodos receptores.

4.2.5 Retardo

El retardo se calcula en dos partes: el retardo normal y el retardo adicional, este último siendo el más relevante para nuestra investigación.

Cada paquete tiene un tiempo de retardo el cual lo calculamos como el tiempo que tarda, desde que sale de la capa de aplicación fuente (*FTP/CBR*) y hasta que llega a su respectivo par destino (*end-to-end*). Lo que es de interés para nosotros es conocer cuál es el tiempo adicional que espera un paquete en los nodos intermedios de su ruta para ser codificado, al que llamamos retardo adicional. Formalmente, denominamos retardo adicional al tiempo promedio que espera un paquete a ser codificado a lo largo de su recorrido desde su nodo fuente hacia su nodo destino. A continuación se definen de manera formal el cálculo del retardo y del retardo adicional.

Sea $\bar{\Delta}$ el retardo promedio de un paquete sobre la red de malla, el cual calculamos como

$$\bar{\Delta} = \frac{\sum_{P^x \in R} (t_a(P^x) - t_s(P^x))}{|R|} \quad (4.11)$$

donde t_a y t_s son funciones que calculan el tiempo de salida y llegada en las capas de aplicación del paquete P^x . R es el conjunto que representa a los paquetes que se recibieron en la capa de aplicación de los nodos destino.

Para el cálculo del retardo adicional sumamos el tiempo que el paquete espera en cada nodo intermedio en su ruta hacia su destino. En cada una de nuestras simulaciones una variable que se mantiene constante es el *timeout*, es decir, no existen distintos valores de tiempo de espera para una misma simulación. Lo anterior es importante ya que el tiempo máximo que puede esperar un paquete para ser codificado es el *timeout* definido en la simulación.

Para definir al retardo adicional, antes definamos una serie de elementos que nos permitirán calcular su valor con más facilidad. Sea δ_{P^x} el conjunto de tiempos de codificación empleados por el paquete P^x a lo largo de su camino desde su salida del nodo fuente hasta su llegada al nodo destino, este conjunto está ordenado y cada posición indica el número de salto que efectuó el paquete. Por defecto para todo conjunto δ , el primer y último elemento del conjunto serán cero, esto indica que los paquetes no esperaron ningún tiempo adicional a ser despachados en sus nodos fuente ni destino. Como ejemplo consideremos al paquete P^x que tiene como conjunto de tiempos de codificación δ_{P^x} como sigue:

$$\delta_{P^x} = \{0, 0.001, 0.12, 0.1, 0.01, 0.075, 0.031, 0.001, 0.077, 0\}$$

lo anterior indica que en el primer salto del paquete P^x , el tiempo de espera fue *1ms*, para el segundo salto 12ms y así sucesivamente. La cardinalidad del conjunto puede variar dependiendo el número de saltos que efectúe el paquete, ya sea hasta llegar a su destino o hasta que se agote su TTL. Dada la topología de la red y el tipo de encaminamiento estático que se implementa en cada nodo, se espera que $|\delta| = 10$.

Denotemos ahora a $\delta_{P^x}^i$ como el tiempo de codificación que espera el paquete P^x en su i -ésimo salto. Finalmente podemos definir al retardo adicional Δ como

$$\Delta = \bar{\Delta} - \frac{\sum_{P^x \in R} \sum_{i=0}^{|\delta_{P^x}|-1} \delta_{P^x}^i}{|R|} \quad (4.12)$$

Es decir, sumamos todos los tiempos de espera por paquete, obteniendo su retardo adicional y hacemos lo mismo para todos los recibidos, haciendo finalmente un promedio entre ellos.

Capítulo 5

Resultados de simulación

En este capítulo mostramos los resultados obtenidos para los escenarios de *Bob y Alice*, así como para el escenario de la malla de nodos. Se enuncia la forma en que se realizaron las simulaciones en cuanto a velocidades de transmisión, tiempos de simulación, entre otros parámetros en el simulador.

5.1 *Bob y Alice*

En este apartado mostramos los resultados de una implementación en *NS-2* para el escenario canónico de la codificación en red, *Bob y Alice*. En este escenario la idea es que *Bob y Alice* compartan información usando a su nodo vecino *R* como pasarela.

Se eligió a *NS-2* como simulador el cual es software libre, de esta manera podemos acceder a su código fuente, modificarlo a nuestras necesidades y distribuir el nuevo software con nuestros cambios a la comunidad interesada. Dado que *NS-2* no soporta la codificación en red entre sus funcionalidades por defecto, se modificó al código para que las soportara. Esto con el fin realizar mediciones en cuanto a ganancia en el caudal de datos.

Se realizaron simulaciones con tráfico cruzado entre los nodos *Bob y Alice* tanto para *TCP* como para *UDP*. Los nodos contaron con aplicaciones *FTP* para *TCP* y *CBR* para *UDP* donde la tasa de transmisión para *UDP* fue de *15kbps* y *30kbps* con paquetes de *500 bytes*. La duración de las

simulaciones fue de 200 segundos para *UDP* y de 700 segundos para *TCP*. Esta duración fue elegida de esta manera ya que se realizaron pruebas con tiempos mayores en los cuales se obtenían resultados similares con tiempos más cortos. Las aplicaciones comenzaron a generar paquetes de datos continuamente a desde el primer segundo de la simulación.

Las mediciones obtenidas durante los experimentos son las de ganancia de codificación para *TCP* y *UDP*, las cuales se refieren a la ganancia en cuanto al número de transmisiones realizadas sin usar la codificación en red sobre el número de transmisiones usando codificación. Como mencionamos en el capítulo 4, uno de los objetivos de la investigación es el reproducir los escenarios de Bob y Alice sobre *NS-2* y comparar los resultados obtenidos en [17].

En la fig. 5.1 se muestra una gráfica de la ganancia de codificación de una simulación usando *TCP*. Recordemos que el valor teórico que podemos esperar del escenario de *Bob* y *Alice* en un caso ideal sería de $g = 4/3 = 1.33\bar{3}$. Observamos que los resultados mostrados tienen una tendencia al valor promedio. Los cambios tan bruscos que se muestran en la gráfica son probablemente una consecuencia de que estos cálculos son el cociente de dos simulaciones ajenas. Como habíamos mencionado estas simulaciones son: una haciendo el uso de la codificación en red y otra sin utilizarla.

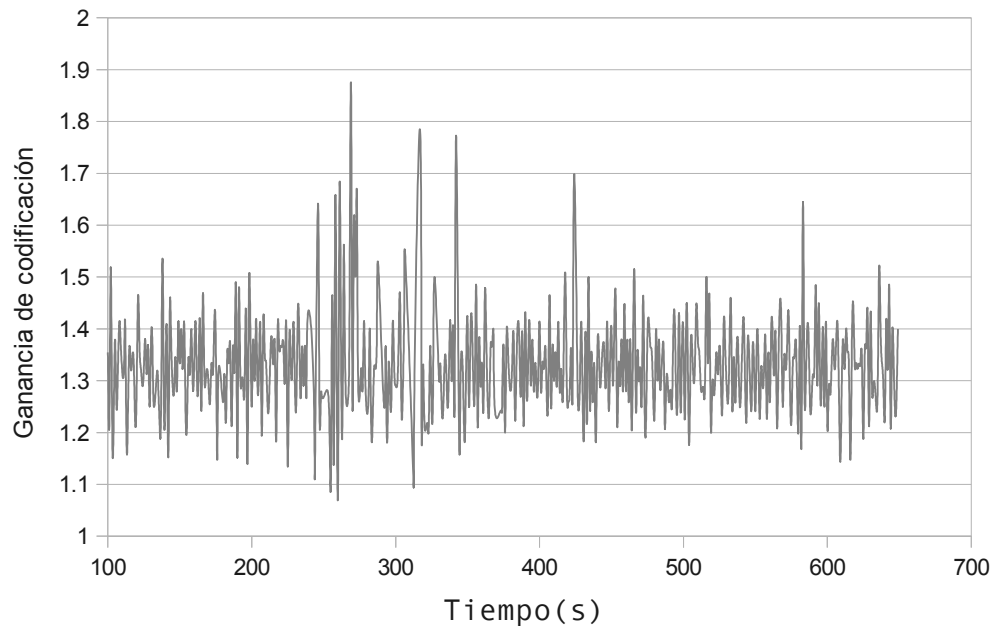


Figura 5.1. Ganancia de codificación sobre *TCP*

Las simulaciones con *UDP* que se muestran en la fig. 5.2, resultaron tener más ganancia de codificación que la obtenida de las simulaciones sobre *TCP*. Esto es ocasionado por las transmisiones extras que existen en *TCP* para los paquetes *ack*, lo cual se refleja con la obtención de un valor menor de ganancia comparado con *UDP*. Los paquetes de *ack* no fueron tomados en cuenta al momento de codificar ya que realizar esta codificación, representa muy poca ganancia a comparación de codificar juntos dos paquetes de datos. Es decir, es preferible que la oportunidad de codificación de un paquete de datos se aproveche con otro paquete de datos, que con un *ack*.

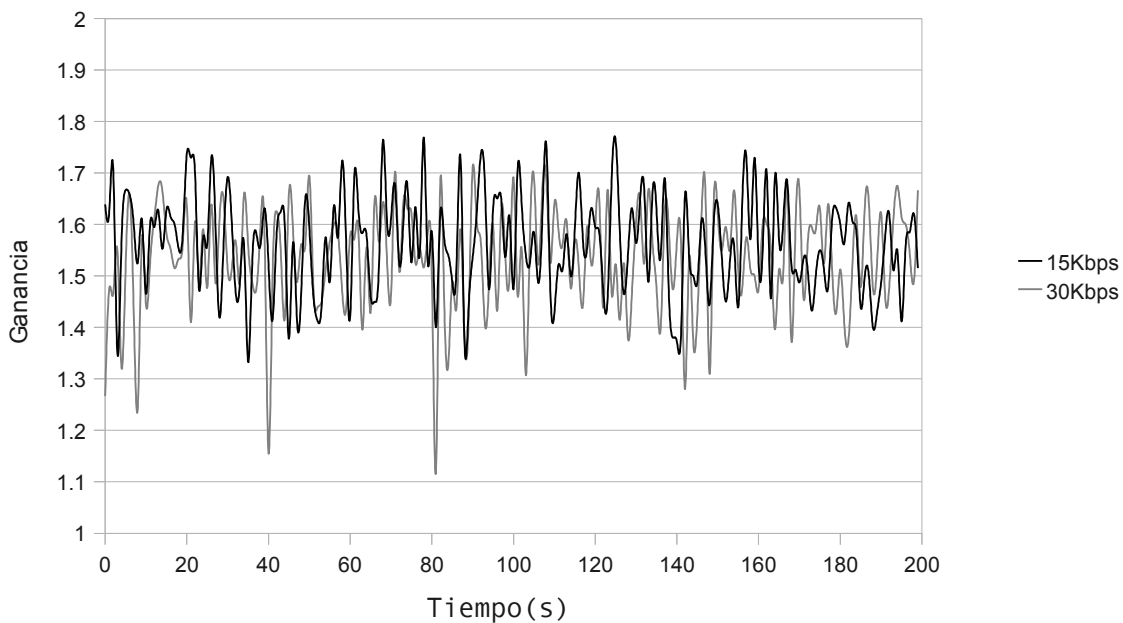


Figura 5.2. Ganancia de codificación sobre UDP

Para observar la tendencia al valor promedio de la ganancia de codificación tanto en *UDP* como en *TCP* en la fig. 5.3 se muestra la gráfica de la función de distribución acumulada o *CDF*. Se observa la tendencia de *UDP* a valores cercanos al 1.6 y de *TCP* a 1.4. Como hemos mencionado, esta diferencia se debe que el codificar paquetes *ack* con paquetes de datos no generan mucha ganancia. Mientras con *UDP* se codifican necesariamente paquetes de datos a pares, en *TCP* puede darse el caso de codificarse un paquete *ack* con uno de datos.

Bob y Alice es un buen escenario para mostrar algunas de los beneficios que ofrece la codificación en red. Escenarios de este tipo pueden encontrarse en muchos lados, por ejemplo en el modo *PCF* de

IEEE 802.11 o en escenarios comunes de redes celulares. En general, se obtuvieron resultados que muestran la ganancia de codificación casi en cualquier instante de las simulaciones tanto en *TCP* como en *UDP*. El número de transmisiones para compartir información entre *Bob* y *Alice* se ahorra un 25% de transmisiones. Este ahorro de transmisiones, es aprovechado por los nodos *Bob* y *Alice* para realizar nuevos intercambios de información.

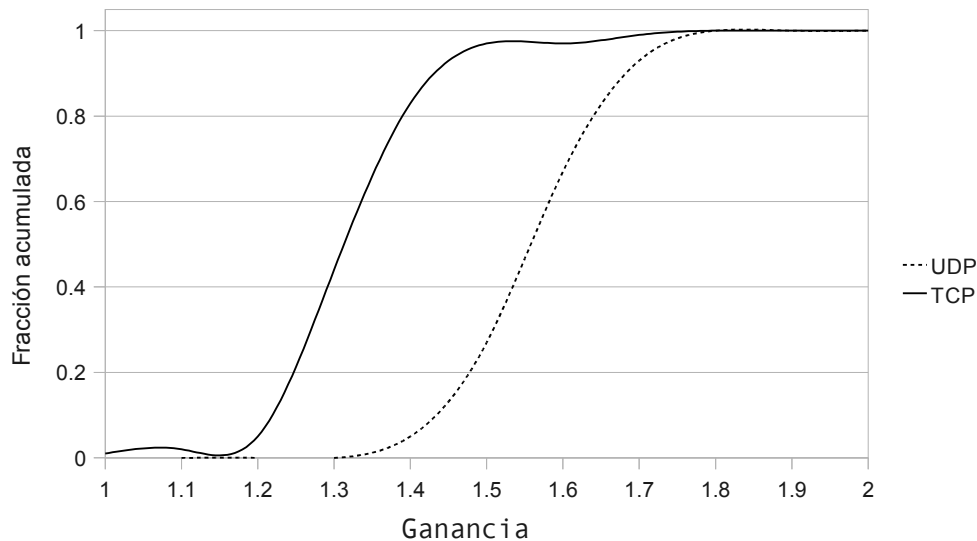


Figura 5.3. CDF de la ganancia en codificación para TCP y UDP a 15kbps

5.2 Malla de nodos

La malla de nodos tiene una estructura que hace que muchos paquetes puedan coincidir en los nodos interiores de la malla, para así poder codificarse. Se probó este escenario bajo tráficos diversos con el objetivo de determinar cómo afecta la carga de datos a las oportunidades de codificación. Intuitivamente podría pensarse que entre más llegadas de paquetes existan en los nodos de la malla, existirían más oportunidades de codificación. Aunque no se implementaron las técnicas de codificación en la malla de nodos, se hicieron adaptaciones al código de *NS-2* para simular la codificación en red.

5.2.1 UDP

5.2.1.1 Oportunidades y tiempos de codificación

Bajo *UDP* se realizaron una gran cantidad de simulaciones. La tasa de transmisión de los agentes *CBR* varió desde el 1kbps hasta los 130kbps. Estas velocidades fueron homogéneas para cada simulación, es

decir, cada uno de los agentes transmitía a la misma tasa. Los paquetes de datos tenían una longitud de 500 *bytes*. Las simulaciones tuvieron una duración de mil segundos y todos los agentes transmitieron desde el primer segundo de la simulación.

Respecto al Tiempo entre arribos de paquetes, en la fig. 5.4 se puede observar que conforme se incrementa la tasa de transmisión, llegan con más frecuencia los paquetes a los nodos. Se podría pensar que este comportamiento fuese el mismo con más altas velocidades de transmisión, cosa que no pasa con velocidades relativamente altas para *NS-2* como a *60kbps* (15.5 paquetes por segundo) en adelante. Con el tiempo de codificación observamos un comportamiento similar al tiempo entre arribos, pero en el sentido de que, a velocidades bajas se obtiene un tiempo de codificación mayor. Mediante se incrementa la tasa de transmisión, el tiempo de codificación baja, pero vuelve a crecer nuevamente. También se puede observar que los valores del tiempo de codificación crecen de una manera más significativa que los tiempos entre arribos, en función de la tasa de transmisión. Estas tasas de transmisión en la práctica son bajas, pero pueden ser útiles para la transmisión de voz con diferente calidad. Ejemplos de estas tasas de transmisión son el estándar de 64kbps o 13kbps usada en redes celulares GSM, por ejemplo.

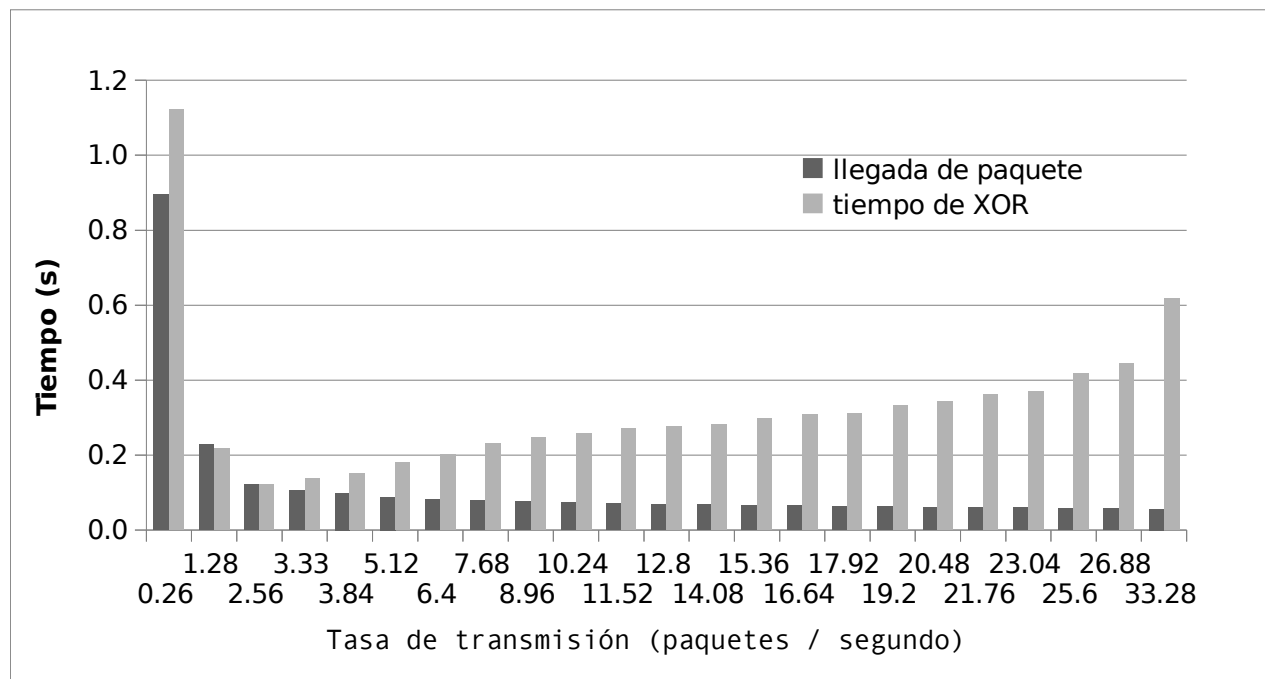


Figura 5.4. Gráfica que muestra los tiempos de arribo contra los tiempos de codificación.

Comparando los valores de los tiempos entre arribos y los de codificación observamos que el tiempo de codificación siempre es mayor al tiempo entre arribos. Si tomamos por ejemplo la tasa de *1kbps* (0.3

paquetes por segundo), podemos decir que, si tenemos un *timeout* de aproximadamente *1250ms* en los nodos, podríamos decodificar casi todos los paquetes. En casos como 10kbps y 15kbps (2.6 y 3.8 paquetes por segundo) observamos que no son necesarios *timeouts* tan grandes ni tan separados de sus correspondientes tiempos entre arribos. Para tasas más altas de transmisión, los tiempos de codificación tienen una diferencia importante con respecto a sus tiempos de arribo, valores que serían inoperables para aplicaciones sensibles al retardo.

En resumen, el tiempo de codificación sería el valor ideal que tendrían que esperar los nodos para poder codificar el mayor número de paquetes (*timeout ideal*), esto sin importar el retardo. Claramente este valor no garantiza el cien por ciento de codificación de paquetes, ya que es un valor promedio, muchos otros paquetes podrían requerir un *timeout* mayor. Es decir, muchos de ellos podrían no ser codificados. El porcentaje de paquetes codificados se muestra en la gráfica de la fig. 5.5.

Podemos decir que, para tasas de transmisión bajas existe una mayor probabilidad de codificación de paquetes que con tasas de transmisión más altas. Observemos que difícilmente se alcanzan grandes porcentajes de paquetes codificados al ir incrementando la velocidad de transmisión, esto puede ser debido a que muchos paquetes, candidatos a ser codificados, requieran más tiempo para ser codificados que el tiempo de codificación promedio.

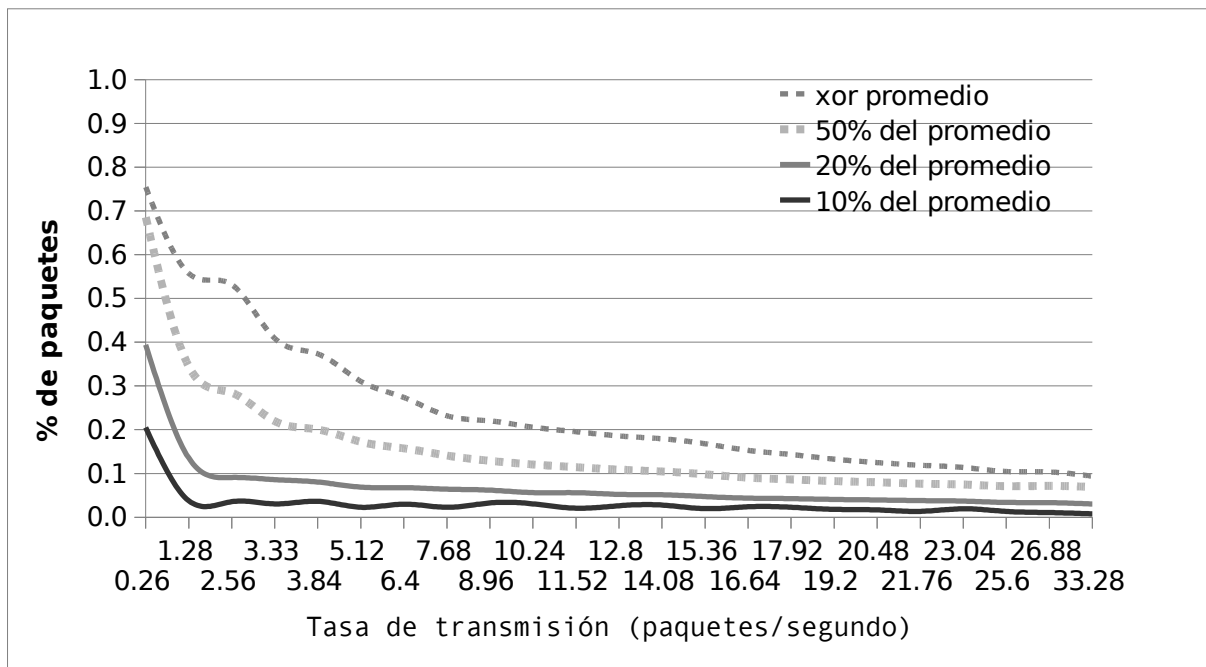


Figura 5.5. Gráfica que muestra el porcentaje de paquetes codificados usando el tiempo de codificación promedio, el 50%, 20% y 10% de éste.

En general la gráfica de la fig. 5.5 nos muestra que el arribo de paquetes se distribuye de manera uniforme. Por ejemplo, si usamos un *timeout* del 50% del tiempo de codificación promedio para digamos *15kbps*, se esperaría obtener la mitad de los paquetes codificados de los obtenidos usando el valor promedio del tiempo de codificación (fig. 5.4), de igual manera para las curvas que representan al 20% y al 10%.

Ahora sería interesante conocer cuál es la cantidad de paquetes codificados, en general para un *timeout* fijo a distintas velocidades de transmisión. Este *timeout* se tomó arbitrariamente desde los 100ms hasta 1s. Lo que podríamos intuir es que a un tiempo de espera menor, menores serían las oportunidades de codificación y, por lo tanto, menor porcentaje de paquetes serían codificados. La gráfica de la fig. 5.6 muestra la tendencia del porcentaje de paquetes con oportunidad de ser codificados, donde se aumentó gradualmente el *timeout*. Como se intuía, entre mayor es tiempo de espera en un nodo, más paquetes tienen la oportunidad de codificarse. Sin embargo, observamos que como se muestra en la gráfica de la fig. 5.6, las tasas de transmisión altas no son beneficiadas con las oportunidades de codificación como en las de baja velocidad. Mientras que a 1kbps con 100ms de *timeout* se obtiene un 10% de paquetes codificados, si aumentamos el *timeout* a 1s sólo se podrían codificar un 20%. En un mejor escenario como los 10kbps, observamos que no son necesarios *timeouts* tan grandes para obtener un buen porcentaje de paquetes codificados. Observamos que a partir de tasas mayores a los 10 paquetes por segundo los paquetes codificados no se incrementan de manera significativa.

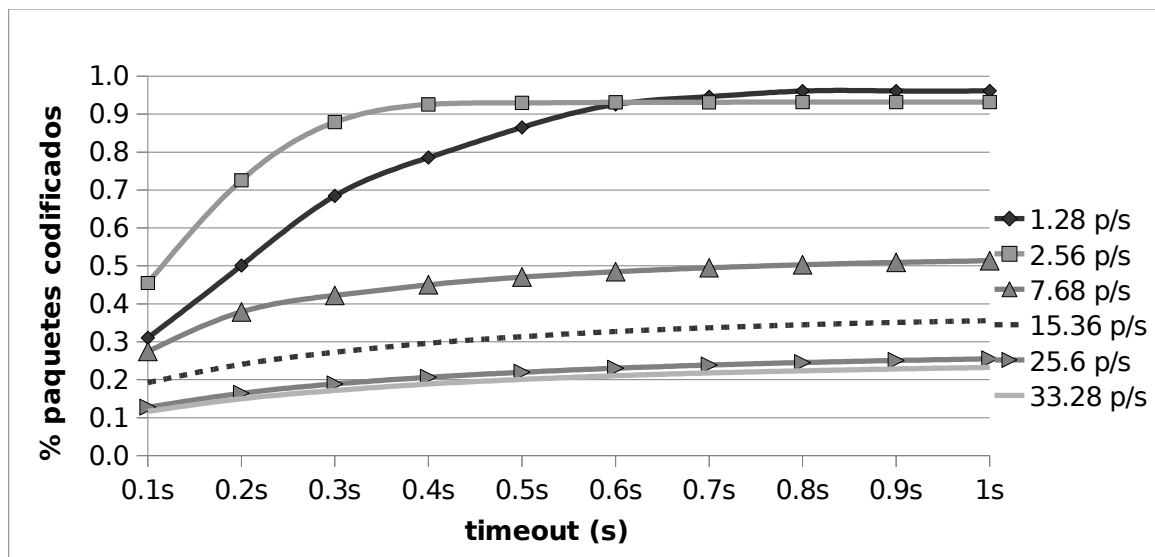


Figura 5.6. Gráfica del porcentaje de paquetes codificados variando el *timeout*

5.2.1.2 Caudal de datos

En lo que se refiere a la medida del caudal de datos, en la gráfica de la fig. 5.7 se presentan una serie de curvas que muestran el valor de τ_R para distintos *timeouts* (100ms, 200ms, 500ms y 900ms). También en la misma gráfica se muestra la curva para las simulaciones sin uso de codificación. Se observa que la tendencia de crecimiento de las curvas es similar, llega un momento donde el caudal de datos toma su valor más alto, para después decrementar cuando la tasa de transmisión se incrementa.

Observamos que el valor más alto del caudal de datos se alcanza con tasas de transmisión de 15kbps y 20kbps, y se hace prácticamente nula a partir de los 60kbps. Esta ganancia nula puede ser ocasionada muy probablemente por la saturación en el canal y la pérdida de paquetes en las colas de espera, lo que impide una comunicación eficiente en la red y grandes retardos de los paquetes. Existen valores de ganancia en el caudal de datos superiores del 50% con respecto a la curva que no usa codificación para *timeouts* desde los 100ms. También se observa que, a contrario de lo que diría la intuición, tener un *timeout* más largo no garantiza un caudal de datos real mayor. Notemos que la curva de 500ms tiene una ganancia mayor con respecto a la curva de los 900ms.

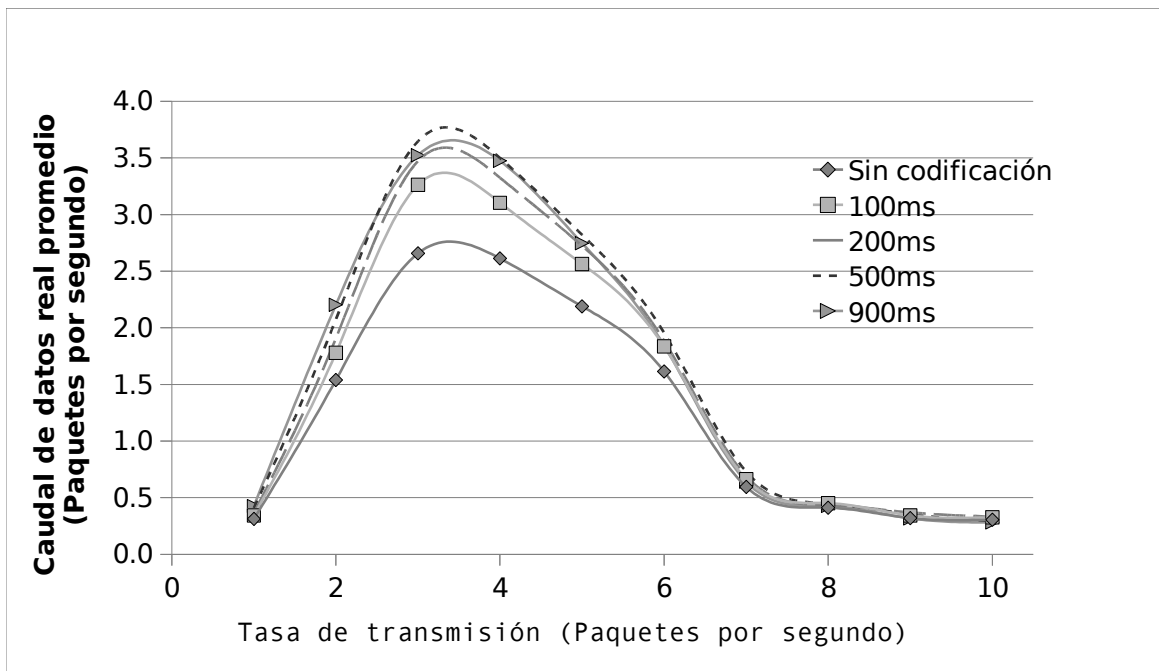


Figura 5.7. Gráfica de la tasa de transmisión contra el caudal de datos real a distintos *timeouts*.

Otro dato interesante que se pudo obtener de las simulaciones fue cómo varía el caudal de datos real en función del *timeout*. En la gráfica de la fig. 5.8 se muestra el comportamiento de la variable τ_R sobre

distintas tasas de transmisión. En la mayoría de las curvas que representan velocidades bajas de transmisión, se muestra un incremento en τ_R considerable al incrementarse el *timeout*, pero en tasas de transmisión mayores, el incremento del *caudal de datos* real es pequeño, hasta para *timeouts* altos. Otra cosa que puede observarse es que existen puntos máximos en los valores de τ_R , los cuales no necesariamente se alcanzan en los *timeouts* más altos, por ejemplo el de *1s*.

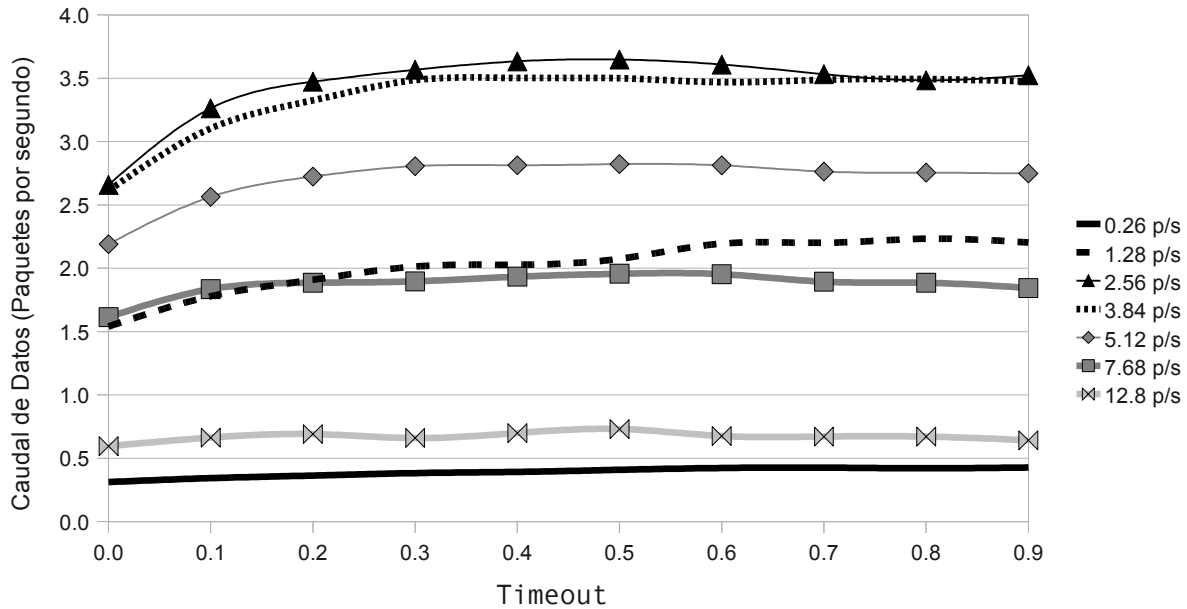


Figura 5.8. Caudal de datos real variando el *timeout*, sobre distintas tasas de transmisión.

5.2.1.3 Retardo

Como se mencionó anteriormente, el valor que calculamos fue el retardo adicional. Este indica la suma de los tiempos de codificación (promedio) de un paquete a lo largo de su ruta. La gráfica de la fig. 5.9 muestra los valores del retardo adicional obtenidos de las simulaciones bajo distintos valores de *timeout*.

Observamos que existe una tendencia creciente con respecto a los tiempos de codificación. Algo natural que podemos observar es que, entre más tiempo espera un nodo para codificar paquetes, más retardo adicional existirá y por lo tanto el retardo será mayor. Otro dato interesante que podemos notar es que, a velocidades bajas, el retardo adicional no crece a gran magnitud. Este dato se debe a que, para velocidades altas, no es necesario tanto tiempo para poder codificar un paquete y como hemos mencionado antes, el *tiempo de codificación* no es muy grande para velocidades bajas.

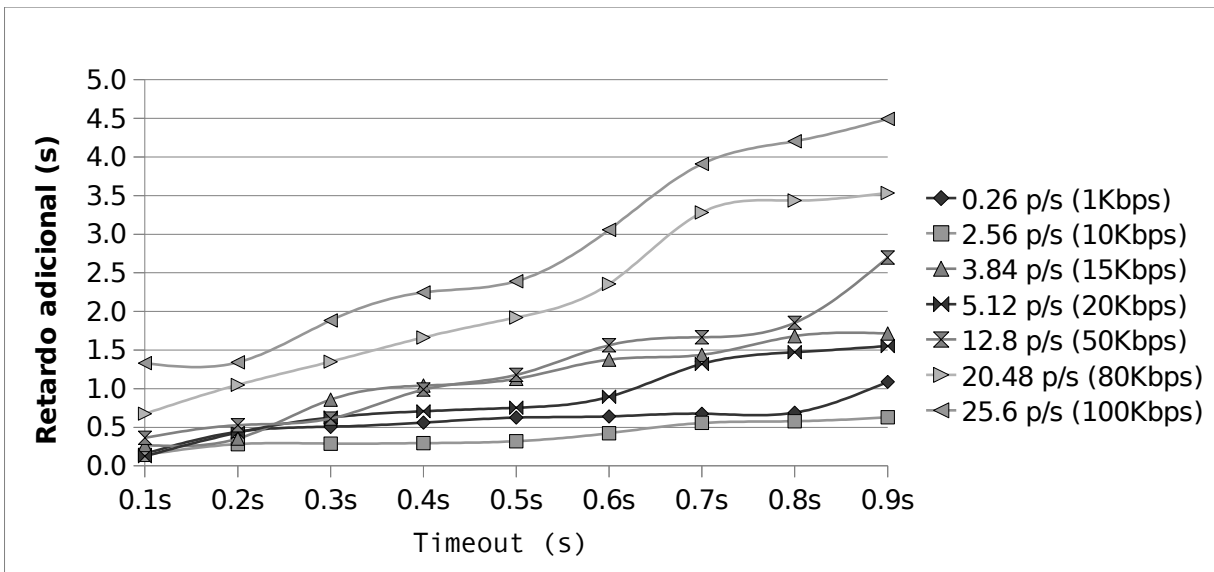


Figura 5.9. Valores del retardo adicional variando el *timeout*, sobre distintas tasas de transmisión.

5.2.2 TCP

5.2.2.1 Oportunidades de codificación

A diferencia del protocolo *UDP*, en *TCP* existe el factor de los paquetes *ack*, por lo que el estudio se dividió en tres tipos de oportunidades de codificación las cuales son:

- *Oportunidad de codificación de datos*. Ocurre cuando los paquetes involucrados en una oportunidad de codificación son únicamente de datos.
- *Oportunidad de codificación mixta*. Ocurre cuando los paquetes involucrados de la oportunidad de codificación son uno de datos y otro de *ack*.
- *Oportunidad de codificación de ack*. Los dos paquetes son de *ack*.

Es importante dividir el estudio en estos tres tipos de codificaciones ya que es de interés conocer qué tan probable es que se codifiquen paquetes de datos, ya que es cuando existe mayor ganancia en cuanto a la cantidad de datos enviados en una transmisión.

En la fig. 5.10 se muestran las curvas que representan el porcentaje de oportunidades de codificación sobre *TCP*. Se observa que existe mayor probabilidad de codificar dos paquetes de tipos distintos (uno de datos y otro de *ack*) que un par de paquetes del mismo (dos de datos o dos *ack*), cuyas curvas son parecidas. Se observa un crecimiento importante para las oportunidades de codificación mixta con el

aumento en los valores del *timeout*, mientras que en las oportunidades de codificación de *ack* y de datos el crecimiento no es tan acelerado.

Este resultado nos dice que existe una mayor probabilidad de codificar dos paquetes de distintos tipos (datos con *ack*), por lo tanto, la ganancia en el caudal de datos podría ser no muy significativa, ya que al combinar un paquete de datos con un paquete de *ack* sólo se ahorraría la transmisión de este último. Como se muestra en la fig. 5.11, en las oportunidades de codificación mixtas sólo existirían ahorros del tamaño de los paquetes de *ack*, las cuales tendrían más ocurrencias que las de datos y las de *ack*.

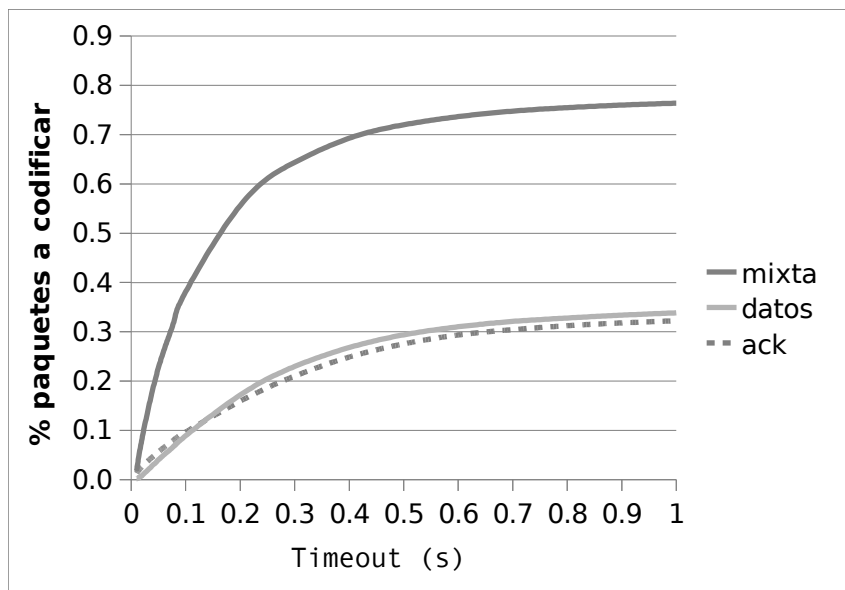


Figura 5.10. Gráfica de las oportunidades de codificación sobre TCP

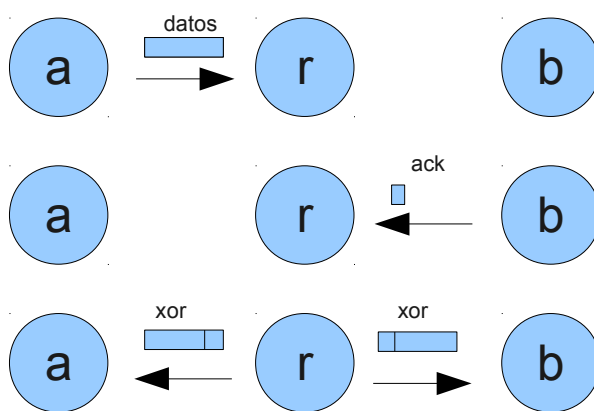


Figura 5.11. Ejemplo de codificación mixta, se ahorra la transmisión de un paquete *ack*

5.2.2.2 Retardo adicional

Los valores para el retardo adicional en *TCP* son pequeños a diferencia de *UDP*. La fig. 5.11 muestra que el valor del retardo mantiene un crecimiento casi lineal conforme se incrementa el valor del *timeout*. Para *TCP* se realizaron simulaciones de dos mil segundos, esto con el fin de que el protocolo llegara a su estado estacionario. Se utilizó el agente *TCP* por defecto en el simulador. Observamos que para los tres tipos de oportunidades de codificación no se muestra diferencia significativa, lo cual nos dice que el valor del retardo no se ve afectado por el tipo de oportunidad de codificación que usemos.

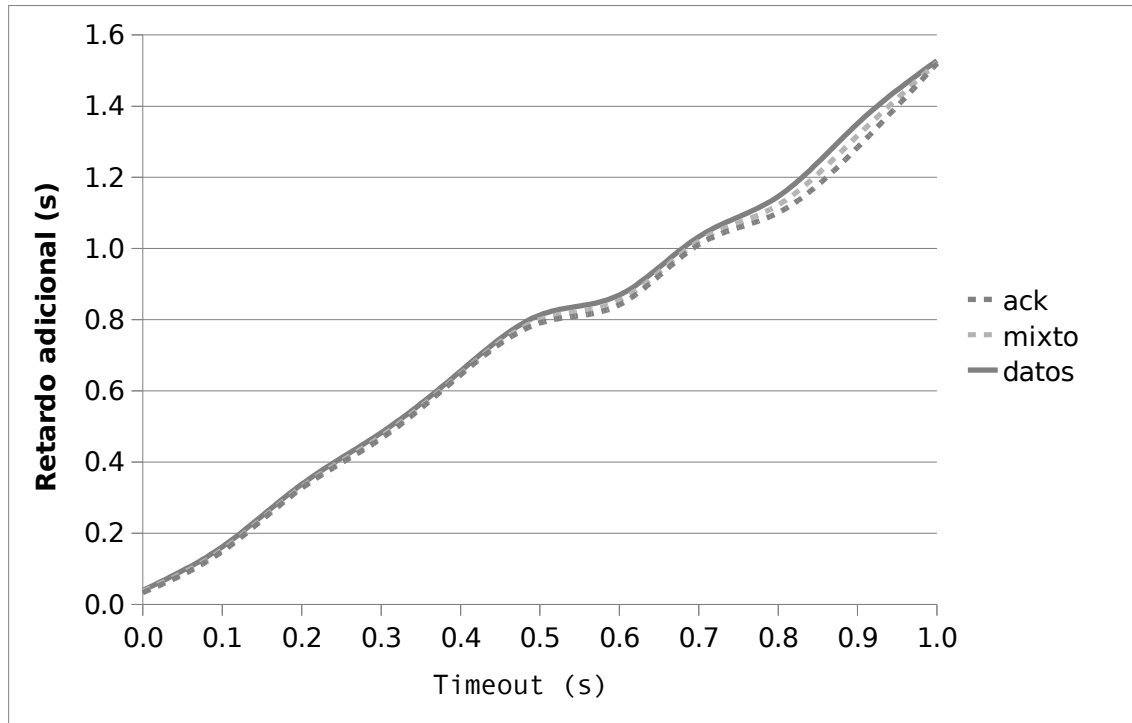


Figura 5.11. Gráfica que muestra el retardo adicional sobre el protocolo *TCP* para las tres oportunidades de codificación: mixta, de datos y de *ack*.

5.2.2.3 Caudal de datos

El caudal de datos se ve directamente afectado por las oportunidades de codificación y por el retardo adicional. Se esperaría ver una ganancia en cuanto valor de τ_R (caudal de datos real promedio), pero de acuerdo a los resultados experimentales mostrados en la fig. 5.12, en las cuales se muestra la comparativa de los valores de $\bar{\tau}$ y τ_R , se puede concluir que el *timeout* es un factor que altera de forma negativa al caudal de datos. El aumento en el valor del *timeout*, hace que los valores de $\bar{\tau}$ y τ_R

disminuyan de manera significativa en lugar de reflejarse como una ganancia.

Mencionemos que para este cálculo se tomaron en cuenta únicamente las oportunidades de codificación de datos, descartando las oportunidades de codificación mixtas y de *ack*. Intuitivamente el tipo de oportunidades de codificación de datos es donde existiría una ganancia más representativa.

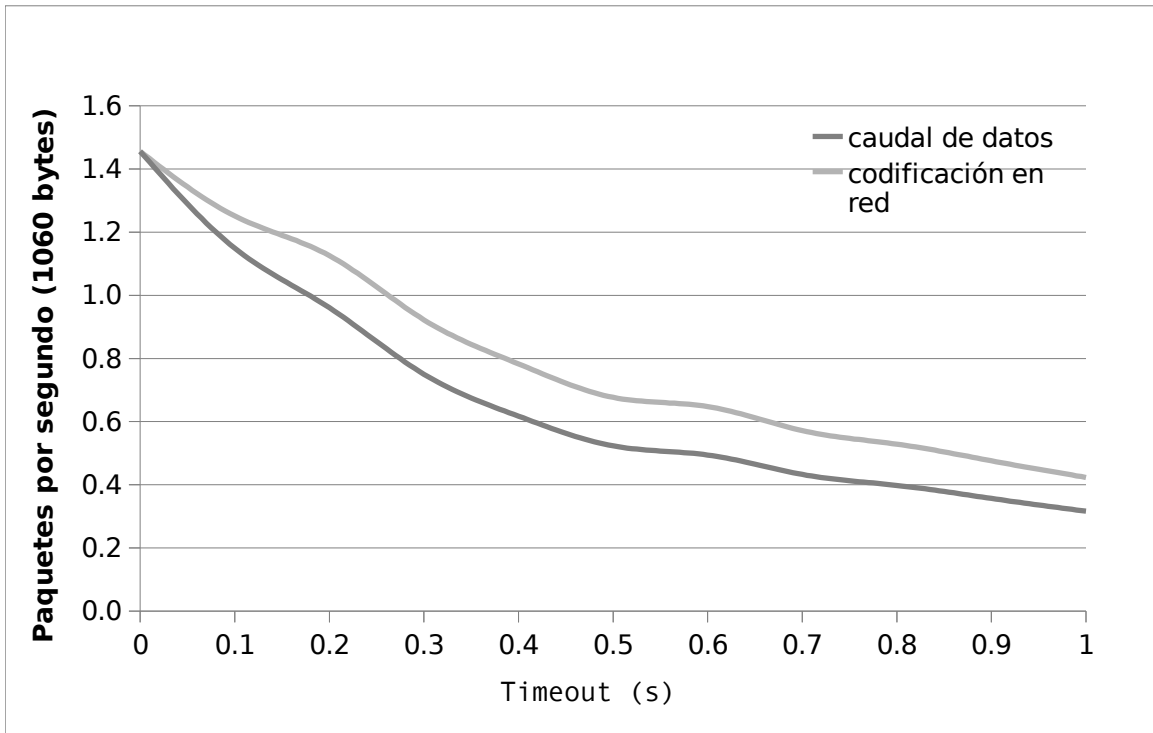


Figura 5.12. Comparativa de los valores del caudal de datos promedio y el caudal de datos promedio real sobre TCP.

De estos resultados podemos concluir que incorporar un *timeout* en los nodos genera un retardo mayor en los paquetes, alterando el desempeño de *TCP* que se refleja en el decremento del caudal de datos en la red. La alteración tan constante del *RTT* podría ocasionar que los agentes *TCP* esperasen retardos menores de sus paquetes y por lo tanto reenvíos innecesarios de éstos. El retraso de paquetes en los nodos crea inestabilidad en el comportamiento del protocolo *TCP*.

Usar la codificación en red en escenarios como la malla de nodos beneficia a velocidades bajas sobre *UDP*. El retardo es un factor que es afectado cuando se usan estas técnicas, lo cual puede verse en nuestros resultados sobre *TCP*, en donde retrasar un paquete puede afectar el comportamiento global del protocolo. Observamos que aunque exista más carga de tráfico en la red, no existen mayores ganancias en la codificación.

En los casos donde se obtienen ganancias, como en *UDP*, se observa que el valor del *timeuot* es determinante para aprovechar los beneficios en el caudal de datos que ofrece la codificación. Tener un *timeout* muy largo genera más oportunidades de codificación, pero genera retardos más grandes, lo cual afecta directamente al caudal de datos.

Capítulo 6

Conclusiones

En este trabajo se realizó un estudio de las oportunidades de codificación en redes inalámbricas tipo *ad hoc*. En nuestro primer escenario de *Bob* y *Alice*, se realizaron simulaciones para crear oportunidades de codificación y así, mostrar la ganancia que se puede obtener al implementar la codificación en red. Se mostró que este tipo de escenarios sencillos, tanto en *TCP* como en *UDP* muestran ganancias importantes en el valor del caudal de datos. Estas ganancias, como hemos verificado de [17], hacen de la codificación en red una técnica útil para poder aplicarse en redes inalámbricas tipo *ad hoc*. Escenarios donde muchos nodos estén suscritos a un nodo pasarela para intercambiar información, son ideales para implementar estas técnicas de codificación.

Pese a que existen paquetes de señalización como los *RTS* y *CTS* para evitar las colisiones, y el algoritmo de *exponential backoff* para hacer más justa la competencia del canal inalámbrico, concluimos que a pesar de la existencia de estos factores, se obtienen buenos resultados en términos de ganancia en el caudal de datos.

De igual forma, para el escenario de *Bob* y *Alice*, se mostró que existe un fenómeno interesante que

refleja una ganancia en *el caudal de datos* mayor para *UDP* que para *TCP*. Este resultado es debido a que para *TCP* existen transmisiones de paquetes *ack*, las cuales también generan contiendas en el canal inalámbrico disminuyendo la probabilidad de codificar dos paquetes de datos.

Las simulaciones realizadas sobre la malla de nodos, tanto para *TCP* como para *UDP*, se crearon para observar el comportamiento de las técnicas de codificación. El objetivo de las simulaciones fue, además de determinar el porcentaje de paquetes se podrían codificar, observar las medidas del caudal de datos y retardo.

Con *UDP* se observaron ganancias altas en el caudal de datos. A mayores velocidades de transmisión no se observaron ganancias significativas en el caudal de datos, inclusive si el *timeout* en los nodos se incrementaba. Podemos concluir que el uso de la codificación en red en la malla bajo el protocolo *UDP* puede hacer más eficiente el uso de la red en condiciones de poca carga de tráfico. Altas velocidades de transmisión no generan mayores ganancias en el caudal de datos. El valor del caudal de datos se ve afectado principalmente por la congestión en la red, ya que si los nodos transmiten a tasas altas, hay más probabilidad de perder paquetes en las colas de espera de los nodos, además de generar más contiendas por el canal y probablemente colisiones.

Entre más se incrementa el *timeout*, el valor del retardo total también se incrementa, haciendo al caudal de datos aún más pequeño. La implementación de la codificación en red en una red de malla sería una mala opción en aplicaciones sensibles al retardo bajo velocidades de transmisión altas.

Con *TCP* se obtuvieron resultados interesantes que indican que el retardo afecta negativamente en el desempeño del protocolo pues no se obtuvieron ganancias en el caudal de datos bajo ningún valor del *timeout*. Podemos decir que colocar un *timeout* en los nodos para intentar codificar paquetes, altera la eficiencia de *TCP* de manera negativa. No se obtiene ganancia alguna aumentando el valor del retardo. Resulta mejor no usar la codificación en red cuando se usa *TCP* en este tipo de escenarios como la red de malla.

Sin embargo, conocer la frecuencia con la que se pueden codificar paquetes de datos con los paquetes *ack*, podría tener alguna aplicación con el fin de reducir el número de contiendas. Los paquetes *ack* podrían ser enviados con algún paquete de datos y así evitar una contienda más por el canal.

Escogiendo un valor de *timeout* adecuado pueden alcanzarse ganancias en el caudal de datos sin alterar en gran medida el retardo. Algunas aplicaciones en tiempo real como *VoIP* podrían soportar el uso de estas técnicas sin alterar su funcionamiento. En *TCP* y *UDP* el escenario de *Bob* y *Alice* obtuvo buenos resultados que podrían ser utilizados en sistemas con infraestructura aprovechando las oportunidades de

codificación que existen, incrementando el caudal de datos.

Trabajo a futuro

La implementación en *NS-2* de *Bob* y *Alice* resultó exitosa, más sin embargo en la malla de nodos no se pudo completar, es por eso que en la malla de nodos se realizó un aproximación en la cual se intentó apegarse de la mejor manera a la realidad.

Se podrían realizar otros estudios con distintos escenarios, por ejemplo en nodos con movimiento que podrían no ser favorables para el desempeño de la codificación en red. Un estudio de escenarios aleatorios con tráfico mezclado entre *TCP* y *UDP* a distintas tasas de tráfico sería interesante para saber las oportunidades de codificación.

En nuestra aproximación se realizó el estudio con la codificación de dos paquetes únicamente, pero en el escenario de malla podría existir una codificación de cuatro paquetes. Un resultado interesante sería conocer qué impacto tiene el poder codificar más paquetes.

Referencias

- [1] B. Crow et al., "IEEE 802.11 Wireless Local Area Networks," IEEE Commun. Mag., vol. 35, no. 9, sept. 1997, pp. 116–26.
- [2] C. Perkins, "Mobile IP," IEEE Commun. Mag., vol. 35, no. 5, may 1997, pp. 84–99.
- [3] ISO/IEC and IEEE Draft International Standard, “ Part II : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, ISO/IEC 8802-11, IEEE P802.11/D10, ene. 1999.
- [4] IEEE Standards for Local Area Networks: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standard 802.3-1985, 1985.
- [5] C. Perkins and E. Royer, Ad hoc on-demand distance vector routing, IEEE WMCSA'99 (1999).
- [6] Tanenbaum Andrew S.: Computer Networks, Prentice Hall, Inc., 1996, ISBN 7-302-02410-3.
- [7] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: The insecurity of 802.11. In The Seventh Annual International Conference on Mobile Computing and Networking (MobiCom 2001), 2001.
- [8] <http://reflexor.com/trac/a51>, consultado en noviembre de 2010.

- [9] www.backtrack-linux.org, consultado en noviembre de 2010.
- [10] Changling Liu, Jörg Kaiser, "A Survey of Mobile Ad Hoc network Routing Protocols" Technique report, University of Magdeburg, <http://www.minema.di.fc.ul.pt/papers.html>, octubre 2010.
- [11] David B. Johnson, David A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks", Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Capítulo 5, páginas 153-181, Kluwer Academic Publishers, 1996.
- [12] Jiang, R., Gupta, V., Ravishankar, C.V.: Interactions Between TCP and the IEEE 802.11 MAC Protocol, in DARPA Information Survivability Conference and Exposition (DISCEX), Volumen I, 2003
- [13] Ahlswede, R., Cai, N., Li, S., And Yeung, R. Network information flow. Information Theory, IEEE Transactions on 46, 4 (2000), 1204–1216.
- [14] Fragouli, C., and Widmer, J. Network coding: an instant primer. ACM SIGCOMM Computer Communication Review 36, 1 (2006), 63–68.
- [15] C. Fragouli and E. Soljanin, "Network coding fundamentals," Foundation and Trends in Networking, vol. 2, no. 1, pp. 1–133, 2007.
- [16] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. IEEE Transactions on Information Theory, 2003.
- [17] S. Katti, H. Rahul, D. Katabi, W. H. M. Médard, and J. Crowcroft. XORs in the Air: Practical Wireless Network Coding. In Proc. of ACM SIGCOMM 2006, Pisa, Italia.
- [18] R. Koetter and M. Médard. An algebraic approach to network coding. IEEE/ACM Transactions on Networking, Volume 11, Issue 5, oct. 2003, Páginas:782 - 795.
- [19] J. Widmer, C. Fragouli, and J. Y. LeBoudec. Energy efficient broadcasting in wireless ad hoc networks. In NetCod, abr. 2005.
- [20] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in Proc. of IEEE Infocom, may 2007, pp. 1028–1036.