



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

UNIDAD IZTAPALAPA

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
POSGRADO EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

TESIS DOCTORAL

Codificación adaptativa de red para sistemas inalámbricos IEEE 802.11s en modo infraestructura

Adaptive Wireless Network Coding for IEEE 802.11s Infrastructure Wireless Mesh Networks

PRESENTADA POR

Carlos Ernesto Carrillo Arellano

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

DIRECTOR DE LA INVESTIGACIÓN DOCTORAL

Dr. Víctor Manuel Ramos Ramos

Departamento de Ingeniería Eléctrica, UAM-Iztapalapa

MIEMBROS DEL JURADO

PRESIDENTE: Dr. Enrique Stevens Navarro

Facultad de Ciencias, Universidad Autónoma de San Luis Potosí

SECRETARIO: Dr. Víctor Manuel Ramos Ramos

Departamento de Ingeniería Eléctrica, Universidad Autónoma Metropolitana - Unidad Iztapalapa

VOCAL: Dr. Ulises Pineda Rico

Facultad de Ciencias, Universidad Autónoma de San Luis Potosí

VOCAL: Dra. Graciela Román Alonso

Departamento de Ingeniería Eléctrica, Universidad Autónoma Metropolitana - Unidad Iztapalapa

VOCAL: Dr. Gerardo Abel Laguna Sánchez

Departamento de Sistemas de Información y Comunicaciones, Universidad Autónoma Metropolitana -
Unidad Lerma

DISERTADA PUBLICAMENTE EN LA UAM-IZTAPALAPA EL DÍA 25 DE ENERO DE 2023.

Codificación adaptativa de red para sistemas inalámbricos IEEE 802.11s en modo infraestructura

Resumen

LAS redes inalámbricas malladas IEEE 802.11s en modo infraestructura, denominadas comúnmente como iWMNs (*Infrastructure Wireless Mesh Networks*), están constituidas por nodos inalámbricos estáticos capaces de trabajar coordinadamente para encaminar paquetes de datos. De esta manera, los nodos colaboran para poder intercambiar información entre sí. Más aún, las iWMNs pueden ser interconectadas con otras tecnologías de red y, de este modo, coadyuvar a extender inalámbricamente la cobertura de estas redes; por ejemplo, las iWMNs se emplean hoy en día para extender la cobertura de redes celulares o de redes cableadas. Gracias a estas características, y también a su bajo costo de infraestructura, las redes iWMNs son consideradas hoy en día como una excelente opción para ofrecer servicios de conectividad inalámbrica a Internet en zonas geográficas donde el uso de otras tecnologías resulta inviable.

A pesar de las prometedoras características de las iWMNs; existen estudios y resultados que plantean dudas sobre su desempeño, ya que se ha documentado que el rendimiento de estas redes puede ser afectado por numerosos factores; tales como el uso de TCP para transportar información en entornos inalámbricos, la tasa de errores en el medio inalámbrico, así como la contienda por el acceso al medio entre usuarios de la red. Todos estos factores pueden degradar las prestaciones de las iWMNs y, consecuentemente, afectar la calidad de la experiencia que reciben los usuarios.

En esta tesis doctoral se atienden algunos de estos problemas de desempeño mediante la técnica denominada como **codificación adaptativa de red**. Esta técnica ayuda a que los nodos de una iWMN puedan combinar varios paquetes de datos y de este modo construir un paquete codificado; al transmitir este paquete se transporta la información contenida en los paquetes originales requiriendo únicamente una transmisión inalámbrica, reduciendo de esta manera el uso del medio inalámbrico y, con ello, se incrementa la capacidad de la red. La técnica propuesta, además, busca que el proceso de codificación se adapte a las condiciones de tráfico en la red a través del ajuste dinámico del tiempo de espera de los paquetes en un nodo antes de poder ser combinados; es así como se puede disminuir el retardo de codificación.

Con esta propuesta se pretende mejorar sustancialmente el desempeño de las iWMNs, resolviendo algunos problemas que las afectan. La evaluación de la propuesta se realiza empleando simulaciones y evaluaciones numéricas. A través de un minucioso análisis de resultados encontramos que las iWMNs pueden mejorar su rendimiento al emplear la técnica de codificación adaptativa de red, ya que se reduce considerablemente el número de transmisiones inalámbricas en la red, y, por consiguiente: *i)* se disminuye la contienda por el medio, *ii)* se reducen las probabilidades de error en el medio y *iii)* se incrementa la capacidad de la red.

Adaptive Wireless Network Coding for IEEE 802.11s Infrastructure Wireless Mesh Networks

Abstract

IEEE 802.11s INFRASTRUCTURE WIRELESS MESH NETWORKS (commonly known as iWMNs) are integrated by static wireless nodes capable of working in coordination to route data packets. In this way, the nodes collaborate to exchange information with each other. In addition, iWMNs can be interconnected with other network technologies and, in this way, help to wirelessly extend the coverage of these networks; for example, iWMNs are used today to extend the coverage of cellular or wired networks. Thanks to this feature, and also to their low infrastructure cost, iWMNs networks are considered today as an excellent option to offer wireless Internet connectivity services in geographical areas where the use of other network technologies is unfeasible.

Despite the promising features of iWMNs, there are studies and results that cast doubt on their performance, since it has been documented that the performance of these networks can be affected by numerous factors; such as the use of TCP to transport information in wireless environments, the transmission errors in the wireless medium, as well as the access contention between network users. All these factors can degrade the performance of iWMNs and, consequently, affect the quality of the experience for the users.

In this doctoral thesis, some of these performance problems are addressed through the technique called **adaptive network coding**. With this technique, the nodes of an iWMN are allowed to combine various data packets and thus build an encoded packet; this packet contains the information from the original packets, requiring only one wireless transmission to transport the original information, reducing the use of the wireless medium and, thereby, increasing the capacity of the network. The proposed technique also seeks to adapt the coding process to the traffic conditions in the network through the dynamic adjustment of the waiting time of the packets in a node before they can be combined.

This proposal aims to substantially improve the performance of iWMNs, solving some problems that affect them. The evaluation of the proposal is carried out through simulations and numerical evaluations. After a detailed analysis of the results, we find that iWMNs can improve their performance by using the adaptive network coding technique, since the number of wireless transmissions in the network is considerably reduced, and, consequently, *i*) the medium access contention decreases, *ii*) the probability of errors in the medium is reduced, and *iii*) the capacity of the network increases.

Índice general

Índice de figuras	10
Índice de tablas	12
Lista de acrónimos	15
1 Introducción	19
1.1 Propuesta de investigación	21
1.2 Objetivos de la investigación	21
1.3 Metodología de la investigación	22
1.4 Resultados esperados	22
1.5 Estructura de la tesis	22
1.6 Contribuciones principales	23
2 Redes IEEE 802.11s iWMNs	25
2.1 Arquitectura de las iWMNs	25
2.2 Clasificación de los nodos de una iWMN	26
2.2.1 Mesh Station (Mesh STA)	26
2.2.2 Mesh Portal (MP)	27
2.2.3 Mesh Access Point (MAP)	27
2.2.4 User Station (STA)	27
2.3 Funciones definidas por la enmienda IEEE 802.11s	28
2.3.1 Mesh discovery	28
2.3.2 Mesh peering management	30
2.3.3 Mesh coordination function	30
2.3.4 Mesh path formation	31
2.3.5 Airtime Link Metric	32
2.3.6 Intra-mesh congestion control	33
2.4 Beneficios de las iWMNs	34
2.5 Problemas de desempeño de las iWMNs	34
3 Codificación de red	37
3.1 Codificación de red	37
3.2 Codificación lineal aleatoria (RLNC)	38
3.2.1 Proceso de codificación	39
3.2.2 Proceso de decodificación	41
3.3 Codificación y decodificación empleando suma aritmética módulo-2	42
3.3.1 Codificación de flujos <i>multicast</i> en una topología <i>butterfly</i>	43
3.3.2 Codificación de flujos <i>unicast</i> en una topología <i>butterfly</i>	44
3.4 Codificación de red en entornos inalámbricos	45
3.5 Codificación en redes inalámbricas <i>multihop</i>	48
3.6 Beneficios de la técnica de codificación de red	50
4 Trabajo relacionado	53
4.1 Retos de desempeño	53

4.1.1	Escalabilidad de la red	53
4.1.2	Tecnología de transmisión	54
4.1.3	Seguridad en la red	54
4.1.4	Interacción con protocolos de la capa de transporte	55
4.2	Codificación de red en entornos inalámbricos	57
4.2.1	COPE	58
4.2.2	BEND	59
4.2.3	iCORE	59
4.2.4	Reutilización espacial, adaptación dinámica de transmisión y codificación de red	60
4.2.5	Oportunidades para codificación de red: esperar o no esperar	60
4.2.6	Fair Coding	60
4.2.7	SNC - Calendarización y codificación de red	61
4.2.8	Análisis de COPE en redes inalámbricas <i>multihop</i>	62
4.2.9	Análisis de la interacción de la codificación de red con IEEE 802.11 DCF	63
4.2.10	MPOF - Múltiples paquetes un solo flujo	63
4.3	Codificación de flujos TCP	64
4.3.1	Piggy Code	65
4.3.2	Code MP	65
4.3.3	Combo Coding	66
4.3.4	TCP Fender	67
4.3.5	ENC - Codificación de red eficiente	68
4.3.6	MACINT-TCP	68
4.3.7	Mejora de TCP con codificación de red en redes inalámbricas	69
4.3.8	TCP y codificación de red	70
4.3.9	Codificación de red para mejorar la equidad para flujos TCP	71
4.3.10	Codificación de red para TCP en redes WMN multi-radio	72
4.4	Codificación utilizando temporizadores para la detección de oportunidades	72
4.4.1	Desempeño de TCP en WMNs codificadas	72
4.4.2	Impacto de la codificación de red en el desempeño de TCP en WMNs	73
4.4.3	Codificación de red con inserción de tiempos de espera	74
4.5	Conclusiones sobre la literatura revisada	75
5	Evaluación del desempeño de diversas variantes de TCP en iWMNs	77
5.1	Fundamentos de TCP	79
5.1.1	Administración de la conexión	79
5.1.2	Control de flujo	80
5.1.3	Estimación del RTT	81
5.1.4	Control de congestión	82
5.2	Retos de desempeño de TCP en iWMNs	84
5.2.1	Modificaciones de TCP para operar en redes inalámbricas	85
5.3	Variantes de TCP en iWMNs	90
5.3.1	TCP Westwood+	90
5.3.2	TCP Veno	92
5.3.3	TCP Vegas	94
5.3.4	TCP Hybla	95
5.3.5	TCP Freeze	97
5.4	Evaluación de desempeño	98
5.4.1	Análisis de desempeño de TCP en iWMN en topología de cadena	98
5.4.2	Análisis de desempeño de TCP en iWMN en topología <i>butterfly</i>	110
5.5	Conclusiones sobre las evaluaciones de desempeño	111
6	Propuesta para la codificación adaptativa en iWMNs y evaluaciones de desempeño	115

6.1	Oportunidades de codificación en iWMNs	116
6.2	Codificación en los puntos de acceso (MAPs)	117
6.3	Retardo de codificación	118
6.4	Modelo del sistema como una DTMC	119
6.4.1	Estado estacionario y transiciones de la DTMC	120
6.4.2	DTMC sin transmisión de paquetes nativos	123
6.5	Codificación de red en un MAP	126
6.5.1	Escenario de simulación	126
6.5.2	Tasa de arribos	130
6.5.3	Oportunidades de codificación	130
6.5.4	Número de transmisiones	131
6.5.5	Retardo de codificación	133
6.5.6	Ajuste del retardo de codificación	136
6.5.7	Retardo de codificación óptimo	138
6.5.8	Ganancia de codificación	139
6.6	Estimación adaptativa del tiempo de espera	140
6.6.1	Estimación por promedios móviles	140
6.6.2	Algoritmo de estimación adaptativa	142
6.6.2.1	Inicialización del vector de muestras	142
6.6.2.2	Calculo de las muestras iniciales del retardo óptimo de codificación	142
6.6.2.3	Estimación dinámica del retardo óptimo de codificación	143
6.6.3	Evaluación del proceso de estimación	144
6.6.3.1	Retardo óptimo de codificación	144
6.6.3.2	Error de estimación	145
7	Conclusiones y trabajo futuro	147
	Bibliografía	149

Índice de figuras

1.1	iWMN desplegada para proporcionar acceso inalámbrico a Internet en una zona comunitaria. . . .	20
2.1	Arquitectura fundamental de una iWMN. Los nodos que componen la red utilizan funciones de encaminamiento para interconectar inalámbricamente diferentes tecnologías de red, p. ej., LTE, 802.11 WLAN y Ethernet.	26
2.2	IEEE 802.11s Mesh Station (<i>Mesh STA</i>).	27
2.3	IEEE 802.11s Mesh Portal (MP)).	27
2.4	IEEE 802.11s Mesh Access Point (MAP).	28
2.5	Dispositivo móvil IEEE 802.11 (STA).	28
2.6	Mecanismo de diferenciación de tráfico implementado por EDCA. Cada categoría de acceso (AC) tiene diferentes prioridades de acceso al medio y estas prioridades son definidas a través de filas de acceso internas y parámetros de configuración para el acceso al medio.	31
2.7	HWMP Reactive Mode	32
2.8	HWMP Proactive Building Tree Mode	33
3.1	El nodo A recibe X_1 y X_2 , con esta información calcula el mensaje codificado $f(X_1, X_2)$ y lo transmite al nodo B . El nodo B realiza la función de decodificación sobre el mensaje codificado para obtener la información original X_1 y X_2 . Con este esquema, se reducen el número de transmisiones sobre el enlace que va de A a B	38
3.2	Paquete extraviado en un sistema de comunicaciones y su respectiva solicitud de retransmisión. . .	39
3.3	Segmentos extraviados en un sistema de comunicaciones empleando RLNC.	39
3.4	Codificación lineal aleatoria de la red (<i>Random Linear Network Coding</i>) empleando dos paquetes nativos P_1 y P_2 y los vectores de codificación \vec{C}_1 y \vec{C}_2	41
3.5	Grafo $G(V, E)$ para una red con <i>topología mariposa</i> . En la red se realiza una transmisión <i>multicast</i> entre el nodo S y los nodos $T1$ y $T2$	44
3.6	Transmisión multicast con un cuello de botella que reduce la capacidad de la red.	44
3.7	Codificación en una topología <i>mariposa</i> para incrementar la capacidad de la red.	45
3.8	Transmisión de flujos <i>unicast</i> en una red con topología <i>mariposa</i>	45
3.9	Los nodos de la red transmiten paquetes al nodo N y realizan <i>escucha oportunista</i> de los mensajes que se transmiten en el medio inalámbrico, almacenando una copia de los paquetes escuchados en el medio.	47
3.10	El nodo N codifica y transmite el paquete codificado P_x a los M nodos vecinos.	47
3.11	Establecimiento de una ruta libre de bucles en redes inalámbricas <i>multihop</i>	48
3.12	<i>Store-and-forward</i> vs codificación de red.	49
4.1	Trayectoria de comunicaciones entre los nodos 1 y 2 con propiedades dinámicas en los enlaces. . .	55
4.2	Problemas de terminal oculta (izq.) y terminal expuesta (der.).	56
4.3	El nodo B , conociendo los paquetes que han recibidos por los nodos vecinos, tiene varias opciones de codificación y debe elegir aquella que sea decodificable y que maximice la capacidad de la red. .	59
4.4	El nodo B encamina dos flujos de datos y puede aplicar una política de codificar paquetes o esperar por una oportunidad de codificación.	60
4.5	El nodo B encamina dos flujos de datos, de A a E y de C a D . De lado derecho se muestra la evolución de la fila de transmisión en el nodo B durante los periodos de tiempo de $T1$ a $T5$; por un lado, el comportamiento desbalanceado cuando se emplea el protocolo IEEE 802.11 MAC y, por otro lado, el comportamiento balanceado empleando FairCoding, incrementando así el número de oportunidades de codificación al priorizar el flujo de C a D	61

4.6	Cada nodo tiene pares de filas virtuales de codificación donde se almacenan los paquetes que van destinados a un nodo vecino.	62
4.7	Modelo de recepción y transmisión de paquetes nativos y codificados en un nodo de la red.	63
4.8	a) Esquema de codificación OPOF con paquetes nativos de diferentes tamaños y utilizando <i>zero-padding</i> para igualarlos. b) Combinación de paquetes para igualar el tamaño de paquetes nativos y evitar usar <i>zero-padding</i>	64
4.9	Arquitectura de un nodo inalámbrico con las funcionalidades de codificación de red definidas por PiggyCode.	66
4.10	Codificación de paquetes TCP DATA y TCP ACK en los nodos intermedios de la trayectoria <i>multihop</i>	66
4.11	TCPFender implementa las funciones de transmisión oportunista de paquetes y codificación de red.	67
4.12	Funcionamiento de las filas virtuales SVQ y LVQ para clasificar paquetes y codificarlos con ENC.	68
4.13	Interacción intercapa del protocolo MACINT-TCP	69
4.14	Módulos del modelo de nodo móvil de NS2 y su interacción con el módulo COPE.	70
4.15	Implementación de los módulos <i>NC</i> entre la capa IP y el protocolo TCP Vegas junto con el sistema de acuses basados en <i>degree of freedom</i> para el enmascarado de pérdidas.	71
4.16	Algoritmos de codificación-decodificación para el transmisor y receptor al incorporar un <i>buffer</i> de codificación.	73
4.17	Implementación de los módulos <i>NC</i> entre la capa IP y la capa MAC y las interacciones entre ellos.	75
5.1	Intercambio de <i>I</i> bits información entre dos procesos empleando los servicios de comunicación de extremo a extremo provistos por TCP	79
5.2	Mecanismos para el establecer y finalizar una sesión TCP	80
5.3	Evolución de la estimación del Timeout para la detección de pérdidas por RTO.	82
5.4	Evolución del valor de la ventana de congestión (<i>cwnd</i>) durante las fases <i>slow-start</i> , <i>congestion avoidance</i> , <i>fast-retransmit</i> y <i>fast-recovery</i>	83
5.5	TCP Freeze evita que se ejecute la fase de <i>slow-start</i> después de un evento de reconexión.	97
5.6	Escenario general de evaluación.	99
5.7	Escenario de simulación para una topología en cadena.	100
5.8	Capacidad de la red como función de la probabilidad de error de paquete $0 \leq P_e \leq 0.1$	102
5.9	Capacidad de la red como función de la probabilidad de error $0.01 \leq P_e \leq 0.05$	102
5.10	Capacidad de la red por segundo durante el tiempo de simulación y con $P_e = 0.0$	103
5.11	Capacidad de la red por segundo durante el tiempo de simulación y con $P_e = 0.015$	103
5.12	Valores de la ventana de congestión (<i>cwnd</i>) cuando $P_e = 0.0$	104
5.13	Valores de la ventana de congestión (<i>cwnd</i>) cuando $P_e = 0.015$	105
5.14	Valores de la estimación RTT cuando $P_e = 0.0$	106
5.15	Valores de la estimación RTT cuando $P_e = 0.015$	106
5.16	<i>Throughput</i> promedio como función de <i>k</i> cuando $P_e = 0.0$	107
5.17	<i>Throughput</i> promedio como función de <i>k</i> cuando $P_e = 0.015$	107
5.18	<i>Aggregated throughput</i> para las diferentes variantes de TCP cuando $N_a(t) = 2, 3, 4, 6, 8, 10, 20$	109
5.19	Índice JFI para las diferentes variantes de TCP en un escenario con $N_a(t) = 2, 3, 4, 6, 8, 10, 20$	110
5.20	Escenario con dos MAPs ofreciendo conectividad a 4 STAs para establecer comunicaciones desde/hacia Internet.	111
5.21	<i>Aggregated</i> e <i>individual throughput</i> como función del número de Mesh STAs entre el MAP1 y el MP.	112
5.22	Índice JFI como función del número de Mesh Points entre el MAP 1 y el MP	112
6.1	Estructuras internas de una iWMNs y la aplicación de la técnica de codificación de red.	116
6.2	Arquitectura de un Mesh Access Point para implementar la técnica WNC.	117
6.3	Cuando el paquete P_i se almacena en la fila S_1 y la fila S_2 está vacía, P_i puede: i) esperar a P_j un tiempo ilimitado, ii) esperar a P_j un tiempo limitado, o iii) no esperara a P_j y es transmitido en su forma nativa.	118
6.4	Diagrama de estados que representa la dinámica del proceso aleatorio bidimensional $N(\Delta_t)$	120
6.5	Probabilidad de estado-estable de la DTCM como función de M_k y $P_{MAP} = P_{MP} = P_{STA} = 1/3$	121

6.6	Número normalizado de transiciones de la DTMC como función de M_k y $P_{MAP} = P_{MP} = P_{STA} = 1/3$	122
6.7	Histogramas de ocurrencias de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$ y $M_k = \{2, 3, 4, 9\}$	124
6.8	Número de transiciones de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$, $P_{STA} = P_{MP} = (1 - P_{MAP})/2$ y $M_k = \{2, 3, 4, 9\}$	125
6.9	Diagrama de estados que representa la dinámica del proceso aleatorio bidimensional $N(\Delta_t)$	126
6.10	Histogramas de ocurrencias de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$ y $M_k = \{2, 3, 4, 9\}$	127
6.11	Número de transiciones de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$, $P_{STA} = P_{MP} = (1 - P_{MAP})/2$ y $M_k = \{2, 3, 4, 9\}$	128
6.12	Escenario de simulación	129
6.13	Tasa de arribos a las filas de codificación Q1 y Q2	130
6.14	Estadísticas de $\chi_\tau(Q_k)$ con una ventana de observación $\tau = 1$ sec	131
6.15	Formación de oportunidades de codificación con los paquetes de las filas de codificación Q1 y Q2	132
6.16	Oportunidades de codificación por segundo como función del número de STAs	132
6.17	Comparación entre el número de transmisiones utilizadas con NC y Store-and-Forward	133
6.18	Oportunidades de codificaciones totales como función del número de STAs en la red	133
6.19	Instantes de arribos a las filas de codificación Q1 y Q2 con un flujo TCP en el escenario de simulación	134
6.20	Instantes de arribos a las filas de codificación Q1 y Q2 con 32 flujos TCP en el escenario de simulación	135
6.21	Muestras iniciales del retardo de codificación $\psi_{i,i}$ como función del número de STAs	136
6.22	Muestras estadísticas del retardo de codificación $\psi_{i,i}$ como función del número de STAs	136
6.23	Incremento del retardo de codificación $\psi_{i,i}$	137
6.24	Retardo de codificación óptimo $\psi_{optimal}$	138
6.25	Retardo de codificación óptimo $\psi_{optimal}$ en un escenario de simulación con una STA	138
6.26	Retardo de codificación óptimo $\psi_{optimal}$ como función del número de STAs	139
6.27	Ganancia de codificación por segundo empleando el retardo de codificación óptimo $\psi_{optimal}$	140
6.28	Método de estimación por promedio móvil	141
6.29	Construcción de los arreglos iniciales S_1 y S_2 con muestras de los instantes de arribo de los paquetes nativos al MAP	143
6.30	Cálculo del vector ψ donde se almacenan las muestras del retardo de codificación óptimo	143
6.31	Estimación del retardo de codificación empleando mecanismos basados en promedios móviles (WMA)	144
6.32	Comparación entre retardo óptimo de codificación y el valor dinámico de $MAX_{codingtime}$ obtenido mediante diferentes métodos de estimación con promedios móviles	145
6.33	Error de estimación utilizando diferentes métodos de media móvil e incrementando el número de STAs	146

Índice de tablas

2.1	Funciones principales de los dispositivos definidos por la enmienda IEEE 802.11s	29
2.2	Ventajas de las redes malladas IEEE 802.11s en modo infraestructura (iWMNs)	35
2.3	Desventajas de las redes malladas IEEE 802.11s en modo infraestructura (iWMNs)	36
3.1	Tabla de verdad del operador lógico XOR	42
3.2	Codificación y decodificación de variables lógicas empleando el operador XOR	43
4.1	Características de los mecanismos de transmisión del estándar IEEE 802.11	54
5.1	Parámetros de simulación para la evaluación de variantes de TCP en iWMNs	101

6.1	Parámetros de simulación para la evaluación de WNC en iWMNs.	129
-----	--	-----

Lista de acrónimos

ACK	TCP Acknowledgment.
AIAD	Additive Increase - Adaptive Decrease.
AIFS	Arbitration Inter-Frame Space.
AIMD	Additive Increase - Multiplicative Decrease.
ALM	Airtime Link Metric.
ANC	Adaptive Network Coding.
AODV	Adhoc On-demand Distance Vector Protocol.
AP	Access Point.
BDP	Bandwidth Delay Product.
BER	Bit Error Rate.
BSS	Basic Service Set.
BW	Bandwidth.
CCN	Congestion Control Notification.
CW	MAC Contention Window.
CWA-CD	Congestion Window Adaptation Through Contention Detection.
CWL	Congestion Window Limit.
DCF	Distributed Coordination Function.
DSR	Dynamic Source Routing.
DT	Delay Threshold.
DTMC	Discrete-Time Markov Chain.
DUPACK	Duplicate Acknowledgment.
EDCA	Enhanced Distributed Channel Access.
ELFN	Explicit Link Failure Notification.
ELN	Explicit Loss Notification.
EWMA	Exponential Weight Moving Average.
ExT	TCP Expected Throughput.
GWMA	Gaussian Weight Moving Average.
HWMP	Hybrid Wireless Mesh Protocol.
IEEE	Institute of Electrical and Electronics Engineers.
IEEE 802.11	IEEE Media access control and physical layer specifications for implementing wireless local area network computer communication.
IP	Internet Protocol.
IRQ	Inter Range Quartile.
ISN	Initial Sequence Number.
iWMN	Infrastructure Wireless Mesh Network.

JFI	Jain's Fair Index.
LL-SACK	Link-Layer Selective Acknowledgement.
LRED	Link Random Early Detection.
LTE	Long Term Evolution.
MAC	Medium Access Control.
MAP	Mesh Access Point.
MBSS	Mesh Basic Service Set.
MCCA	Mesh Controlled Channel Access.
MCCAOP	MCCA Opportunity.
MCCASR	MCCA Setup Request frame.
MCF	Mesh Coordination Function.
MDP	Markov Decision Process.
MP	Mesh Portal.
MPM	Mesh Peering Management.
MSDU	MAC Service Data Unit.
MSS	Maximum Segment Size.
MWMA	Modified Weight Moving Average.
NICTA	National Information and Communications Technology Research Center of Australia.
OWD	One-Way Delay.
PER	Packet Error Rate.
PHY	Physical Layer.
PREP	HWMP Path Reply message.
PREQ	HWMP Path Request message.
QoS	Quality of Service.
RANN	HWMP Root Announcement Message.
RED	Random Early Detection.
RLNC	Random Linear Network Coding.
RTO	Retransmission Timeout.
RTT	Round Trip Time.
RWS	Receiver Window Size.
SACK	Selective Acknowledgement.
SIR	Signal-to-Interference Ratio.
STA	IEEE 802.11 Station.
SWMA	Simple Weight Moving Average.
TCP	Transmission Control Protocol.
TCP-NCL	TCP for Non Congestive Loss.
TCPF	TCP Freeze.
TCPH	TCP Hybla.
TCPNR	TCP New Reno.

TCPV	TCP Vegas.
TCPVN	TCP Veno.
TCPW	TCP Westwood.
TR-ACK	Triplicate Reconnection ACK.
TWMA	Triangular Weight Moving Average.
TXOP	MCCA Transmission Opportunity.
VCRH	Variance of Contention RTT per hop.
WLAN	Wireless Local Area Network.
WMN	Wireless Mesh Network.
WNC	Wireless Network Coding.
WWW	World Wide Web.
ZWA	Zero Window Advertisement.
ZWP	Zero Window Probe.

CAPÍTULO 1

Introducción

“La educación es el paso del hombre de la ignorancia engreída a la incertidumbre reflexiva.”

Kenneth Johnson

INTERNET se ha consolidado hoy en día como la red de comunicaciones más utilizada por la humanidad. **La conectividad a esta red se ha convertido en una necesidad imperativa para muchas de las aplicaciones de la vida cotidiana**, por ejemplo, Internet se emplea para consumir contenidos multimedia; también para enviar mensajes instantáneos y correos electrónicos; para buscar y compartir información en la WWW (*World Wide Web*); para realizar llamadas de voz y video; e incluso para establecer relaciones académicas y laborales. Este amplio abanico de aplicaciones demanda disponibilidad de conectividad inalámbrica a Internet a toda hora y en todo lugar. Pero, ¿cómo lograr satisfacer eficientemente esta demanda de conectividad creciente?

Existen diferentes tecnologías de red capaces de atender este reto. Primero están las redes celulares o satelitales, que por su costo solo pueden ser empleadas por grandes proveedores de servicios. Las redes Wi-Fi, por otro lado, cuentan con una tecnología inalámbrica económica; sin embargo, esta tecnología tiene un alcance de transmisión limitado, lo que las descarta para ser una solución eficiente. Por fortuna, existe una novedosa alternativa económica y accesible para proveer acceso inalámbrico a Internet: las **redes mallas IEEE 802.11s en modo infraestructura** (*IEEE 802.11s Infrastructure Wireless Mesh Networks*, iWMNs por sus siglas en inglés) [1].

Las iWMNs funcionan como una **red de distribución inalámbrica** (conocida comúnmente como *backbone*). Esta infraestructura está constituida por nodos inalámbricos estáticos capaces de trabajar conjuntamente para encaminar paquetes de datos. Además, las iWMNs pueden conectarse eficientemente con otras tecnologías de red y esta característica permite que puedan ser utilizadas como un sistema de interconexión para diferentes tecnologías de red. Más aún, las iWMNs tienen un bajo costo de infraestructura, son confiables, y pueden ser desplegadas y escaladas fácilmente. Todas estas características las posicionan como una opción viable para extender la cobertura de las redes cableadas y así dar respuesta a la demanda de conectividad inalámbrica a Internet.

La Figura 1.1 ejemplifica la manera en la que una iWMN puede proporcionar acceso inalámbrico a Internet en una zona comunitaria. Para que un usuario móvil pueda acceder a Internet, primeramente, el tráfico del usuario es enviado a un nodo de la iWMN; posteriormente, el tráfico es encaminado inalámbricamente con ayuda de los nodos del *backbone* hasta llegar al acceso cableado de banda ancha a Internet; finalmente, el tráfico es transmitido por la red cableada hacia Internet. Para los usuarios, este proceso de interconexión inalámbrica es totalmente transparente, es decir, los usuarios no perciben que la iWMN actúa como intermediaria entre dos tecnologías de red, es decir, la red IEEE 802.11 Wi-Fi y la red IEEE 802.3 Ethernet.



Figura 1.1: iWMN desplegada para proporcionar acceso inalámbrico a Internet en una zona comunitaria.

A pesar de las características destacadas de las iWMNs, se ha demostrado que **sus prestaciones pueden degradarse por numerosos factores**; tales como: la alta densidad de usuarios en la red [2], la distribución sesgada del tráfico [3], el incremento de la contienda por el acceso al medio [4], así como por el pobre desempeño que tiene el protocolo TCP en redes inalámbricas [5, 6]. Estos hallazgos ponen en duda los beneficios que pudieran obtenerse al utilizar esta tecnología. Por tal motivo, actualmente se exploran **varias alternativas para atender estos problemas y mejorar así el rendimiento de estas redes**.

Una prometedora solución para mejorar el desempeño de las iWMNs es la técnica denominada como **codificación de red inalámbrica** [7] (*Wireless Network coding-WNC*). Esta es una propuesta diseñada para mejorar el funcionamiento de redes inalámbricas; pues con ella se aprovechan las propiedades de difusión del medio inalámbrico de la siguiente forma: a diferencia del esquema de conmutación tradicional, *store-and-forward*, (en donde cada *paquete de datos* es tratado individualmente durante el proceso de transmisión), WNC permite que un nodo pueda combinar varios paquetes de datos para construir un paquete codificado. Este paquete transporta la información de los paquetes originales, esto significa que basta con solo transmitir el paquete codificado en lugar de transmitir todos los paquetes que lo conformaron. De esta manera se reducen: el uso del medio inalámbrico, la contienda por el medio y los errores de transmisión. Es así como se espera que este nuevo paradigma maximice los límites de desempeño de las redes inalámbricas actuales [8, 9], incluyendo a las iWMNs.

Ahora bien, la técnica WNC no siempre ofrece beneficios, porque al realizar el proceso de combinación de paquetes, se agrega un retardo de procesamiento y se debe satisfacer condiciones precisas para que la técnica funcione correctamente. Si no se cumplen estas condiciones, WNC puede significar una medida perjudicial para el desempeño de la red [10]. Una variante utilizada para mejorar el funcionamiento de WNC se conoce como **codificación adaptativa de red** (*Adaptive Network Coding-ANC*). Al ser empleada logra que cada elemento de la red aprenda de su entorno; detecte oportunidades de codificación; y determina la viabilidad de éstas para maximizar la utilización del medio. Sin embargo, y como bien es señalado por Katti *et al.* [11], existen importantes complicaciones para la aplicación efectiva de la codificación en redes en entornos altamente dinámicos como los son las iWMNs.

A pesar de la relevancia de las iWMNs y de los beneficios de la codificación adaptativa, **al momento de realizar el estado del arte, no se encontraron estudios que compaginen estos dos elementos**. En gran medida se ha debido a que las aplicaciones de la codificación de red se han centrado en atender problemáticas propias de redes inalámbricas *multihop*, y estas aplicaciones no se pueden aplicar a la estructura propia de las iWMNs. Se han identificado cuatro problemas que afectan seriamente el desempeño de las iWMNs que podrían atenderse con la técnica ANC, éstos son:

1. La alta probabilidad de errores de transmisión en el medio y las pérdidas de paquetes asociadas a este problema.
2. El bajo desempeño de los protocolos diseñados para redes cableadas; resaltando el protocolo de control de la transmisión (TCP), protocolo *de facto* para las aplicaciones que usan Internet.
3. La contienda por acceso al medio entre una gran cantidad de usuarios, incrementa el retardo y las colisiones.
4. La distribución sesgada del tráfico produce congestión en las rutas de comunicación y pérdidas de paquetes, particularmente en los nodos que son responsables de procesar una gran cantidad de información.

En esta investigación doctoral se diseña una técnica de codificación adaptativa que, al aplicarse específicamente en iWMNs, reduce los problemas de desempeño, mejorando significativamente el rendimiento de la red. Tomando en cuenta que una iWMN tiene una infraestructura fija y sin acotaciones de procesamiento o de energía en los nodos que la componen, se puede afirmar que estas redes reúnen las condiciones ideales para utilizar los métodos de codificación de red. Además, las iWMNs tienen una estructura jerárquica que multiplica las oportunidades de codificación de paquetes. Más aún, la heterogeneidad de los dispositivos que componen la infraestructura de las iWMNs y sus funciones, requiere de un análisis de las múltiples oportunidades de codificación que ofrecen. Con base en la revisión del estado del arte, se propone **una técnica adaptativa de codificación de red que explote la estructura de las iWMNs, orientada a reducir el retardo de codificación y a disminuir la contienda por el medio entre los usuarios y los nodos de la red**. Esta técnica de codificación, además, permitirá reducir los niveles de congestión en la red y propiciará un incremento importante de la capacidad de la red.

1.1 Propuesta de investigación

El éxito de las iWMNs depende en gran medida de la capacidad de éstas para ofrecer un servicio de calidad a una gran cantidad de usuarios. Sin embargo, ha de señalarse que el desempeño de estas redes puede verse afectado por diversos factores. Se ha demostrado teórica y experimentalmente que la codificación de la red es una técnica útil para mejorar el funcionamiento de redes inalámbricas, ya que ayuda a *i*) mejorar la utilización del medio, *ii*) a disminuir la probabilidad de errores de transmisión, y *iii*) a reducir la contienda por el medio y la carga de tráfico en la red. Tras una amplia revisión de la literatura se puede afirmar que, a pesar de sus beneficios, la codificación de red aún no ha sido explotada en la arquitectura jerárquica de las iWMNs, representando esto un área de oportunidad muy importante. Considerando que una iWMN tiene una infraestructura fija y sin acotaciones de procesamiento o de energía, se puede concluir que esta arquitectura ofrece ventajas importantes y novedosas estructuras donde es factible aplicar la técnica de codificación de red. Por lo tanto, la correcta aplicación de la técnica de codificación de red en iWMNs promete resolver múltiples problemas de desempeño e incrementar sustancialmente la capacidad de estas redes.

1.2 Objetivos de la investigación

Objetivo general

Diseñar un mecanismo adaptativo de codificación de red para iWMNs y evaluar su desempeño.

Objetivos particulares

1. Idear un método de codificación de red que explote las características de las iWMNs para incrementar las oportunidades de codificación.
2. Generar una función de adaptabilidad que permita incrementar los beneficios de la codificación de red.
3. Identificar los efectos del mecanismo de auto regulación de TCP sobre la codificación en iWMNs.
4. Evaluar analíticamente los efectos de la codificación de red en iWMNs.

1.3 Metodología de la investigación

La metodología para el desarrollo de este proyecto es la siguiente.

1. Estudiar el desempeño de TCP en iWMNs y proponer un marco de evaluación.
2. Desarrollar un esquema de codificación para iWMNs y evaluar su desempeño vía simulación.
3. Identificar las características de los flujos de datos en la red, y los nodos donde convergen, para poder incrementar las oportunidades de codificación.
4. Diseñar un algoritmo que aproveche las propiedades de los nodos idóneos para implementar la codificación de red.
5. Concebir una función adaptativa para el algoritmo de codificación y evaluar sus beneficios.

1.4 Resultados esperados

1. Una evaluación de la interacción entre TCP e iWMNs.
2. Un esquema de codificación adaptativo para iWMNs.
3. Un marco de referencia para valorar diversos esquemas de codificación en iWMNs.
4. Un modelo analítico que caracterice la eficiencia del proceso de codificación.
5. Reportes de investigación y la divulgación de resultados en al menos una revista internacional indexada.

1.5 Estructura de la tesis

1. En el Capítulo 2 se expone la arquitectura de las iWMNs con base en la enmienda IEEE 802.11s.
2. En el Capítulo 3 se detallan los mecanismos de codificación y decodificación de red.
3. En el Capítulo 4 se muestran diferentes trabajos relacionados, destacando las propuestas más relevantes que se hayan en la literatura.
4. En el Capítulo 5 se expone la evaluación de desempeño de TCP en iWMNs. Los resultados obtenidos conforman el marco de referencia para la evaluación de desempeño pretendido.
5. El Capítulo 6 describe la propuesta de codificación adaptativa para iWMNs y presenta los resultados de la evaluación.
6. El Capítulo 7 presenta las conclusiones de este proyecto, subrayando las líneas de investigación que deja abiertas.

1.6 Contribuciones principales

Los resultados de esta investigación se publicaron en los siguientes foros internacionales.

Artículo en revista internacional indexada

- *Adaptive Wireless Network Coding for Infrastructure Wireless Mesh Networks*, **Ernesto Carrillo** and Víctor Ramos. KSII Transactions on Internet and Information Systems, Vol. 13, Num. 7, Julio, 2019, DOI: [10.3837/tiis.2019.07.008](https://doi.org/10.3837/tiis.2019.07.008).

Artículo en conferencia internacional

- *On the impact of network coding delay for IEEE 802.11s infrastructure wireless mesh networks*, **Ernesto Carrillo** and Víctor Ramos. In the Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 305-312, Victoria, Canadá, Mayo, 2014, DOI: [10.1109/AINA.2014.39](https://doi.org/10.1109/AINA.2014.39).

Redes IEEE 802.11s iWMNs

“La máquina no aísla al hombre de los grandes problemas de la naturaleza, sino que lo sumerge más profundamente en ellos.”

Antoine de Saint-Exupery

EL estándar IEEE 802.11 [12] define mecanismos de la capa control de acceso al medio (*Medium Access Control - MAC*), así como las especificaciones de la capa física (PHY), con el objetivo de proporcionar servicios de movilidad y comunicaciones inalámbricas para los protocolos que residen en capas superiores del modelo TCP/IP. Desde su aparición en 1997, el estándar ha evolucionado e incrementado sus capacidades, a grado tal que hoy en día los escenarios de aplicación y las prestaciones de esta tecnología son muy variados [13]. Una contribución significativa ocurrió en 2012, fecha en la que apareció la enmienda IEEE 802.11s, también conocida como **mesh networking amendment** [14].

La enmienda IEEE 802.11s incorpora nuevas funciones que permiten utilizar el estándar en nuevos escenarios de aplicación. Dichos escenarios van desde la creación de redes de comunicación para ciudades inteligentes [15], redes para el Internet de las cosas [16] o redes de comunicación para extender la cobertura de otras tecnologías de red [17]. Una aplicación notable de enmienda IEEE 802.11s son las **redes malladas en modo infraestructura** (*Infrastructure Wireless Mesh Networks-iWMNs*). Esta novedosa arquitectura de red puede ser aprovechada para brindar acceso inalámbrico a Internet en regiones donde el uso de otras tecnologías resulta impráctico. A continuación se detallan: la arquitectura de una iWMN, los dispositivos de red que las componen, así como los mecanismos necesarios para la operación de estas redes.

2.1 Arquitectura de las iWMNs

Una iWMN está integrada por nodos inalámbricos heterogéneos. Estos dispositivos forman colectivamente un sistema de distribución inalámbrico que comúnmente es denominado como el **mesh backbone**. Estos nodos son capaces de realizar **funciones de enrutamiento** de paquetes de datos. Es así como una iWMN puede ser utilizada para interconectar diferentes tecnologías de red [18]. Para ilustrar esta característica, la Figura 2.1 muestra la arquitectura y operación de una iWMN. En este ejemplo, la iWMN interconecta diferentes tecnologías de red, es decir, tres redes inalámbricas operando con tecnología IEEE 802.11 WLAN; una red celular operando con tecnología LTE; y una red cableada IEEE 802.3 Ethernet. Gracias a la capacidad de la iWMN para encaminar paquetes de datos, la red puede interconectar inalámbricamente a los usuarios de las redes, aun cuando estos utilizan diferentes tecnologías. Al contar la iWMN con una **topología flexible y escalable**, los nodos que la componen solo requieren de una fuente de energía y añadirse a la iWMN, reduciendo sustancialmente los costos de infraestructura, así como los tiempos de despliegue y configuración de la red. Gracias a estas propiedades, una iWMN puede emplearse en numerosos y diversos escenarios de aplicación.

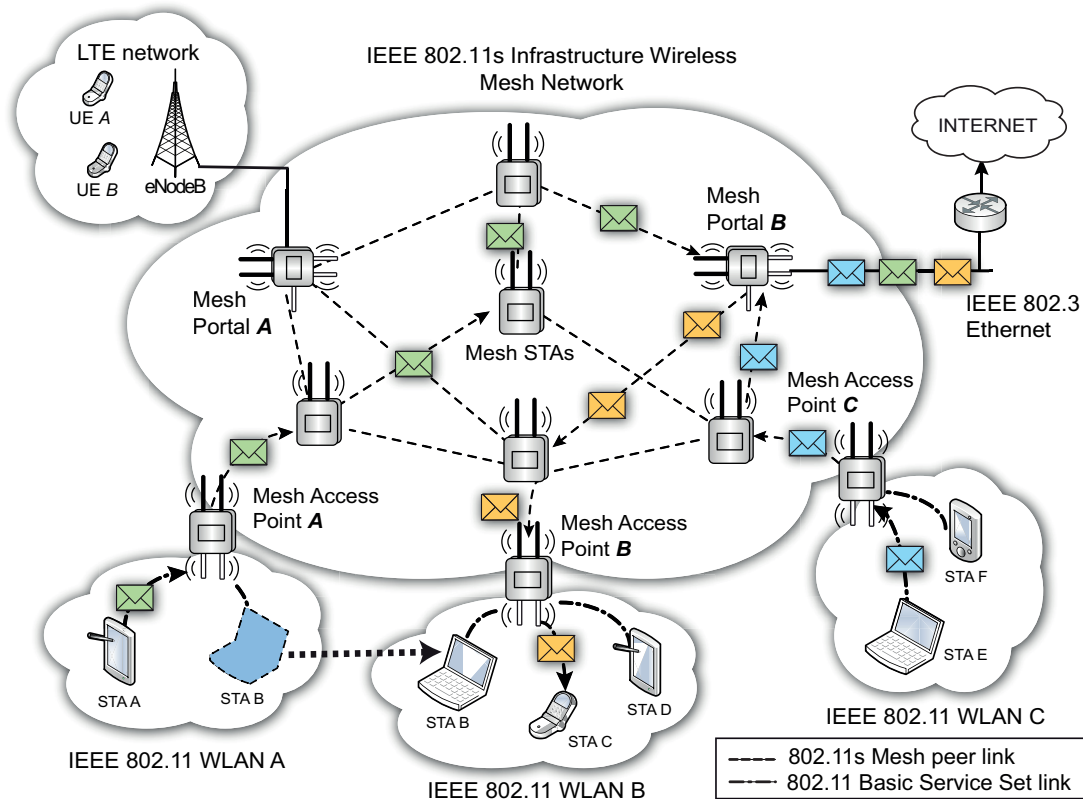


Figura 2.1: Arquitectura fundamental de una iWMN. Los nodos que componen la red utilizan funciones de encaminamiento para interconectar inalámbricamente diferentes tecnologías de red, p. ej., LTE, 802.11 WLAN y Ethernet.

2.2 Clasificación de los nodos de una iWMN

La enmienda IEEE 802.11s define **tres diferentes categorías** para clasificar a los nodos que componen una iWMN. A continuación se define cada categoría y se describen las características más significativas de cada una de ellas.

2.2.1 Mesh Station (Mesh STA)

Una *Mesh Station* (Mesh STA) es el **nodo fundamental** para la construcción de una iWMN. Este es un dispositivo inalámbrico que cumple con las funciones definidas por la enmienda IEEE 802.11s para participar en la formación y operación de un *Mesh Basic Service Set* (MBSS), (término técnico que define a una WMN). Una Mesh STA puede realizar diferentes acciones dentro de una iWMN: puede *i*) ser una **fuentes de tráfico**, *ii*) un **nodo receptor de tráfico** o *iii*) puede actuar como un dispositivo que **encamina tráfico** en beneficio de otras Mesh STAs. Debido a esta última funcionalidad, las Mesh STAs, incluso aquellas que están fuera del alcance de transmisión directa, pueden comunicarse entre sí. Una restricción definida por la enmienda IEEE 802.11s es que las Mesh STAs solo pueden comunicarse con otras Mesh STAs; así es que, se requerirá de un componente lógico extra que coadyuve a que una Mesh STA pueda comunicarse con dispositivos que no cumplen con la enmienda IEEE 802.11s. La Figura 2.2 muestra la estructura de una Mesh STA; la capa física (PHY Layer) cumple con la norma IEEE 802.11, mientras que la capa de enlace (Link Layer) cumple con la enmienda IEEE 802.11s. En la Figura 2.1 se puede apreciar que la iWMN está integrada mayoritariamente por Mesh STAs, las cuales se encargan, entre otras cosas, de encaminar paquetes dentro de la red.

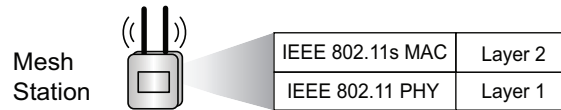


Figura 2.2: IEEE 802.11s Mesh Station (*Mesh STA*).

2.2.2 Mesh Portal (MP)

Un *Mesh Portal* (MP) es un dispositivo que debe cumplir con **dos funciones**: *i*) debe actuar como una Mesh STA y **encaminar paquetes** en la red, tal y como lo hace todas las Mesh STA, y *ii*) debe poder **intercambiar tráfico con otras tecnologías de red**. Por lo tanto, este dispositivo requiere de dos componentes esenciales: un componente que incorpore las funciones de una Mesh STA, y un componente que permita la comunicación con otras tecnologías de red. Dando origen a un sistema de distribución interno en un MP que interconecta dos tecnologías de red. Por consiguiente, el MP soporta mecanismos MAC/PHY para tecnologías que no son 802.11 y mecanismos MAC/PHY para interactuar con las Mesh STAs. Gracias a las prestaciones de interconexión que tienen los MPs, las Mesh STAs pueden comunicarse con dispositivos que son ajenos a la iWMN. La Figura 2.3 ejemplifica la estructura de un Mesh Portal que puede interactuar con Mesh STAs y con una red IEEE 802.3.

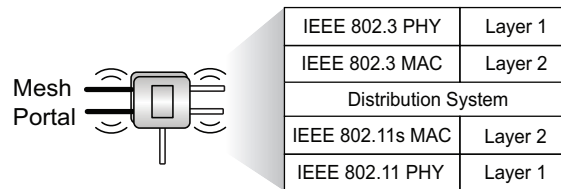


Figura 2.3: IEEE 802.11s Mesh Portal (MP)).

La iWMN mostrada en la Figura 2.1 tiene dos MPs: el MP A conecta la iWMN con una red celular con tecnología LTE, mientras que el MP B conecta la iWMN con una red IEEE 802.3 Ethernet con acceso a Internet. De este modo, cualquier Mesh STA puede enviar/recibir mensajes hacia/desde los dispositivos que residen en las redes LTE o Ethernet.

2.2.3 Mesh Access Point (MAP)

Un *Mesh Access Point* (MAP) es un dispositivo que opera como Mesh STA y además ofrece **conectividad inalámbrica a usuarios móviles que emplean tecnología IEEE 802.11** (denominados como *stations* (STAs)). Por ello, un MAP requiere de dos componentes: *i*) el que permite la comunicación con otras Mesh STAs y *ii*) el componente que permite la operación como *Access Point* (AP) para las STAs. Es así que cada MAP configura un *Infrastructure Basic Service Set* (BSS) para la conexión de las STAs [19]. Por consiguiente, los usuarios móviles que quieran hacer uso de los servicios de la iWMN deben asociarse a un MAP para que éste envíe y reciba paquetes a la red. La iWMN que se muestra en la Figura 2.1 tiene tres MAPs (MAP A, B y C). Estos son responsables de **establecer conexiones inalámbricas con las STAs, recibir el tráfico proveniente de las STAs e incorporarlo al MBSS, y también recibir tráfico proveniente del MBSS y transmitirlo a las STAs**. La Figura 2.4 muestra la estructura de un MAP que puede interconectar, nótese que un MAP interconecta dos tecnologías de red, IEEE 802.11s e IEEE 802.11.

2.2.4 User Station (STA)

Los usuarios de la red, definidos como *Stations* (STAs), residen en la capa más baja de la estructura jerárquica de las iWMNs. Estos dispositivos cumplen con la enmienda IEEE 802.11 y pueden **utilizar los servicios provistos por la iWMN para intercambiar información entre STAs, o bien con otras tecnologías de red** (La Figura 2.5 muestra la estructura de una STA.). En el ejemplo que se muestra en la Figura 2.1 se puede ver que la STA A y STA E envían paquetes de datos hacia Internet. Para completar este proceso de comunicación,

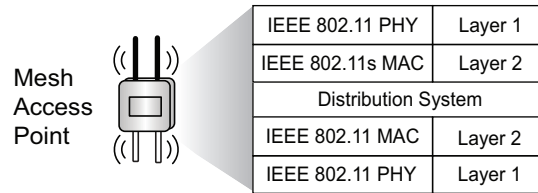


Figura 2.4: IEEE 802.11s Mesh Access Point (MAP).

primeramente, las STAs deben asociarse con un MAP y enviar los paquetes de datos a estos dispositivos. Es así como los MAPs *A* y *C* reciben los paquetes transmitidos por las STAs y los reexpiden a otras Mesh STAs. Empleando funciones de encaminamiento, las Mesh STAs reexpiden los paquetes desde los MAPs hasta el MP *B*. Finalmente, el MP *B* transmite los paquetes de datos a la red cableada con acceso a Internet. Otro ejemplo que se muestra en la Figura 2.1, es la STA *C* recibiendo paquetes desde Internet; en este caso, el proceso de comunicación se da en sentido inverso, es decir, Internet → Ethernet IEEE 802.3 → MP *B* → Mesh STAs → MAP *B* → STA *C*.

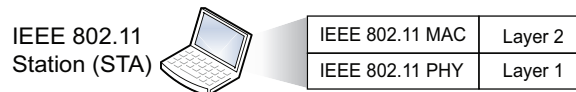


Figura 2.5: Dispositivo móvil IEEE 802.11 (STA).

Los dispositivos de red definidos por la enmienda IEEE 802.11s utilizan mecanismos especiales para operar y coordinar el funcionamiento de una iWMN (véase la Tabla 2.1). En la siguiente sección se detallan las funciones más importantes que establece la enmienda. El siguiente análisis presenta una visión integral de la operación de una iWMNs para resaltar los retos de desempeño que existen en dicha arquitectura. El mismo análisis permite descubrir las funciones que pueden explotarse para mejorar el desempeño de las iWMNs.

2.3 Funciones definidas por la enmienda IEEE 802.11s



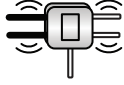

La enmienda IEEE 802.11s define un conjunto de funciones para la capa MAC (Layer 2) orientadas a establecer y operar una WMN. Estas funciones son utilizadas y aprovechadas para redefinir protocolos y funciones de red que permitan mejorar el desempeño de las iWMNs. En esta subsección se muestran las funciones más destacadas de la enmienda IEEE 802.11s.

2.3.1 Mesh discovery

Para **detectar una WMN activa**, una Mesh STA emplea la función denominada *Mesh Discovery*. Esta función puede realizarse de dos modos: *i) escaneo activo* o *ii) escaneo pasivo*.

Escaneo pasivo

Cuando una Mesh STA utiliza el modo de **escaneo pasivo**, debe esperar y escuchar las transmisiones realizadas por otras Mesh STAs, hasta recibir una trama *Beacon*, transmitidas periódicamente por las Mesh STAs que pertenecen a una WMN activa. La trama *Beacon* transporta un *Mesh Profile*, conteniendo los parámetros operativos de la WMN, tales como: el Mesh ID; el protocolo de selección de rutas con el que opera la red; la métrica de selección de rutas óptimas; entre otros. La recepción de este mensaje ayuda a identificar las WMNs activas y las Mesh STAs viables para establecer una conexión; dichas Mesh STAs son catalogadas como *Candidate Peer Mesh STAs*.

Dispositivo	Funciones
Mesh Station 	<ul style="list-style-type: none"> ■ Ejecutar las funciones definidas por la enmienda IEEE 802.11s. ■ Recibir y encaminar paquetes inalámbricamente dentro del MBSS[†].
Mesh Access Point 	<ul style="list-style-type: none"> ■ Ejecutar las funciones definidas por la enmienda IEEE 802.11s. ■ Recibir y encaminar inalámbricamente paquetes dentro del MBSS[†]. ■ Establecer conexiones Wi-Fi con los usuarios móviles. ■ Recibir y encaminar paquetes de los usuarios hacia el MBSS[†] y viceversa.
Mesh Portal 	<ul style="list-style-type: none"> ■ Ejecutar las funciones definidas por la enmienda IEEE 802.11s. ■ Recibir y encaminar inalámbricamente paquetes dentro del MBSS[†]. ■ Intercambiar paquetes destinados a redes que utilizan otras tecnologías.
Station 	<ul style="list-style-type: none"> ■ Asociarse con un Mesh Access Point para utilizar de manera transparente las funciones de encaminamiento y conectividad provistas por el MBSS[†]. ■ Transmitir y recibir paquetes hacia y desde el Mesh Access Point.

[†] *Mesh Basic Service Set*, término definido por la enmienda IEEE 802.11s para referirse a una WMN.

Tabla 2.1: Funciones principales de los dispositivos definidos por la enmienda IEEE 802.11s

Escaneo activo

En cambio, si una Mesh STA utiliza el modo de **escaneo activo**, la Mesh STA debe transmitir mensajes de prueba denominados como *Probe Requests*. Las Mesh STAs pertenecientes a una WMN activa que reciban un mensaje *Probe Request* deberán responder a dicha solicitud con un mensaje denominado como *Probe Response*. Esta respuesta también proporciona el *Mesh Profile* de operación de la WMN. De este modo, la Mesh STA que busca una WMN activa puede recibir respuestas de diferentes Mesh STAs y de distintas WMNs, por lo tanto, la Mesh STA que cumpla con criterios de elegibilidad es denominada como una *Candidate Peer Mesh STAs*.

Por último, la Mesh STA que realizó la detección de una WMN activa, puede establecer una, y solo una conexión (es decir, *Mesh Peering Connection*) con una *Candidate Peer Mesh STA*. Para poder establecer esta conexión, la Mesh STA que realizó el proceso de detección, debe primero configurar su *Mesh Profile* y hacerlo coincidir con el de la WMN activa. De este modo, todas las Mesh STAs de una WMN utilizarán los mismos parámetros operativos, garantizando así la interoperabilidad en la red. Cuando la Mesh STA haya configurado el *Mesh Profile* podrá iniciar el procedimiento de conexión con una *Candidate Mesh STA*.

2.3.2 Mesh peering management

La función *Mesh Peering Management (MPM)* se utiliza para **establecer, mantener y terminar conexiones entre dos Mesh STA** [20]. Como se ha señalado, la comunicación directa entre dos Mesh STAs solo puede realizarse si ambas han acordado y establecido una conexión entre ellas. Para consumir este proceso, las dos Mesh STA deben transmitir, recibir y procesar dos tramas: la denominada *Mesh Peering Open* para iniciar la conexión y la respuesta correspondiente denominada *Mesh Peering Confirm*.

Es de señalar que el establecimiento de la conexión se debe realizar en dos sentidos: ambas Mesh STAs deben transmitir la trama *Mesh Peering Open* y deben responder con la trama de respuesta *Mesh Peering Confirm*. Al iniciar el proceso de establecimiento de conexión entre Mesh STAs, la trama *Mesh Peering Open* transporta las capacidades operativas de cada Mesh STA, p. ej., tasas de transmisión soportadas, posibles canales de operación, funciones de seguridad, información específica del fabricante, entre otras muchas propiedades. De esta manera, ambas Mesh STAs comparten sus atributos operativos permitiendo, o no, aceptar el establecimiento de la conexión al responder con la trama *Mesh Peering Confirm*.

Por último, cuando una Mesh STA abandona una WMN activa, se deben finalizar las conexiones activas y eliminar la configuración establecida por el *Mesh Profile*. Para concluir una conexión entre dos Mesh STAs, ambas deben enviar, recibir y procesar la trama denominada *Mesh Peering Close*. Esta trama contiene diferentes campos con información de la WMN, incluyendo el *Mesh ID*, datos específicos del fabricante, así como la razón específica por la que se concluye la conexión.

2.3.3 Mesh coordination function

Para **definir las reglas de acceso al medio** las Mesh STA utilizan la función *Mesh Coordination Function (MCF)*. Esta define dos métodos de acceso al medio: el denominado *Enhanced Distributed Channel Access (EDCA)*, el cual es definido como mecanismo **obligatorio y por defecto para todas las Mesh STAs** y, como mecanismo opcional, el denominado *Mesh Controlled Channel Access (MCCA)*. A continuación se describen ambos métodos.

- **EDCA es un método de acceso aleatorio** diseñado para proporcionar calidad de servicio para aplicaciones heterogéneas [21]. EDCA define cuatro categorías de acceso (*Access Categories - AC*): *background*, *best-effort*, *video*, y *voice*. Cuando una trama es procesada en capa 2, dicha trama es denominada como una *MAC Service Data Unit (MSDU)*. Cuando una MSDU es recibida, EDCA clasifica la MSDU en su respectiva fila de acceso, tal como se muestra en la Figura 2.6. Cada fila tiene configurado sus propios parámetros para el acceso al medio, tales como: el *Arbitration Inter-Frame Space (AIFS)* y los límites de la ventana de contienda del mecanismo MAC (*Contention Window - CW*), es decir, CW_{\min} y CW_{\max} . Estos parámetros son configurados especialmente para priorizar el acceso al medio de una AC sobre otra. Si dos o más AC ganan el acceso al medio al mismo tiempo, el acceso al medio se otorga a la AC con la mayor prioridad, mientras que las otras ACs deben asumir que ha ocurrido una colisión en el medio. La Figura 2.6 muestra las cuatro filas de prioridad de EDCA, junto con sus respectivos parámetros de configuración (para mayores detalles, refiérase a la enmienda IEEE 802.11e).
- **MCCA es un método de acceso al medio que utiliza reservaciones** [22]. Este método permite a las Mesh STAs obtener el acceso al medio por un periodo de tiempo definido. De este modo, MCCA **reduce la contienda** por el acceso al medio y brinda Calidad de Servicio (*Quality of Service - QoS*) para cargas de tráfico intensas y periódicas. Cuando una Mesh STA utiliza MCCA envían una trama de reservación de canal a las Mesh STAs vecinas, dicha trama se define como una *MCCA Setup Request frame (MCCASR)*. El periodo de tiempo solicitado en un MCCASR se denomina como una *MCCA Opportunity (MCCAOP)*. Las Mesh STA que reciben un MCCASR deben: *i)* advertir el periodo de reservación a todas las Mesh STAs vecinas mediante la transmisión de una trama denominada como *MCCAOP Advertisement*, y, *ii)* deben enviar una trama *MCCA Setup Reply* a la Mesh STA que ha solicitado la reservación, para informar así que el periodo MCCAOP ha sido aceptado. De este modo, todas las Mesh STA que reciben, ya sea la trama MCCASR o la trama *MCCAOP Advertisement*, deben diferir sus transmisiones para evitar interferir las transmisiones que se realizaran durante el periodo de reserva MCCAOP. Finalmente, cuando el periodo MCCAOP inicia, la Mesh STA que solicito la reserva del canal obtiene una oportunidad de transmisión

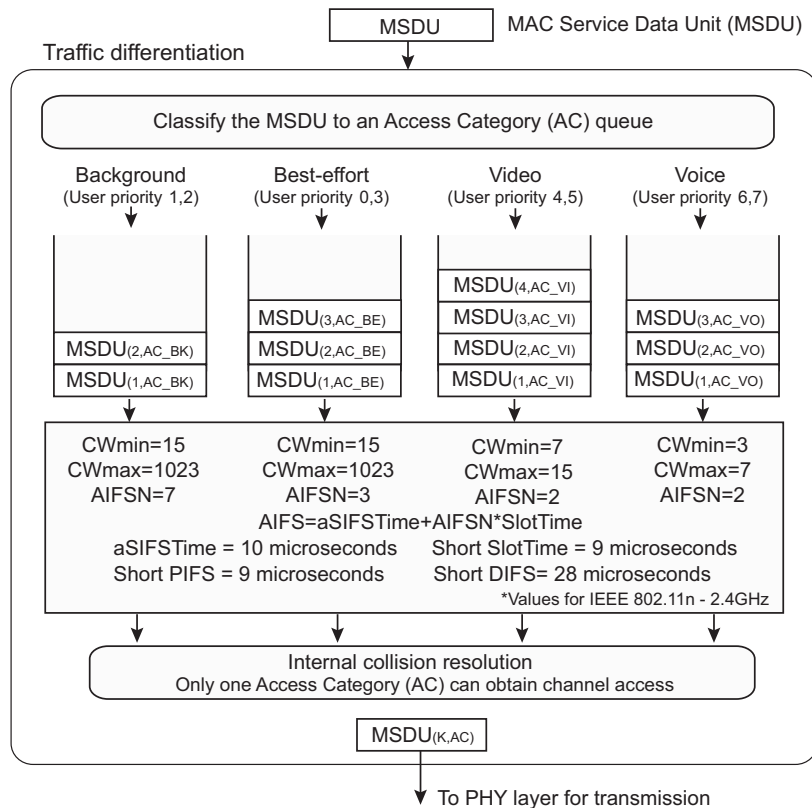


Figura 2.6: Mecanismo de diferenciación de tráfico implementado por EDCA. Cada categoría de acceso (AC) tiene diferentes prioridades de acceso al medio y estas prioridades son definidas a través de filas de acceso internas y parámetros de configuración para el acceso al medio.

(*Transmission Opportunity* - TXOP) al ganar una instancia de contienda de EDCA pero sin competencia por otras Mesh STAs.

2.3.4 Mesh path formation

La enmienda IEEE 802.11s define un **protocolo obligatorio para establecer rutas de comunicación en una WMN**. Además, la enmienda también permite la inclusión de otros protocolos y métricas para el establecimiento de rutas [23]. Así, las Mesh STAs pueden seleccionar entre diferentes protocolos de selección de rutas; pero solo uno de estos protocolos debe ser utilizado por todas las Mesh STAs de una WMN. De esta manera se garantiza la operatividad en la red. El **protocolo por defecto** definido por la enmienda IEEE 802.11s para la **construcción de rutas de comunicación en una WMN** es el denominado *Hybrid Wireless Mesh Protocol* (HWMP). Este protocolo de encaminamiento soporta **dos modos de operación**: el **modo reactivo** (*HWMP Reactive Mode*) y el **modo proactivo** (*HWMP Proactive Building Tree Mode*). Ambos modos de operación construyen rutas de comunicación utilizando direcciones MAC en lugar de direcciones IP.

HWMP Reactive Mode

El modo de operación *HWMP Reactive Mode* se utiliza para construir rutas de comunicación *a demanda*. Esto es, cada vez una Mesh STA requiere de una trayectoria de comunicación para comunicarse con una Mesh STA destino, debe transmitir en modo *broadcast* un mensaje denominado como *Path Request* (PREQ). Este mensaje se reexpide por otras Mesh STAs hasta contactar a la Mesh STA destino. Cuando esta Mesh STA recibe el mensaje PREQ, esta debe enviar –a la Mesh STA que originó el PREQ–, un mensaje en modo *unicast* denominado como

Path Reply (PREP). Finalmente, cuando la Mesh STA que ha solicitado la ruta recibe el mensaje PREP, el proceso de descubrimiento de trayectorias ha finalizado. La Figura 2.7 ilustra el funcionamiento del protocolo HWMP en modo reactivo; en la imagen superior se ilustra la transmisión en modo *broadcast* del mensaje PREQ, y en la imagen inferior se muestra la transmisión de la respuesta PREP en modo *unicast*.

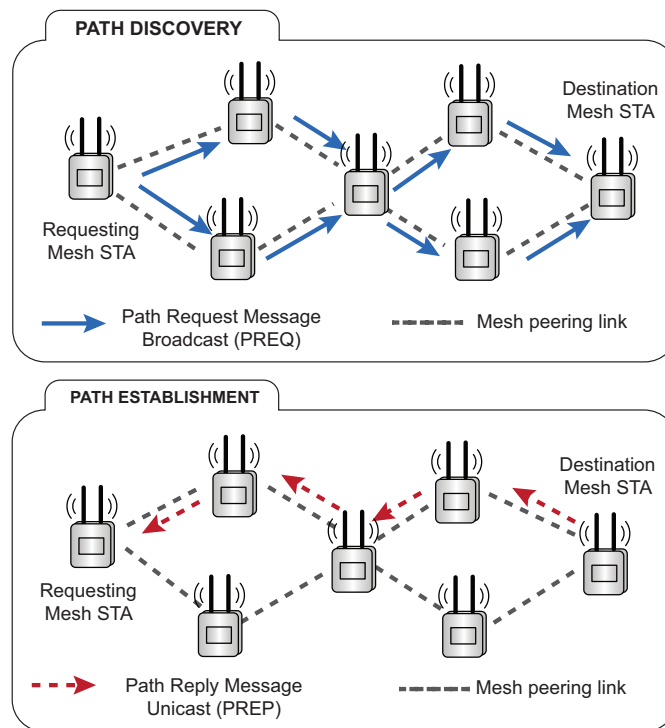


Figura 2.7: HWMP Reactive Mode

HWMP Proactive Building Tree Mode

El modo de operación denominado como *HWMP Proactive Building Tree* **construye un árbol de encaminamiento para establecer rutas de comunicación de manera proactiva**. Este modo requiere que una Mesh STA sea configurada como el nodo raíz del árbol (comúnmente un Mesh Portal asume el rol de nodo raíz de la red (*Root Node*)). El nodo raíz transmite periódicamente y en modo *broadcast* un mensaje denominado como *Root Announcement* (RANN). El objetivo de este mensaje es notificar la configuración de un Root Node a las Mesh STAs de la red. El mensaje RANN va recolectando información a medida que inunda la red; así, cada vez que una Mesh STA recibe el mensaje RANN, la Mesh STA puede seleccionar la Mesh STA que proporciona la ruta más eficiente para llegar hasta el nodo raíz. La Figura 2.8 muestra el modo de operación HWMP Proactive Building Tree; la imagen superior muestra la manera en la que el Root Node transmite el mensaje RANN en modo broadcast a todas las Mesh STAs de la red; la imagen inferior muestra como cada Mesh STA es responsable de seleccionar la Mesh STA que proporciona una ruta óptima hacia el nodo raíz.

2.3.5 Airtime Link Metric

La métrica de selección de trayectorias definida por la enmienda IEEE 802.11s es denominada como *Airtime Link Metric* (ALM) y cualquier modo de operación de HWMNP debe utilizarla por defecto. Esta métrica **identifica la ruta con mejor eficiencia de radiofrecuencia**, estimando la cantidad de recursos consumidos en el

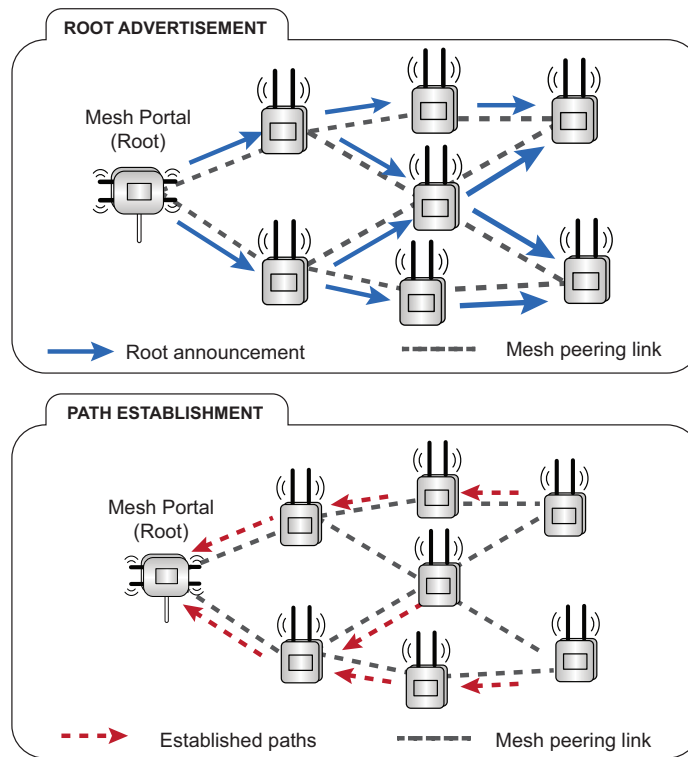


Figura 2.8: HWMP Proactive Building Tree Mode

canal inalámbrico por cada transmisión [24]. De manera tal que el consumo de recursos del canal se estima como

$$c_a = \left[O + \frac{B_t}{r} \right] \frac{1}{1 - e_f}, \quad (2.1)$$

donde r representa la tasa de bits del enlace; e_f , la tasa de error de trama para una trama de tamaño B_t y con un *overhead* de acceso al canal O .

2.3.6 Intra-mesh congestion control

El mecanismo *Intra-mesh congestion control* es utilizado para **controlar el flujo de datos en la red para evitar que se pierdan paquetes, debido a eventos de congestión**. Este mecanismo está constituido por tres funciones fundamentales, nombrados como *Local Congestion Monitoring and Congestion Detection*, *Congestion Control Signaling*, y *Local Rate Control*.

Local Congestion Monitoring and Congestion Detection

Con la función *Local Congestion Monitoring and Congestion Detection*, una Mesh STA puede monitorizar la carga de tráfico que procesa y detectar eventos de congestión inminentes cuando el tráfico de datos supera un umbral de detección. Si se prevé una inminente congestión, la Mesh STA determina cuales Mesh STAs deben reducir sus tasas de transmisión para evitar la congestión. Ante tal circunstancia, la Mesh STA detectora envía un mensaje de advertencia a las Mesh STAs causantes de esta congestión, para que reduzcan su tasa de transmisión hasta resolver el problema [25].

Congestion Control Signaling

La función *Congestion Control Signaling* especifica el formato de los mensajes que deben intercambiar las Mesh STAs para advertir un inminente evento de congestión. Cuando una Mesh STA prevé un evento de congestión, debe transmitir una trama de notificación a las Mesh STAs que están propiciando este evento. Esta trama de advertencia se denomina como *Congestion Control Notification* (CCN). La trama CCN indica a otras Mesh STAs que existe la posibilidad de congestionar la red, el tiempo aproximado de duración del evento de congestión, y que Mesh STAs están provocando este evento. Cuando una Mesh STA recibe la trama de notificación CCN, debe detener la reexpedición de paquetes o bien reducir su tasa de transmisión del tráfico destinado a las Mesh STAs listadas en la trama CCN. Cuando el tiempo de duración de la congestión notificado en el CNN termina, todas las Mesh STAs que detuvieron o redujeron sus tasas de transmisión deben retomar su comportamiento habitual.

Local Rate Control

El mecanismo *Local Rate Control* es el responsable de determinar el modo en que las Mesh STAs deben reducir su tasa de transmisión. Sin embargo, es importante señalar que la enmienda IEEE 802.11s únicamente especifica los mecanismos de señalización para el control de la congestión, los algoritmos que monitorizan y detectan inminentes eventos de congestión, así como los algoritmos de reducción de tasas de transmisión, están fuera del alcance de la enmienda y están abiertos para que cada fabricante pueda implementar sus propios algoritmos.

Existen otras múltiples funciones definidas por la enmienda IEEE 802.11s para la operación de una WMN. Sin embargo, estas otras funciones no afectan el desempeño de los protocolos que operan en capas superiores de la pila TCP/IP, y, por lo tanto, no son de importancia para este trabajo. Dichas funciones incluyen mecanismos para el ahorro de energía, para establecer funciones de seguridad en la red, etcétera ¹.

2.4 Beneficios de las iWMNs

Hasta ahora se han revisado las características y funciones más significativas de la enmienda IEEE 802.11s para mostrar que esta tecnología otorga numerosos beneficios al aplicarse en diversos escenarios. En la Tabla 2.2 se describen las ventajas más importantes que ofrecen las iWMNs.

2.5 Problemas de desempeño de las iWMNs

Las iWMNs ofrecen beneficios prácticos y económicos para administradores de la red, así como para sus usuarios. Sin embargo, a la fecha no existen estudios sobre los efectos negativos derivados de la interacción entre las iWMNs y los protocolos de comunicaciones de la pila TCP/IP, así como tampoco sobre las dificultades que trae consigo el uso de una topología jerárquica en la red. **No obstante, si existen evidencias que ponen en entredicho el grado de eficacia de las iWMNs** [26]; es por lo tanto que a continuación se revisan algunos de los problemas de desempeño que se han documentado en la literatura.

La **organización jerárquica de los nodos de una iWMN provoca efectos complejos**, como lo es que gran parte del tráfico de la red se concentre en nodos específicos, particularmente en los MPs y en los MAPs. Generando un incremento de la contienda por el medio entre los nodos vecinos a los MPs y MAPs y, por ende, aumenta el retardo de acceso al medio, e incrementa la probabilidad de pérdidas de paquetes por colisiones o por congestión en estos nodos. Estos problemas terminan por afectar severamente el desempeño de los protocolos de la capa de transporte [27]. Por lo tanto, se requieren mecanismos que logren el balance de carga en la

¹Si se desea ahondar en otras funciones definidas por enmienda IEEE 802.11s, se sugiere revisar la Sección intitulada *MAC Sub-Layer Management Entity Mesh Procedures* del estándar IEEE 802.11 [12]

Ventajas de las redes iWMNs
<ul style="list-style-type: none"> ■ Las iWMNs requieren de una infraestructura de muy bajo costo, y tienen una topología sumamente flexible, de manera tal que esta tecnología puede ser utilizada en diversos escenarios de aplicación y por diferentes usuarios o entidades. ■ Los mecanismos empleados por una iWMN permiten que la red sea auto-configurable y auto-organizada, lo que implica simplificar significativamente la administración de la red e incrementa su confiabilidad en escenarios dinámicos. ■ Comúnmente, los nodos de una iWMN son estáticos, esto significa que estos nodos no tengan restricciones de energía, procesamiento o almacenamiento. Dando pie a la incorporación de múltiples tareas de procesamiento en estos nodos. ■ Las iWMNs ofrecen un servicio orientado a ofrecer QoS para los usuarios de la red y a resolver los problemas de congestión en la red. ■ Las redes iWMN operan comúnmente en bandas espectrales de acceso libre, así que pueden utilizarse sin las regulaciones gubernamentales del espectro radioeléctrico. ■ Las iWMNs tienen mecanismos de seguridad que permiten que los paquetes de datos solo sean manipulados por nodos autorizados, por lo que garantiza que la información solo sea manipulada por los nodos autorizados para ese fin.

Tabla 2.2: Ventajas de las redes mallas IEEE 802.11s en modo infraestructura (iWMNs).

red, requiriendo nuevos algoritmos de encaminamiento [28]. Por otro lado, se puede dar un acceso inequitativo de los servicios de la red; es decir, los nodos cercanos a un MP tendrán un mejor acceso a los servicios de conectividad de la iWMN, que aquellos nodos cuyo tráfico debe ser retransmitido por varias Mesh STAs hasta llegar al MP [29].

Sumado a lo anterior, tenemos que **el conjunto heterogéneo de aplicaciones que utilizarán los servicios de conectividad de una iWMN demandarán requerimientos elevados de QoS**, mientras que **el ancho de banda disponible en redes inalámbricas es un recurso limitado** [30, 31]. La conjugación de estos factores impactará seriamente en la calidad de la experiencia para los usuarios de la red. El reto es complejo, y requiere de una alta eficiencia de la pila de protocolos para aprovechar al máximo los recursos existentes y limitados de la red. Con el fin de satisfacer los requerimientos de desempeño de las aplicaciones de los usuarios.

Además, **la movilidad de los usuarios móviles es un factor que ocasiona frecuentes pérdidas de conectividad**, retardos para ejecutar eficientemente el *handoff vertical*, pérdidas y retardos para los paquetes; todo esto desencadena la ejecución recurrente de mecanismos de restablecimiento de rutas y un desperdicio constante de los recursos de la iWMN [32].

Las aplicaciones de los usuarios móviles utilizan ampliamente el protocolo de transporte TCP para transportar datos desde y hacia Internet. **La interacción del protocolo TCP en redes inalámbricas ha sido ampliamente estudiada en la literatura**, p. ej., [33, 34, 35], y **se han encontrado numerosos problemas de desempeño**. En [36], los autores sintetizan cuatro causas fundamentales que inducen el bajo desempeño de TCP en redes inalámbricas:

1. TCP asume que las pérdidas de paquetes siempre son originadas por congestión en la red. Esto significa que TCP es incapaz de identificar las diferentes causas de pérdidas de paquetes, ya que se pueden atribuir a: errores de transmisión, colisiones, errores de ruta, o bien por desconexiones en la red.

2. TCP sufre de constantes pérdidas de conectividad debido a la movilidad de los nodos, afectando así la tasa de transmisión de este protocolo.
3. La contienda por el medio genera variabilidades en el retardo y pérdidas de paquetes –ocasionados por colisiones o por los efectos de la terminal oculta–. Estos eventos afectan el mecanismo de autorregulación de TCP, y por ende la utilización eficiente del medio inalámbrico.
4. TCP, en combinación con los protocolos de acceso al medio, tiende a distribuir inequitativamente el ancho de banda disponible en la red.

Para resumir los problemas de desempeño asociados a la operación de las iWMNs, en la Tabla 2.3 se han sintetizado de los problemas descritos.

Problemas de desempeño en iWMNs
<ul style="list-style-type: none"> ■ Contienda por el medio: Las iWMNs tienen una infraestructura organizada de manera jerárquica, lo que incrementa la contienda por el medio en ciertas zonas de la red. ■ Inequidad: La interacción entre diferentes protocolos puede conducir a una distribución inequitativa de los recursos de la red para aquellos nodos más alejados de los MPs. ■ Calidad de servicio: El tráfico de los usuarios móviles es grande y deben atenderse con diferentes requerimientos de QoS. ■ Alta movilidad: La movilidad de los usuarios ocasiona constantes pérdidas de conectividad, afectando intermitentemente las conexiones en la red y el funcionamiento de protocolos tales como TCP. ■ Capacidad limitada: La capacidad de los enlaces inalámbricos es un recurso limitado, las iWMNs deben utilizarlo eficientemente para dar servicio a una alta cantidad de tráfico en la red. ■ Errores en el medio: El elevado número de transmisiones inalámbricas en la red incrementa significativamente la probabilidad de colisiones y pérdidas de paquetes. ■ Interacción con TCP: TCP transportará gran parte del tráfico en una iWMN; sin embargo, TCP opera ineficientemente en redes inalámbricas pues no distingue entre pérdidas por congestión de la red y pérdidas debido a factores asociados al medio. ■ Balance de carga: Las iWMNs requieren algoritmos de ruteo que realicen eficientemente el balance de carga para reducir la carga de tráfico en los MAPs.

Tabla 2.3: Desventajas de las redes malladas IEEE 802.11s en modo infraestructura (iWMNs).

CAPÍTULO 3

Codificación de red

“En teoría, no hay diferencia entre teoría y práctica; en la práctica sí.”

Benjamin Brewster

LA codificación es una técnica muy común que se ha empleado con diferentes objetivos. Por ejemplo, la **codificación de la fuente** se utiliza como un mecanismo de compresión para reducir la redundancia de información y así disminuir los recursos necesarios para transportarla. La **codificación de canal**, es la que se utiliza como un mecanismo para la generación de redundancia que ayuda a transmitir información de manera confiable en canales con pérdidas. Y la **codificación de red** es un mecanismo que reduce la utilización de la red, y también se emplea para mejorar el proceso de recuperación de pérdidas de información [37].

3.1 Codificación de red

En el año 2000, Ahlswede, Cai, Li, y Yeung publicaron el artículo intitulado “*Network information flow*” [38], artículo considerado como el documento seminal donde se define por primera vez la técnica denominada como **codificación de red**. Esta herramienta está basada en una idea sencilla, pero con un enorme potencial para redes de conmutación de paquetes, es decir, en lugar de que los nodos de la red actúen simplemente como re-expedidores de paquetes. **Con la técnica de codificación de red, un nodo no solo es capaz de reexpedir paquetes, sino que también puede calcular una función de codificación con los paquetes que recibe y transmitir el resultado de dicha función a otros nodos de la red.** El cálculo de dicha función, la cual no es más que un mapeo entre los paquetes recibidos y el paquete transmitido, es lo que Ahlswede et al. han definido como **codificación de red**. Con esta técnica, los nodos no son solo capaces de codificar información con los paquetes que reciben, sino que también pueden recibir paquetes codificados y decodificarlos para obtener información útil. Uno de los mayores beneficios de la codificación de red es que coadyuva a incrementar la *capacidad de la red*, ya que con el uso de esta técnica se puede transportar más información con un menor número de transmisiones en los enlaces de comunicación.

La Figura 3.1 muestra cómo opera la técnica de codificación de red. En este ejemplo, el nodo *A* recibe dos mensajes que contienen la información X_1 y X_2 respectivamente; cuando el nodo *A* recibe esta información, calcula una función de codificación para formar un mensaje codificado, definido como $f(X_1, X_2)$. Este mensaje codificado es calculado con una función de codificación que opera con la información X_1 y X_2 . Ahora *A* puede transmitir el mensaje codificado $f(X_1, X_2)$ hacia el nodo *B* en lugar de transmitir individualmente X_1 y X_2 ; en otras palabras, **solo se requiere una transmisión sobre el enlace de comunicaciones que une a *A* y *B*, en lugar de dos transmisiones que se emplearían de manera tradicional.** Cuando *B* recibe el mensaje codificado $f(X_1, X_2)$, se puede decodificar la información recibida y obtener la información original X_1 y X_2 . Con este ejemplo sencillo se muestra como la codificación de red puede reducir significativamente el número

de transmisiones en el enlace de comunicación; sin embargo, nótese que se requiere de una nueva tarea de procesamiento en los nodos de la red para poder codificar y decodificar información. A continuación se presentan dos de los esquemas más importantes para realizar el proceso de codificación y decodificación de red: la **codificación lineal aleatoria de red** y **codificación con suma aritmética módulo-2**.

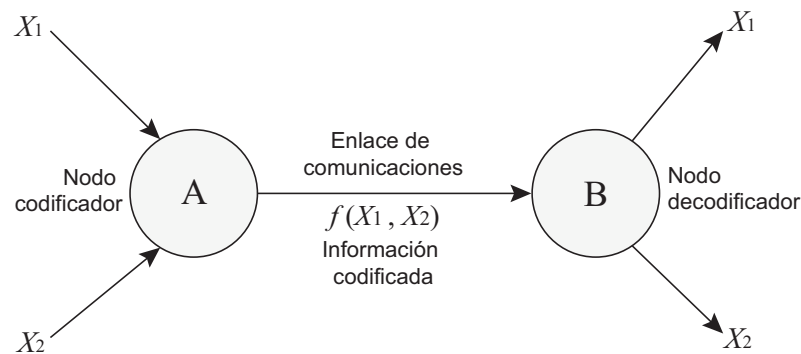


Figura 3.1: El nodo A recibe X_1 y X_2 , con esta información calcula el mensaje codificado $f(X_1, X_2)$ y lo transmite al nodo B. El nodo B realiza la función de decodificación sobre el mensaje codificado para obtener la información original X_1 y X_2 . Con este esquema, se reducen el número de transmisiones sobre el enlace que va de A a B.

3.2 Codificación lineal aleatoria (RLNC)

Tradicionalmente, si una fuente de información quiere transmitir un bloque de datos a un destino, se deben realizar cuatro tareas fundamentales: *i*) la fuente debe dividir la información original en K segmentos, *ii*) la fuente debe enviar los K segmentos al sistema de comunicaciones, *iii*) el sistema de comunicaciones debe transportar todos los segmentos desde la fuente hasta el destino y, finalmente, *iv*) el destino debe recibir los K segmentos y reconstruir la información original. Si solo uno de los K paquetes no llega al destino, entonces no se puede reconstruir la información original transmitida por la fuente. Si este fuese el caso, la única forma de recuperar los paquetes perdidos es solicitando explícitamente a la fuente que retransmita dichos paquetes mediante un mensaje de solicitud de retransmisión. Es claro que **cuando se pierden paquetes en el sistema de comunicaciones se generan dos efectos que degradan el desempeño: i) se reutiliza el sistema de comunicaciones para transportar las solicitudes de retransmisión así como las retransmisiones, y ii) se agrega un retardo al proceso de comunicación**. La Figura 3.2 ilustra claramente estos efectos negativos; en este ejemplo, la fuente transmite cuatro paquetes, de los cuales se pierde uno en el sistema de comunicaciones, en este caso el destino debe solicitar explícitamente a la fuente la retransmisión del paquete perdido, reutilizando así el sistema de comunicaciones e incorporando un retardo hasta obtener el paquete perdido. Una vez que se recibe el paquete perdido, el destino puede obtener la información generada en la fuente.

La **codificación lineal aleatoria de red** o RLNC, por sus siglas en inglés (Random Linear Network Coding), puede resolver el problema de las pérdidas mediante un esquema de redundancia. Con RLNC se pueden combinar algebraicamente K segmentos de información y crear nuevas representaciones denominadas como **paquetes codificados** (véase la Figura 3.3). Estos nuevos paquetes codificados se transmiten por sistema de comunicaciones, el cual transportará a dichos paquetes del mismo modo que lo hace con los paquetes no codificados, es decir, los encaminará desde la fuente hasta el destino. Aún si en el sistema de comunicaciones se pierden paquetes codificados, la información original se puede obtener con solo recibir suficientes paquetes. **El destino solo necesita recibir cualesquiera K paquetes codificados para descifrar la información original generada por la fuente**. Esto significa que **no importa que paquetes codificados se pierdan en el sistema de comunicaciones, basta con que K paquetes lleguen al destino para que se pueda decodificar la información original**. Por lo tanto, la fuente puede transmitir de manera proactiva más de K paquetes codificados que servirán como redundancia. En la Figura 3.3 se aprecia como la fuente calcula y transmite $K + 2$ paquetes

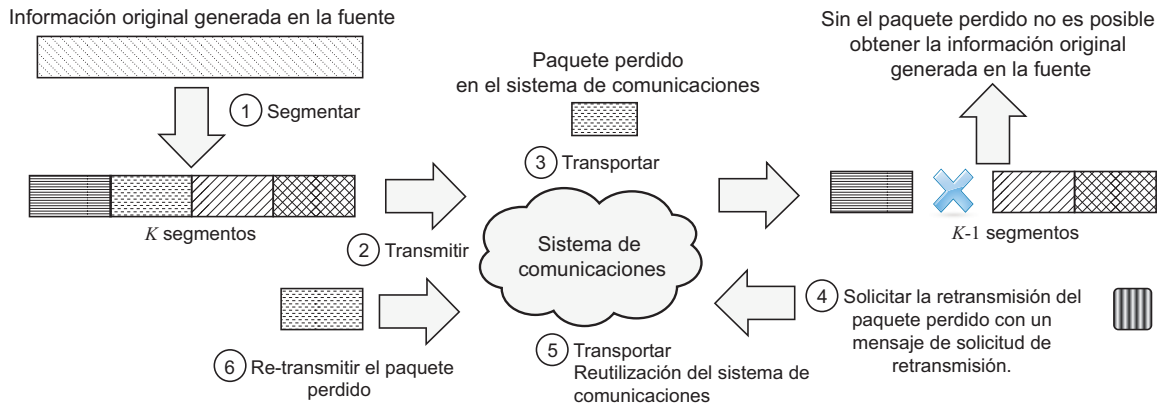


Figura 3.2: Paquete extraviado en un sistema de comunicaciones y su respectiva solicitud de retransmisión.

codificados, esto significa que dos paquetes funcionan como redundancia. Sin importar que par de paquetes codificados se pierdan en la red, basta con que K paquetes codificados lleguen al destino para que éste pueda decodificar la información original transmitida por la fuente. De este modo, se solucionan los efectos negativos de la pérdida de paquetes, pues se evita la reutilización de la red y la retransmisión de información.

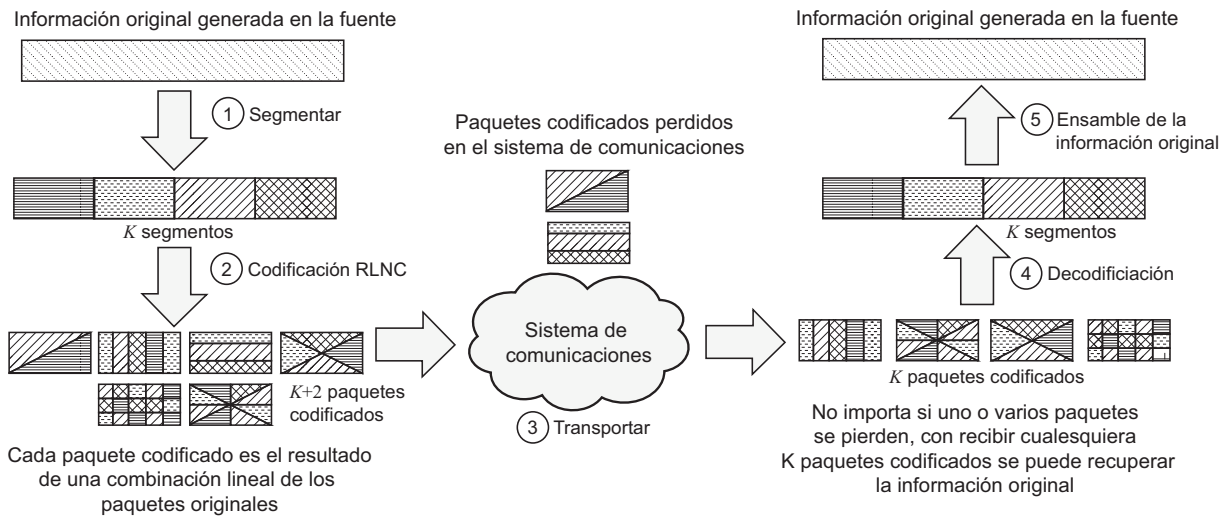


Figura 3.3: Segmentos extraviados en un sistema de comunicaciones empleando RLNC.

3.2.1 Proceso de codificación

Asuma que un nodo genera, o recibe, n paquetes de datos, P_1, P_2, \dots, P_n . Estos n paquetes aún no codificados se denominan como **paquetes nativos**. Cada paquete nativo está compuesto por una cadena de L bits, y estos bits se pueden dividir en sub-cadenas de m bits de manera que cada sub-cadena puede ser interpretada como un símbolo que reside sobre un campo finito F_{2^a} . Por lo tanto, cada paquete nativo se constituye por un conjunto de L/m símbolos a los que se les puede aplicar una transformación lineal antes de ser transmitidos [39].

El i -ésimo símbolo del paquete j -ésimo se representa con la notación $P_j(s_i)$. **La codificación lineal de red permite generar paquetes codificados X_l , donde $l \geq j$, utilizando sumas y multiplicaciones sobre los símbolos de un conjunto paquetes nativos.** Para eso se requieren n vectores de codificación, \vec{C}_j , cuyos símbolos de $C_j(1), C_j(2), \dots, C_j(n)$ pueden ser seleccionados de manera aleatoria o determinista. El objetivo es crear un sistema de paquetes codificados que pueden ser analizados como si constituyeran un sistema de ecuaciones lineales como el que se muestra a continuación. Los n paquetes codificados, X_1, X_2, \dots, X_n , son calculados empleando operaciones sobre los símbolos de los n paquetes nativos y n vectores de codificación.

$$\begin{aligned} X_1 &= C_1(1)P_1 + C_1(2)P_2 + \dots + C_1(n)P_n \\ X_2 &= C_2(1)P_1 + C_2(2)P_2 + \dots + C_2(n)P_n \\ &\vdots \\ X_n &= C_n(1)P_1 + C_n(2)P_2 + \dots + C_n(n)P_n \end{aligned}$$

El cálculo de cada uno de los símbolos de los paquetes codificados, X_l , se realiza con base en el siguiente lema.

Lema 1

Sea $n \in \mathbb{N}$ el número de paquetes nativos seleccionados para realizar el proceso de codificación lineal y sea \vec{C}_i el i -ésimo vector de codificación constituido por los n símbolos $C_i(1), C_i(2), \dots, C_i(n)$, estos símbolos son tomados del campo F_{2^q} pueden ser definidos de manera determinista o aleatoria. Si cada paquete nativo está compuesto por L/m símbolos, entonces $P_i(s_k)$ es el k -ésimo símbolo del paquete nativo i , de tal modo que el k -ésimo símbolo del j -ésimo paquete codificado, $X_j(s_k)$, se calcula como:

$$X_j(s_k) = \sum_{i=1}^n C_j(i)P_i(s_k), \quad \text{para } k = 1, 2, \dots, L/m \quad (3.1)$$

Cuando los símbolos del vector de codificación, \vec{C}_i , son seleccionados aleatoriamente, a cada paquete codificado X se adjunta, comúnmente, el vector de codificación \vec{C} antes de ser transmitido; así, el receptor podrá tener toda la información necesaria para poder decodificar los n paquetes nativos. Las operaciones suma y multiplicación para la generación del paquete codificado X se realizan sobre el campo finito F_{2^q} y sobre todos los símbolos de los paquetes nativos.

Para clarificar el proceso de codificación lineal, en la Figura 3.4 se muestra la codificación de dos paquetes nativos, P_1 y P_2 . Cada uno de estos paquetes está constituido por dos símbolos, $P_1 = \{P_1(s_1), P_1(s_2)\}$ y $P_2 = \{P_2(s_1), P_2(s_2)\}$. Para codificar los paquetes nativos se generan aleatoriamente dos vectores de codificación, \vec{C}_1 y \vec{C}_2 , cada uno conformado por dos símbolos (pues es el número de paquetes que se codificarán), es decir, $\vec{C}_1 = \{C_1(1), C_1(2)\}$ y $\vec{C}_2 = \{C_2(1), C_2(2)\}$. Cada uno de los vectores de codificación, \vec{C}_i , se emplean para construir un paquete codificado, cuya formulación algebraica se muestra en las ecuaciones 2.3 y 2.4. Nótese que a los paquetes codificados, X_1 y X_2 , se les ha adjuntado los símbolos de los vectores de codificación en las cabeceras de dichos paquetes.

$$X_1 = \{C_1(1), C_1(2)\}, \{C_1(1)P_1(s_1) + C_1(2)P_2(s_1)\}, \{C_1(1)P_1(s_2) + C_1(2)P_2(s_2)\} \quad (3.2)$$

$$X_2 = \{C_2(1), C_2(2)\}, \{C_2(1)P_1(s_1) + C_2(2)P_2(s_1)\}, \{C_2(1)P_1(s_2) + C_2(2)P_2(s_2)\} \quad (3.3)$$

En el ejemplo que se muestra en la Figura 3.4 se generan los paquetes codificados X_1 y X_2 . Es importante tener en cuenta que se pueden generar más paquetes codificados empleando más vectores de codificación. De este modo, el envío de más paquetes codificados garantiza que el receptor recibirá suficiente información como para poder decodificar los paquetes nativos, aún si se pierden algunos paquetes codificados. Por ejemplo, si se emplearan los vectores de codificación \vec{C}_3 y \vec{C}_4 , se podrían formar los paquetes codificados X_3 y X_4 , empleando el procedimiento descrito en el Lema 1. La ventaja es que con cualquier par de paquetes codificados que reciba el receptor se pueden decodificar los paquetes nativos P_1 y P_2 . De tal modo que se pueden transmitir más de dos paquetes codificados y minimizar así los efectos de las probables pérdidas de paquetes en la red.

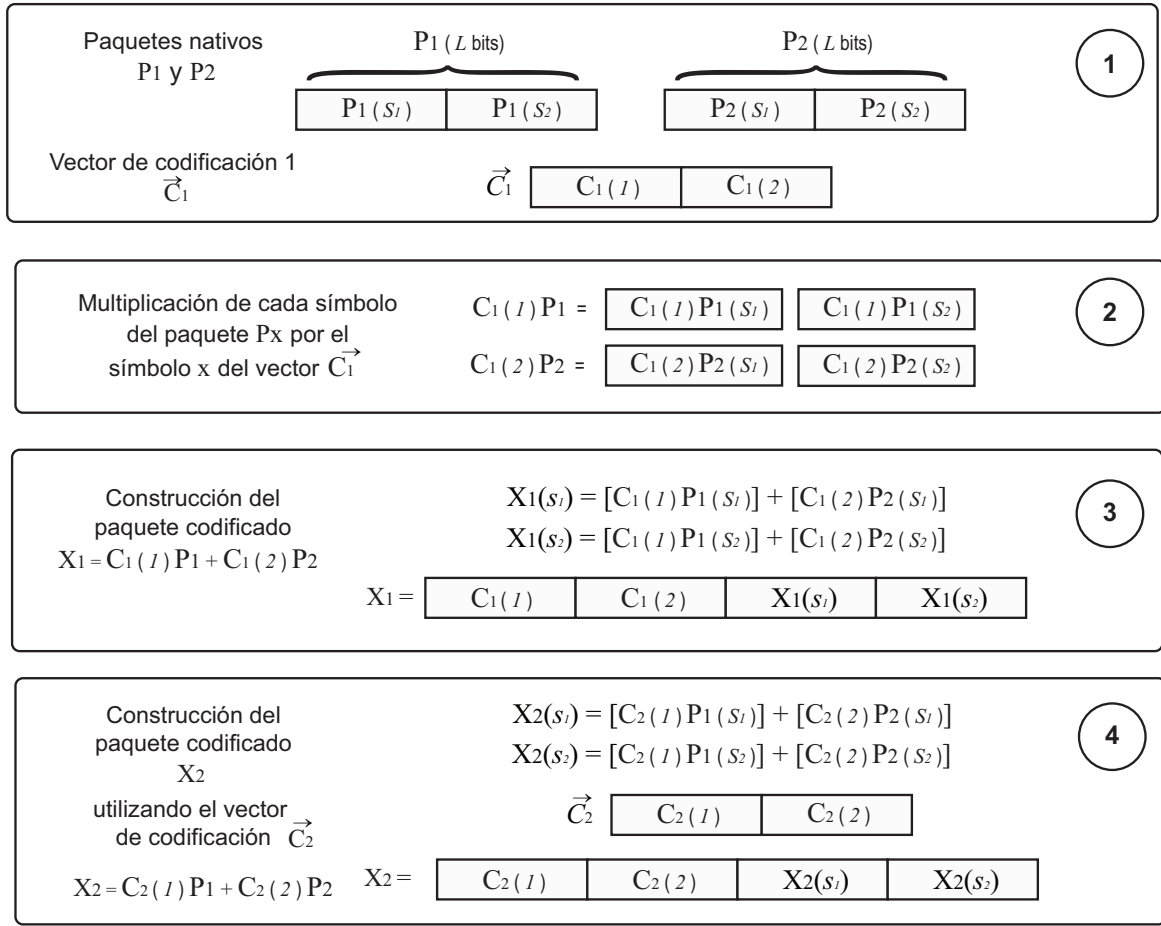


Figura 3.4: Codificación lineal aleatoria de la red (*Random Linear Network Coding*) empleando dos paquetes nativos P_1 y P_2 y los vectores de codificación \vec{C}_1 y \vec{C}_2 .

3.2.2 Proceso de decodificación

El proceso de decodificación es realizado en el dispositivo destino tan pronto como se hayan recibido, al menos, n paquetes codificados, X_1, X_2, \dots, X_n . Para recuperar los n paquetes nativos, el nodo receptor de los paquetes codificados tendrá que resolver el siguiente sistema de ecuaciones lineales:

$$\begin{aligned}
 X_1 &= C_1(1)P_1 + C_1(2)P_2 + \dots + C_1(n)P_n \\
 X_2 &= C_2(1)P_1 + C_2(2)P_2 + \dots + C_2(n)P_n \\
 &\vdots \\
 X_n &= C_n(1)P_1 + C_n(2)P_2 + \dots + C_n(n)P_n
 \end{aligned}$$

Este sistema de ecuaciones puede ser expresado como un producto matricial. El resultado queda de la siguiente forma:

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} = \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix} \begin{pmatrix} C_1(1) & C_1(2) & \dots & C_1(n) \\ C_2(1) & C_2(2) & \dots & C_2(n) \\ \vdots & \vdots & & \vdots \\ C_n(1) & C_n(2) & \dots & C_n(n) \end{pmatrix}$$

Este producto matricial puede manipularse algebraicamente para para obtener el siguiente sistema de ecua-

ciones.

$$\begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \begin{pmatrix} C_1(1) & C_1(2) & \dots & C_1(n) \\ C_2(1) & C_2(2) & \dots & C_2(n) \\ \vdots & \vdots & & \vdots \\ C_n(1) & C_n(2) & \dots & C_n(n) \end{pmatrix}^{-1}$$

El proceso de decodificación se realiza al resolver este sistema de ecuaciones. Para hallar la solución la matriz C^{-1} se debe manipular para convertirla en una matriz triangular superior o inferior, y así obtener el contenido de los n paquetes nativos [40]. Comúnmente, la solución de este sistema de ecuaciones se realiza con la técnica de eliminación de Gauss-Jordan. Tras solucionar el sistema de ecuaciones, la matriz C^{-1} se convierte en una matriz triangular superior, y el valor de sus coeficientes ha cambiado debido a la manipulación algebraica. Lo importante es que, después de la manipulación, el sistema de ecuaciones se resuelve de manera muy sencilla y es posible obtener la información original de los n paquetes nativos.

$$\begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \begin{pmatrix} C'_1(1) & C'_1(2) & C'_1(3) & \dots & C'_1(n) \\ 0 & C'_2(2) & C'_2(3) & \dots & C'_2(n) \\ 0 & 0 & C'_3(3) & \dots & C'_3(n) \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & C'_n(n) \end{pmatrix}^{-1}$$

3.3 Codificación y decodificación empleando suma aritmética módulo-2

Una de las formas más comunes de implementar la codificación de red es utilizando, como función de codificación, la operación *suma módulo 2*, es decir, la operación lógica XOR, cuyo operador lógico se representa como \oplus . y cuya tabla de verdad se muestra en la Tabla 3.1. El funcionamiento del operador lógico XOR puede sintetizarse como un operador que funciona a nivel de bit y que capaz de identificar si los operandos son diferentes; es decir, el operador XOR da como resultado un 1 lógico si el valor lógico de los operandos es diferente y un 0 si el valor lógico de ambos operandos es igual.

A	B	A \oplus B
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 3.1: Tabla de verdad del operador lógico XOR.

Al emplear la operación lógica XOR se simplifica considerablemente las operaciones de codificación y decodificación de información. Con base en las expresiones lógicas 2.5 y 2.6, se pueden realizar las operaciones de codificación y decodificación de información. Por ejemplo, asuma que X es el resultado de la operación lógica XOR entre las variables lógicas A y B (véase la ecuación 2.7). X puede ser visto como la información codificada, mientras que A y B pueden interpretarse como la información nativa. Para realizar el proceso de decodificación, la variable lógica A puede calcularse si se conocen X y B (la ecuación 2.8 muestra el procedimiento), o bien, la variable lógica B puede calcularse si se conocen X y A (la ecuación 2.9 muestra el procedimiento). La demostración de los procesos de codificación y decodificación empleando el operador XOR se muestra en la Tabla 3.2.

$$A \oplus A = 0 \tag{3.4}$$

$$A \oplus 0 = A \quad (3.5)$$

$$X = A \oplus B \quad (3.6)$$

$$A = B \oplus X \implies A = B \oplus (A \oplus B) \implies A = (B \oplus B) \oplus A \implies A = 0 \oplus A \implies A = A \quad (3.7)$$

$$B = A \oplus X \implies B = A \oplus (A \oplus B) \implies B = (A \oplus A) \oplus B \implies B = 0 \oplus B \implies B = B \quad (3.8)$$

Información		Codificación	Decodificación de A	Decodificación de B
A	B	$X = A \oplus B$	$A = X \oplus B$	$B = X \oplus A$
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

Tabla 3.2: Codificación y decodificación de variables lógicas empleando el operador XOR.

Las propiedades del operador XOR, mostradas en las ecuaciones 2.5 y 2.6, pueden extender el proceso de decodificación de información tal y como se muestra en el siguiente ejemplo. Si se asume que la operación $X = A \oplus B \oplus C$ es el proceso de codificación de las variables lógicas A , B y C , entonces A puede calcularse conociendo X , B y C tal y como se desarrolla en la ecuación 2.9. Empleando un procedimiento similar al que se muestra en la ecuación 2.9, se puede demostrar que B puede calcularse si se conocen X , A y C , mientras que C puede calcularse conociendo X , A y B .

$$A = X \oplus B \oplus C \implies A = (A \oplus B \oplus C) \oplus B \oplus C \implies A = A \oplus (B \oplus B) \oplus (C \oplus C) \implies A = A \oplus 0 \implies A = A \quad (3.9)$$

3.3.1 Codificación de flujos *multicast* en una topología *butterfly*

Para ilustrar el potencial de la codificación de red empleando la operación lógica XOR, a continuación se analiza uno de los ejemplos más significativos y revisados en la literatura: la codificación de red en una topología *mariposa* (*butterfly networks*) [38, 41, 42]. Asumamos que se desea aplicar la técnica de codificación en una red que utiliza como paradigma de comunicación la conmutación de paquetes. El grafo dirigido $G(V, E)$, que se muestra en la Figura 3.5, modela la red, siendo esta topología comúnmente conocida como *topología mariposa*. El nodo S actúa como una fuente de tráfico y desea transmitir los bits b_1 y b_2 a dos nodos destinos, es decir, T1 y T2. Los nodos T1 y T2 deben recibir los dos bits generados por S , lo cual se conoce comúnmente como una transmisión en modo *multicast* de S a T1 y T2. En la red, los nodos 1, 2, 3 y 4 únicamente actúan como nodos que reexpiden tráfico en la red, es decir, si dichos nodos reciben un bit entonces reexpedirán ese bit a sus nodos vecinos. Cada una de las aristas del grafo G representa un canal de comunicaciones libre de errores, consideración pertinente para modelar enlaces cableados de hoy en día, y dicho enlace es capaz de transmitir un bit durante cada uso del canal.

En este ejemplo, el nodo S transmite el bit b_1 al nodo 1 por el enlace ($S \rightarrow 1$) y el bit b_2 por el enlace ($S \rightarrow 2$) hacia el nodo 2. Cuando el nodo 1 recibe el bit b_1 , lo retransmite por los enlaces ($1 \rightarrow 3$) y ($1 \rightarrow T1$). Cuando el nodo 2 recibe el bit b_2 , lo retransmite por los enlaces ($2 \rightarrow 3$) y ($2 \rightarrow T2$) (véase la Figura 3.6). En este instante de tiempo, el nodo T1 ya ha recibido el bit b_1 y el nodo T2 ya ha recibido el bit b_2 ; sin embargo, el nodo 3 deberá determinar que bit transmitirá por el enlace ($3 \rightarrow 4$). Si el nodo 3 transmite b_1 y, posteriormente, el nodo 4 reexpide dicho bit por los enlaces ($4 \rightarrow T1$) y ($4 \rightarrow T2$), entonces T2 habrá recibido los bits b_1 y b_2 ,

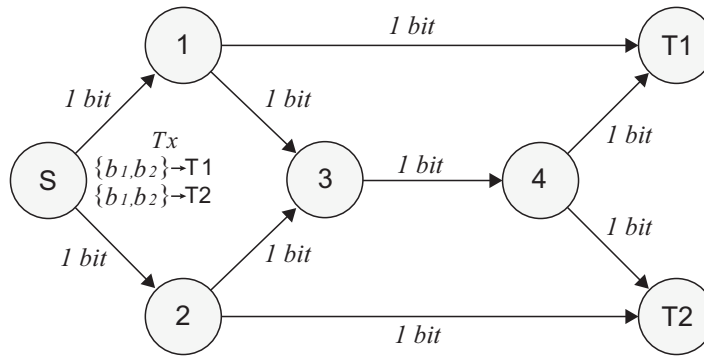


Figura 3.5: Grafo $G(V, E)$ para una red con *topología mariposa*. En la red se realiza una transmisión *multicast* entre el nodo S y los nodos $T1$ y $T2$.

y $T1$ solo tendrá dos copias del bit b_1 . Por otro lado, si el nodo 3 transmite b_2 y, posteriormente, el nodo 4 reexpide dicho bit, entonces $T1$ tendrá los bits b_1 y b_2 mientras que $T2$ solo habrá recibido dos copias del bit b_2 . Este esquema impide realizar eficientemente la transmisión *multicast* de S a $T1$ y $T2$, ya que el enlace $(3 \rightarrow 4)$ actúa como un cuello de botella. La única manera de completar la transmisión *multicast* es utilizando, en más de una ocasión, los enlaces $(3 \rightarrow 4)$, $(4 \rightarrow T1)$ y $(4 \rightarrow T2)$. Esto quiere decir que se requieren un total de ocho transmisiones para enviar dos bits en modo *multicast*.

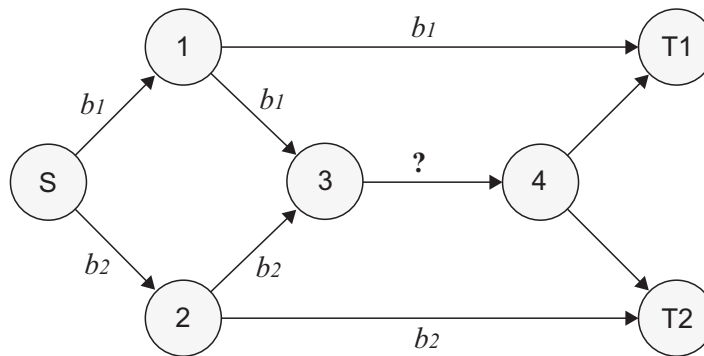


Figura 3.6: Transmisión *multicast* con un cuello de botella que reduce la capacidad de la red.

La codificación de red puede mejorar el desempeño de la red de la siguiente forma. Refiérase a la Figura 3.7, en esta imagen se aprecia que el nodo fuente, S , ha transmitido los bits b_1 y b_2 al nodo 1 y al nodo 2 respectivamente. Posteriormente, los nodos 1 y 2 retransmiten los bits b_1 y b_2 hacia el nodo 3 y, al mismo tiempo, han transmitido b_1 a $T1$ por el enlace $(1 \rightarrow T1)$ y b_2 a $T2$ por el enlace $(2 \rightarrow T2)$. En este instante, el nodo 3 puede crear un mensaje codificado de un solo bit empleando la operación XOR con los bits b_1 y b_2 . El bit codificado es transmitido por el enlace $(3 \rightarrow 4)$ y, una vez que es recibido por el nodo 4 se retransmite a los nodos $T1$ y $T2$. Los nodos $T1$ y $T2$ pueden decodificar el paquete empleando la información recibida previamente, es decir, $T1$ ha recibido a b_1 y $T2$ a b_2 . De esta manera, $T1$ puede obtener el bit b_2 empleando la operación lógica $b_1 \oplus (b_1 \oplus b_2)$ y $T2$ puede obtener el bit b_1 realizando la operación lógica $b_2 \oplus (b_1 \oplus b_2)$. De esta manera se concluye la transmisión *multicast* empleando únicamente seis transmisiones en la red.

3.3.2 Codificación de flujos *unicast* en una topología *butterfly*

La técnica de codificación de red también puede beneficiar a flujos *unicast* en una red con topología *mariposa*. El grafo dirigido $G(V, E)$ que se muestra en la Figura 3.8 representa una red de conmutación de paquetes con una topología que también se conoce como *mariposa*. En este escenario el nodo $S1$ debe transmitir el bit b_1 al

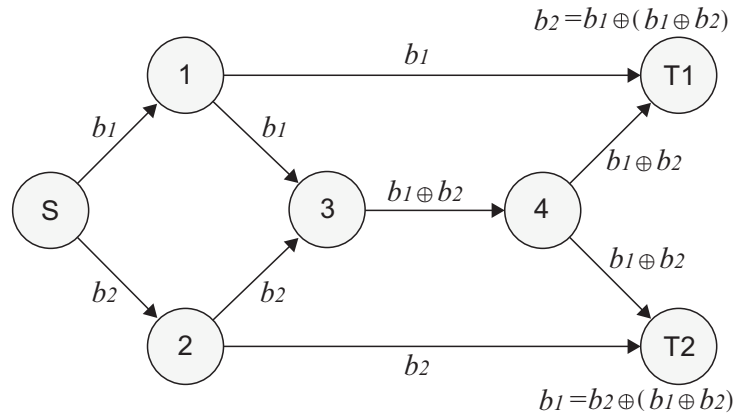


Figura 3.7: Codificación en una topología *mariposa* para incrementar la capacidad de la red.

nodo T1, mientras que el nodo S2 debe transmitir el bit b_2 al nodo T2. Cada arista del grafo G representa un canal de comunicaciones libre de errores y puede transmitir un bit durante cada uso del enlace. Los nodos 1 y 2 únicamente actúan como dispositivos que retransmiten la información que reciben.

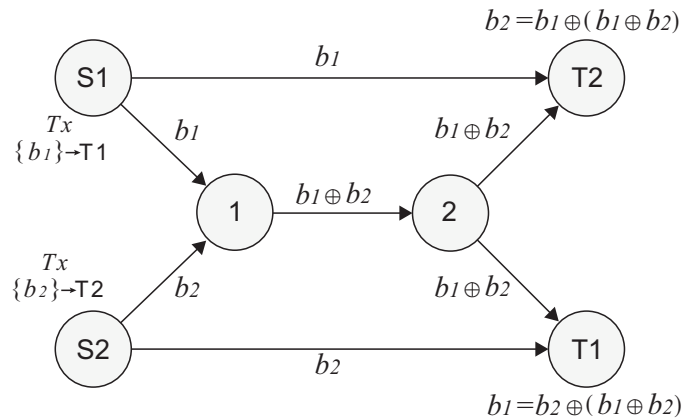


Figura 3.8: Transmisión de flujos *unicast* en una red con topología *mariposa*.

El nodo S1 transmite el bit b_1 por los enlaces $(S1 \rightarrow 1)$ y $(S1 \rightarrow T2)$. El nodo T2 ha recibido el bit b_1 y, aunque no es un bit destinado para él, debe almacenarlo para poder realizar el proceso de decodificación. En seguida, asuma que el nodo S2 transmite por los enlaces $(S2 \rightarrow 1)$ y $(S2 \rightarrow T1)$ el bit b_2 . En este caso, T1 recibe el bit b_2 y lo guarda para utilizarlo en el proceso de decodificación. El nodo 1, después de haber recibido los bits b_1 y b_2 , realiza el proceso de codificación empleando la operación lógica XOR, obteniendo como resultado el bit $b_1 \oplus b_2$. Este bit codificado es transmitido al nodo 2, y este nodo únicamente retransmite el bit codificado a los nodos T1 y T2. Los nodos T1 y T2 ahora pueden obtener los bits b_1 y b_2 empleando el proceso de decodificación mediante la operación lógica XOR. Es decir, el nodo T1 realiza la operación $b_2 \oplus (b_1 \oplus b_2)$ y el nodo T2 realiza la operación $b_1 \oplus (b_1 \oplus b_2)$. Finalmente, cada nodo ahora tiene la información original destinada para ellos, y puede descartar el bit auxiliar que utilizó en el proceso de decodificación.

3.4 Codificación de red en entornos inalámbricos

La codificación de red también puede ser utilizada en redes inalámbricas; particularmente por que la propiedad de difusión del medio inalámbrico resulta ser ideal para la aplicación de esta técnica. Al utilizar la codificación en redes inalámbricas (técnica conocida como *Wireless Network Coding* - WNC) se reducen el número

de transmisiones en el medio inalámbrico y, por consiguiente, se incrementa la capacidad de la red; además, se reduce la probabilidad de colisiones y la contienda por el medio [40]. Sin embargo, tal y como ha sido señalado por Katti et al. [7, 40], para que la codificación de red pueda operar en redes inalámbricas se debe satisfacer un conjunto de requerimientos importantes. El Lema 2 tiene por objetivo formalizar y sintetizar estos requerimientos.

Lema 2

Asuma que existe un nodo N ; dicho nodo recibe m paquetes para ser retransmitidos. Cada uno de los m paquetes ha sido transmitido por uno de los M nodos vecinos de N . Cada uno de los m paquetes se puede identificar como $P_{i,j}$, donde i identifica al nodo vecino que transmitió el paquete, y j identifica el nodo vecino hacia donde debe ser retransmitido dicho paquete. Asuma que \mathbf{P} es el conjunto de los m paquetes recibidos en el nodo N , es decir, $\mathbf{P} = \{P_{i,j} | \forall i, \forall j\}$; entonces, N puede crear un paquete codificado P_x utilizando un subconjunto \mathbf{K} de n paquetes (donde $\mathbf{K} \subseteq \mathbf{P}$ y $n \leq m$) si el nodo N puede verificar las siguientes dos condiciones.

1. Los n paquetes del subconjunto \mathbf{K} provienen de diferentes nodos vecinos y divergen en el siguiente salto. Esto significa que los índices i de todos los paquetes $P_{i,j} \in \mathbf{K}$ son diferentes y lo mismo debe cumplirse para todos los índices j .
2. Los j nodos que recibirán el paquete codificado P_x tienen información suficiente para decodificar dicho paquete y obtener así el paquete que está destinado para cada uno de ellos. Por ende, N debe tener la certeza que todos los nodos j han recibido, de algún modo, los paquetes contenidos en el conjunto \mathbf{K} excepto el paquete que deben recibir.

Si estas condiciones son verificadas por N , entonces se puede codificar y transmitir por el medio inalámbrico el paquete codificado P_x . Al transmitir este paquete codificado, N puede ahorrar $n - 1$ transmisiones en el medio; pues una sola transmisión del paquete P_x transportará la información contenida en n paquetes.

La Figuras 3.9 y 3.10 ilustran como pueden satisfacerse las condiciones estipuladas por el Lema 2 en una red inalámbrica. En este ejemplo, el nodo N es el responsable de realizar la técnica WNC en la red. El proceso de codificación se puede separar en tres etapas diferentes.

La primera etapa consiste en la recepción de paquetes en el nodo N . En la Figura 3.9 se muestra que el nodo N tiene como nodos vecinos a los nodos $1, 2, 3, \dots, M$. Cada uno de los M nodos vecinos transmite al nodo N un paquete para que este sea retransmitido hacia otro nodo vecino. El nodo N debe retransmitir los paquetes que recibe hacia sus respectivos destinos. Por ejemplo, en la imagen se aprecia que el nodo 1 transmite al nodo N un paquete destinado hacia el nodo 3, es decir, el paquete $P_{1,3}$; el nodo 4 transmite el paquete $P_{4,2}$, el nodo 2 el paquete $P_{2,4}$ y el nodo 3 transmite el paquete $P_{3,1}$. En la imagen también se muestra que cuando el nodo 1 transmite su paquete en el medio inalámbrico, el alcance de esta transmisión permite que N y otros nodos vecinos puedan recibir $P_{1,3}$, en este caso los nodos N , 4 y 2 pueden escuchar el paquete $P_{1,3}$. Este mismo tipo de *escucha oportunista* se da en el nodo 1 y en el nodo 3, que escuchan las transmisiones de los nodos 2 y 4, y en los nodos 2 y 4, que escuchan las transmisiones de los nodos 1 y 3. Los nodos que realizan la *escucha oportunista* deben almacenar una copia de los paquetes que transmiten sus nodos vecinos para poder realizar posteriormente el proceso de decodificación.

La segunda etapa consiste en seleccionar los paquetes con los cuales se realizará el proceso de codificación. Es decir, de los m paquetes recibidos, N debe seleccionar un conjunto de n paquetes con los cuales formar el paquete codificado P_x . La selección de los n paquetes se debe determinar considerando la información que han escuchado los nodos vecinos y si tienen los paquetes necesarios para decodificar P_x . En otras palabras, el nodo N debe conocer la información que han recibido previamente los posibles receptores del paquete codificado P_x . Con la *escucha oportunista* que realizó el nodo 1, éste ha podido escuchar en el medio inalámbrico la transmisión del nodo 2 al nodo N y también la transmisión del nodo 4 al nodo N , esto quiere decir que el

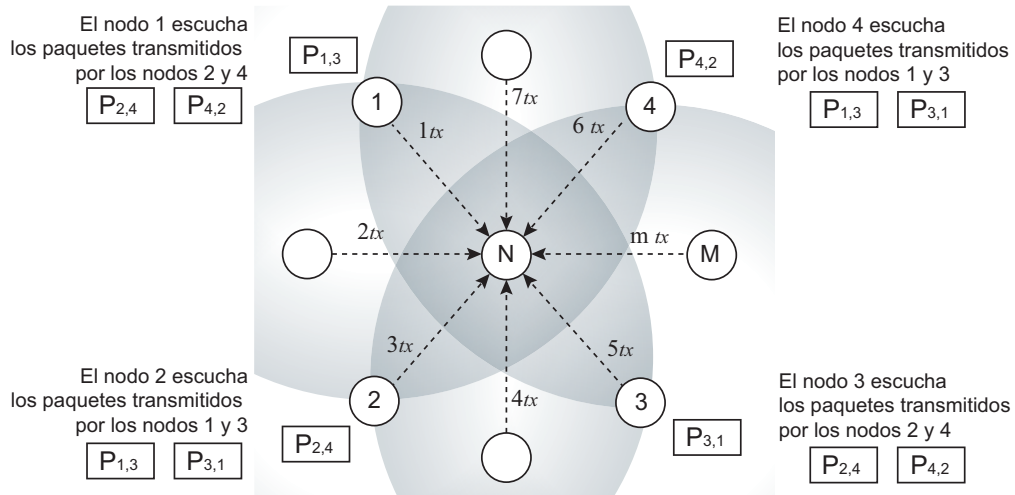


Figura 3.9: Los nodos de la red transmiten paquetes al nodo N y realizan *escucha oportunist* de los mensajes que se transmiten en el medio inalámbrico, almacenando una copia de los paquetes escuchados en el medio.

nodo 1 conoce el contenido de los paquetes $P_{1,3}$ (su propia transmisión al nodo N), $P_{2,4}$ y $P_{4,2}$. Tal como se muestra en la Figura 3.9, el nodo 2 ha escuchado las transmisiones de los paquetes $P_{1,3}$ y $P_{3,1}$, el nodo 3 ha escuchado las transmisiones de los paquetes $P_{2,4}$ y $P_{4,2}$ y el nodo 4 ha escuchado los paquetes $P_{1,3}$ y $P_{3,1}$. Es importante notar que ninguno de los nodos 1, 2, 3 y 4 ha recibido el paquete destinado para cada uno de ellos, y el único nodo que conoce el contenido de todos esos paquetes es el nodo N . El nodo N , al tener conocimiento del estado de la información recibida por los nodos de la red, forma el paquete codificado P_x utilizando la operación XOR entre los $n = 4$ paquetes seleccionados; es decir, $P_x = P_{1,3} \oplus P_{2,4} \oplus P_{3,1} \oplus P_{4,2}$ (este proceso de codificación se muestra en la Figura 3.10). Finalmente el nodo N transmite el paquete codificado P_x a los M nodos vecinos. **Nótese que** en lugar de que el nodo N realice 4 transmisiones inalámbricas en el medio para enviar cada uno de los paquetes $P_{1,3}$, $P_{2,4}$, $P_{3,1}$ y $P_{4,2}$ a sus respectivos destinos; ahora N **solo debe realizar una transmisión** inalámbrica para enviar simultáneamente el paquete codificado P_x a los nodos 1, 2, 3 y 4, tal y como se muestra en la Figura 3.10.

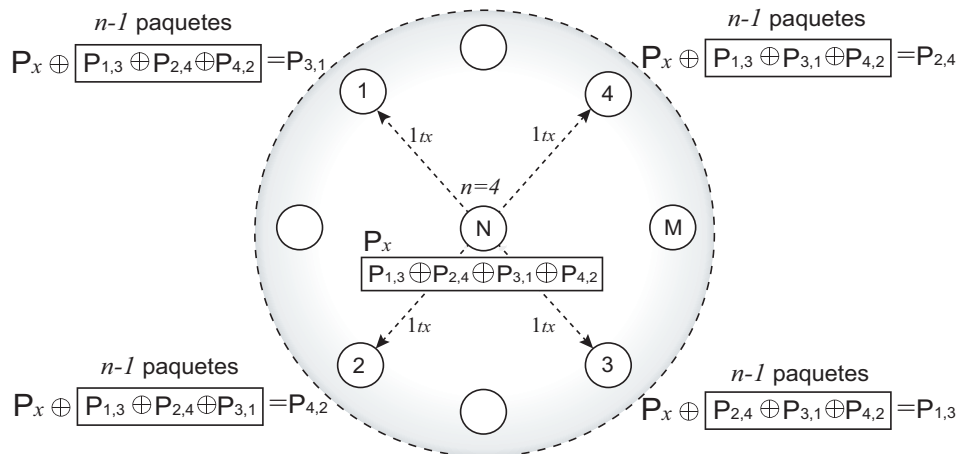


Figura 3.10: El nodo N codifica y transmite el paquete codificado P_x a los M nodos vecinos.

La tercera y última etapa es la decodificación del paquete P_x . Aquí los nodos que reciben el paquete codificado pueden decodificarlo empleando los $n - 1$ paquetes que obtuvieron previamente a través de la *escucha oportunist*. Por ejemplo, el nodo 1 realiza la operación $P_x \oplus P_{1,3} \oplus P_{2,4} \oplus P_{4,2}$ para obtener como resultado el paquete

$P_{3,1}$, tal y como se muestra en la Figura 3.10. La demostración algebraica del proceso de decodificación en el nodo 1 es la siguiente.

$$P_{3,1} = (P_x) \oplus (P_{1,3} \oplus P_{2,4} \oplus P_{4,2}), \quad (3.10)$$

dado que $P_x = P_{1,3} \oplus P_{2,4} \oplus P_{3,1} \oplus P_{4,2}$, entonces,

$$P_{3,1} = (P_{1,3} \oplus P_{2,4} \oplus P_{3,1} \oplus P_{4,2}) \oplus (P_{1,3} \oplus P_{2,4} \oplus P_{4,2}), \quad (3.11)$$

$$P_{3,1} = P_{3,1} \oplus (P_{1,3} \oplus P_{1,3}) \oplus (P_{2,4} \oplus P_{2,4}) \oplus (P_{4,2} \oplus P_{4,2}), \quad (3.12)$$

$$P_{3,1} = P_{3,1} \oplus (0) \oplus (0) \oplus (0), \quad (3.13)$$

$$P_{3,1} = P_{3,1} \quad (3.14)$$

El procedimiento de decodificación es realizado por los nodos que reciben el paquete codificado P_x , es decir, los nodos 1, 2, 3 y 4. Al decodificar el paquete, cada nodo obtiene el paquete original. Con esto se demuestra que la aplicación de WNC en una red inalámbrica ayuda a reducir el número de transmisiones en el medio y, a la vez, incrementa la capacidad de la red, se reduce la probabilidad de colisiones y también la contienda por el acceso al medio.

3.5 Codificación en redes inalámbricas *multihop*

Una red inalámbrica *multihop* está constituida por nodos inalámbricos capaces de cooperar entre sí para encaminar paquetes en la red. Gracias a esta capacidad de encaminamiento, cualquier par de nodos distantes pueden intercambiar datos, aun si se encuentran fuera de su alcance de transmisión. Es importante subrayar que, para poder establecer una ruta de comunicaciones entre un par de nodos en una red *multihop*, se debe emplear un protocolo de encaminamiento. El objetivo primario de estos protocolos es construir una ruta libre de bucles entre un nodo fuente y un nodo destino. Es decir, entre cualquier par de nodos se establece una cadena de nodos que de manera colectiva reciben y retransmiten paquetes desde el nodo fuente hasta el nodo destino. La Figura 3.11a muestra una red *multihop* en la que los nodos S y D desean intercambiar información. Se puede advertir que estos nodos están fuera de su alcance de transmisión, de manera tal que la única manera que tienen de comunicarse es con el apoyo de los otros nodos de la red.

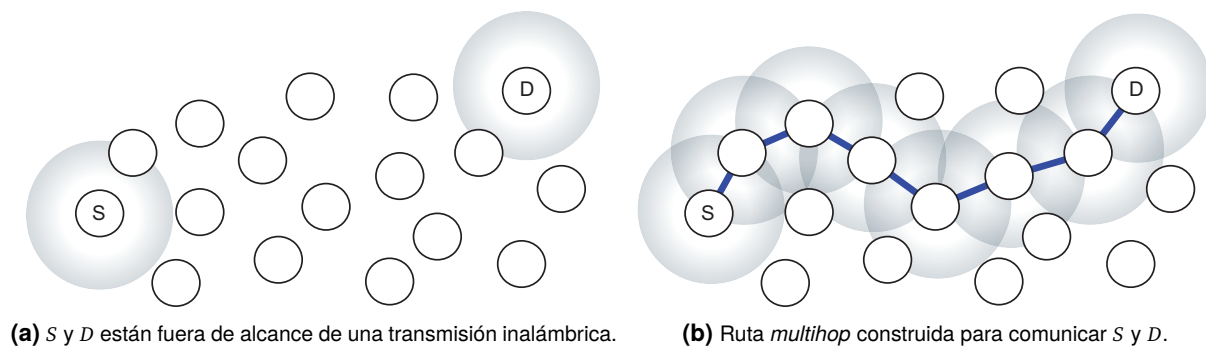


Figura 3.11: Establecimiento de una ruta libre de bucles en redes inalámbricas *multihop*.

Al emplear colectivamente un protocolo de encaminamiento, los nodos de la red construyen una trayectoria de comunicaciones entre S y D , tal como se muestra en la Figura 3.11b. Los nodos que forman la trayectoria de comunicación entre S y D son responsables de recibir información y retransmitirla al nodo siguiente (*next-hop*) de la cadena, este modelo de comunicaciones se conoce como *store-and-forward*. Es así como S y D pueden intercambiar información gracias al apoyo de los nodos de la red.

La técnica de codificación de red inalámbrica (WNC) puede implementarse fácilmente en una red inalámbrica *multihop*. Pues, como se ha mostrado, las trayectorias de comunicación en una red *multihop* son cadenas de nodos, y en este tipo de topologías se pueden satisfacer con facilidad los requerimientos planteados por el Lema 2. Para analizar los beneficios de la codificación en una red inalámbrica *multihop* se debe contrastar el esquema de comunicaciones tradicional *store-and-forward* y el esquema WNC. La Figura 3.12 contrasta estos dos modelos de comunicación.

Asuma que existe una cadena de nodos que colaboran para comunicar a los nodos S y D . En este ejemplo solo se analiza en una porción de la cadena. Esta porción incluye tres nodos, A , R y B , como se muestra en la Figura 3.12. El nodo A y el nodo B deben intercambiar los paquetes de datos P_1 y P_2 , respectivamente. El paquete P_1 pertenece flujo de datos que va de S a D y el paquete P_2 pertenece al flujo de datos que va de D a S . El nodo inalámbrico R colabora para que A y B puedan intercambiar dichos paquetes, pues A y B se encuentran fuera de su alcance de transmisión (**obsérvese que el análisis se puede reproducir a lo largo de toda la cadena de nodos**).

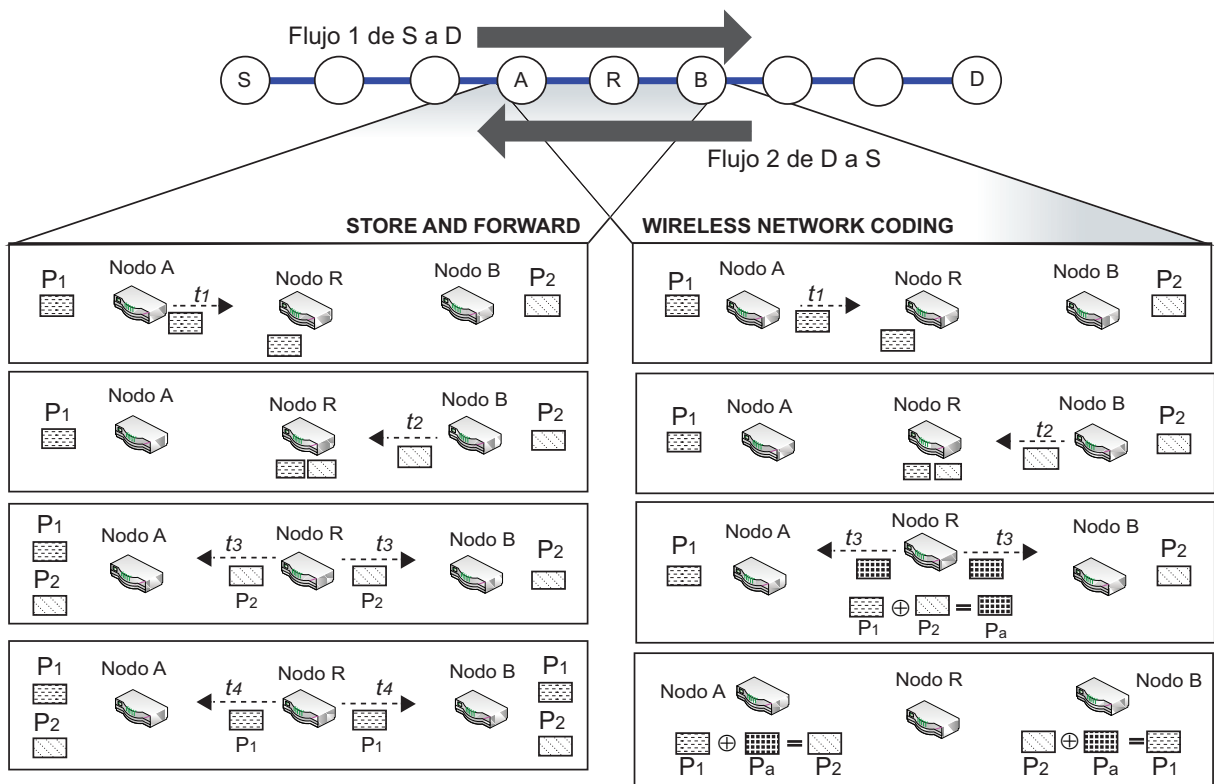


Figura 3.12: *Store-and-forward* vs codificación de red.

Al utilizar el modelo tradicional *store-and-forward*, el nodo A y el nodo B deben transmitir los paquetes P_1 y P_2 al nodo R . En la Figura 3.12 se muestran estas transmisiones como eventos que suceden en los instantes t_1 y t_2 . El nodo R recibirá estos paquetes y deberá determinar cuál es el siguiente nodo hacia donde deben ser transmitidos. En este caso, el paquete P_1 debe ser transmitido al nodo B , y el paquete P_2 debe ser transmitido hacia el nodo A . Así, el nodo R debe realizar dos transmisiones para enviar estos paquetes a sus respectivos nodos siguientes. En total se deben realizar cuatro transmisiones inalámbricas en el medio para completar el proceso de comunicación de dos paquetes entre A y B . Esto quiere decir que si A y B tuvieran que intercambiar N paquetes cada uno, entonces el número de transmisiones en el medio de comunicaciones compartido por A , B y R sería $4N$.

La naturaleza de difusión del medio inalámbrico permite utilizar WNC en este escenario. Básicamente, un nodo que debe encaminar dos flujos de datos que viajan en sentidos opuestos puede codificarlos y transmitirlos utilizando una sola transmisión inalámbrica, siempre y cuando los nodos receptores sean capaces de decodificar la información contenida en el paquete codificado. Retomando el ejemplo que se muestra en la Figura 3.12, al emplear WNC, el nodo R recibe los paquetes P_1 y P_2 en los instantes de tiempo t_1 y t_2 (nótese que estos dos paquetes viajan en sentidos opuestos). Posteriormente, el nodo R realiza una operación de codificación $P_1 \oplus P_2$; el paquete resultante, P_a , es transmitido simultáneamente a los nodos A y B en el instante t_3 . Finalmente, los nodos A y B efectúan un proceso de decodificación con el paquete recibido P_a y el paquete transmitido previamente. Por ejemplo, el nodo A había transmitido P_1 a R , con el paquete P_1 y el paquete recibido P_a , el nodo A puede realizar la operación $P_1 \oplus P_a$ lo que es igual a $P_1 \oplus (P_1 \oplus P_2) = P_2$. Por consiguiente, si se emplea WNC se utilizan solo 3 transmisiones en el medio en lugar de las cuatro que se utilizan con el modelo *store-and-forward*. Si A y B tuvieran que intercambiar N paquetes cada uno, entonces el número de transmisiones en el medio de comunicaciones compartido por A , B y R sería $3N$. Las N transmisiones que se ahorran con WNC pueden ser utilizadas para intercambiar más información entre A y B , es decir, los nodos A y B podrían transmitir cada uno hasta $N/3$ paquetes extra. De esta forma se aprecia que **la capacidad de esta topología de red se incrementa a 1.33 paquetes/por unidad de tiempo al utilizar WNC, lo cual representa una mejora del 33% con respecto al esquema *store-and-forward* [7].**

3.6 Beneficios de la técnica de codificación de red

La técnica de codificación de red es muy útil para mejorar diferentes características de las redes de computadoras. A continuación se resumen y se fundamentan los beneficios ofrecidos por la técnica de codificación de red.

- **Capacidad de la red:** Como se ha demostrado en este capítulo, uno de los mayores beneficios que ofrece la codificación de red consiste en aprovechar los enlaces de comunicaciones para transmitir más información de la que se puede transmitir con el esquema tradicional *store-and-forward*. Al transportar más información en los enlaces, se incrementa la capacidad de la red; de manera tal que la red puede transportar más información entre los nodos que la componen.
- **Robustez y resiliencia ante las pérdidas:** Un problema común en redes es la pérdida de paquetes causadas por numerosos factores como: sobrecarga en los *buffers* de recepción de los nodos, interferencia electromagnética en los enlaces o colisiones en los enlaces compartidos. Hay que tener en cuenta que la pérdida de paquetes tiende a agravarse en redes cuyos enlaces de comunicación tienen un alto índice de pérdidas, tal y como sucede en redes inalámbricas. La técnica de codificación de red incorpora robustez ante las pérdidas. Primero, porque reduce el número de transmisiones en el medio y, consecuentemente, la probabilidad de pérdidas. Segundo, debido a que al reducir el número de transmisiones, disminuye la contienda por el medio y la probabilidad de colisiones. Retome el ejemplo mostrado en la Figura 3.12 para ilustrar como la técnica de codificación de red puede disminuir la tasa de errores en una red inalámbrica. Asuma que el enlace inalámbrico entre el nodo A y el nodo R tiene una probabilidad de pérdidas de paquetes ε_{AR} , mientras que la probabilidad de pérdidas ε_{RB} está asociada al enlace que va del nodo R al nodo B . Por lo tanto, la probabilidad de que un paquete sea transmitido exitosamente del nodo A al nodo B será $(1 - \varepsilon_{AR})(1 - \varepsilon_{RB})$. Del mismo modo, la probabilidad de que un paquete sea transmitido exitosamente del nodo B al nodo A será $(1 - \varepsilon_{RB})(1 - \varepsilon_{AR})$; por lo tanto, la transmisión exitosa de dos paquetes es $[(1 - \varepsilon_{AR})(1 - \varepsilon_{RB})]^2$. Si $\varepsilon_{AR} = \varepsilon_{RB}$, entonces la probabilidad exitosa de dos paquetes, uno de A a B y el otro de B a A será igual a $(1 - \varepsilon_{AR})^4$. Se puede notar que al utilizar la técnica de codificación de red, se ahorra una transmisión en el medio y por lo tanto la probabilidad de error se reduce a $(1 - \varepsilon_{AR})^3$. La capacidad de adaptación ante los eventos de pérdidas se obtiene utilizando RLNC. Con este mecanismo de codificación se pueden tolerar pérdidas, y recuperar fácilmente la información original, pues el receptor solo necesita un conjunto mínimo de paquetes codificados para obtener la información original.

- **Seguridad:** La técnica de codificación de red puede ofrecer algunos beneficios en términos de seguridad. Por un lado, si un atacante en la red pudiese capturar paquetes codificados, por ejemplo, el paquete codificado $P_1 \oplus P_2$ en la Figura 3.12, éste no podría decodificar ninguno de los paquetes originales, ya que para decodificar la información contenida en el paquete codificado, se necesitaría conocer al menos uno de los paquetes nativos, es decir, el atacante debe conocer P_1 ó P_2 . Este esquema de protección es muy similar al modelo de llaves de cifrado que se emplea para proteger datos sensibles en sistemas de la información.

Trabajo relacionado

“La medida de la grandeza de una idea científica es en qué medida estimula el pensamiento y abre nuevas líneas de investigación.”

Paul Dirac

LAS iWMNs ofrecen un amplio abanico de beneficios cuando se emplean como redes de acceso inalámbrico a Internet; especialmente en aquellos escenarios donde el uso otras tecnologías es inviable. Gracias a su bajo costo de infraestructura y al uso de rutas *multihop*, las iWMNs son flexibles, fáciles de desplegar y permiten interconectar redes con tecnologías heterogéneas. Pese a estos beneficios, hay documentación que confirma que **el desempeño de las iWMNs puede resultar insuficiente para atender al creciente número de usuarios y a las aplicaciones que éstos utilizan**; debido a que, como se viene advirtiendo, las nuevas aplicaciones demandan altos requerimientos para su desempeño y ofrecer la experiencia de calidad que esperan los usuarios. Por ello, es importante: *i)* analizar cuáles son los diferentes **retos de desempeño que reducen las prestaciones de las iWMNs**, y *ii)* explorar aquellas experiencias que emplean la **técnica de codificación de red para mejorar el rendimiento de redes inalámbricas**.

4.1 Retos de desempeño

Las iWMNs enfrentan crecientes retos de desempeño que reducen negativamente la calidad de sus prestaciones y servicios que ofrecen a los usuarios. A continuación se revisan los retos de desempeño más importantes que enfrentan las redes inalámbricas y a la luz de ello las oportunidades que ofrece la **técnica de codificación de red** al emplearse en estas redes.

4.1.1 Escalabilidad de la red

A medida que una red inalámbrica *multihop* escala, es decir, cuando incrementa el número de nodos que la componen, pueden inducirse efectos adversos en el desempeño de la red. Uno de los más notorios es la reducción de la capacidad, definida como la cantidad de datos que la red puede transportar eficazmente por unidad de tiempo. Este problema puede afectar seriamente el servicio que proporciona la red y la calidad de la experiencia de los usuarios, porque la cantidad de datos que pueden transmitir por la red se reduce significativamente [43]. Gupta y Kumar [44] demuestran que: **la capacidad en redes inalámbricas multihop está acotada por el número de nodos que componen la red**. La razón es que cuando n nodos idénticos son posicionados aleatoriamente y con capacidad de transmitir W bits por segundo, la tasa de transmisión efectiva que puede alcanzar cada nodo es una función de n de orden $\Theta(W/\sqrt{n})$. Esto significa que **a medida que incrementa el número de nodos en la red, la capacidad de transmisión de cada nodo disminuye**. Este inconveniente es heredado por las iWMNs y, se incrementa a medida que la red escala, pues: *i)* se incrementa el número de saltos en las trayectorias de comunicaciones, *ii)* se incrementa la contienda por el acceso al medio,

así como el retardo de extremo a extremo y, por si fuera poco, también *iii*) se afecta el funcionamiento de los protocolos de encaminamiento [45, 46].

Las iWMNs tienen un comportamiento dual. Por un lado, la conectividad entre usuarios móviles (STAs) y los *Mesh Access Points* (MAPs) tiene un comportamiento idéntico al de las redes Wi-Fi (WLAN); y, por otro lado, el *backbone* de la red tiene un comportamiento similar al de una red multihop con nodos donde se centraliza el tráfico, es decir, en los MAPs (*Mesh Access Points*) y los MPs (*Mesh Portals*). Si en una iWMN se incrementa la densidad de nodos, la contienda por el medio aumenta, así como las colisiones y la interferencia. Estos factores reducen la capacidad de la red y afectan el funcionamiento de los protocolos que le dan soporte operativo, estos son principalmente los protocolos de acceso al medio y los de encaminamiento [4, 47, 48]. Así es que cuando una iWMN escala, es altamente probable que el tráfico también lo haga, de manera que puede afectarse la equidad con la que se reparten los recursos de la red e incrementar las pérdidas de paquetes por congestión [49].

4.1.2 Tecnología de transmisión

La velocidad y el alcance máximo de las transmisiones entre dispositivos inalámbricos dependen de la tecnología utilizada en la capa física. La evolución de las tecnologías de comunicación ha traído consigo mejoras importantes. La Tabla 4.1 detalla las principales características de los mecanismos de transmisión del estándar IEEE 802.11. Actualmente, el estándar IEEE 802.11ax ofrece tasas de transmisión superiores a 1Gbps, con límites máximos de transmisión que pueden alcanzar algunas decenas de metros. Existen otros ejemplos que ilustran la evolución de la tecnología de transmisión, tales como sistemas de antenas inteligentes [50, 51], sistemas de antenas coordinadas MIMO (*multiple input-output*) [52, 53], e incluso el uso de radios definidos por software [54]. Todas estas mejoras **requieren de ajustes en capas superiores de la pila de protocolos** (especialmente de los protocolos de la capa 2 y de la capa 3), lo que continúa siendo un reto importante de investigación [55, 56].

Protocolo	Frecuencia	Ancho de banda	MIMO	Tasa de transmisión
802.11ax	2.4 or 5GHz	20, 40, 80, 160MHz	Multi User (MU-MIMO)	2.4 Gbps ¹
802.11ac wave 2	5 GHz	20, 40, 80, 160MHz	Multi User (MU-MIMO)	1.73 Gbps ²
802.11ac wave 1	5 GHz	20, 40, 80MHz	Single User (SU-MIMO)	866.7 Mbps ²
802.11n	2.4 or 5 GHz	20, 40MHz	Single User (SU-MIMO)	450 Mbps ³
802.11g	2.4 GHz	20 MHz	N/A	54 Mbps
802.11a	5 GHz	20 MHz	N/A	54 Mbps
802.11b	2.4 GHz	20 MHz	N/A	11 Mbps
Legacy 802.11	2.4 GHz	20 MHz	N/A	2 Mbps

Tabla 4.1: Características de los mecanismos de transmisión del estándar IEEE 802.11.

4.1.3 Seguridad en la red

Las transmisiones inalámbricas se realizan sobre un medio de difusión, por esto es que la información transmitida puede recibirse por varios nodos, lo cual, de no atenderse, pone en riesgo la confidencialidad e integridad de la información [57]. Incluso, pueden ocurrir ataques que lleguen a dañar directamente las transmisiones inalámbricas (conocidos como *jamming*), o ataques de suplantación de identidad [58]. Por esto, la seguridad es un elemento clave para las iWMNs y puede ser considerado como un factor crítico por diferentes razones. **Primera**, una iWMN puede ser atacada en los MAPs, ya que estos nodos son los que reciben paquetes de los usuarios y, por lo tanto, son vulnerables; por lo tanto requieren políticas de seguridad que no solo restrinjan el acceso a la red, sino que, además, garanticen la autenticidad, confidencialidad, integridad y disponibilidad del servicio para los usuarios [59]. **Segunda**, el *backbone* de la una iWMNs debe garantizar que el tráfico de los

usuarios sea transportado de manera segura, garantizando que la información no sea capturada por atacantes que pretendieran actuar como nodos legítimos del *backbone* [60, 61]. Para poder satisfacer la seguridad de una iWMN se requiere: *i*) proteger el proceso de asociación y autenticación de los nodos del *backbone*, *ii*) proteger los mensajes de control y administración de la red para evitar que sean capturados y procesados por agentes maliciosos, *iii*) implementar mecanismos para la detección y prevención de intrusiones y, finalmente, *iv*) diseñar mecanismos de seguridad que sean compatibles con los ya definidos por la enmienda IEEE 802.11i, es decir, WPA y WPA2 [62].

4.1.4 Interacción con protocolos de la capa de transporte

TCP (*Transmission Control Protocol*) es el protocolo más importante de la capa de transporte. Actualmente, un alto porcentaje de las aplicaciones que envían y/o reciben datos por Internet utilizan los servicios que ofrece TCP. Este protocolo fue concebido considerando que operaría primordialmente en redes cableadas [63]. Desde ese entonces, y hasta la fecha, TCP ha funcionado eficientemente en redes cuyos enlaces tienen una baja tasa de errores a nivel de bit (BER), tal como sucede en redes cableadas. No obstante, existe evidencia documentada que exhibe el **pobre desempeño de TCP en redes inalámbricas** [34, 64, 65, 66]. Este pobre desempeño es ocasionado por diferentes factores que se listan a continuación.

- **Errores de transmisión** - En redes inalámbricas las pérdidas de paquetes pueden ser causadas por diferentes eventos, tales como: errores a nivel de bit, interferencias, colisiones, atenuación de la señal, entre otros. **TCP es incapaz de detectar la causa de las pérdidas en redes inalámbricas, y asume que toda pérdida es causada por congestión en la red.** Esto provoca que TCP reduzca la tasa de transmisión para resolver el problema de congestión. En redes inalámbricas esta acción correctiva conduce al desaprovechamiento del ancho de banda (pues TCP podría reducir innecesariamente la tasa de transmisión) y a interrupciones innecesarias de la transmisión [33].
- **Asimetrías** - TCP estima el ancho de banda disponible en la red utilizando acuses, los cuales indican que los paquetes de datos han llegado exitosamente al destino. Por consiguiente, TCP utiliza dos rutas de comunicación: *i*) la ruta por donde viajan los paquetes de datos desde la fuente hasta el destino, y *ii*) la ruta por donde se transmiten los acuses de recepción desde el destino hasta la fuente. Sin embargo, **en redes inalámbricas existen asimetrías que pueden provocar la incorrecta estimación del ancho de banda disponible**, dado que es altamente probable que la ruta de comunicaciones en una dirección tenga propiedades distintas a la ruta en sentido contrario.

En la Figura 4.1 se ilustra este problema. El nodo 1 es la fuente de información y establece una conexión TCP con el nodo 4. Entonces, se requiere de una trayectoria, $P_{1,4}$ (constituida por el conjunto de nodos v_1 y enlaces e_1) para la transmisión de datos de 1 a 4. Además, se necesita la trayectoria, $P_{4,1}$, (constituida por el conjunto de nodos v_2 y enlaces e_2) para la transmisión de acuses.

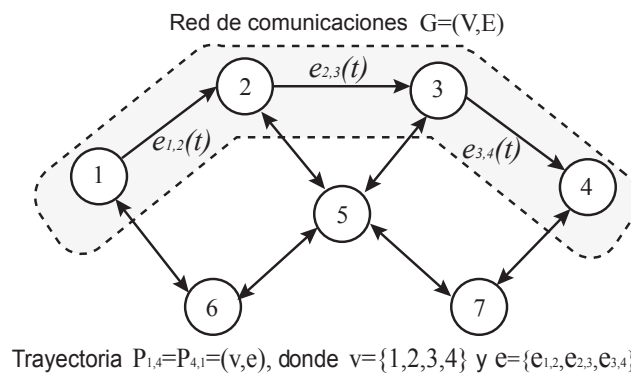


Figura 4.1: Trayectoria de comunicaciones entre los nodos 1 y 2 con propiedades dinámicas en los enlaces.

En redes inalámbricas las propiedades de los enlaces no solo son dinámicas, sino que el enlace que une a dos nodos podría tener **propiedades dinámicas distintas en la dirección contraria**. También, en redes inalámbricas se pueden tener rutas diferentes para la transmisión de datos y acuses ($P_{1,4} \neq P_{4,1}$), lo cual afecta la estimación del ancho de banda que realiza TCP [36].

- Terminal oculta y terminal expuesta** - El rendimiento de TCP es afectado por dos problemas comunes en redes inalámbricas: la terminal oculta y terminal expuesta. Estos dos problemas son provocados por dos propiedades de la tecnología de transmisión, el *alcance de transmisión* y el *alcance de detección de portadora*. El alcance de transmisión define la distancia mínima para realizar una transmisión exitosa entre dos nodos. En cambio, el alcance de detección de portadora es la distancia en la cual un nodo puede detectar las transmisiones que realizan otros nodos.

El problema de la **terminal oculta** es ocasionado cuando un nodo no detecta una transmisión en curso y decide transmitir en el medio, dando como resultado una colisión. En el ejemplo de la Figura 4.2, el nodo *A* escucha el medio y, dado que no detecta una transmisión en curso, decide transmitir al nodo *B*. Nótese que el nodo *A* está fuera del *alcance de detección de portadora* del nodo *C*, y, por lo tanto, *C* es incapaz de detectar la transmisión de *A* a *B*. Ahora, suponga que el nodo *C* quiere transmitir a *B*: como *C* no puede escuchar la transmisión de *A*, entonces transmite a *B*; causando una colisión en el nodo *B*, porque éste no puede recibir correctamente ninguna de las dos transmisiones. Es así como la **terminal oculta** reduce la capacidad de la red y la tasa de transmisión de TCP. Este es un problema difícil de erradicar, incluso si se emplean técnicas tales como la detección de portadora virtual [67, 68].

El problema de la **terminal expuesta** es ocasionado cuando un nodo transmite, y esta transmisión es detectada por otro nodo, gracias a que se encuentra dentro del *alcance de detección de portadora*. El nodo que desea transmitir debe retrasar su transmisión hasta que termina la que está en curso en el medio. Sin embargo, se dan condiciones en las que su transmisión no afectaría a la transmisión en curso. Considere el ejemplo que se muestra en la Figura 4.2. El nodo *D* inicia una transmisión al nodo *E*. Cuando el nodo *C* quiere transmitir al nodo *B*, no lo hace, dado que detecta la transmisión en curso del nodo *D*. En la imagen se muestra que la transmisión de *C* a *B* no interfiere con las comunicaciones de *D* a *E*, sin embargo, el protocolo de acceso al medio impide que se den estas transmisiones de manera simultánea; afectando la utilización del medio y limitando innecesariamente las transmisiones en el medio [69].

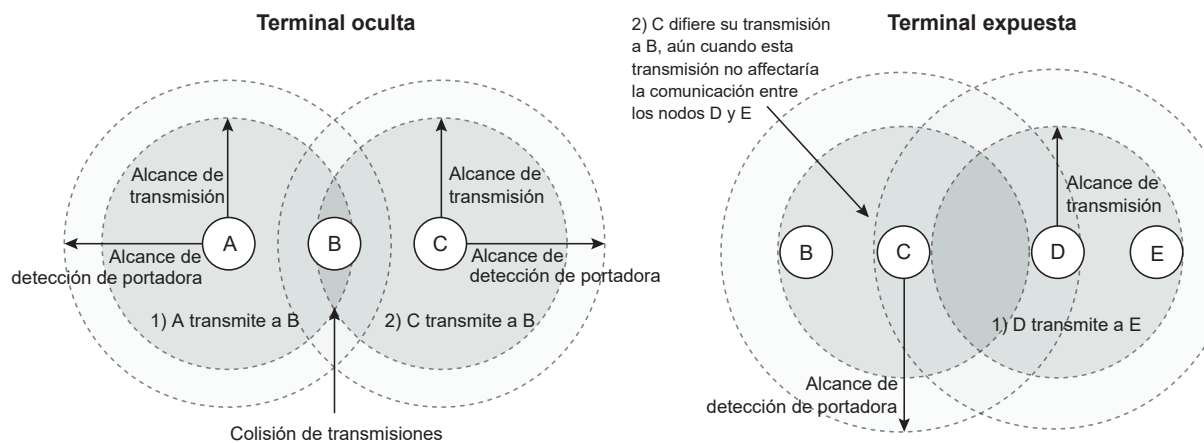


Figura 4.2: Problemas de terminal oculta (izq.) y terminal expuesta (der.).

- Inequidad** - La contienda entre varios flujos de datos transportados por TCP puede conducir a un acceso inequitativo al ancho de banda disponible en redes inalámbricas [70]. Este efecto negativo es ocasionado por dos factores: *i*) terminales ocultas y expuestas en entornos inalámbricos; y *ii*) el algoritmo de *backoff* exponencial de la capa MAC; además, un esquema de acceso aleatorio al medio, tal como CSMA/CA (*Carrier*

Sense Multiple Access with Collision Avoidance), que puede conducir a uno de los casos extremos de inequidad: la captura del medio [71, 72]. Estos dos problemas pueden ocurrir con mucha probabilidad en la operación de las iWMNs. Particularmente porque las *Mesh STAs* emplean el algoritmo de acceso aleatorio al medio EDCA. Además, la inequidad en iWMNs puede agravarse, ya que los usuarios más cercanos a los *Mesh Portals* (MPs) pueden capturar el ancho de banda disponible en la red [29].

- **Movilidad**- En redes inalámbricas los nodos tienen la libertad de moverse, lo cual podría alejar a dos nodos y causar que el enlace de comunicación entre estos desaparezca. Ahora bien, si los nodos son parte de una ruta de comunicación *multihop* activa, la ruta se divide e impide que los nodos que la usan puedan comunicarse. Si el protocolo de encaminamiento no es lo suficientemente rápido para crear una nueva ruta, entonces se degrada el desempeño de las conexiones TCP. Pues este protocolo detectará pérdidas de paquetes y reducirá la tasa de transmisión. De acuerdo con [34], un evento de partición de rutas, aún si es de corta duración, puede detener y afectar la tasa de transmisión de conexiones TCP por un largo periodo de tiempo. Esto se debe a que TCP detecta la reconexión una vez que comienza a recibir acuses; por lo tanto, la ausencia de estos acuses induce largos periodos de espera [36]. En iWMNs, la movilidad es una característica intrínseca de los usuarios de la red; por lo tanto, las particiones de rutas son comunes [73]. Esto afecta significativamente la calidad de la experiencia que perciben los usuarios debido a que la tasa de transmisión de TCP se interrumpe constantemente.
- **Contienda *inter-flow* e *intra-flow*** - La contienda *inter-flow* es percibida por un nodo inalámbrico cuando comparte el medio de comunicación con otros nodos y todos compiten por el acceso. En cambio, la contienda *intra-flow* es causada por la competencia de acceso al medio de paquetes de datos y acuses del mismo flujo TCP [74]. En ambos casos, el protocolo MAC define una política de acceso al medio que reduce las colisiones, de la siguiente manera: cuando un nodo quiere transmitir, debe escuchar el medio antes de hacerlo; si el medio está ocupado, ningún nodo puede transmitir para evitar colisiones, y deben esperar hasta que termine la transmisión en curso. Sin embargo, cuando el número de nodos que comparten el medio es alto, pueden experimentar un mayor retardo de acceso al medio. Provocando que el ancho de banda se distribuya entre los nodos en competencia; así, si los nodos que comparten el medio tienen una tasa de transmisión elevada, entonces la tasa efectiva de cada nodo se reduce considerablemente.

Estudios revelan que la ventana de congestión (*cwnd*) de TCP debe configurarse con un valor bajo para reducir los efectos negativos de una contienda agresiva entre nodos inalámbricos [75], dado que TCP no controla apropiadamente la ventana de congestión en entornos inalámbricos. En iWMNs, la contienda *inter-flow* se agrava por varias razones. Por un lado, en iWMNs existe altos niveles de contienda por el medio en ciertos nodos de la red; especialmente en los Mesh AP (MAP) –que son los nodos de acceso para los usuarios– y en los Mesh Portals (MP) –que son los nodos que conectan la iWMN con otras tecnologías de red–. Más aún, cuando los flujos TCP que viajan por una cadena de n nodos, experimentarán n contiendas por el medio, lo cual incrementa el retardo, afectando directamente la tasa de transmisión de TCP.

Actualmente tienen vigencia diversas propuestas con el propósito de mejorar el desempeño de las iWMNs; que van desde: la multicanalización del medio [76, 77], la redistribución del tráfico en la red [78, 79, 28] o bien el diseño de protocolos inter-capas (*Cross-Layer*) [56, 80]. En el contexto del presente proyecto de investigación se han identificado dos alternativas prometedoras que ayudarían a resolver varios de los problemas en iWMNs. Una de ellas consiste en utilizar la codificación de red como técnica para combinar paquetes de datos para incrementar la capacidad de una iWMN. Otra propuesta consiste en mejorar el funcionamiento de TCP utilizando la técnica de codificación de red, aprovechando los flujos bidireccionales que usa este protocolo de transporte. La siguiente sección presenta algunas de las propuestas más significativas de la literatura.

4.2 Codificación de red en entornos inalámbricos

Los usuarios móviles de una iWMN requieren enviar/recibir información de manera ágil y confiable hacia/desde Internet. Por lo que resulta imperativo que estos dispositivos utilicen un protocolo de transporte compatible con los protocolos de Internet. Por ello, TCP transportará un mayor volumen del tráfico en una iWMN. Sin

embargo, TCP tiene notorias limitaciones para su desempeño en redes inalámbricas. A lo largo de las últimas décadas han surgido diversas propuestas con el objetivo de mejorar el desempeño de TCP [81, 82, 83, 84, 85]. Como: i) modificar las funciones de TCP para adaptarlas a los entornos inalámbricos, ii) modificar el funcionamiento de la red para ofrecer mejores prestaciones y mejorar indirectamente el desempeño de TCP, y iii) modificar los protocolos subyacentes de la capa de transporte que permita adecuarse a entornos inalámbricos.

Es mejorando los servicios que brinda la infraestructura de red como se puede mejorar de manera indirecta el funcionamiento de TCP. Esta estrategia no violenta el principio de operación de extremo a extremo con la que opera dicho protocolo, y no se afecta la modularidad y funciones de ninguna de las capas de la pila de protocolos TCP/IP. Para mejorar la operación de la infraestructura de red no se requiere modificar el funcionamiento de los dispositivos finales. Por tanto, esta estrategia es fácil de implementar, operar y modificar. La codificación de red en entornos inalámbricos, WNC (*Wireless Network Coding*), es una estrategia sobresaliente que apunta en esa dirección [86, 87]. De acuerdo con [10], TCP puede mejorar su desempeño en redes inalámbricas gracias a las ventajas que ofrece WNC. Al utilizar WNC se pueden obtener beneficios en redes inalámbricas tales como: enmascarar los errores de transmisión en el medio, incrementar la información transportada en cada transmisión inalámbrica, reducir la contienda por el medio, optimizar el proceso de auto regulación de TCP, así como mejorar la equidad con la que se distribuyen los recursos de la red. Estas aseveraciones han sido documentadas e implementadas a través de diversas propuestas, a continuación se revisan algunas de ellas.

4.2.1 COPE

Uno de los trabajos con mayor reconocimiento es el que presentan Katti *et al.* en [7]. En este trabajo los autores proponen un mecanismo de codificación denominado como COPE. Este mecanismo identifica oportunidades de codificación, y, al encontrarlas, transmite múltiples paquetes nativos con una sola transmisión codificada. COPE determina inteligentemente la combinación de paquetes nativos que maximiza la utilización del medio; para esto, se utilizan tres métodos importantes.

- **Escucha Oportunista** (*Opportunistic listening*) - Cada nodo escucha y almacena todas las transmisiones que recibe durante un periodo de tiempo limitado, T . El tiempo T por defecto es de 500 ms. Así mismo, cada nodo envía *reception reports* para notificar a los nodos vecinos los paquetes almacenados que se han escuchado (estas notificaciones pueden ser transmitidas empleando paquetes de control).
- **Codificación Oportunista** (*Opportunistic Coding*) - Para maximizar la entrega de paquetes nativos a sus nodos vecinos utilizando una transmisión inalámbrica se deben conocer los paquetes que almacenan. Con esta información se puede determinar la mejor combinación de paquetes nativos que puede ser decodificada por la mayor cantidad de nodos. Esto significa que un nodo puede tener varias opciones de codificación, pero debe seleccionar aquella opción que maximice el número de paquetes nativos que pueden ser entregados utilizando una sola transmisión inalámbrica. La Figura 4.3 ilustra este concepto; el nodo B tiene en su fila de transmisión los paquetes $P1, P2, P3$ y $P4$; y debe seleccionar el esquema de codificación que ofrezca el máximo beneficio para el desempeño de la red. B tiene varias opciones de codificación, pero debe seleccionar la que brinda el mayor beneficio, en este caso $P1 \oplus P2 \oplus P3$.
- **Aprendizaje del Estado de los Vecinos** (*Learning Neighbor State*) - Este mecanismo permite que cada nodo transmita reportes a sus nodos vecinos en donde se indica los paquetes que se tienen almacenados para realizar el proceso de decodificación. El uso de *reception reports* puede fallar si existen retardos y errores en la transmisión, por ello también se aprovecha la información de encaminamiento para calcular el grado de eficiencia de los enlaces e inferir que paquetes han sido *escuchados* de manera oportunista.

COPE es evaluado utilizando una implementación en *hardware*. Los resultados muestran que la ganancia en capacidad puede variar mucho. Esta alta variabilidad depende de: el patrón de tráfico en la red, el nivel de congestión, la topología de la red y del protocolo de transporte utilizado. COPE funciona bajo el principio denominado *never delaying packets*: cada que un nodo obtiene acceso al medio transmite un paquete codificado, si puede crearlo, o bien un paquete nativo, si no tiene información suficiente para de crear un paquete

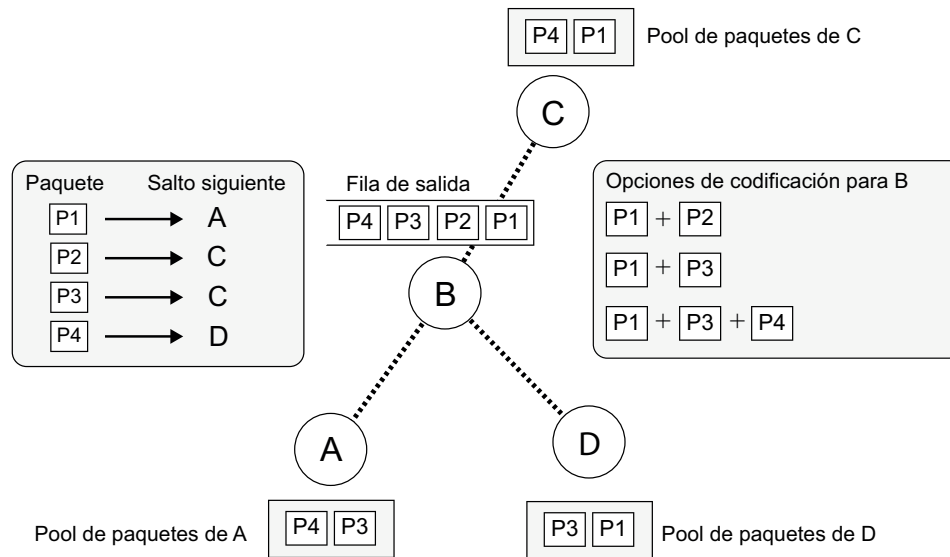


Figura 4.3: El nodo B, conociendo los paquetes que han recibido por los nodos vecinos, tiene varias opciones de codificación y debe elegir aquella que sea decodificable y que maximice la capacidad de la red.

codificado. Para implementar COPE se requieren tres propiedades para los nodos: *i*) memoria para almacenar los paquetes obtenidos al emplear *Opportunistic listening*, *ii*) antenas omnidireccionales para aprovechar la propiedad de difusión del medio y *iii*) no tener limitantes energéticas.

4.2.2 BEND

La codificación de red depende en gran medida de la convergencia de flujos para la creación de oportunidades; sin embargo, los protocolos de encaminamiento pueden limitar dichas oportunidades al crear rutas fijas con enlaces disjuntos. Para solucionar este problema, Zhang *et al.* [88] proponen un esquema de envío oportunista denominado BEND. Esta propuesta modifica dinámicamente y oportunamente el proceso de reexpedición de paquetes para maximizar las oportunidades de codificación (denominado como *Opportunistic forwarding*). Así se busca atender las necesidades de separación y concentración de tráfico dinámicamente. Para alcanzar este objetivo, BEND permite que cada nodo sea capaz de identificar paquetes fluyendo en sentidos contrarios (utilizando información de la capa de red) para conocer el segundo salto en la trayectoria de un paquete (2^{nd} next hop). Con esta información, cada nodo puede utilizar un criterio de codificación que asegura la decodificación en los nodos vecinos. BEND prioriza aleatoriamente paquetes codificados para transmitirse antes que la transmisión de paquetes nativos, con este mecanismo, la información codificada tiene mayores probabilidades de ser transmitida en el medio inalámbrico. Los resultados experimentales muestran que BEND puede superar la ganancia en capacidad que ofrece COPE hasta en un 30% en escenarios muy particulares, tales como redes jerarquizadas.

4.2.3 iCORE

En [89] se sintetizan los beneficios y dificultades que conlleva la codificación de red en redes WLAN. Se demuestra que la codificación de flujos *unicast* mejora sustancialmente la equidad con la que se distribuyen los recursos de la red, así como también la capacidad que se ofrece a cada usuario. Chieochan *et al.* proponen un esquema de codificación de red denominado iCORE (*interface COoperation Repeater-aided network coding Engine*). Esta propuesta utiliza la cooperación entre dispositivos con múltiples interfaces de comunicación para incrementar el alcance de las WLAN (*repeater-aided WLAN*). iCORE utiliza la codificación lineal *intra-flow* desde el nodo transmisor, combinado con un esquema de codificación *inter-flow* en los nodos que actúan como

dispositivos auxiliares. Los resultados demuestran que *iCORE* puede mejorar significativamente la capacidad de la red hasta en un 30%.

4.2.4 Reutilización espacial, adaptación dinámica de transmisión y codificación de red

Para explotar de mejor manera los recursos espectrales en WMNs, Ning *et al.* [90] proponen la combinación de técnicas tales como adaptación dinámica de tasa de transmisión, reutilización espacial del espectro y codificación de red. Esta propuesta incrementa el número de transmisiones simultáneas en el medio y la capacidad de la red. El método propuesto modifica la técnica de codificación de red para que identifique el conjunto de enlaces libres de interferencias y se utilicen concurrentemente. El modelo de optimización propuesto tiene una alta complejidad computacional; no obstante, se demuestra que la codificación de red puede combinarse, bajo ciertas condiciones, con otras estrategias para optimizar el desempeño de la red.

4.2.5 Oportunidades para codificación de red: esperar o no esperar

Hsu *et al.* [91] identifican los retos que enfrenta la técnica de codificación de red en entornos altamente dinámicos, ya que estos escenarios limitan la creación de oportunidades de codificación. Una forma en la que se pueden incrementar las oportunidades de codificación consiste en incorporar un retardo de transmisión a los paquetes. Esto significa que cuando no hay oportunidades de codificación los paquetes pueden ser retrasados hasta detectar la creación de una oportunidad. Los autores utilizan una política de control para reducir los costos asociados al retardo inducido para no afectar seriamente a las aplicaciones sensibles al retardo. Los autores formulan el problema como un MDP (*Markov Decision Process*) para derivar se él una política de optimización. El modelo planteado parte del escenario que se muestra en la Figura 4.4. Aquí, el nodo B es responsable de encaminar los flujos f_1 y f_2 . Los paquetes de estos flujos son almacenados en las filas de transmisión q_1 y q_2 respectivamente antes de ser retransmitidos. El modelo propuesto por los autores cuantifica el costo asociado a la decisión de retardar o no la transmisión de un paquete.

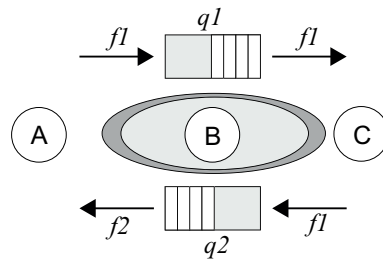


Figura 4.4: El nodo B encamina dos flujos de datos y puede aplicar una política de codificar paquetes o esperar por una oportunidad de codificación.

Si las filas q_1 y q_2 están vacías, el nodo B no debe realizar acción alguna, si ambas filas tienen paquetes entonces el nodo B puede transmitir un paquete codificado. El caso de interés emerge cuando una fila está vacía y la otra tiene paquetes almacenados. Los autores definen una *función costo* que considera el estado presente en las filas q_1 y q_2 , así como la acción tomada; es decir, transmitir o esperar. Los autores muestran que la política que minimiza la *función costo promedio* es aquella que toma en consideración umbrales que contabilizan el número de paquetes en las filas de transmisión. Además, los autores evalúan su propuesta a través de simulaciones numéricas y muestran que el modelo propuesto supera a estrategias tales como *Opportunistic Coding* (Codificación oportunista sin retardo), QLT (Queue-length Threshold), e incluso WT (Waiting time - tiempo fijo de retardo).

4.2.6 Fair Coding

El mecanismo de control de acceso al medio (MAC) es muy importante para la implementación de la codificación de red. En [92], Yang *et al.* demuestran que la inequidad de este mecanismo es un factor que reduce

las oportunidades de codificación, limitando significativamente los beneficios de la codificación de red. Los autores proponen un mecanismo denominado como *FairCodig* para atender este problema. Esta propuesta se basa en interacciones intercapa, y utiliza una aproximación heurística que busca balancear la probabilidad de acceso al medio entre diversos nodos para incrementar las oportunidades de codificación. El método busca que un nodo identifique los flujos con el menor número de paquetes en las filas de transmisión, a estos flujos se les asigna una mayor prioridad para acceder al medio y de este modo se balancea el tráfico y se incrementan el número de oportunidades de codificación.

La Figura 4.5 muestra el funcionamiento de *FairCoding*. En el ejemplo, el nodo B es responsable de encaminar dos flujos de datos, el flujo A→E y el flujo C→D. Si el flujo C→D genera una menor cantidad de tráfico que el flujo de A→E, entonces el nodo B ve reducidas las oportunidades de codificación al emplear el protocolo de acceso al medio IEEE 802.11 MAC, pues el nodo A transmite una mayor cantidad de paquetes a B y estos tienen pocas oportunidades de encontrar un paquete proveniente de C para ser codificados. *FairCoding* resuelve este problema asignando prioridades a los flujos de datos, propiciando que ciertos flujos tengan mayores probabilidades de acceder al medio. La capa IP es responsable de marcar flujos como *poor* o *rich*. Un flujo identificado como *rich* es un flujo con suficientes paquetes en espera de poder ser codificados y un flujo *poor* es un flujo sin los suficientes paquetes para ser codificados. Con esta información la capa MAC ajusta los parámetros de acceso al medio para cada flujo. La evaluación de *FairCoding* muestra que al balancear flujos se incrementan las oportunidades de codificación y se mejora la capacidad de la red hasta un 20%. No obstante, existe un compromiso de equidad con la que se distribuye el ancho de banda disponible en la red.

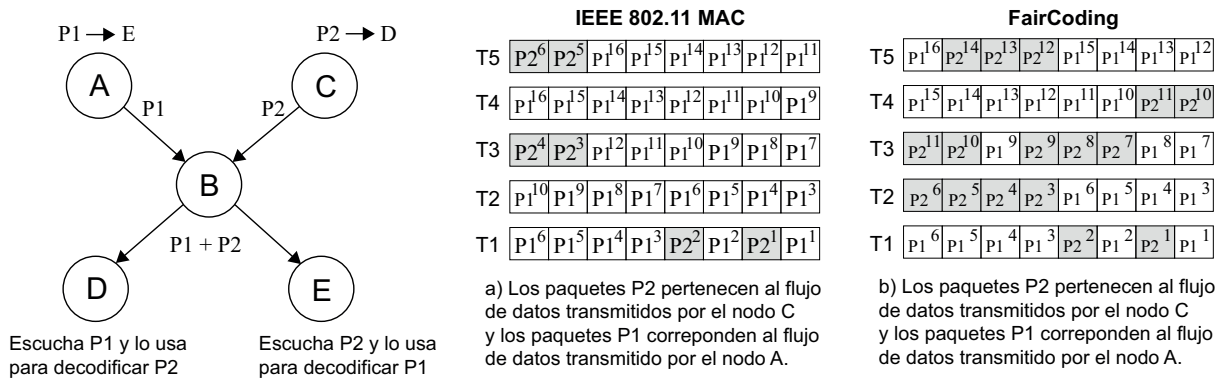


Figura 4.5: El nodo B encamina dos flujos de datos, de A a E y de C a D. De lado derecho se muestra la evolución de la fila de transmisión en el nodo B durante los periodos de tiempo de T1 a T5; por un lado, el comportamiento desbalanceado cuando se emplea el protocolo IEEE 802.11 MAC y, por otro lado, el comportamiento balanceado empleando FairCoding, incrementando así el número de oportunidades de codificación al priorizar el flujo de C a D.

4.2.7 SNC - Calendarización y codificación de red

El acceso aleatorio al medio y la discordancia entre las tasas de transmisión de diferentes flujos reducen la probabilidad de que se puedan formar oportunidades de codificación. Aunque la incorporación de un tiempo de espera puede ser una estrategia para incrementar las oportunidades de codificación, esta estrategia puede afectar al tráfico de aplicaciones sensibles al retardo. Liu *et al.* [93] proponen la operación conjunta entre la calendarización de transmisiones y la codificación de red para maximizar así las oportunidades de codificación para este tipo de tráfico. El mecanismo propuesto por los autores se denomina *Scheduling and Network Coding* (SNC) y consta de dos políticas principales. La primera es denominada como *Delayed Packet Transmission Policy* (DTP), la cual dinámicamente determina si se debe retrasar o no la transmisión de paquetes, y la segunda es denominada como *Network Coding based Scheduling Policy* (GSP), la cual toma como referencia los resultados obtenidos por DTP y los requerimientos de retardo de los flujos de aplicaciones en tiempo real con el objetivo de asignar un peso a cada NCG (Network Coding Group) y calendariza la transmisión de paquetes codificados con base en el peso asignado.

La política DTP es la responsable de determinar si un conjunto de paquetes pertenecientes a un NCG debe ser transmitido en su forma nativa, o bien si deben ser retrasados para esperar una oportunidad de codificación. Modelando el problema como un Proceso de Decisión de Markov (*Markov Decision Process*) los autores encuentran la acción óptima que maximiza el set de recompensas del modelo. Finalmente, los autores finalizan el problema estableciendo todo un algoritmo que cumpla con el resultado del modelo. Los resultados de ambas políticas muestran que se SNC incrementa el número promedio del grado de codificación (*average encoding degree*) y bajo condiciones controladas en la topología, el tipo de tráfico y la tasa de transmisión se puede obtener hasta un incremento del 24% de *throughput* de la red.

4.2.8 Análisis de COPE en redes inalámbricas *multihop*

COPE resulta ser una propuesta para aplicar la técnica de codificación de red en entornos inalámbricos y mejorar así la capacidad de la red. En [94], Chi *et al.* proponen un marco teórico que permita analizar de manera teórica los beneficios que puede ofrecer COPE en redes inalámbricas. Los autores desarrollan un estudio a nivel de nodo para contrastar los beneficios que ofrece COPE contra el encaminamiento tradicional denominado como *store-and-forward*. COPE mantiene un sistema de filas virtuales de espera como el que se muestra en la Figura 4.6. El modelo propuesto por los autores parte de la derivación de la PDF (*Probability Density Function*) de la probabilidad de que un nodo pueda escuchar las transmisiones de dos nodos vecinos. El modelo propuesto por los autores aproxima el PER (*Packet Error Rate*) con la probabilidad de que el SNR (*Signal-Noise Ratio*) sea menor a un umbral definido como γT . De manera que el PER puede expresarse como:

$$\text{PER} = 1 - \exp\left(-\frac{\gamma T}{K} d^\alpha\right)$$

Donde d es la distancia del enlace entre dos nodos y K es una constante que depende de los parámetros físicos de la interfaz de comunicaciones tales como la potencia de transmisión, la ganancia de la antena, etcétera. Considerando esta derivación del PER los autores estiman geoméricamente la PDF de $p_{i,j}$ que es la probabilidad de entrega de del paquete P_j de un nodo vecino previo al nodo N_i , la cual es calculada periódicamente por el nodo N_i y difundido a todos los nodos de la red.

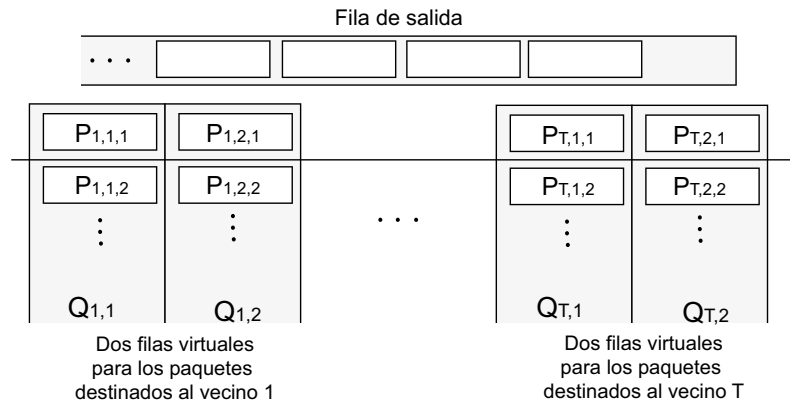


Figura 4.6: Cada nodo tiene pares de filas virtuales de codificación donde se almacenan los paquetes que van destinados a un nodo vecino.

Tomando como base la PDF obtenida, los autores calculan la probabilidad de codificar múltiples paquetes en un nodo en el escenario en el que los nodos vecinos son posicionados aleatoriamente. Los autores muestran el cálculo de probabilidades para hasta 5 flujos codificables en un nodo. A través de simulaciones numéricas los autores estiman que, para obtener un beneficio sustancial al utilizar COPE, se deben tomar en cuenta parámetros tales como el umbral de codificación, la distancia radial de transmisión R del nodo codificador y la probabilidad de transmisión. A través de diferentes parámetros en el modelo, los autores muestran que la codificación de red puede ofrecer una mejora de hasta el 50% de la tasa de transmisión promedio.

4.2.9 Análisis de la interacción de la codificación de red con IEEE 802.11 DCF

Una contribución primordialmente orientada a analizar los beneficios de la codificación de red en entornos inalámbricos junto con la interacción con la Función de Coordinación Distribuida (IEEE 802.11 DCF) es presentada en [95]. Los autores argumentan que los protocolos de red, especialmente los protocolos de control de acceso al medio (MAC) y los mecanismos de la capa física (PHY), tienen un efecto directo sobre los beneficios que ofrece la técnica de codificación de red. Además, los autores enfatizan que muchos de los trabajos existentes asumen simplificaciones que impiden establecer claramente los beneficios de la codificación; por esa razón, los autores proponen un modelo analítico empleando teoría de filas y lo aplican en un escenario con flujos *unicast* bidireccionales en una red inalámbrica *multihop* en una topología de cadena con k nodos. El modelo propuesto tiene por objetivo evaluar no solo los beneficios del *throughput* en la red, sino también el retardo de extremo a extremo. El modelo toma en consideración especificaciones MAC/PHY del estándar IEEE 802.11 y los posibles eventos asociados a las transmisiones inalámbricas bajo CSMA/CA, tales como las retransmisiones, la probabilidad de colisión, la calidad de los enlaces, el algoritmo de *back-off*, entre otros. El modelo propuesto considera a cada nodo que compone la red como una fila de espera donde los paquetes almacenados en el *buffer* de transmisión son considerados como los clientes de la fila. Por otro lado, dentro de la topología en cadena el nodo N_i tienen una probabilidad de transmisión exitosa al nodo N_j definida como:

$$p_{i,j} = (1 - C_{i,j})(1 - p_e \times L_p). \quad (4.1)$$

Donde p_e es la tasa de error de bit, $C_{i,j}$ es la tasa de error o colisión de paquetes y L_p es la longitud del paquete transmitido por el enlace. Tomando en consideración que existe un conjunto de nodos I_j cuyas transmisiones interfieren las transmisiones del nodo N_i al nodo N_j , entonces la probabilidad de transmitir exitosamente del nodo N_i al nodo N_j se puede estimar tomando en consideración la probabilidad de transmisión h_x de un nodo N_x perteneciente a I_j quedando como:

$$p_{i,j} = (1 - p_e L_p) \prod_{N_x \in I_x - \{N_i\}} (1 - h_x). \quad (4.2)$$

El modelo mostrado en la Figura 4.7 ilustra el funcionamiento del módulo de codificación. Cuando el nodo N_i recibe paquetes codificados son procesados por un bloque de decodificación (*decoder*), responsable de decodificar los paquetes y determinar el *next-hop* de los paquetes nativos que los conforman. En seguida, los paquetes nativos pasan por el bloque de codificación (*encoder*), responsable que recibir paquetes nativos y determinar si pueden ser codificados o bien si deben ser transmitidos en su forma nativa. El *encoder* pasa los paquetes codificados o nativos a sus respectivas filas de espera (Q_{native} y Q_{coded}), cada una con su respectiva tasa de arribos y tasa de servicio. Finalmente, el módulo *forwarder* es el responsable de tomar los paquetes almacenados en las filas y transmitirlos.

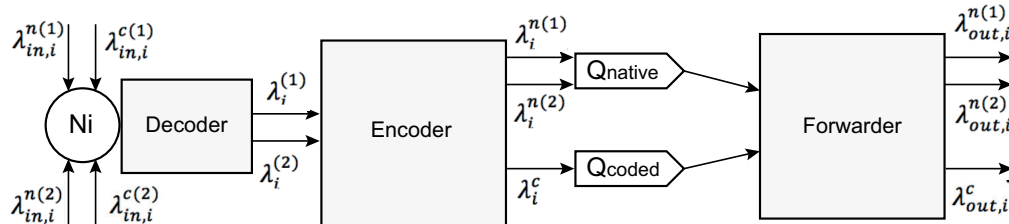


Figura 4.7: Modelo de recepción y transmisión de paquetes nativos y codificados en un nodo de la red.

4.2.10 MPOF - Múltiples paquetes un solo flujo

Un punto de acceso (AP - *Access Point*) es un dispositivo que fácilmente puede convertirse en cuello de botella y consecuentemente limitar la capacidad de la red inalámbrica de área local (WLAN - *Wireless Local Area*

Network). Cuando dos nodos de una WLAN quieren comunicarse entre sí, deben de utilizar los servicios de encaminamiento provistos por el AP. De acuerdo con Chi *et al.* [96], en este escenario la codificación de red podría mejorar el desempeño de las redes WLAN. Sin embargo, si los tamaños de los paquetes que se codifican son diferentes, no se podría obtener un beneficio sustancial de esta técnica, ni explotar eficientemente la capacidad de la red. De acuerdo con los autores, las soluciones más comunes son aquellas que toman un paquete perteneciente a un flujo de datos para codificarlo con un paquete perteneciente a otro flujo de datos, esta técnica se denomina comúnmente como OPOF (*one-packet-one-flow*).

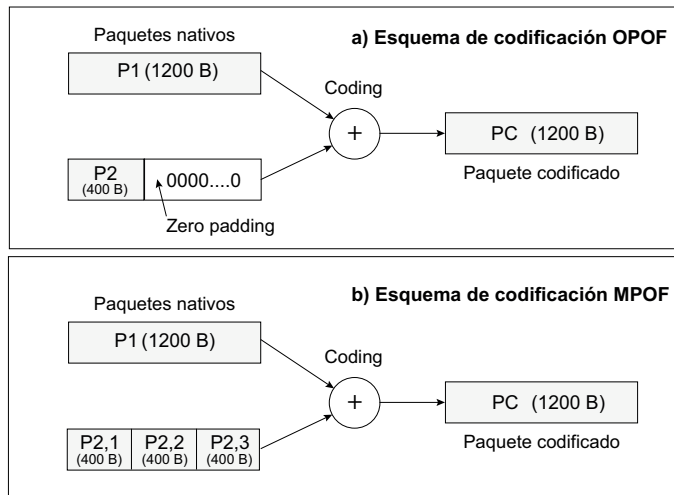


Figura 4.8: a) Esquema de codificación OPOF con paquetes nativos de diferentes tamaños y utilizando *zero-padding* para igualarlos. b) Combinación de paquetes para igualar el tamaño de paquetes nativos y evitar usar *zero-padding*.

Una limitante de la técnica OPOF se ilustra en la Figura 4.8, en esta imagen se muestra que puede darse el caso de que se codifiquen paquetes nativos con distintos tamaños. En ese caso, el paquete nativo de menor tamaño es rellenado con ceros (técnica conocida como *zero-padding*) para que iguale el tamaño del otro paquete nativo. Si el paquete nativo P_1 tiene un tamaño de 1200 bytes y el paquete nativo P_2 tiene 400 bytes, al codificarlos se desperdician 800 bytes por utilizar *zero padding*. La propuesta de los autores consiste en utilizar un mayor número de paquetes pertenecientes a un mismo flujo de datos para igualar el tamaño de los paquetes nativos que participaran en la codificación. En la Figura 4.8 se muestra como en lugar de usar *zero-padding*, el paquete nativo de menor tamaño, $P_{2,1}$ se rellena con los paquetes nativos $P_{2,2}$ y $P_{2,3}$ para igualar el tamaño del paquete nativo P_1 y de esta manera maximizar los beneficios de la codificación de red. Esta estrategia es denominada por los autores como MPOF (*Multiple Packet - One Flow*). Los autores presentan un algoritmo heurístico para encontrar la codificación óptima de paquetes. Dicha solución busca maximizar la diferencia ΔT , la cual es igual al tiempo de transmisión de los paquetes de forma nativa T menos el tiempo que tomaría la transmisión de los mismos paquetes al ser codificados T_{nc} , es decir, $\Delta T = T - T_{nc}$. La simulación del algoritmo MPOF se realiza en un escenario con un AP que recibe Q flujos de datos. Los resultados muestran que al incrementar el número de flujos Q se puede obtener una ganancia de hasta el 46% en el *throughput* de una WLAN si se emplea MPOF.

4.3 Codificación de flujos TCP

TCP es el protocolo más utilizado en redes de datos. Las características del tráfico transportado por este protocolo son apropiadas para utilizar la técnica de codificación de red, ya que una conexión TCP requiere dos flujos bidireccionales; el primero, el flujo de datos (TCP-DATA) y el segundo, el flujo de acuses de recepción (TCP-ACK). Ambos flujos viajan en direcciones opuestas, lo cual representa una oportunidad interesante para explotar la técnica de codificación de red en los nodos intermedios de la trayectoria de comunicaciones entre

el nodo fuente y el nodo destino. A continuación, se muestran algunas propuestas de codificación de red cuyo objetivo es mejorar las prestaciones de redes inalámbricas para este tipo de flujos de datos.

4.3.1 Piggy Code

Gerla *et al.* presentan en [97] un esquema de codificación para los flujos bidireccionales TCP, el mecanismo es denominado *PiggyCode*. Con este mecanismo se codifican paquetes nativos de datos (TCP-DATA) junto con paquetes nativos de acuse (TCP-ACK) pertenecientes a un mismo flujo TCP. El objetivo al que apunta esta propuesta es reducir el tiempo de entrega de paquetes TCP-ACK para mejorar el desempeño de TCP y al mismo tiempo reducir la competencia por el medio de inalámbrico. *PiggyCode* debe interactuar con el protocolo de encaminamiento para asegurar que los paquetes nativos TCP-DATA y los paquetes nativos TCP-ACK se transporten por la misma secuencia de nodos intermedios entre fuente y destino, y así poder crear oportunidades de codificación en los nodos intermedios de la trayectoria. Este esquema aprovecha el medio de difusión y propone que cada nodo escuche oportunamente en su alcance de detección de portadora. Utilizando *PiggyCode*, los autores aseguran que la probabilidad de obtener una oportunidad de decodificación es igual a 1, debido a que cada nodo siempre recibirá un paquete que ha sido codificado con al menos un paquete nativo almacenado en su *buffer* de decodificación.

Cuando el algoritmo *PiggyCode* alcanza un punto de estabilidad, queda asegurado que cada nodo intermedio en la ruta de comunicaciones recibirá dos paquetes codificados TCP DATA \oplus TCP ACK –uno con dirección hacia el nodo destino y otro con dirección hacia el nodo fuente–. Estos paquetes codificados deben ser decodificados e intercambiada su información para generar un nuevo paquete codificado, TCP DATA \oplus TCP ACK. De este modo, con la transmisión de un paquete codificado, se transmite simultáneamente el paquete TCP-ACK dirigido al nodo fuente y el paquete TCP-DATA camino al nodo destino.

La Figura 4.9 muestra la arquitectura propuesta para implementar *PiggyCode* en un nodo inalámbrico. En la imagen se muestra la incorporación de una capa intermedia entre la capa de red y la capa MAC, esta capa es denominada como la Capa de Codificación de Red (*Network Coding Layer*). El funcionamiento de *PiggyCode* es de la siguiente manera. Cuando un nodo recibe un paquete y debe retransmitirlo, una copia de este paquete de ser almacenada en el *buffer* de decodificación. Posteriormente, debe buscar en la fila de transmisión si es posible realizar la codificación con algún paquete almacenado en ella (*TX Queue*). Para realizar el proceso de decodificación, el nodo utiliza un módulo *sniffer* que envía los paquetes escuchados al buffer de decodificación. El módulo *decodificador* es el responsable de recibir paquetes, si se recibe un paquete codificado, el módulo busca en el *buffer* de decodificación los paquetes necesarios para decodificar el paquete. Los resultados planeados por los autores demuestran una ganancia en la capacidad de la red del 17%, junto con una reducción significativa del número de ACKs reexpedidos por la red.

4.3.2 Code MP

En [98], Chen *et al.* proponen *CodeMP*, es una adaptación del trabajo presentado en [97] pero ajustado especialmente para el encaminamiento multi-trayectoria y adecuado para escenarios disruptivos. En escenarios disruptivos, TCP enfrenta numerosos retos debido a la frecuente ruptura de rutas de comunicación, a la variabilidad de errores en la transmisión, y a las constantes pérdidas de paquetes. *CodeMP* fue diseñado con el objetivo de mitigar las pérdidas aleatorias utilizando la codificación *intra-flow* de paquetes, entendida esta última como la codificación sobre paquetes nativos pertenecientes a un mismo flujo TCP. *CodeMP* consta de tres mecanismos fundamentales, i) un mecanismo de codificación lineal aleatoria, ii) un mecanismo de encaminamiento multi-trayectoria y iii) un mecanismo para la codificación de paquetes nativos TCP-DATA y TCP-ACK, tal como es presentado en [97] e ilustrado en la Figura 4.10. Con esto, *CodeMP* adquiere la capacidad de adaptarse a los cambios frecuentes de la calidad de los enlaces inalámbricos y sin necesitar de mensajes de control explícitos. *CodeMP* brinda una solución para mitigar las pérdidas aleatorias de paquetes al utilizar codificación *inter-flow* y, además, reduce el número de transmisiones en el medio y la interferencia. *CodeMP* asocia a cada paquete TCP-DATA, en el *buffer* de codificación, un temporizador T denominado como *Piggy coding timer*. Si un paquete TCP-ACK es recibido antes de que el temporizador T expire, dicho paquete se codifica con el paquete TCP-DATA. *CodeMP* también agrega redundancia para atender las pérdidas en la

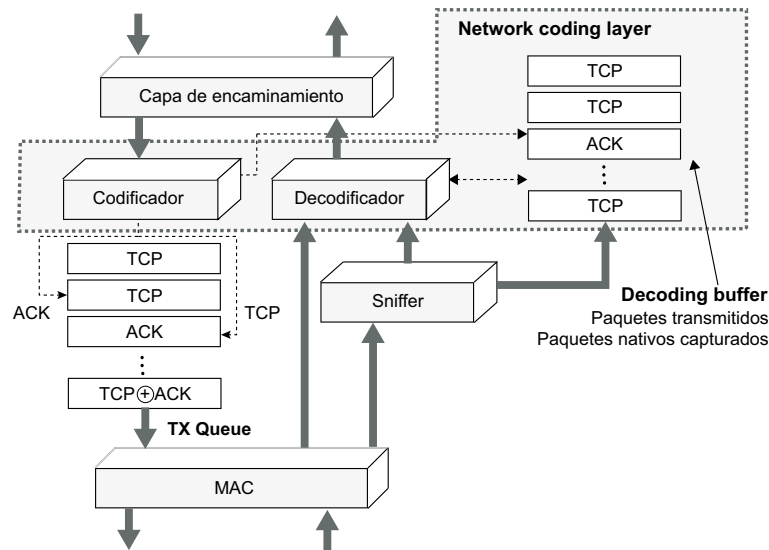


Figura 4.9: Arquitectura de un nodo inalámbrico con las funcionalidades de codificación de red definidas por Piggy-Code.

red. Esta redundancia se adapta a los cambios en la red utilizando un algoritmo de estimación heurístico. El algoritmo agrupa los nodos en grupos denominados como *virtual slices*. Cada nodo se asocia a un grupo considerando la distancia que existe entre los nodos y la fuente de datos. Una vez agrupados los nodos se estima las condiciones de la red para ajustar la redundancia necesaria para cada *virtual slice* tomando como referencia el número de transmisiones exitosas a dicho grupo. Los resultados muestran que *CodeMP* mejora la capacidad de la red en entornos con un alto PER (*Packet Error Rate*), sin embargo, una observación importante de los resultados presentados es que *CodeMP* no se adapta eficientemente a condiciones ideales en la red y ofrece un menor desempeño ante la opción de no codificar. No obstante, *CodeMP* ofrece un buen desempeño aún en escenarios altamente dinámicos, el caso más destacado se presenta con un PER del 40% y dos sesiones TCP en competencia, en este escenario la propuesta de codificación alcanza a obtener una capacidad agregada de 700 Kbps y un índice de equidad cercano a 0.99.

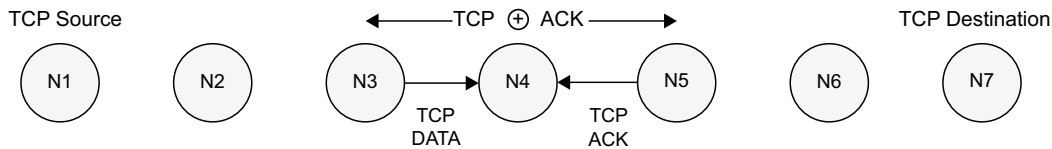


Figura 4.10: Codificación de paquetes TCP DATA y TCP ACK en los nodos intermedios de la trayectoria *multihop*.

4.3.3 Combo Coding

Chen *et al.* presentan en [99] un esquema de codificación de red denominado *ComboCoding*. Esta propuesta codifica paquetes en flujos bidireccionales (TCP DATA + TCP ACK), véase Figura 4.10, agregando un esquema adaptativo de redundancia de codificación. *ComboCoding* resuelve dos problemas importantes de TCP en redes inalámbricas; el problema de auto interferencia entre los paquetes nativos TCP-DATA y los acusos TCP-ACK, y el problema de las pérdidas aleatorias. La idea fundamental de *ComboCoding* consiste en utilizar la técnica de codificación *inter-flow*, es decir, codificación de distintos flujos de datos encaminados en direcciones opuestas, combinada con la técnica de codificación *intra-flow*, es decir, codificación lineal aleatoria únicamente de los paquetes TCP-DATA de un mismo flujo de datos. *ComboCoding* integra, además, un mecanismo adaptativo que

se basa en la estimación de la tasa de pérdidas en cada enlace, para calcular la redundancia de codificación. Los resultados muestran que los flujos TCP que utilizan el esquema de codificación *ComboCoding* obtienen una mayor capacidad en redes que tienen una alta probabilidad de errores de transmisión (entre 0.2 y 0.4); en el mejor de los casos, la mejora va de 200 Kbps sin emplear codificación de red, llegando a 2 Mbps utilizando *ComboCoding*. Además, *ComboCoding* ofrece un alto índice de equidad para escenarios multiflujo, aún en altas condiciones de errores en la red.

4.3.4 TCP Fender

En [100, 101], Zhang *et al.* argumentan que en entornos en donde existen fluctuaciones en la calidad de los enlaces inalámbricos, la codificación de red puede ser útil para mejorar el desempeño de la red. Sin embargo, el comportamiento de TCP puede verse afectado seriamente debido a la pérdida de paquetes y al arribo desordenado de paquetes en el nodo destino. Consecuentemente, TCP no podría operar eficientemente aún si se emplea la codificación de red en la infraestructura de comunicaciones. Los autores enfatizan que la codificación de red no es una estrategia eficiente cuando se combina con tráfico TCP, especialmente por el retardo que esta técnica induce y por los posibles casos donde se pierdan paquetes nativos debido a la imposibilidad de decodificar paquetes codificados. Por tal motivo es que proponen el mecanismo denominado *TCPFender*. Dicho mecanismo está diseñado para que el control de congestión de TCP funcione eficientemente en una red inalámbrica que presente fluctuaciones en la calidad de los enlaces y en donde opera el mecanismo de codificación de red. El mecanismo propuesto, es una capa intermedia entre la capa de transporte y la capa de red, véase la Figura 4.11. Los módulos propuestos operan en los hosts fuente y destino respectivamente, así como en los nodos intermedios de la trayectoria *multihop*. El objetivo de *TCPFender* es que **el host transmisor mantenga su ventana de congestión abierta incluso cuando hay eventos de pérdidas de paquetes o arribos desordenados en el destino**. Para conservar la ventana de congestión abierta, el módulo receptor puede enviar de manera anticipada TCP-ACKs para que informe al módulo transmisor que ha recibido paquetes TCP-DATA, aun cuando estos no se hayan decodificado. La capa de *TCPFender* tiene el objetivo de dar soporte a las funciones de reexpedición de paquetes y codificación de red. Esta capa de adaptación interpreta eventos en la capa de red y notifica sus observaciones a TCP. Los resultados de simulación presentados muestran que al utilizar *TCPFender* se puede obtener una ganancia del 100% del *throughput* de TCP. Más aún, los autores sugieren que su propuesta puede, además, interactuar con otros mecanismos que induzcan pérdidas de paquetes o arribos desordenados en el destino.

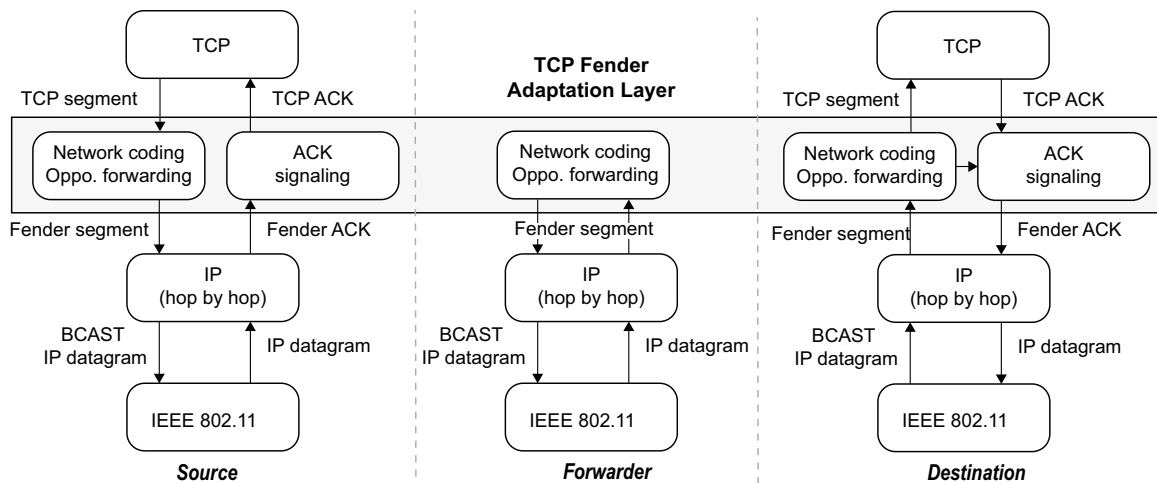


Figura 4.11: TCPFender implementa las funciones de transmisión oportunista de paquetes y codificación de red.

4.3.5 ENC - Codificación de red eficiente

Jannesari *et al.* proponen un algoritmo de codificación de red denominado ENC (Efficiency Network Coding) [102]. ENC permite codificar paquetes con diferencias marginales en su tamaño, lo cual permite utilizar eficientemente el ancho de banda disponible en la red. Al clasificar paquetes por tamaño y por destino, ENC incrementa el número de oportunidades de codificación y reduce el tiempo de espera para realizar la codificación de paquetes nativos. Para realizar la codificación, cada nodo utiliza 2 filas virtuales por cada nodo vecino. La primera fila, denominada como SVQ, se emplea para almacenar paquetes clasificados como pequeños; por otro lado, la segunda fila, denominada como LVQ, se utiliza para almacenar paquetes clasificados como grandes, en la Figura 4.12 se muestra este esquema de filas virtuales para el nodo E. Este nodo toma los paquetes de su fila de salida y los organiza en filas virtuales tomando en consideración el siguiente salto de los paquetes y el tamaño de estos. La clasificación de tamaño se realiza con base en un umbral cuyo el objetivo es que los paquetes nativos que se emplean para codificar tengan tamaños similares. La Figura 4.12 muestra cómo se organizarían los paquetes en las filas virtuales del nodo E tomando 100 bytes como valor de referencia para la clasificación de paquetes por tamaño. En este escenario, el nodo E debe transmitir inicialmente P1, este paquete puede codificarse con paquetes de tamaños similares y con destinos distintos, es decir, P10, P3 y P8. Sin embargo, en este caso no es posible realizar la codificación de P1, puesto que los nodos que reciban el paquete codificado no podrán decodificar la información debido a que su respectivo Pool de paquetes no cuentan con los paquetes necesarios para la decodificación. Incluso si P1 se trata de codificar con los paquetes en las filas virtuales SVQ, la codificación no es posible dado que los nodos receptores no podrían decodificar el paquete, por lo tanto P1 se transmite sin codificar. El siguiente paquete, P2, podría codificarse con P11 o con P5, debido a que estas codificaciones pueden ser decodificadas por los nodos receptores B, C y A. Los resultados obtenidos tras evaluar ENC en el simulador NS2 muestran una ganancia marginal observada en el *throughput* de TCP. Además, se advierte que la complejidad en tiempo de ENC puede incluso ser mayor a la observada en COPE; se omiten los beneficios en las oportunidades de codificación.

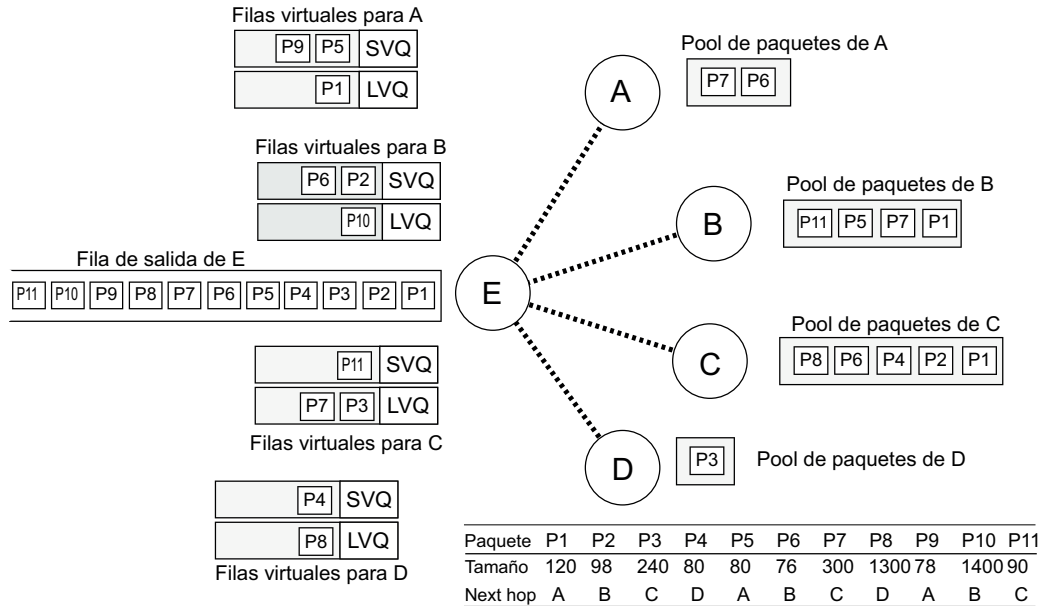


Figura 4.12: Funcionamiento de las filas virtuales SVQ y LVQ para clasificar paquetes y codificarlos con ENC.

4.3.6 MACINT-TCP

Alferaidi y Piechocki [103], afirman que TCP no puede obtener un beneficio sustancial cuando opera en una red inalámbrica en la que se ha implementado la codificación de red. Los autores señalan que no existe una

colaboración efectiva entre los protocolos de la capa MAC y los protocolos de la capa de transporte para mejorar la interacción de la técnica de codificación de red y TCP. Además, los autores enfatizan que el protocolo TCP y los protocolos de la capa MAC no fueron diseñados para aprovechar la técnica de codificación de red, impidiendo obtener los máximos beneficios de esta técnica. Los autores proponen una solución *cross-layer* denominada como MACINT-TCP, cuyo objetivo es **mejorar el throughput**, así como reducir el retardo y la tasa de paquetes perdidos de TCP empleando la codificación de red. MACINT-TCP comienza cuando el transmisor envía paquetes de datos al receptor y recibe los respectivos ACKs. La ventana de congestión incrementa su valor $1.5 * MSS$ tomando en consideración la ganancia teórica que ofrece la codificación de red al *throughput* de la red. En el caso de recibir tres ACKs duplicados, MACINT-TCP reduce la ventana de congestión a $1MSS$ (*Maximum Segment Size*) y el umbral del *slow-start* es calculado utilizando la siguiente relación $SSThresh = cwnd \times \frac{1+\sqrt{5}}{2}$. MACINT-TCP no solo modifica la manera en la que TCP ajusta el valor de la ventana de congestión y el umbral de *slow-start* sino que además incorpora una interacción *cross-layer* entre TCP y la función DCF de la capa MAC. Según los autores, cada que un acuse duplicado es recibido por MACINT-TCP es porque es muy probable que se deba al uso de la codificación de red, por tal motivo, MACINT-TCP alerta a la función DCF para que actualice su propia ventana de contienda utilizando la expresión $CW = (CW \times 0.618) + 3$. La interacción intercapa MACINT-TCP se muestra en la Figura 4.13. De esta forma, MACINT-TCP asigna los valores mínimos de la ventana de contienda CW de la capa MAC para que los nodos intermedios en una trayectoria puedan acceder al medio frecuentemente y reduzcan los eventos de *Timeout* de TCP ocasionados por la codificación de red. Los resultados de simulación muestran que MACINT-TCP puede reducir hasta en un 18% el PLR (*Packet Loss Ratio*) e incrementar el *throughput* hasta en un 23%.

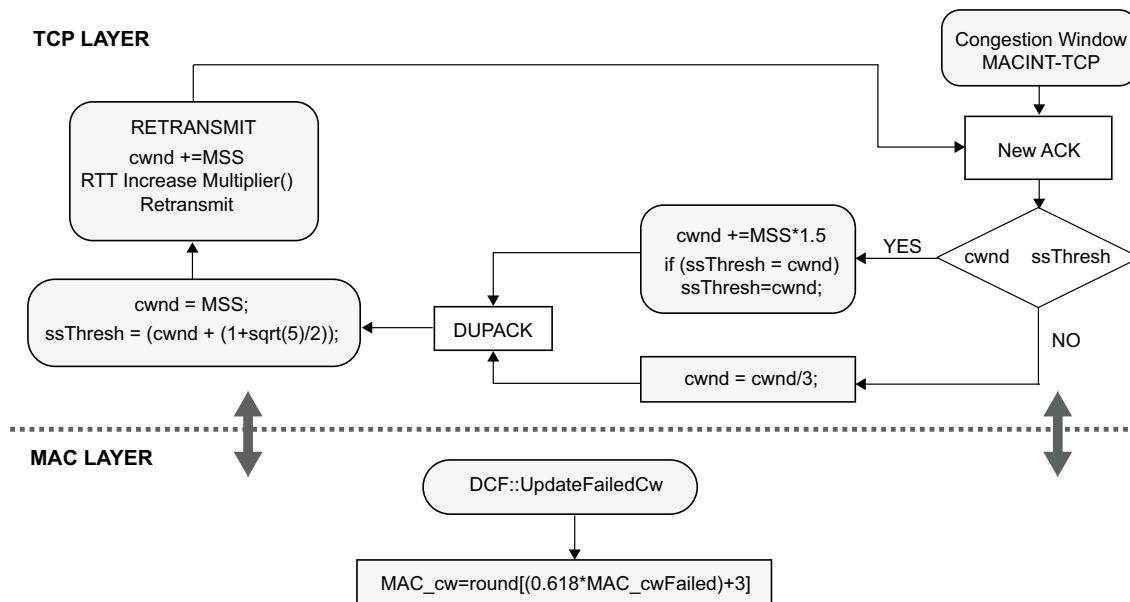


Figura 4.13: Interacción intercapa del protocolo MACINT-TCP.

4.3.7 Mejora de TCP con codificación de red en redes inalámbricas

En [104], Xu *et al.* evalúan, mediante simulaciones con NS2, los beneficios del esquema de codificación COPE con diferentes versiones del protocolo TCP tales como TCP-NewReno, TCP-FeW (*Fractional Window Increment*) y TCP-AP (*TCP with Adaptive Pacing*). Los protocolos de transporte se evalúan con y sin el uso de COPE en tres topologías de redes inalámbricas *multihop*, topología en cadena y topología de *mariposa* con uno y dos saltos. Los autores modificaron la capa MAC y el protocolo IFq (*Interface Queue*) de NS2 para poder implementar COPE. La interacción de los módulos propuestos por los autores se ilustra en la Figura 4.14. Los resultados

presentados muestran ganancias marginales en el *throughput* de TCP aun empleando COPE como mecanismo de mejora. Los autores argumentan que estas ganancias marginales son causadas particularmente por los mecanismos de control de transmisión de TCP. Para mejorar esos resultados, la propuesta que presentan consiste en modificar el mecanismo de retransmisión de COPE para que los paquetes nativos solo puedan ser codificados una vez, técnica denominada como *Encoded Once*. Con esta propuesta, la ganancia en el *throughput* de TCP incrementa del 2% al 5%. Los autores agregan a su propuesta un mecanismo adaptativo que toma en cuenta la longitud del *buffer* de transmisión y utiliza esta información para incrementar dinámicamente la tasa de transmisión de TCP, técnica denominada como *Network Coding Aware TCP*. Al incrementar la tasa de transmisión de TCP se incrementa el número de paquetes en la fila de transmisión y consecuentemente se incrementan las oportunidades de codificación. Con esta propuesta los autores muestran que bajo ciertas condiciones de simulación pueden obtener un incremento del 12% en el *throughput* de TCP.

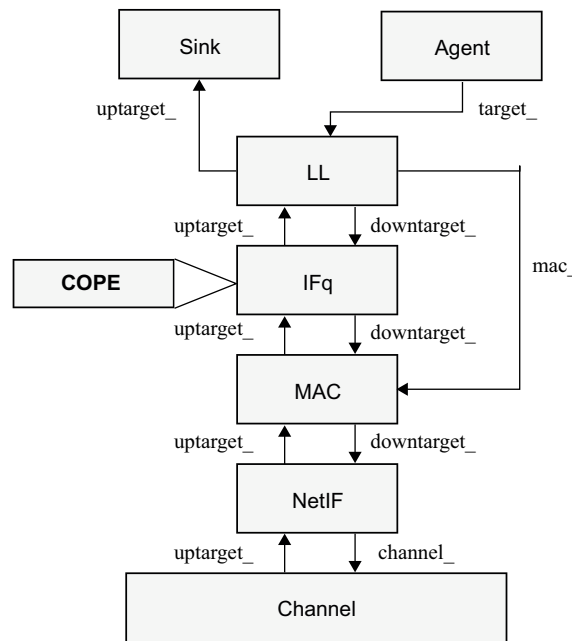


Figura 4.14: Módulos del modelo de nodo móvil de NS2 y su interacción con el módulo COPE.

4.3.8 TCP y codificación de red

En [87], Sundararajan *et al.* proponen una técnica denominada como TCP/NC. La propuesta consiste en aplicar codificación lineal de red sobre un flujo TCP en entornos con alta probabilidad de pérdidas, de este modo se busca minimizar el efecto de las pérdidas sobre el algoritmo de control de congestión de TCP. Para la implementación de esta propuesta se requieren cambios mínimos en la pila de protocolos, pues solo es necesario la incorporación de una capa de codificación entre la capa de red y la capa de transporte de los dispositivos fuente y destino. Al emplear TCP/NC, el nodo fuente del flujo TCP transmite codificaciones, lineales y aleatorias, del conjunto de paquetes de datos permitidos por la ventana de congestión de TCP. Esta propuesta preserva la filosofía de comunicación de extremo a extremo y puede ser fácilmente implementada pues no requiere modificar el funcionamiento de la infraestructura de red. De este modo, el nodo receptor no recibe los paquetes originales generados por TCP en el nodo fuente, sino que recibe un conjunto de combinaciones lineales que deberán ser decodificadas para obtener los paquetes originales. Los autores proponen modificar el esquema de acuses de TCP, de esta forma el nodo destino envía un acuse por cada combinación lineal, que revela una unidad de información, nueva (*degree of freedom*) y no por cada paquete recibido. La idea detrás de esta propuesta consiste en adaptar TCP en entornos con alta probabilidad de pérdidas, enmascarando las pérdidas, tal como se muestra en la Figura 4.15. Se puede apreciar en la imagen que las pérdidas ahora son

enmascaradas como un retardo y solamente afectan al cálculo del RTT y no a la ventana de congestión de TCP. Con este esquema se evita que TCP reduzca la ventana de congestión debido a las pérdidas y de este modo se puede mejorar el desempeño del protocolo. La única forma de identificar la congestión en la red será a través de la estimación del RTT, algo que TCP Vegas realiza proactivamente. Por esta razón, los autores lo han seleccionado para implementar su propuesta. Los resultados de simulación demuestran que esta propuesta ofrece un buen nivel de equidad para flujos en competencia y una ganancia significativa en la capacidad de la red si es elegido adecuadamente el factor de redundancia de TCP Vegas. Además, los resultados presentados también muestran que TCP/NC mejora el *throughput* de TCP aun en entornos con altas probabilidades de pérdidas.

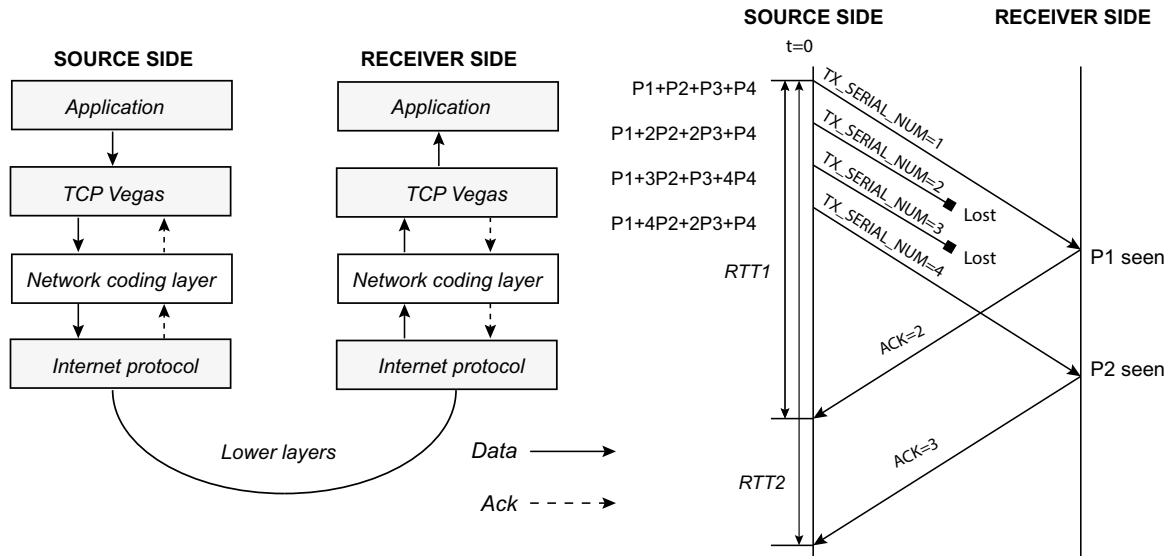


Figura 4.15: Implementación de los módulos NC entre la capa IP y el protocolo TCP Vegas junto con el sistema de acusos basados en *degree of freedom* para el enmascaramiento de pérdidas.

4.3.9 Codificación de red para mejorar la equidad para flujos TCP

Un problema evidente en iWMNs es la equidad que se ofrece para los flujos TCP que compiten en la red, dado que esta puede ser afectada por el número de saltos que existen entre los usuarios móviles y el MPP. En otras palabras, aquellos usuarios que se encuentran más cercanos al MPP obtendrán una mayor capacidad de la red que la capacidad que obtendrán los usuarios que se encuentran más alejados, esto se debe a que la tasa de transmisión de TCP depende de la estimación del RTT y este parámetro es influenciado en gran medida por el número de saltos en una trayectoria *multihop*. En [105] los autores enfrentan esta problemática utilizando dos propuestas de manera conjunta. Por un lado, los autores proponen implementar transmisiones *pseudo-broadcast* combinadas con el mecanismo de *handshake* RTS/CTS de 802.11. Los autores opinan que el uso de transmisiones en modo *broadcast* no son útiles si se quiere implementar la técnica de codificación de red, ya que las colisiones degradarán invariablemente el desempeño de la red. Por tal razón, las transmisiones en modo *pseudo-broadcast* utilizan el mecanismo de *handshake* RTS/CTS (*Request To Send/Clear To Send*). Por otro lado, los autores se inclinan por una técnica denominada *codificación de red temporizada*. Utilizando esta técnica, se crean ventanas aleatorias de tiempo de duración τ , donde τ es una variable aleatoria uniformemente distribuida entre $[0, \tau_{\max}]$. Cada paquete es combinado dentro de una ventana de tiempo, proporcionando un tiempo de espera para la codificación de paquetes provenientes de nodos lejanos. Los autores demuestran que su propuesta mejora la equidad que ofrece la red a flujos TCP, aunque ellos mismos señalan que la equidad sigue siendo un problema de gravedad para las iWMNs.

4.3.10 Codificación de red para TCP en redes WMN multi-radio

En [106] se presenta un mecanismo de codificación de red denominado TCP – I²NC para Multi-Radio WMNs. En Multi-Radio WMNs network coding podría no operar eficientemente por no existir posibilidad de realizar el proceso de escucha oportunista de transmisiones. TCP – I²NC atiende dos objetivos: minimizar las pérdidas de paquetes debidas a las características dinámicas del medio y maximizar la capacidad de la red para flujos transportados por TCP. Para resolver estas problemáticas los autores proponen dos mecanismos.

Primero, un método de codificación salto tras salto (*hop by hop*) sobre N paquetes TCP (*inter flujo*). Este esquema de codificación utiliza la estimación de la tasa de pérdidas sobre el enlace inalámbrico para definir la redundancia de codificación como $R = 1/(1 - P_e)$, donde P_e es la tasa de pérdidas de paquetes sobre el enlace. El nodo receptor, al decodificar los N paquetes, reexpide un ACK al nodo transmisor para informar al transmisor que se han podido decodificar los N paquetes codificados. Con este mecanismo de codificación: i) se ocultan las pérdidas para TCP, ii) se garantiza la confiabilidad en cada transmisión, iii) se ofrece una utilización equitativa para los flujos en competencia, y iv) se evita la espera de oportunidades de codificación. Segundo, los autores proponen un esquema de control de flujo distribuido que permite minimizar la interacción del algoritmo de control de congestión de TCP y la WMN, permitiendo utilizar de manera racional los recursos de la red. Esta propuesta consiste en ajustar, de manera adaptativa, el número de pares codificador-decodificador en un salto de la trayectoria. Los autores evalúan esta propuesta con TCP Vegas, con base el argumento presentado en [87]. Los resultados demuestran que TCP – I²NC ofrece una capacidad muy cercana a la ideal; aun en un escenario de red con un alto índice de pérdidas en la trayectoria. Es de destacar que el retardo es minimizado y la equidad con la que se distribuyen los recursos de la red obtiene un JFI (*Jain's Fair Index*) de 1 a pesar del incremento del número de nodos en la red y de la tasa de pérdidas. No obstante sus indudables aportes, esta propuesta presenta dos argumentos que limitan su utilidad en escenarios inalámbricos. Si bien, los autores asumen que en la red existen enlaces inalámbricos *full-duplex*, se debe asumir que las transmisiones inalámbricas son libres de interferencias debido a la existencia de canales ortogonales ilimitados. Los resultados muestran que TCP – I²NC utiliza aproximadamente un 90% de la capacidad de los enlaces aun cuando estos tengan una tasa de pérdidas de hasta el 35%.

4.4 Codificación utilizando temporizadores para la detección de oportunidades

La detección de oportunidades de codificación es sumamente importante para obtener beneficios sustanciales de la técnica de codificación de red. Una oportunidad de codificación se detecta en un nodo cuando éste ha recibido los paquetes nativos necesarios para construir un paquete codificado. Es importante enfatizar que aun cuando un nodo pueda haber recibido múltiples paquetes nativos no significa que logre codificarlos, pues los paquetes nativos deben satisfacer ciertas condiciones para poder construir con ellos un paquete codificado. Para incrementar la probabilidad de crear la oportunidad de codificación se puede emplear la estrategia de incorporación de tiempos de espera para los paquetes nativos. Cuando un nodo ha recibido paquetes nativos, pero no puede codificarlos, puede incorporar tiempos de espera para estos paquetes antes de ser transmitidos. Estos periodos de espera permiten que un nodo pueda recibir otros paquetes y de este modo incrementar la probabilidad de generar una oportunidad de codificación. Esta técnica resulta muy útil para incrementar las oportunidades de codificación y de este modo maximizar los beneficios de la codificación de red y mejorar el desempeño de la red. El costo a pagar, sin embargo, es el incremento del retardo de extremo a extremo para los paquetes en la red.

4.4.1 Desempeño de TCP en WMNs codificadas

En [10], Huang *et al.* cuantifican los beneficios de la técnica de codificación de red para tráfico TCP en WMNs. La evaluación presentada por los autores se realiza utilizando un banco de pruebas (*testbed*). Los autores identifican **dos beneficios** significativos que ofrece la codificación de red: i) la reducción de la probabilidad de pérdidas, debido particularmente a la disminución de la contienda por el medio que ofrece la codificación de red, así como ii) el incremento en la capacidad de la red. En esta propuesta los autores incluyen un *buffer* de codificación, denominado *NCBuffer*, para almacenar paquetes nativos que aún no tienen oportunidad de ser

codificados. Si un paquete nativo se recibe, y existen paquetes nativos en espera en el *NCBuffer* ambos paquetes son codificados con una operación lógica XOR. A cada paquete almacenado en el *NCBuffer* se le debe asociar un temporizador *NCBufTimeout* que determina el tiempo de espera máximo que puede mantenerse en el buffer en espera de una oportunidad de codificación; al expirar este temporizador, el paquete nativo es transmitido sin ser codificado. El algoritmo implementado por los autores para el proceso de recepción y transmisión se muestra en la Figura 4.16. Los resultados muestran que la capacidad de la red, en una topología lineal, puede alcanzar hasta un incremento del 70% cuando el *NCBufTimeout* es seleccionado adecuadamente, es decir, un valor constante de 1.5ms para el escenario propuesto por los autores. Sin embargo, es claro que al incorporar tiempos de espera, existe un compromiso entre el retardo inducido por el proceso de codificación y la estimación del RTT (*Round Trip Time*) que realiza TCP. En las pruebas realizadas por los autores, al incrementar el valor del *NCBufTimeout* se obtiene una menor ganancia en la capacidad de la red, siendo éste únicamente del 20%. Esta reducción es originada por el incremento del RTT, el cual alcanza hasta 200% en el peor de los casos, y su efecto sobre la tasa de crecimiento de la ventana de congestión de TCP. Estos resultados ponen de manifiesto que **la configuración del tiempo de espera puede, bajo ciertas condiciones, afectar el desempeño de TCP y no brindar ningún beneficio para la red.**

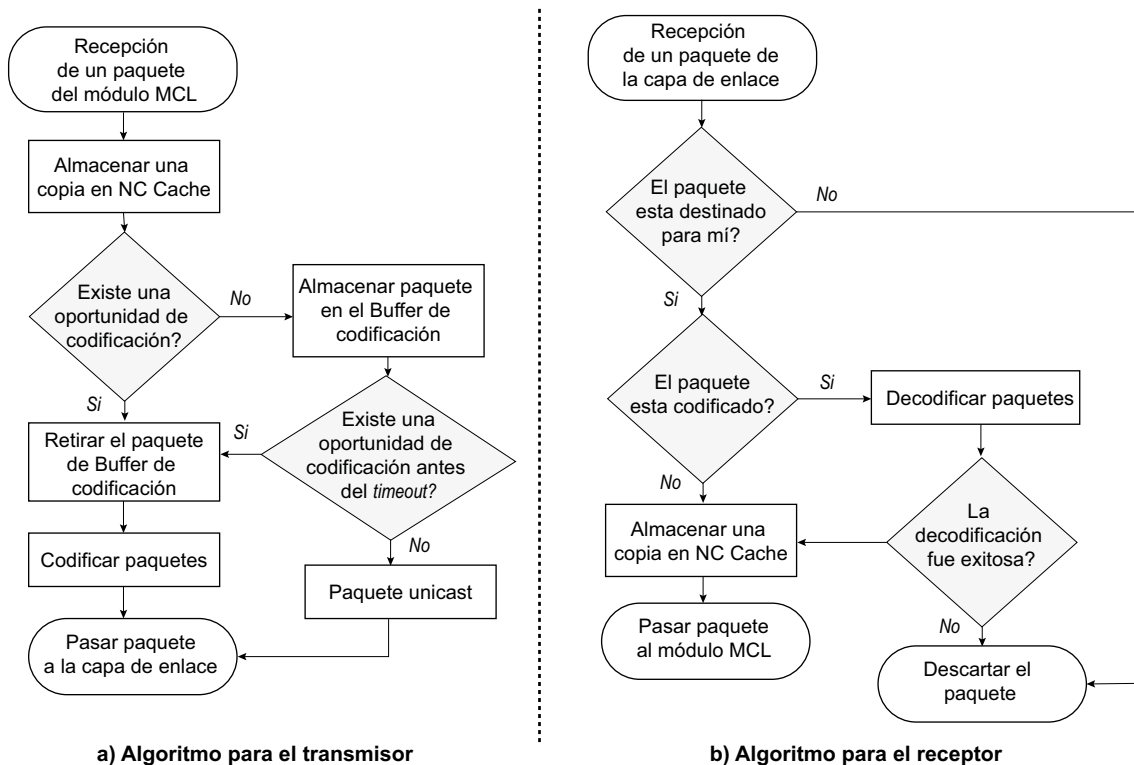


Figura 4.16: Algoritmos de codificación-decodificación para el transmisor y receptor al incorporar un *buffer* de codificación.

4.4.2 Impacto de la codificación de red en el desempeño de TCP en WMNs

En [107], los autores diseñan e implementan una entidad para implementar la codificación de red, esta es implementada entre la capa de enlace y la capa de red. A ella se le asigna el proceso de codificación y decodificación de paquetes. Para simplificar, los autores sugieren codificar únicamente paquetes nativos de datos TCP, dejando de lado la codificación de los paquetes de acuse de TCP (*ACKs*). En la propuesta los nodos intermedios deben almacenar los paquetes de datos TCP en un *buffer* de codificación de tamaño $N \geq 1$. A cada paquete nativo se le asignará un temporizador de permanencia en el *buffer*, dicho tiempo de espera es denominado

por los autores como el *Coding Timeout- TC*. Este factor limita el tiempo máximo que un paquete nativo puede mantenerse almacenado en el *buffer* de codificación.

Cada vez que un nodo recibe un paquete nativo, se verifica si dicho paquete se puede codificar con un subconjunto de paquetes almacenados en el *buffer* de codificación. Si el paquete recibido se puede codificar con paquetes del *buffer*, el proceso se realiza inmediatamente y el paquete codificado pasa a las capas inferiores para su transmisión. En caso contrario, el paquete se almacena en el *buffer* de codificación y se le asigna su respectivo *TC*. Si no llegará a presentarse una oportunidad de codificación durante el tiempo *TC*, el paquete nativo pasa a las capas inferiores para su transmisión sin codificar.

Además, si un nodo *escucha* la transmisión de un paquete que no está destinado para él, el paquete igualmente es almacenado temporalmente en el *buffer* de decodificación; esta técnica es denominada por los autores como *Opportunistic listening*, y es una de las diferencias principales respecto a la propuesta presentada en [10]. Los autores argumentan que existe la posibilidad de que el *buffer* de codificación alcance su capacidad máxima, y en ese escenario, el paquete con un mayor tiempo de espera en el *buffer* deberá ser transmitido para que un nuevo paquete pueda ser almacenado en el *buffer*. La propuesta, evaluada con simulaciones, considera parámetros tales como el tamaño del *buffer* de codificación así como el *TC*.

Los resultados muestran que aun incrementando el tamaño del *buffer* de codificación y el tiempo de espera *TC*, la ganancia en capacidad no mejora; por lo tanto, los autores concluyen que el incremento de estos factores no se traduce en mayores oportunidades de codificación y enfatizan la importancia que tienen estos factores para el desempeño de la red y de la codificación de red, afirmando que el incremento de *TC* puede afectar el desempeño de las conexiones TCP al incrementar sustancialmente la estimación del RTT (*Round Trip Time*). Otro de los importantes resultados de este trabajo muestra que este método de codificación de red no resulta útil en entornos con alta probabilidad de errores de transmisión, ya que la ganancia en capacidad es insignificante. Finalmente, los autores muestran que solo en algunos escenarios, donde se considera la topología de la red, el *TC* y el *buffer* de codificación, podría incrementar la capacidad de la red hasta en un 33%.

4.4.3 Codificación de red con inserción de tiempos de espera

En [108], Takimoto *et al.* identifican que la propuesta presentada en [97], denominada como *PiggyCode*, tiene beneficios para el desempeño de TCP al codificar paquetes de datos y paquetes de acuse que viajan en flujos bidireccionales; sin embargo, advierten que dichos beneficios dependen de la tasa de transmisión. Afirman que *PiggyCode* ofrece mayores beneficios cuando la red está cerca del punto de saturación; pero, si la tasa de transmisión es baja, la probabilidad de encontrar oportunidades de codificación en los nodos intermedios se reduce significativamente y de este modo se reducen también los beneficios de la codificación de red. Para resolver este dilema, los autores introducen un tiempo de espera en los nodos intermedios de la red y ajustándolo tomando en consideración la tasa de transmisión y el número de saltos entre el nodo fuente y el nodo destino. El método propuesto requiere de una fila de codificación y una fila para la interfaz. En esta propuesta los autores implementan una capa de codificación (*NC*) entre la capa de red y la capa de enlace, tal y como se muestra en la Figura 4.17.

Cuando la capa (*NC*) recibe un paquete de datos (*DATA*) desde la capa IP, el módulo *NC* busca en la fila de la interfaz si existe algún paquete de acuse (*ACK*) para realizar la codificación. Si se encuentra un paquete *ACK*, entonces se realiza el proceso de codificación, y el paquete codificado se posiciona al frente de la fila para la interfaz. Cuando no se encuentra un paquete *ACK* en la fila de interfaz, entonces el paquete *DATA* es almacenado en la fila de codificación con un tiempo de permanencia predefinido que considera el número de saltos y la tasa de transmisión. La tasa de transmisión se estima considerando la cantidad de paquetes almacenados en la fila para la interfaz. La tasa de transmisión se considera baja si la fila de la interfaz está vacía y alta si la fila se satura. Tomando en consideración el número de saltos entre el nodo fuente y el nodo destino, los autores establecen tres tiempos de espera para tres diferentes escenarios: 10 ms para topologías con dos o tres saltos y baja tasa de transmisión, 30 ms para escenarios con alta tasa de transmisión y dos saltos, y 90 ms

para topologías con 3 saltos y altas tasas de transmisión. Al evaluar esta propuesta mediante simulaciones, los autores señalan que obtienen hasta un 10% de incremento del *throughput* de TCP con *PiggyCode* en escenarios con bajas tasas de transmisión. Es importante señalar que esta solución funciona únicamente en escenarios donde TCP no utiliza la función conocida como *Delayed ACKs*.

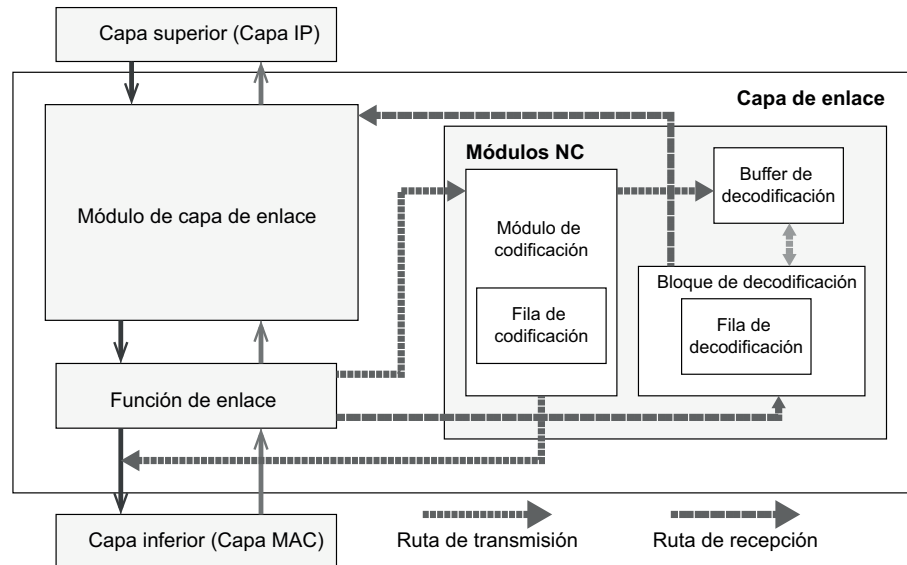


Figura 4.17: Implementación de los módulos NC entre la capa IP y la capa MAC y las interacciones entre ellos.

4.5 Conclusiones sobre la literatura revisada

En este capítulo se han mostrado diferentes propuestas que utilizan la técnica de codificación de red y cuyo propósito es mejorar el desempeño de diversas redes de datos. Se ha revisado como con esta técnica se puede mejorar el funcionamiento de redes inalámbricas; ya que al utilizarla: *i)* se reduce la contienda por el medio inalámbrico, *ii)* se disminuye el número de transmisiones en el medio logrando incrementar la capacidad de la red. Por otro lado, se ha mostrado también como la técnica de codificación de red permite mejorar el desempeño del protocolo TCP, esta técnica es utilizada para construir combinaciones lineales de paquetes e incrementar la tolerancia a las pérdidas del protocolo, logrando reducir los efectos adversos de las pérdidas en la tasa de transmisión de TCP.

Resulta claro que las propuestas analizadas en este capítulo no fueron concebidas para atender específicamente los problemas de desempeño de las iWMNs, puesto que estas redes combinan mecanismos operativos de las redes inalámbricas multihop junto con mecanismos de las redes inalámbricas en modo infraestructura, los cuales dificultan la implementación de la técnica de codificación de red.

Es entendido que en iWMNs existen patrones de tráfico centralizado en ciertos nodos junto con una mayor movilidad de los usuarios de la red; además las iWMNs tienen un *backbone* estático sin limitaciones de energía o de procesamiento, y, por último, utilizan un mecanismo de encaminamiento *multihop*. Por estas características es que para poder obtener el mayor beneficio de la técnica de codificación de red es necesario contar con análisis detallados de las interacciones de los dispositivos de la red que hagan posible seleccionar aquellos esquemas de codificación factibles de ser integrados eficientemente en esta arquitectura.

Del conjunto de propuestas analizadas en este capítulo se puede destacar lo siguiente.

- La técnica de codificación de red requiere de la operación de diferentes mecanismos para funcionar en redes inalámbricas, es decir, *Opportunistic listening*, *Opportunistic coding* y *Learning Neighbor State*

–los cuales han sido propuestos y ampliamente analizados por Katti *et al.* en [7]–. Estos mecanismos pueden ser modificados para que funcionen de manera eficiente en iWMNs, pues se debe destacar que en este tipo de red tiene una infraestructura estática que facilita la adaptación e implementación de los mecanismos citados.

- Tal como se señala en [88], la convergencia de flujos incrementa el número de oportunidades de codificación. En iWMNs existen diferentes nodos en la red en donde convergen una gran cantidad de flujos, tales como en los MAPs y los MPs, esto representa una oportunidad ideal para aplicar la técnica de codificación de red en iWMNs porque incrementarían significativamente las oportunidades de codificación.
- En [91, 98, 10, 108] se demuestra que las oportunidades de codificación en un nodo se pueden incrementar si se incorpora un temporizador para los paquetes nativos en espera de una oportunidad de ser codificados. Si bien esta propuesta puede teóricamente mejorar los beneficios obtenidos por la técnica de codificación de red, ninguno de los autores señala las dificultades que se enfrentan para adaptar este temporizador a las condiciones de la red.
- Gerla *et al.* [99, 98] muestran de manera fundamentada como la técnica de codificación de red puede ser implementada en un nodo inalámbrico. Los resultados presentados por los autores muestran que dicha implementación resulta útil en escenarios con altas probabilidades de errores en los enlaces de transmisión. Lamentablemente, la evaluación presentada por los autores no contempla interacciones complejas en la red, tales como movilidad, tráfico sesgado, número reducido de saltos en la red, y únicamente presentan resultados para un número reducido de flujos TCP. Esto reduce el alcance de esta propuesta para utilizarse en redes con alta densidad de usuarios, en donde, como se sabe, la capacidad y la equidad sufren una degradación importante, tal como sucede en iWMNs.
- El fundamento operativo de los mecanismos ComboCoding [99] y CodeMP [98] puede ser utilizado para atender flujos *unicast intra-malla* y *outer-malla* para minimizar las interferencias entre flujos duales (TCP DATA y TCP ACK) frecuentes en iWMNs.

Evaluación del desempeño de diversas variantes de TCP en iWMNs

“Eso es un experimento de tontos. Pero me encantan los experimentos de los tontos. Siempre los estoy haciendo.”

Charles Darwin

EL protocolo TCP transporta un amplio volumen del tráfico en Internet. Sus servicios para la transferencia confiable de información, así como sus funciones de detección y respuesta ante la congestión de la red, le han conferido una enorme popularidad. TCP es tan versátil que actualmente es utilizado por un amplio abanico de aplicaciones, tales como HTTP, FTP, Telnet, SSH, entre otras.

TCP fue diseñado para operar en redes cableadas, y una de las características principales de su diseño es la **interpretación de las pérdidas de paquetes como un indicador de congestión en la red** [63]. Esta interpretación funciona eficientemente en redes con muy baja tasa de error de bit, tal como sucede en redes cableadas; no obstante, en redes inalámbricas puede ser perjudicial [109, 110]. Las pérdidas de paquetes en redes inalámbricas no solo son producidas por congestión, también pueden ser causadas por otras características del medio inalámbrico y de la red. Cada que TCP detecta pérdidas de paquetes, disminuye la tasa de transmisión para poder revertir la congestión; medida correctiva que resulta ineficiente en redes inalámbricas, ya que no todas las pérdidas de paquetes en redes inalámbricas son ocasionadas por congestión. Esto da como resultado que TCP reduzca innecesariamente la utilización de la red.

Además, no solo las pérdidas de paquetes propician el pobre desempeño de TCP en redes inalámbricas, **existen otros factores que limitan la utilización óptima del medio inalámbrico**. Esto representa un serio problema, ya que el ancho de banda disponible en redes inalámbricas es un recurso limitado. Varios factores influyen en este problema; por ejemplo, la variabilidad del retardo de extremo a extremo propicia un lento incremento de la tasa de transmisión de TCP y consecuentemente la baja utilización del ancho de banda. Por otro lado, dado que el medio inalámbrico es *half-duplex* y solo permiten el tránsito de un paquete a la vez, los acuses deben competir por el acceso al medio con los paquetes de datos de TCP, esto reduce drásticamente la utilización del canal, generando una gran cantidad de interrupciones a la cadencia de transmisión. Otro caso es que entre la ruta para el flujo de paquetes de datos y la ruta para los acuses existen diferentes tipos de asimetrías [36] que afectan el mecanismo de autorregulación de la tasa de transmisión, por la razón de que este mecanismo tiene una estrecha relación con la tasa de recepción de los acuses. Mencionando por último que la movilidad de los nodos puede ocasionar la partición de una red inalámbrica y propiciar la pérdida de conectividad de extremo a extremo, interrumpiendo temporalmente el flujo de los paquetes de datos y acuses. Estos ejemplos son solo una muestra de las diversas dificultades que enfrenta TCP en redes inalámbricas.

A pesar de los efectos negativos que se han enunciado, TCP seguirá predominando como protocolo de transporte en redes inalámbricas, debido principalmente a la enorme popularidad de este protocolo en Internet. Hoy en día existen diversas propuestas que buscan mejorar el desempeño de TCP en redes inalámbricas. Las modificaciones más comunes adaptan el funcionamiento de TCP a escenarios específicos, dando como resultado diferentes versiones del protocolo, tales como: TCP para redes satelitales, TCP para redes inalámbricas *ad hoc*, TCP para redes inalámbricas WLAN, TCP para redes celulares o bien TCP para redes heterogéneas.

Las diferentes versiones de TCP son diseñadas empleando distintos enfoques, entre los que destacan dos: *i) división de red* y *ii) conectividad de extremo a extremo* [34]. Así tenemos el enfoque de **división de red** (*split mode*) que se emplea en redes heterogéneas, y tiene por objetivo crear sub-conexiones TCP para separar porciones de red que emplean distintas tecnologías. Así, mediante sub-conexiones TCP es posible separar redes con enlaces cableados y redes con enlaces inalámbricos, empleando una conexión para TCP para la sección cableada y otra para sección la inalámbrica, p. ej., I-TCP [111] y Split TCP [112]. Sin embargo, merece mencionarse que este tipo de propuestas violan el principio de operación de extremo a extremo (*end-to-end*) de TCP. A diferencia del enfoque de **conectividad de extremo a extremo** en donde solo los dispositivos fuente y destino participan en el proceso de transmisión de TCP, es decir, la red y su infraestructura es ajena a este proceso. De este modo, solo el nodo destino debe notificar al nodo fuente la recepción de los paquetes de datos para que este último sea capaz de determinar el estatus de la red y adaptar su tasa de transmisión de manera dinámica, p. ej., TCP New Reno [113].

Otra característica del diseño de las diferentes variantes de TCP es: **la forma en la que el protocolo responde a las pérdidas de paquetes en redes inalámbricas**. La respuesta se puede realizar de dos formas: de manera **proactiva** o bien **reactiva**. Con el modo **reactivo**, el transmisor reduce su tasa de transmisión justo cuando es detectada la congestión de la red. Mientras que con el modo **proactivo**, el transmisor interpreta constantemente la información recibida desde el emisor para prever la congestión y reducir la tasa de transmisión antes de que esta ocurra [34].

Paralelo al problema de desempeño de TCP en redes inalámbricas, han surgido nuevas arquitecturas de red que diversifican aún más las dificultades de TCP en estos escenarios. Tal es el caso de las iWMNs. El estudio sobre el desempeño de las iWMNs y su relación con los diferentes protocolos de la pila de protocolos es una labor que se ha analizado en años recientes, p. ej., [114, 19, 115]. Estos esfuerzos se han enfocado en evaluar y diseñar propuestas novedosas para mejorar el funcionamiento de las iWMNs, por ejemplo: protocolos de encañamiento, p. ej., [3, 116, 117], mecanismos de multi-canalización para maximizar la reutilización espacial del medio, p. ej., [76, 77], protocolos inter-capa para mejorar y enriquecer la interacción con TCP [110], mecanismos para mejorar la equidad y capacidad de TCP [5, 118, 119], entre otros.

Con el objetivo de identificar los retos de TCP en iWMNs, en este capítulo evaluamos la interacción entre estos dos elementos. **En esta evaluación analizamos el desempeño de algunas variantes de TCP** considerando parámetros tales como la máxima capacidad de la red y la equidad con la que se distribuye este recurso. **Para esta evaluación se seleccionan las variantes de TCP denominadas como: TCP Westwood+, TCP New Reno, TCP Vegas, TCP Venó, TCP Hybla y TCP Freeze**. Estas variantes han sido seleccionadas por tres razones principales. Primera, algunas de estas variantes fueron diseñadas para funcionar en entornos heterogéneos, lo cual se adapta a las características de las iWMNs. Segunda, las variantes utilizan métodos tanto proactivos como reactivos para detectar la congestión en la red, lo cual le brinda diversidad al estudio. Finalmente, estas variantes cumplen con el modelo de conectividad de extremo a extremo, lo cual se ajusta al principio de operación de extremo a extremo de TCP. A través de simulaciones computacionales analizamos el desempeño de estos protocolos en iWMNs, se analiza minuciosamente la interacción entre iWMNs y TCP. Este estudio coadyuvará al proceso de diseño de la técnica de codificación de red para iWMNs.

TCP *server-side* (véase la Figura 5.2). Este mensaje se denomina como SYN y especifica, entre otras cosas, el número de secuencia inicial (*Initial Sequence Number- ISN*), el número de puerto del *server-side* y el tamaño del buffer de recepción (conocido como *Reception Window*). Cuando la entidad TCP *server-side* recibe el mensaje SYN, responde con un mensaje de acuse y de sincronización denominado como SYN-ACK. Este mensaje incluye el ISN que utilizará el *server-side*, el número de secuencia que espera recibir a continuación (ACK) y el tamaño del buffer de recepción (*Reception Window*). Finalmente, cuando la entidad TCP *client-side* recibe el mensaje SYN-ACK acusa esta recepción con un mensaje ACK; ajustando el valor del ISN y el tamaño del buffer de recepción. De esta forma se establece una conexión TCP entre las dos entidades y ahora pueden realizar el intercambio de datos. El procedimiento descrito para el establecimiento de una conexión TCP es comúnmente conocido como *three way handshake*.

Finalización de la conexión

TCP es un protocolo full-duplex, por tanto se deben finalizar dos sesiones; por un lado, la conexión TCP que va del TCP *client-side* al TCP *server-side* y, por otro lado, la conexión que va del TCP *server-side* al TCP *client-side*. Para cerrar cada conexión se emplean dos mensajes, esto quiere decir que para finalizar completamente una conexión TCP se emplean 4 mensajes, por ello este proceso también es conocido como *4-way-handshake*. Primero, la entidad TCP *server-side* debe enviar un mensaje de finalización de conexión denominado como *finish packet* (FIN-ACK). Cuando la entidad TCP *client-side* recibe el mensaje FIN-ACK debe transmitir un mensaje de acuse (ACK); así se da por concluida la primera conexión. Finalmente, el TCP *client-side* debe enviar su respectivo mensaje FIN-ACK al TCP *server-side*. Cuando la entidad TCP *server-side* recibe este mensaje envía un ACK para cerrar completamente la sesión TCP. La Figura 5.2 ilustra cómo se realiza el proceso.

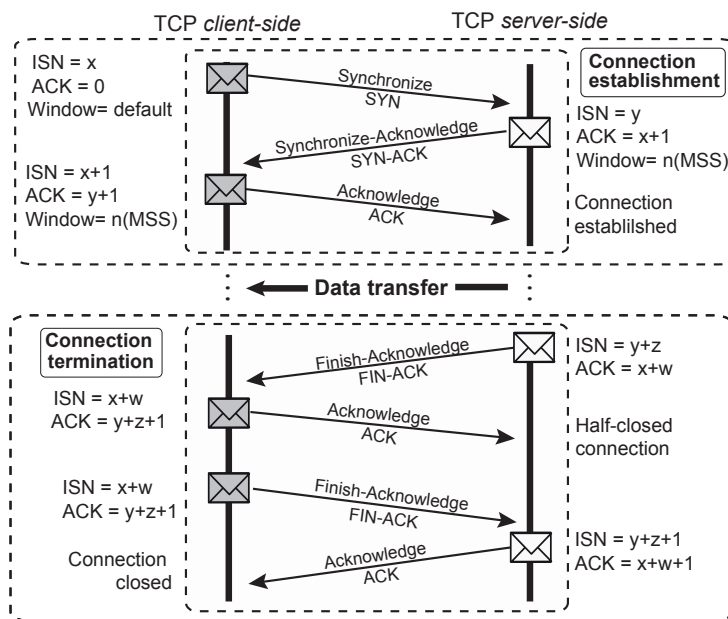


Figura 5.2: Mecanismos para el establecer y finalizar una sesión TCP.

5.1.2 Control de flujo

El mecanismo de control de flujo se encarga de limitar la tasa de transferencia entre las entidades TCP *server-side* y TCP *client-side* para no saturar el *buffer* de recepción. Un evento de saturación ocurrirá cuando el *buffer* de recepción se llena a una velocidad mayor de la que TCP puede procesar y transferir los paquetes a los protocolos de la capa de aplicación. Para evitar este problema, la entidad TCP *client-side* debe informar continuamente a la

entidad TCP *server-side* el **espacio disponible en el buffer de recepción** empleando un parámetro denominado como el *tamaño de la ventana de recepción* (receiver window size - RWS). Así, TCP garantiza que la entidad TCP *server-side* transmitirá una cantidad de segmentos que no superarán la cantidad de información que puede almacenar el TCP *client-side* en su buffer de recepción. Para notificar el valor RWS, la entidad TCP *client-side* agrega el RWS en cada ACK. De esta manera, cuando el TCP *server-side* recibe un ACK obtiene el valor RWS que limita la tasa de transmisión, ese límite se mantendrá hasta recibir un nuevo valor RWS. Si el valor RWS es igual a cero (indicador de *buffer* lleno), entonces se detiene la transmisión de segmentos y se entra en una etapa denominada como **condición de persistencia de TCP** (*TCP persist condition*). En esta etapa TCP no solo detiene la transmisión, sino que además mantiene el valor de todas sus variables hasta no recibir un RWS diferente de cero. La condición de persistencia de TCP utiliza un temporizador para evitar un bloqueo, ocasionado si el TCP *server-side* se encuentra esperando un RWS diferente de cero y el TCP *client-side* también se encuentra esperando un nuevo segmento de información para así enviar un ACK y actualizar el RWS. Cuando el temporizador de la condición de persistencia de TCP expira, el TCP *server-side* envía periódicamente al TCP *client-side* **segmentos especiales** denominados como ZWP (*Zero Window Probe*) para solicitar un ACK con una actualización del RWS.

5.1.3 Estimación del RTT

TCP detecta la pérdida de paquetes empleando un evento denominado como *retransmission timeout* (RTO) y funciona de la siguiente manera. Cada que un paquete de datos es transmitido por TCP, se asigna un temporizador (*Timeout*) que establece un tiempo de espera para recibir el respectivo ACK. **Si el acuse no se recibe y el Timeout expira, entonces TCP asume que el paquete se ha perdido. TCP utiliza un mecanismo adaptativo que permite calcular dinámicamente el valor del Timeout.** Para realizar este cálculo, TCP obtiene continuamente muestras del RTT, definidas como *Sample_RTT*. Esta muestra mide el tiempo que transcurre desde que un paquete de datos se transfiere a la capa de red y hasta que se recibe su respectivo acuse (ACK). El valor de las muestras *Sample_RTT* varían de segmento a segmento, ya que el RTT para cada segmento depende de la densidad de tráfico en la red. Esto significa que el valor *Sample_RTT* puede presentar valores atípicos, que pueden ser mucho mayores que el RTT. TCP, entonces, realiza una estimación del RTT, definida como *Estimated_RTT*. Para realizar esta estimación, TCP utiliza un promedio móvil ponderado exponencial (EWMA - *Exponential Weighted Moving Average*) que, con base en el RFC 6298, se calcula de la siguiente manera:

$$\text{Estimated_RTT} = (1 - \alpha) \times \text{Estimated_RTT} + \alpha \times \text{Sample_RTT}. \quad (5.1)$$

El valor del peso α debe configurarse en 0.125 de acuerdo al RFC 6298. Además de calcular el *Estimated_RTT*, TCP calcula la variabilidad del RTT, definida como *Dev_RTT*. Esta medida permite estimar la desviación que presenta cada muestra *Sample_RTT* con respecto al valor *Estimated_RTT*. El valor de *Dev_RTT* se calcula empleando un EWMA de la siguiente forma:

$$\text{Dev_RTT} = (1 - \beta) \times \text{Dev_RTT} + \beta \times |\text{Sample_RTT} - \text{Estimated_RTT}|. \quad (5.2)$$

Con base en el RFC 6298, el valor de β debe ser configurado con 0.25. Es evidente que cuando una muestra *Sample_RTT* tiene una diferencia grande con respecto a la estimación *Estimated_RTT*, entonces la variabilidad *Dev_RTT* será mayor, si la diferencia es menor entonces ocurrirá lo contrario, es decir, disminuirá la variabilidad *Dev_RTT*. Con los valores que se calculan con cada *Sample_RTT*, es decir, *Estimated_RTT* y *Dev_RTT*, TCP define el valor del *Timeout* para cada paquete. Es claro que el *Timeout* debe ser mayor que el *Estimated_RTT* y debe tomar en consideración los valores del *Dev_RTT*. De tal modo, TCP calcula el *Timeout* como:

$$\text{Timeout} = \text{Estimated_RTT} + 4 \times \text{Dev_RTT}. \quad (5.3)$$

La Figura 5.3 muestra cómo evolucionan los valores *Estimated_RTT*, *Dev_RTT* y el *Timeout* con base en valores *Sample_RTT* de una sesión de transferencia de datos con TCP. En la imagen se aprecia que el valor del *Timeout* se ajusta de manera dinámica a las variaciones que presentan las *Sample_RTT*. Cabe destacar que el valor del *Timeout* es una aproximación, y por lo tanto puede tener errores; por ejemplo cuando el *Timeout* es menor que el *Sample_RTT*; tal y como se señala en la figura. Cuando ocurre un evento RTO, el valor de *Timeout* se duplica para evitar subsecuentes eventos RTO hasta recibir un acuse que permita continuar con la estimación [120].

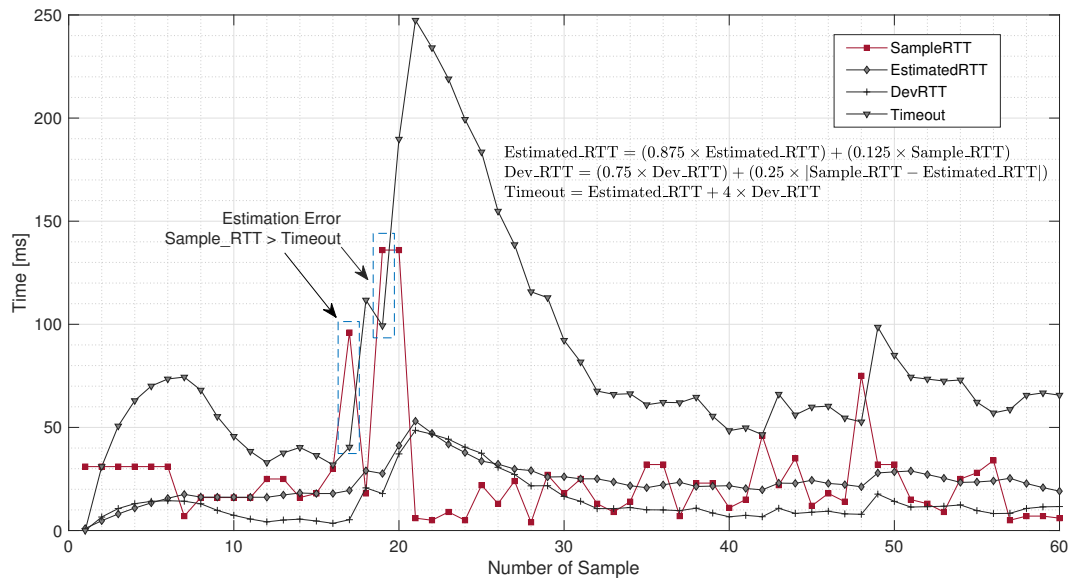


Figura 5.3: Evolución de la estimación del Timeout para la detección de pérdidas por RTO.

5.1.4 Control de congestión

El mecanismo de control de flujo limita la tasa de transmisión del TCP *server-side* con el objetivo de evitar la congestión del *buffer* de recepción del TCP *client-side*; sin embargo, este mecanismo no previene la congestión en los nodos que componen la red. Para solucionar esta limitante, TCP utiliza la función de control de congestión. El objetivo de esta función es **adaptar la tasa de transmisión de la entidad TCP *server-side* para maximizar la utilización del ancho de banda disponible y, simultáneamente, evitar congestionar la red.** Para completar esta tarea, el control de congestión está constituido por cuatro algoritmos: *slow-start*, *congestion avoidance*, *fast recovery*, and *fast retransmit* [121, 113].

El TCP *server-side* utiliza dos variables que limitan su tasa de transmisión: i) el tamaño de la *ventana de recepción* (RWS) y ii) el tamaño de la *ventana de congestión* (*congestion window size - cwnd*). La variable *cwnd* establece el límite máximo de bytes que pueden ser transmitidos por la red sin que estos todavía sean acusados por el TCP *client-side*. El valor de la *cwnd* normalmente se define como base el MSS (*Maximum Segment Size*), valor que define el tamaño en bytes de los segmentos que pueden ser transmitidos por la red. Así, la entidad TCP *server-side* puede transmitir simultáneamente varios segmentos. El número de bytes que el TCP *server-side* puede enviar a la red es limitado por γ , donde γ se puede calcular con la ecuación 5.4.

$$\gamma = \min(\text{RWS}, \text{cwnd}). \quad (5.4)$$

Una de las características destacadas de TCP es la adaptación dinámica de la ventana de congestión *cwnd*. Empleando este valor, la entidad TCP *server-side* ajusta dinámicamente su tasa de transmisión para aprovechar eficientemente el ancho de banda disponible. Los objetivos del mecanismo de control de congestión son: detectar el ancho de banda disponible en la red y evitar la congestión de la red. Para cumplir con estos objetivos, el valor de la *cwnd* permite que TCP adapte su tasa de transmisión a las condiciones de la red. El comportamiento de la *cwnd* se divide en dos fases: *slow start* y *congestion avoidance*. La Figura 5.4 muestra la evolución del valor de la *cwnd* en ambas fases y se analiza a continuación.

- **Slow-start (SS):** Durante esta fase la entidad TCP *server-side* prueba el ancho de banda disponible incrementando exponencialmente el valor de la *cwnd*. Al comenzar la transmisión inicial de segmento

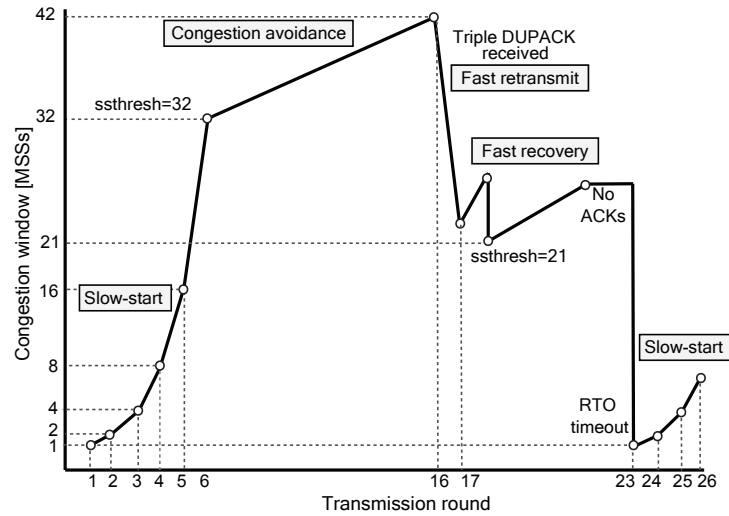


Figura 5.4: Evolución del valor de la ventana de congestión (*cwnd*) durante las fases *slow-start*, *congestion avoidance*, *fast-retransmit* y *fast-recovery*.

el valor de la *cwnd* se inicia en 1 MSS, cada que un mensaje de acuse ACK es recibido, el valor de la *cwnd* se incrementa en un MSS. De esta manera el valor de la *cwnd* se duplica cada ciclo RTT (*Round Trip Time*), donde el RTT es el tiempo desde que un paquete se transmite por el TCP *server-side* hasta que se recibe su ACK. El crecimiento exponencial la *cwnd* se detiene en tres posibles casos: i) si el valor de la *cwnd* llega al valor denominado como el umbral de *slow-start*, es decir, *ssthresh*; ii) si el valor de la *cwnd* es mayor que el valor que el RWS; y iii) se detecta la pérdida de un segmento. En la Figura 5.4 el valor de la *cwnd* inicia en la fase de *slow-start* (*cwnd*= 1 MSS) y se duplica cada RTT hasta la ronda 6, donde el valor de la *cwnd* alcanza el umbral *ssthresh*, establecido en 32 MSSs para este ejemplo.

- Congestion avoidance (CA):** Esta fase cumple con uno de los roles más importantes de TCP. Cuando el valor de la ventana de congestión (*cwnd*) alcanza el umbral de *slow-start* (*ssthresh*), la entidad TCP *server-side* entra en la fase denominada como *congestion avoidance*. Durante esta fase el ajuste de la *cwnd* se define con base en el algoritmo *additive increase multiplicative decrease* (AIMD). De acuerdo con Chiu et al. [122], este algoritmo alcanza un punto de estabilidad con el que TCP opera con un bajo retardo y aprovechando de mejor manera la capacidad de la red. Durante la fase de *congestion avoidance*, TCP prueba de manera gradual el ancho de banda disponible en la red. El valor de la *cwnd* se incrementa de manera fraccionaria, es decir, un $\text{MSS}/cwnd$ bytes cada que se recibe un acuse. De este modo, la *cwnd* incrementa un MSS si todos los paquetes transmitidos son acusados. Así, el valor de la *cwnd* incrementa linealmente en esta fase. En la Figura 5.4, la fase *congestion avoidance* comienza en la ronda seis, es decir cuando el valor de la *cwnd*=32 MSSs. Durante la etapa de *congestion avoidance* el valor de la *cwnd* incrementa linealmente hasta la ronda 16, en este instante TCP detecta la pérdida de paquetes y, por lo tanto, asume que se ha dado un evento de congestión en la red; es decir, se ha consumido el ancho de banda disponible. Para solucionar la congestión TCP debe reducir la tasa de transmisión, es decir, disminuir el valor de la *cwnd*.

Sintetizando, el valor de la *cwnd* durante las fases de *slow-start* (SS) y *congestion avoidance* (CA) se establece con base en las siguientes expresiones:

$$cwnd_{(i+1)} = \begin{cases} cwnd_{(i)} + \text{MSS}, & \text{fase Slow-Start} \\ cwnd_{(i)} + \frac{\text{MSS}}{cwnd_{(i)}}, & \text{fase Congestion Avoidance,} \end{cases} \quad (5.5)$$

donde el índice *i* representa la *i*-ésima ronda de transmisión y cada incremento de la *cwnd* es ocasionado por la recepción de un ACK no duplicado.

TCP interpreta la pérdida de un paquete como un indicador inequívoco de congestión en la red, y para resolver este problema reduce la tasa de transmisión. El algoritmo AIMD determina que la reducción debe ser una fracción del valor de la *cwnd* al momento de detectar la congestión [121]. TCP utiliza dos mecanismos para afrontar las pérdidas: *fast-retransmit* y *fast-recovery*.

- **Fast-retransmit:** TCP establece el temporizador de *Timeout* cada que un paquete es transmitido a la red. Si no recibe el respectivo ACK y el temporizador expira (evento conocido como *retransmission timeout*, RTO) entonces se asume que el paquete se ha perdido en la red y debe retransmitirse. Un evento RTO es considerado por TCP como un signo de congestión severa en la red; cuando este evento ocurre, TCP i) reduce el valor del umbral *ssthresh* a la mitad del valor de la *cwnd* al momento de detectarse la pérdida, ii) iguala el valor de la *cwnd* a un MSS, y, finalmente, iii) reinicia la fase de *slow-start*. Es así como la entidad TCP reinicia el algoritmo de transmisión con una configuración menos agresiva cuyo objetivo es evitar congestionar rápidamente la red.

La detección de pérdidas mediante eventos RTO no es efectiva, debido al tiempo requerido para detectar la pérdida de un paquete. Para solucionar este problema, TCP emplea un mecanismo de detección de pérdidas denominado como *fast-retransmit* con el objetivo es detectar rápidamente la pérdida de un paquete sin necesidad de esperar a que suceda un evento RTO. Si la entidad TCP *client-side* recibe un paquete fuera de orden, es decir, con un número de secuencia que no se esperaba recibir, entonces notifica este evento al TCP *server-side* con un ACK denominado como acuse duplicado, DUPACK. La recepción de un DUPACK en el TCP *server-side* ayuda a detectar la pérdida de un paquete sin necesidad de esperar a que se dé un evento RTO. La recepción de un DUPACK indica: la pérdida de un paquete o bien un paquete que llegó al destino fuera de orden. Si esto sucede, el TCP *client-side* enviará un DUPACK por cada nuevo paquete de datos que reciba. La recepción consecutiva de varios DUPACKs en el TCP *server-side* es interpretada como una señal de la probable pérdida del paquete. El algoritmo *fast-retransmit* determina que la recepción de tres DUPACKs consecutivos indica que el paquete de datos se ha perdido. En tal caso, el TCP *server-side* retransmite el paquete perdido antes de que incluso se dé el evento RTO para ese paquete [123]. En la Figura 5.4 se muestra como en la ronda de transmisión 16 se detecta la posible pérdida de un paquete debido a la recepción de tres DUPACKs. En este instante se procede a la retransmisión inmediata del paquete perdido –de ahí el nombre de este mecanismo, *fast-retransmit*–.

- **Fast-recovery:** Después de haberse efectuado el procedimiento *fast-retransmit* se procede a la ejecución del mecanismo *fast-recovery*. El objetivo de este mecanismo es regular la transmisión de nuevos paquetes de datos hasta dejar de recibir DUPACKs. Durante la ejecución del mecanismo *fast-recovery*, el TCP *server-side* coloca el valor de la *cwnd* en $ssthresh+3$, e incrementa el valor de la *cwnd* un MSS por cada DUPACK recibido. El objetivo de este mecanismo es incrementar artificialmente la *cwnd* con el propósito de compensar la recepción de los paquetes enviados por el TCP *server-side*. Una vez que se recibe un ACK no duplicado, finaliza el mecanismo *fast-recovery* y el TCP *server-side* i) establece el valor *ssthresh* y *cwnd* al valor $cwnd/2$, ii) se reinicia la fase *congestion avoidance*. La Figura 5.12 muestra cómo evoluciona el valor de la *cwnd* durante las etapas *fast-recovery* y *fast-retransmit*. Ambos mecanismos evitan que la *cwnd* reduzca drásticamente, evitando que sucedan eventos RTO. El funcionamiento de *fast recovery* y *fast retransmit* se sintetizan en el Algoritmo 1.

5.2 Retos de desempeño de TCP en iWMNs

Las iWMNs tienen características que representan diversos retos de desempeño para TCP. Por un lado, la topología jerárquica de la red y el uso de enlaces inalámbricos incrementan la contienda por el medio y son frecuentes las pérdidas y errores en el medio inalámbrico, factores que afectan el funcionamiento de los mecanismos de TCP. Por otro lado, el conjunto heterogéneo de aplicaciones de los usuarios tiene altos requerimientos de QoS, esto representa un problema debido a que el ancho de banda disponible en redes iWMNs es limitado y TCP debe distribuirlo equitativamente. La movilidad de los usuarios agrava la situación, ya que pueden presentarse frecuentes pérdidas de conectividad y como consecuencia pérdidas de paquetes, lo cual provoca la ejecución

Algoritmo 1: Mecanismos *fast-retransmit* y *fast-recovery*

```

1: if Se reciben 3 DUPACK then
2:   Retransmitir el paquete "perdido"// Retransmisión rápida antes de un RTO
3:    $ssthresh = cwnd/2$ 
4:    $cwnd = ssthresh + (3 * MSS)$ 
5:   while Reciben un DUPACK do
6:      $cwnd = cwnd + MSS$  // Recuperación rápida de la cwnd
7:   end while
8:    $cwnd = ssthresh$ 
9:   Iniciar fase congestion avoidance
10: end if

```

recurrente de mecanismos de restablecimiento de rutas y un desperdicio constante de los recursos de la red. Además, la variabilidad del RTT en redes inalámbricas reduce la capacidad de TCP para ejecutar el control de congestión.

Además, gran parte del tráfico en iWMNs se concentra en los nodos cercanos a los *Mesh Portals* (MPs) y *Mesh Access Points* (MAP), propiciando la formación de cuellos de botella y, consecuentemente, un incremento en el retardo y pérdidas por congestión en estas zonas de la red. Más aún, la contienda por el medio degrada las prestaciones de la red, ya que a medida que incrementa la densidad de tráfico hay un mayor número de pérdidas por colisiones. Estos problemas aumentarán la variabilidad del retardo y el retardo de extremo a extremo, lo cual impacta directamente en la estimación del producto retardo-ancho de banda de TCP [27]. Ya que la capacidad de la red está estrechamente ligada al: retardo de acceso al medio, el algoritmo de acceso al medio, al número de usuarios que compiten, a la tasa de generación de tráfico y al retardo de propagación en el medio [124].

Sintetizando, en la literatura se han documentado cuatro causas fundamentales que inducen el bajo desempeño de TCP sobre redes inalámbricas [36]:

1. TCP no tiene funciones que permitan identificar la causa de pérdidas de paquetes en redes inalámbricas; ya sea por errores de transmisión, colisiones en el medio, errores de trayectoria o bien congestión en la red. Lo cual propicia que TCP reduzca su tasa de transmisión aun cuando en algunos casos no es necesario.
2. TCP sufre de constantes pérdidas de conectividad causadas por la movilidad de los nodos, propiciando pérdidas de paquetes y reducción de la tasa de transmisión de TCP.
3. La contienda por el medio genera variabilidades en el retardo y pérdidas de paquetes ocasionados por colisiones o por los efectos de la terminal oculta y expuesta. Esto afecta el mecanismo de autorregulación de TCP y por ende la utilización eficiente del medio.
4. La interacción entre TCP y los protocolos de acceso al medio tiende a distribuir inequitativamente el ancho de banda disponible en el medio inalámbrico.

Actualmente existen numerosas propuestas cuyo objetivo es atender los problemas de desempeño de TCP en redes inalámbricas. El rediseño de las funciones de TCP es el enfoque más recurrente en la literatura, a continuación se revisan algunas de las propuestas más representativas. Estos ejemplos nos permiten comprender detalladamente los efectos del medio inalámbrico sobre el desempeño de TCP y ejemplifican diferentes enfoques que permiten mejorar esta interacción.

5.2.1 Modificaciones de TCP para operar en redes inalámbricas

Los retos para el desempeño de TCP en redes inalámbricas son diversos y existen diferentes propuestas para resolverlos. En esta sección se detallan algunas de las propuestas más representativas. Estas propuestas

nos permitirán adquirir una visión más profunda de los diferentes retos de desempeño para TCP en redes inalámbricas y nos permitirán conocer diferentes perspectivas para solucionarlos.

Interacción entre TCP y la capa MAC

En [110], Gerla *et al.* investigan la interacción entre TCP y los mecanismos de la capa MAC en redes inalámbricas *multihop*, así como su efecto en el desempeño de la red. Se seleccionaron para este estudio dos protocolos de acceso al medio, CSMA (*Carrier Sense Multiple Access*) y FAMA (*Floor Acquisition Multiple Access*), cuyas causas más importantes para la degradación de desempeño de TCP son el efecto de pérdidas por la terminal oculta y la inequidad causada por la captura del medio. Mediante la simulación de diversos escenarios de evaluación (es decir, red *multihop* en configuración de cadena, red *multihop* en configuración de anillo y red *multihop* en configuración de malla) se muestra que el desempeño de TCP se degrada considerablemente a medida que la trayectoria de datos pasa por varios *saltos*. Esto se debe particularmente a que los protocolos de acceso al medio acentúan el problema de captura y contienda en el medio, propiciando una mayor cantidad de pérdidas por interferencias entre flujos. Los resultados demuestran que TCP no puede lidiar con estos efectos, aun si se modifica el tamaño máximo de la ventana de congestión; por el contrario, se demuestra que una ventana de congestión mayor a 2 degrada el desempeño de la red. Los resultados han sido incluso validados por Z. Fu y K. Xu en [35] y [125] respectivamente.

La inequidad y captura del medio son ocasionadas por el proceso de *backoff* de los algoritmos de acceso y por la contienda entre los paquetes de datos y los ACKs que utiliza el protocolo TCP. Los resultados demuestran que FAMA ofrece un mejor *throughput* para la red, pero acentúa el problema de captura e inequidad; por otro lado, CSMA ofrece un menor *throughput*, pero resulta más eficiente para evitar los problemas de captura del medio e inequidad. Finalmente, los autores modelan el *throughput*, R , como una función que depende de: el número de *saltos* h de la trayectoria, la probabilidad de pérdidas de un enlace p , y el *throughput* en ausencia de pérdidas R_0 . El modelo propuesto es el siguiente:

$$R = (R_0) * (2^{-\left(\frac{1}{(1-p)^{2h}} - 1\right)}). \quad (5.6)$$

Ajustes de la ventana de congestión

En [126], Fu *et al.* afirman que existe un tamaño óptimo para la ventana de congestión, denominado como W^* , con el cual TCP operaría con una máxima tasa de transmisión sin ocasionar congestión ni pérdidas en una red inalámbrica *multihop*. El valor de W^* es una función del número de saltos por los que atraviesa un flujo TCP en la red *multihop*. Se puede demostrar que TCP no opera ni cercanamente a W^* , sino que incrementa el tamaño de la ventana de congestión por encima de W^* e inevitablemente propicia la pérdida de paquetes, lo cual se traduce en un decremento abrupto de la tasa de transmisión. Uno de los mayores efectos causados por la sobrecarga de la red es la pérdida de paquetes, ocasionadas por el mecanismo de contienda del medio. Este problema impide que TCP opere constantemente con el valor óptimo W^* . Los autores modelan la probabilidad de pérdidas como un proceso Markoviano que depende de los efectos de la terminal oculta y expuesta. Con este modelo se demuestra la estrecha relación que existe entre W^* y los algoritmos de acceso al medio. Se proponen dos alternativas para mantener el valor de la ventana de congestión operando con valores óptimos.

Por un lado, *Link RED* (LRED) mejora: la estimación de la probabilidad de pérdidas para controlar la ventana de congestión W de TCP y la equidad para flujos en contienda. Utilizando la estimación del conteo de retransmisiones (*avg*), LRED utiliza el algoritmo *Random Early Detection* (RED) y estima la probabilidad de colisión de un paquete. Cuando el nivel *avg* supera un umbral predefinido, cada paquete es marcado con una probabilidad de pérdida y este es pasado a la capa de enlace para ser tratado de manera diferenciada. Este proceso controla las pérdidas y permite que TCP adapte su ventana de congestión para operar cerca del punto óptimo W^* . Por otro lado, el algoritmo de *cadencia adaptativa* (*adaptive pacing*) incrementa el tiempo de espera de *backoff* de la capa MAC en un tiempo de transmisión de paquete. Este incremento tiene por objetivo reducir las pérdidas ocasionadas por el efecto de **terminal expuesta**. No obstante, esta propuesta se basa fundamentalmente en el

tamaño de los paquetes de datos para estimar el tiempo de *backoff* agregado, sus efectos no son claramente detallados por los autores.

En [27], Chen *et al.* analizan el problema de ajuste del tamaño máximo de la ventana de congestión de TCP (*Congestion Window Limit - CWL*). Se utiliza como referencia el límite superior del producto retardo-ancho de banda (*Bandwidth Delay Product - Upper Bound - BDP-UB*) para estimar dinámicamente el tamaño máximo de la ventana de congestión. El BDP-UB determina el número máximo de paquetes que pueden fluir en una trayectoria sin competencia entre flujos; este valor representa el *throughput* máximo que se puede alcanzar en dicha trayectoria. Los autores demuestran que este parámetro tiene una estrecha relación con la forma de transmitir en el medio inalámbrico, ya que en el medio solo permite el tránsito de un paquete de datos a la vez. Con base en esta premisa se demuestra que el BDP-UB está limitado por el número de saltos en la trayectoria de ida y vuelta (*round-trip hops*) y por el retardo de transmisión entre cada salto de la trayectoria. Se sabe que en una trayectoria con N saltos intermedios, el máximo número de paquetes de datos que pueden fluir simultáneamente en la trayectoria son N ; cuando este límite se sobrepasa por i paquetes, los i paquetes excedentes serán almacenados en las filas de espera de los nodos de la trayectoria. Esta suposición asume que pueden existir transmisiones concurrentes entre nodos vecinos lo cual no siempre es válido. Para optimizar este modelo, los autores retoman los resultados presentados en [109] en donde se demuestra que el máximo número de transmisiones concurrentes en una cadena de nodos es solo $1/4$ del tamaño N de la cadena. Los autores proponen dos límites del BDP-UP.

Los autores determinan que el BDP-UB estará limitado por kN para $1/8 < k < 1/4$ debido al límite de reutilización en una cadena de nodos y a la interferencia causada por los paquetes ACK del protocolo; esto es validado a través de simulaciones y se define como la **estimación BDP-UP soportada por el número de saltos**.

La **estimación del BDP-UB tomando como base el retardo RTT** es calculada en combinación con el efecto de interferencias entre nodos adyacentes. Los autores proponen que el BDP-UP puede ser estimado como una función de: *i*) el tamaño del paquete, S , *ii*) la suma de retardos en la trayectoria entre transmisor y receptor, d_i , *iii*) la suma de los retardos de la trayectoria de los ACKs, d_j , y *iv*) el retardo máximo, $d_{\max} = \max(d_i, d_j)$. De este modo, el BDP-UB es calculado como:

$$\text{BDP-UB} = S * \left(\sum_{i=0}^m d_i + \sum_{j=0}^n d_j \right) / (4d_{\max}) \quad (5.7)$$

Finalmente, la estimación del BDP-UB se utiliza para establecer el valor límite de CWL. Los resultados presentados afirman que el mecanismo de estimación del BDP-UP soportada por el número de saltos otorga una mejora del 8% al 16% de *throughput* de TCP, mientras que la estimación basada en el retardo ofrece una mejora del 3% al 12%.

Estimación del RTT

En [127], Ming *et al.* proponen un mecanismo de control para TCP que utiliza dos parámetros clave: el RTT de contienda y la desviación del RTT de contienda por salto en la ruta de comunicaciones (*Variance of contention RTT per hop- VCRH*). El primer parámetro sirve para estimar el ancho de banda disponible en la red y el segundo para calcular los niveles de contienda a nivel MAC. Los autores demuestran que el BDP únicamente está determinado por el tiempo de duración de un flujo continuo de paquetes en la ruta de comunicaciones. Considerando esta premisa, **el RTT se divide en dos parámetros: el RTT de contienda y el RTT de congestión**. El primero representa la suma de los retardos ocasionados por la contienda del acceso medio, y el segundo representa los retardos de transferencia de extremo a extremo para un flujo continuo de datos. Con el **RTT de congestión se estima con mayor precisión el BDP**, ya que se puede determinar el RTT asociado al cuello de botella (*bottleneck*) con el valor máximo de este retardo, definido como $D_{\text{congestion_max}}$ y con el tamaño de los paquetes S . Es así como los autores estiman el BDP de la siguiente forma:

$$\text{BDP} = (S/D_{\text{congestion_max}}) \times \text{RTT}_{\text{congestion}} \quad (5.8)$$

A diferencia de la propuesta presentada en [27], esta ecuación no depende de la topología de la red. Los autores, además, proponen un mecanismo de control denominado TCP CWA-CD (Congestion Window Adaptation Through Contention Detection). Con este mecanismo, la capa MAC mide el intervalo de tiempo desde que un paquete llega a la fila de espera hasta que es entregado a la capa PHY (retardo de contienda) y este retardo es anotado como una estampa de tiempo en el paquete de datos. El retardo de contienda se va acumulando cada que el paquete pasa por un nodo en la trayectoria de comunicaciones. Cuando el paquete llega a su destino, el retardo de contienda acumulado es copiado en el ACK respectivo. El mismo procedimiento es realizado con el ACK, y cuando este paquete llega al nodo fuente se puede determinar el RTT de contienda en trayectoria. Para calcular con mayor certeza el RTT de contienda, se estima la varianza de este parámetro (*variance of contention RTT* - VCR) y se divide entre el número de saltos de la trayectoria, estimando así el VCRH. Finalmente, los autores definen el número de muestras n –necesarias para estimar el VCR– mediante un conjunto de ecuaciones recursivas. Para definir el comportamiento de la ventana de congestión de TCP, los autores utilizan el umbral $VCRH_{th}$, empleado para determinar la intensidad de la contienda por el medio. Cada que un ACK es recibido por el nodo fuente, se estima el VCRH y se compara con el umbral $VCRH_{th}$, si $VCRH > VCRH_{th}$ la ventana de congestión disminuye por un MSS. En caso de expirar de darse un evento de *Timeout* se asume que las condiciones de la red son severas, por lo tanto: si $VCRH > VCRH_{th}$, la ventana de congestión se reinicia en $2 * MSS$; por el contrario, si $VCRH < VCRH_{th}$, la ventana de congestión solo se divide a la mitad. Los autores evalúan los efectos de su propuesta al variar $VCRH_{th}$, n y el escenario de simulación. Los resultados que presentan demuestran la eficiencia de su propuesta sobre la versión original de TCP, ofreciendo un mayor *throughput* y un menor retardo de extremo a extremo.

Desconexiones por movilidad

Holland *et al.* [128] investigan los efectos de la pérdida de conectividad en el desempeño de TCP. Los autores definen el límite superior del *throughput* de TCP como el *throughput* esperado (ExT -*Expected Throughput*) y es calculado para una cadena de n nodos estáticos que emplean el algoritmo DCF de IEEE 802.11. El valor del ExT sirve como referencia para determinar los efectos de la movilidad en el desempeño de TCP. Se demuestra que el ExT de una cadena de nodos de longitud i proporciona una capacidad de C/i cuando $1 < i < 3$, y $C/3$ si $i > 3$, lo cual es válido si la capacidad de los enlaces, C , es homogénea. Tomando como referencia este valor, los autores demuestran que, bajo determinados escenarios de movilidad, el *throughput* incrementa a media que también lo hace la velocidad promedio de los nodos; mientras que existen escenarios cuyo *throughput* es sumamente bajo para el mismo valor promedio de la velocidad. Para justificar estos resultados, los autores encuentran que la ineficiencia del protocolo de ruteo (DSR - *Dynamic Source Routing*) para depurar las rutas obsoletas (lo cual causa constantes fallas en las rutas de comunicación) propician un bajo *throughput*. Para los resultados en donde el *throughput* de TCP es elevado a pesar de la movilidad, los autores encuentran que este resultado se debe a que en la mayor parte del tiempo de simulación los nodos fuente y destino se encuentran a corta distancia, ya que existe una alta probabilidad de que el movimiento de los nodos siga un patrón similar debido al algoritmo de movilidad empleado.

Para mejorar el desempeño de TCP, los autores proponen emplear una notificación explícita de fallas en los enlaces de comunicación, denominada como ELFN (*Explicit Link Failure Notification*). Esta notificación requiere de la interacción entre TCP y el protocolo de encaminamiento. Su objetivo consiste en notificar explícitamente las fallas de conectividad en las rutas de comunicación para evitar que las pérdidas de paquetes sean interpretadas como efectos de la congestión en la red. Cada que el protocolo de encaminamiento envía una notificación explícita de error al nodo fuente, también se envía un mensaje ELFN en las cabeceras TCP. Cuando el nodo fuente recibe una notificación ELFN, los temporizadores y las variables de estado del protocolo TCP entran en un periodo de espera, es decir, se mantienen sus valores previos a la recepción de la notificación ELFN. El nodo fuente envía paquetes de prueba para poder detectar la reactivación de la conectividad. En el momento en el que se recibe el acuse de un paquete de prueba, el transmisor reanuda la operación normal de TCP. Los resultados de simulación demuestran que las notificaciones ELFN mejoran el *throughput* de la red, incluso en escenarios con alta movilidad de los nodos.

Pérdidas de paquetes

Con el objetivo de distinguir las pérdidas de paquetes causadas por la congestión de la red de aquellas que son causadas por las propiedades del medio inalámbrico, Lai *et al.* proponen una variante de TCP denominada como TCP para la detección de pérdidas por no-congestión (*TCP for Non Congestive Loss*, TCP-NCL) [129, 81]. Existen dos tipos de señalizaciones que TCP utiliza para la detección de pérdidas: la expiración del *Timeout* (RTO) y la recepción de acuses (ACKs) duplicados (*3 DUPACKs*). TCP considera ambos indicadores como señales de congestión en la red y reduce la tasa de transmisión del protocolo para resolver este problema. Asumir que la recepción de ACKs duplicados es un signo de congestión, puede ser un error en redes inalámbricas, ya que las propiedades del medio inalámbrico pueden ocasionar con mucha probabilidad la llegada desordenada de paquetes de datos en el nodo destino. Los autores proponen utilizar un par de temporizadores serializados para diferenciar las pérdidas ocasionadas por congestión de las que son causadas por errores de transmisión en el medio inalámbrico. El mecanismo inicializa un primer temporizador, RD_i , cada que un paquete, P_i , es enviado en la red. Si el acuse ACK_i se recibe antes que expire el temporizador RD_i , se asume entonces que la red no tiene problemas y se inicia un nuevo temporizador, RD_{i+1} , para el siguiente paquete, P_{i+1} . Por otro lado, si el temporizador RD_i expira antes de recibir el ACK_i , el paquete P_i es retransmitido y se inicializa un segundo temporizador denominado CD_i . Si el ACK_i se recibe antes de que expire el temporizador CD_i , se asume que la red solo atraviesa por pérdidas ajenas a la congestión y se evita que TCP reduzca la tasa de transmisión. Si expira el temporizador CD_i , entonces se asume que la red que ha entrado en un periodo de congestión y se realizan las medidas reactivas del control de congestión típicas de TCP. Para validar la eficiencia de esta propuesta, los autores modelan analíticamente los valores óptimos de RD_i y CD_i con el objetivo de hallar la configuración óptima que se evite las retransmisiones innecesarias y que prevenga la expiración del temporizador del protocolo. La propuesta es evaluada a través de simulaciones, y comparada con otras alternativas tales como TCP-RENO, TCP Westwood, RR-TCP, TCP-DCR, TCP-DOOR, TCP-PR y TCP-Veno. Finalmente, se demuestra que esta propuesta mejora significativamente el desempeño del TCP en diferentes escenarios con enlaces inalámbricos (WLAN, ad hoc y redes cableadas). Cuando se incrementa la tasa de errores *PER Packet Error Rate*, la propuesta puede incrementar hasta en 80% el *goodput* de la red.

Otra solución es presentada por Kulkarni *et al.* [85]. Los autores buscan identificar y diferenciar las pérdidas ocasionadas por la congestión y las pérdidas causadas por errores en la transmisión inalámbrica. Esta propuesta solo requiere la medición de parámetros desde el nodo fuente sin utilizar algún tipo de asistencia de la red o del nodo receptor. Los autores han denominado la propuesta como TCP-E2N; con ella el receptor estima el retardo en una dirección (*OWD-One Way Delay*) utilizando las estampas de tiempo entre el paquete enviado y su respectivo ACK –se requieren las estampas de tiempo descritas en el RFC1323–. El nodo fuente puede definir un umbral para el retardo (*DT-Delay Threshold*) que contemple el OWD ideal y la variabilidad de los retardos de propagación en el medio para estimar un retardo mínimo en la red. Los autores sugieren que el nodo fuente puede predecir el estatus de la red, ya que si se calcula un OWD muy grande puede ser un indicador de inminente congestión en la red. El algoritmo definido por los autores mide constantemente el OWD para cada paquete de datos; en cuanto el nodo fuente recibe un ACK duplicado se asume que ha existido una pérdida. Para determinar la causa de esta solo es necesario comparar el OWD con el DT. Cuando el OWD es mayor que DT se asume que la pérdida es causada por la congestión de la red y TCP-E2E emplea los mismos mecanismos de control de congestión de TCP-Reno. Cuando el OWD es menor que DT, se asume que la pérdida que es causada por errores de transmisión en el medio. Los autores no definen los métodos utilizados para estimar el DT óptimo; la selección de este parámetro tendrá un efecto importante en la equidad del protocolo y en la capacidad alcanzada por el mismo. La propuesta es evaluada en tres escenarios con pérdidas inducidas por tres tipos de errores: de transmisión, por congestión e híbridas. Los resultados demuestran que TCP-E2N es útil en escenarios donde la magnitud y variabilidad del OWD sea pequeña. En caso contrario, la propuesta ofrecerá un desempeño muy similar al que ofrece TCP-Reno, y la ganancia en el desempeño será mínima.

Finalmente en [130], Balakrishnan *et al.* presentan un análisis comparativo de diversas técnicas que mejoran el desempeño de TCP. Las técnicas son clasificadas en tres categorías: protocolos de extremo a extremo (donde el nodo transmisor está al tanto del comportamiento de los enlaces inalámbricos); protocolos con algoritmos pa-

ra mejorar la interacción con la capa de enlace; y protocolos de división de conectividad (diseñados para aislar las secciones de la red inalámbricas de las cableadas para así ocultar los efectos del medio inalámbrico al nodo transmisor). Los autores evalúan 11 diferentes propuestas, entre las que destacan: TCP Reno con notificación explícita de pérdidas (ELN - *Explicit Loss Notification*); TCP Reno con acuses selectivos (LL-SACK); y TCP Reno con división de conectividad y acuses selectivos (SPLIT-SACK). Mediante dos bancos de pruebas (*testbeds*), se evalúan el *throughput* alcanzado por cada variante de TCP en una red de área local y una red de área amplia. Los autores sugieren que existen factores que afectan el desempeño de cada protocolo, sin embargo, el mejor comportamiento lo otorgan las versiones de TCP que utilizan algoritmos para mejorar la interacción con capa de enlace, es decir, LL-TCP-AWARE (*Duplicate ACK Suppression*) y LL-OPT (*SACKS-Duplicate ACK Suppression*). Los resultados obtenidos muestran que estas variantes pueden ofrecer un *throughput* promedio que supera a las otras variantes en un intervalo entre el 10% y el 30%

Las ideas presentadas en esta sección exponen diferentes técnicas cuyo objetivo es mejorar el desempeño de TCP en redes inalámbricas. Podemos concluir que existen diferentes propuestas, entre las que destacan: optimizar el tamaño de la ventana de congestión para aproximarse eficientemente al BDP; el diseño de métodos para discernir entre las pérdidas ocasionadas por la congestión y aquellas ocasionadas por el medio de transmisión; opciones para mejorar las relaciones de desempeño entre TCP y los algoritmos de control de acceso al medio (MAC); y finalmente alternativas para manejar las pérdidas ocasionadas por la movilidad de los nodos. En la siguiente sección se evalúan diferentes variantes de TCP en iWMNs. El objetivo de esta evaluación es indagar y descubrir las dificultades que esta arquitectura de red impone al desempeño de TCP y, al mismo tiempo, identificar aquellas variantes de TCP que ofrecen un mejor desempeño en iWMNs.

5.3 Variantes de TCP en iWMNs

Evaluar el desempeño de TCP en iWMNs coadyuva a comprender cuáles son los retos que impone esta estructura de red jerárquica con enlaces inalámbricos. En esta sección **se evalúan seis variantes de TCP en topologías de red representativas de una iWMN**. A continuación, se analizan las características más significativas de las variantes de seleccionadas, las cuales son: TCP New-Reno, TCP Westwood+, TCP Venó, TCP Vegas, TCP Hybla y TCP Freeze. Estas variantes fueron elegidas por dos razones principales. La primera, por ser variantes de TCP ampliamente analizadas en la literatura y por ser empleadas de manera constante en numerosos estudios de evaluación de desempeño. La segunda, por ser variantes heterogéneas que han sido diseñadas para operar sobre diferentes tecnologías de red, ya sea para redes cableadas, redes inalámbricas e incluso para redes satelitales.

5.3.1 TCP Westwood+

TCP Westwood (TCPW) [131] es una variante de TCP cuyo objetivo es la detección y reacción ante la congestión en la red. TCPW cumple con el paradigma de diseño de extremo a extremo y no requiere de interacciones entre capas de la pila de protocolos. Estas características cumplen con los principios de modularidad de la pila de protocolos y garantiza su funcionamiento sobre redes heterogéneas. TCPW modifica las funciones del TCP *server-side* y tiene por objetivo ajustar la ventana de congestión (*cwnd*) y el umbral de *slow-start* (*ssthresh*) de manera eficiente cuando se detectan pérdidas de paquetes en la red.

TCPW utiliza una variante del mecanismo AIMD denominada como AIAD (*Additive Increase / Adaptive Decrease*). La entidad TCP *server-side* debe estimar dinámicamente el ancho de banda de extremo a extremo y utilizar esta aproximación para discriminar las posibles causas de pérdidas de paquetes, ya sea por congestión o por errores de transmisión. Para poder estimar el ancho de banda en la conexión de extremo a extremo, la entidad TCP *server-side* monitoriza la tasa de recepción de ACKs. Esto significa que cuando se recibe un ACK en el instante de tiempo t_k , se asume que una cantidad de datos d_k ha llegado al TCP *client-side*, consecuentemente, se estima una muestra instantánea del ancho de banda b_k empleando el tiempo inter-arribos de los paquetes ACK, Δ_k . Si el ACK previo fue recibido en el instante t_{k-1} y el ACK actual se recibe en el instante t_k , entonces la muestra instantánea del ancho de banda b_k en el instante t_k estará dada por:

$$b_k = \frac{d_k}{t_k - t_{k-1}} = \frac{d_k}{\Delta_k}. \quad (5.9)$$

TCPW utiliza un filtro pasa-bajos discreto para promediar dinámicamente las muestras b_k del ancho de banda disponible. Este cálculo es de suma importancia ya que la congestión ocurre cuando la tasa de tráfico en bajas frecuencias excede el ancho de banda disponible en la trayectoria de comunicaciones [132, 133]. Utilizando la aproximación de Tustin, los autores proponen discretizar un filtro continuo pasa-bajos para obtener la siguiente aproximación del ancho de banda promedio disponible \hat{b}_k en el instante k ,

$$\hat{b}_k = \alpha_k \hat{b}_{k-1} + (1 - \alpha_k) \left(\frac{b_k + b_{k-1}}{2} \right), \quad (5.10)$$

donde

$$\alpha_k = \frac{(2\tau - \Delta_k)}{(2\tau + \Delta_k)}, \quad (5.11)$$

y donde $1/\tau$ es la frecuencia de corte del filtro y $\Delta_k = t_k - t_{k-1}$ es el tiempo inter-arribos de los ACKs.

Si el tiempo inter-arribos Δ_k incrementa, entonces α_k decrementa y, consecuentemente, el peso que multiplica al promedio de las dos muestras del ancho de banda, es decir, $(b_k + b_{k-1})/2$, incrementa, mientras que el peso de la estimación promedio del ancho de banda en el instante $k-1$, es decir, \hat{b}_{k-1} , decrementa. TCPW emplea la estimación promedio del ancho de banda en el instante, \hat{b}_k , para adaptar de manera eficiente los valores de la ventana de congestión, $cwnd$, y el umbral de *slow-start*, $ssthresh$, cuando se detecta la pérdida de un paquete. Los ajustes de la $cwnd$ y al $ssthresh$ tienen el objetivo de incrementar la velocidad de recuperación ante las pérdidas y ajustar con mayor precisión la tasa de transmisión de TCP. De este modo, TCPW evita la drástica reducción de la tasa de transmisión que realiza el mecanismo *fast-recovery* de TCP, es decir, $cwnd/2$. Para concluir, TCPW mantiene la operación tradicional de los mecanismos *slow-start* y *congestion avoidance* establecidos para TCP.

Faster Recovery

Cuando TCPW detecta n ACKs duplicados (DUPACKs) puede significar dos cosas diferentes: este evento se ha dado por una pérdida asociada a la congestión de la red o bien por una pérdida esporádica asociada a un error de transmisión en el medio inalámbrico. En cualquiera de los dos casos, TCPW utiliza el algoritmo denominado como *faster recovery*. Este algoritmo manifiesta claramente el decremento adaptivo soportado por la estimación dinámica del ancho de banda promedio de extremo a extremo, \hat{b}_k . Los ajustes que realiza TCPW a los valores $cwnd$ y $ssthresh$ al detectar n DUPACKs se describen en el Algoritmo 2. Se debe considerar que RTT_{\min} es la muestra más pequeña del RTT de la conexión TCPW. Se aprecia que el $ssthresh$ se ajusta al valor $(\hat{b}_k * RTT_{\min})/TCP_{\text{SegmentSize}}$, es decir, se reduce de manera adaptativa. Lo mismo sucede con la ventana de congestión, $cwnd$, si su valor al momento de recibir n DUPACKs es mayor que el $ssthresh$, entonces se reduce de manera adaptativa al valor $(\hat{b}_k * RTT_{\min})/TCP_{\text{SegmentSize}}$; en caso contrario se mantiene su valor.

Por otro lado, cuando TCPW detecta paquetes perdidos mediante la expiración del *Timeout* (RTO) la pérdida es interpretada como indicador de congestión y se deben ajustar valores de $cwnd$ y $ssthresh$ con base en el Algoritmo 3. Se aprecia que el valor del $ssthresh$ se ajusta tomando en consideración la estimación \hat{b}_k y en este caso la $cwnd$ se reduce a 1. Los autores han evaluado TCPW empleando simulaciones computacionales y un banco de pruebas (*testbeds*), los resultados demuestran que TCPW ofrece un mejor *throughput* que el que se obtiene con las versiones TCP Reno y TCP SACK en redes con enlaces inalámbricos con tasas de pérdidas variables.

Para finalizar, en [134], Ferorelli *et al.* muestran que el efecto de compresión de los ACKs genera componentes de alta frecuencia que no pueden ser filtrados por el filtro utilizado en TCPW (Ecuación 5.10), esto provoca la sobre estimación del ancho de banda promedio \hat{b}_k . Para solucionar este problema, los autores proponen modificar la estimación dinámica del ancho de banda instantáneo b_k tomando como referencia el RTT en

Algoritmo 2: *Faster recovery* tras la detección de n DUPACKS

```

1: if  $n$  DUPACKs received then
2:    $ssthresh = (\hat{b}_k * RTT_{\min}) / TCP_{\text{SegmentSize}}$  // Ajuste con base en la estimación  $\hat{b}_k$ 
3:   if  $cwnd > ssthresh$  then
4:      $cwnd = ssthresh$ 
5:   end if
6: end if

```

Algoritmo 3: Detección de pérdidas por la expiración del temporizador (RTO)

```

1: if RTO event then
2:    $ssthresh = (\hat{b}_k * RTT_{\min}) / TCP_{\text{SegmentSize}}$  // Ajuste con base en la estimación  $\hat{b}_k$ 
3:   if  $ssthresh < 2$  then
4:      $ssthresh = 2$ 
5:   end if
6:    $cwnd = 1$ 
7: end if

```

lugar del tiempo inter arribo de los ACKs Δ_k . En esta propuesta los autores estiman la muestra b_k utilizando la cantidad total de datos acusados D_k durante el último periodo RTT. Esta modificación es lo que se conoce en la literatura como TCP Westwood+ (TCPW+). La estimación instantánea del ancho de banda se redefine como se muestra en la ecuación 5.12 y se conservan los mecanismos propuestos por TCPW.

$$b_k = \frac{D_k}{\Delta_k} = \frac{D_k}{RTT} \quad (5.12)$$

5.3.2 TCP Venó

TCP Venó (TCPVN) [135] modifica los mecanismos que realiza la entidad TCP *server-side*. TCPVN adapta el algoritmo AIMD (*Additive Increase/Multiplicative Decrease*) para reaccionar eficientemente ante los diferentes tipos de pérdidas que pueden presentarse en redes inalámbricas. TCPVN estima el nivel de congestión de la red para poder discernir si las pérdidas de paquetes son propiciadas por eventos de congestión o por errores de transmisión en el medio inalámbrico. Además, TCPVN utiliza la estimación del nivel de congestión para mantener la tasa de transmisión cerca de la zona de congestión durante un mayor tiempo. Esto permite que TCPVN pueda maximizar el *throughput* que obtiene en la red.

Para realizar la estimación del nivel de congestión, TCPVN utiliza una medida denominada como *backlog*, N , definida como:

$$N = cwnd \left(1 - \frac{\text{Base RTT}}{RTT} \right), \quad (5.13)$$

donde Base RTT, es el RTT mínimo detectado en la conexión TCP. Cada que se detecta la pérdida de un paquete, ya sea por un evento de expiración del *Timeout* o por la recepción de n DUPACKS, el valor de Base RTT es reiniciado y recalculado hasta que una nueva pérdida es detectada. Si $RTT > \text{Base RTT}$, se infiere que ha surgido un cuello de botella en la trayectoria de comunicaciones en donde los paquetes se están acumulando. Si este es el caso, el valor N se utiliza para determinar si TCPVN se encuentra operando en una etapa de inminente congestión en la red. TCPVN emplea un umbral, definido como β , para identificar esta etapa crítica. Con base en Fu et al. [135], un valor apropiado para el umbral β es tres. Si $N > \beta$, TCPVN asume que la red opera en una etapa crítica y es probable que se congestione. Con base en esta inferencia, TCPVN adapta de manera eficiente la tasa de transmisión y también puede determinar la causa de las pérdidas en la red.

Algoritmo Additive Increase (AI)

TCPVN modifica el algoritmo *additive increase* (AI) durante la fase de *congestion avoidance*. El objetivo de esta modificación es que TCPVN mantenga la transmisión de datos operando en una región en donde se aproveche al máximo el ancho de banda disponible. Cuando N es mayor que el umbral β , la ventana de congestión $cwnd$ incrementa en una unidad por cada dos RTTs, de modo que la conexión se mantiene operando en la región de inminente congestión un mayor tiempo que lo que normalmente lo haría TCP. Así, TCPVN experimenta un menor número de oscilaciones de la $cwnd$ por que difiere las pérdidas auto inducidas que normalmente realiza TCP. Así se mantiene la conexión en la región de óptima tasa de transmisión. En otras palabras, TCPVN incrementa el valor de la $cwnd$ aun cuando se detecta una inminente congestión, pero el incremento de este parámetro se realiza lentamente. El algoritmo AI de TCPVN se sintetiza en el Algoritmo 4.

Algoritmo Multiplicative Decrease (MD)

Si TCPVN detecta un paquete perdido por un evento RTO, la pérdida se interpreta como una señal inequívoca de una congestión severa en la red; consecuentemente, TCPVN reacciona empleando el mismo algoritmo propuesto por TCP, es decir, el valor $ssthresh$ se ajusta a $cwnd/2$ y la ventana de congestión se reinicia, es decir, $cwnd = 1$. Si la pérdida de un paquete se detecta mediante la recepción de n DUPACKs, TCPVN modifica el ajuste del umbral de *slow-start*. Si la detección de la pérdida ocurre fuera de la región de inminente congestión, es decir, $N < \beta$, el algoritmo *fast-recovery* de TCP se emplea normalmente, pero la reducción del $ssthresh$ se realiza de manera menos agresiva para evitar disminuir drásticamente la tasa de transmisión. En este caso, dado que la pérdida del paquete ha sido ocasionada por factores ajenos a la congestión de la red, el valor $ssthresh$ se ajusta a $4/5$ del valor de $cwnd$. Si la detección de la pérdida ocurre dentro de la región de inminente congestión, es decir, $N \geq \beta$, la pérdida se asume que fue causada por congestión en la red y el $ssthresh$ se ajusta tal como normalmente se ajusta por TCP, es decir, $ssthresh = cwnd/2$. En el Algoritmo 5 se sintetizan los ajustes realizados por TCPVN cuando se detecta una pérdida por n DUPACKs recibidos.

Algoritmo 4: TCP Veno Additive Increase (AI)

```

1: if  $cwnd \geq ssthresh$  then
2:   if  $N < \beta$  then
3:      $cwnd = cwnd + \frac{1}{cwnd}$  // Ancho de banda disponible
        // Cada que se recibe un nuevo ACK
4:   else
5:      $cwnd = cwnd + \frac{1}{cwnd}$  // Ancho de banda totalmente utilizado
        // Cada que un acuse distinto se recibe
6:   end if
7: end if

```

Algoritmo 5: TCP Veno Multiplicative Decrease (MD)

```

1: if  $n$  DUPACKs then
2:   if  $N < \beta$  then
3:      $ssthresh = \frac{4}{5}cwnd$  // Pérdida aleatoria
4:   else
5:      $ssthresh = \frac{cwnd}{2}$  // Pérdida por congestión
6:   end if
7: end if

```

5.3.3 TCP Vegas

TCP Vegas (TCPV) [136] modifica funciones de la entidad TCP *server-side*. TCPV utiliza dos técnicas para incrementar el *throughput* y para reducir las pérdidas. Por un lado, un mecanismo de retransmisión que reduce el tiempo necesario para detectar una pérdida. Esta reducción permite detectar pérdidas con un menor número de DUPACKs ya que basta con recibir un DUPACK para asumir que un paquete se ha perdido en la red. Por otro lado, TCPV incorpora un mecanismo en la etapa *congestion avoidance* que mantiene una tasa de transmisión óptima. Este mecanismo proporciona dos beneficios importantes: *i*) evita la congestión de la red y *ii*) aprovecha eficientemente el ancho de banda disponible en la red. Para alcanzar estos objetivos, TCPV usa la diferencia entre el *throughput* esperado (*Expected throughput*) y el *throughput* actual para estimar el ancho de banda disponible. Esta estimación permite que TCPV evite la congestión ya que permite actualizar eficientemente el valor de la *cwnd*.

Mecanismo de retransmisión

TCPV mide el RTT de cada paquete para calcular el tiempo de *Timeout*. TCPV emplea este *Timeout* para determinar si se debe realizar la retransmisión de un paquete. La retransmisión puede darse en dos posibles casos, los cuales se enlistan a continuación.

- Cuando TCPV recibe un DUPACK verifica si la diferencia entre el RTT actual y el RTT anterior es mayor que el valor de *Timeout*; si es así, entonces TCPV retransmite el paquete sin necesidad de esperar hasta un tercer DUPACK, tal como lo hace TCP.
- Cuando TCPV recibe un ACK, después de una retransmisión, verifica si el intervalo de tiempo entre el momento en el que el paquete fue retransmitido y el ACK es recibido es mayor que el valor de *Timeout*; si es así, TCPV retransmite de nueva cuenta el paquete.

En caso de presentarse múltiples pérdidas y más de una retransmisión de paquetes, el valor de la *cwnd* se reduce únicamente al valor en el que se dio la primera retransmisión.

Mecanismo Congestion Avoidance

TCPV utiliza un mecanismo proactivo para detectar y evitar la congestión en la red. TCPV define el valor Base RTT, como el mínimo RTT detectado a lo largo de la conexión. Así, el *throughput* esperado se calcula como:

$$\text{EXP_Throughput} = \frac{cwnd}{\text{Base RTT}}. \quad (5.14)$$

TCPV calcula el *throughput* actual (ACT_Throughput) registrando el instante de tiempo t_i en el que se transmite el segmento i , y cuantos segmentos son transmitidos desde el instante t_i hasta el momento de la recepción del ACK del segmento i . Con estos datos, el ACT_Throughput es calculado como el valor de la *cwnd* dividida por el RTT del segmento i , es decir, $\text{ACT_Throughput} = cwnd/\text{RTT}$. TCPV estima la diferencia entre el *throughput* esperado y el *throughput* actual para poder calcular así el ancho de banda disponible de extremo a extremo. La diferencia se estima como:

$$\text{Diff} = \text{EXP_Throughput} - \text{ACT_Throughput} = \frac{cwnd}{\text{Base RTT}} - \frac{cwnd}{\text{RTT}}. \quad (5.15)$$

Finalmente, TCPV establece dos umbrales: α and β . Estos umbrales determinan hay pocos o muchos datos transitando en la red. De esta forma el valor de la *cwnd* se ajusta de manera que se maximiza la utilización del ancho de banda disponible en la red. La *cwnd* incrementa de manera lineal si hay pocos datos transitando en la red, es decir, $\text{Diff} < \alpha$, y decrementa de manera lineal si existen muchos datos en la red, es decir, $\text{Diff} > \beta$. La manera en la que se ajusta la ventana de congestión se sintetiza en el Algoritmo 6.

Algoritmo 6: TCP Vegas Congestion Avoidance

```

1: Diff = EXP_Throughput – ACT_Throughput
2: if Diff <  $\alpha$  then
3:    $cwnd = cwnd + 1$  // Incremento lineal de la ventana de congestión
4: else if Diff >  $\beta$  then
5:    $cwnd = cwnd - 1$  // Decremento lineal de la ventana de congestión
6: else
7:    $cwnd = cwnd$  // La ventana de congestión se mantiene constante
8: end if

```

Mecanismo Slow-Start

TCPV utiliza mecanismos de la etapa de *congestion avoidance similares* para la etapa de *slow-start*. Para tener comparaciones validas entre el valor EXP_Throughput y ACT_Throughput, la ventana de congestión solo puede incrementar su valor después de un RTT durante la etapa de *slow-start*.

5.3.4 TCP Hybla

TCP Hybla (TCPH) [137] incorpora modificaciones a la entidad TCP *server-side*. TCPH mejora el desempeño tradicional de TCP en redes cuyos enlaces tienen mayores tiempos de propagación, esto se logra eliminando la dependencia del RTT. TCPH ha sido diseñado para optimizar sistemas de comunicaciones inalámbricas en los que el retardo RTT es lo suficientemente grande como para degradar la tasa de crecimiento de la ventana de congestión de TCP. Una trayectoria de comunicaciones en la que los enlaces tienen largos tiempos de propagación afecta la tasa de crecimiento de la *cwnd*, es decir, la etapa de *slow-start* tiene una mayor duración, propiciando que TCP no aproveche rápidamente el ancho de banda disponible en la red, y, por otro lado, durante la etapa de *congestion avoidance* la *cwnd* crece lentamente hasta detectar la congestión de la red. TCPH modifica los mecanismos de *congestion control* y obtiene en redes con un RTT alto RTT la misma tasa de transmisión que la que se puede obtener en una conexión TCP con RTTs menores. Además, TCPH elimina la transmisión en ráfagas mediante técnicas de distribución de transmisiones, reduciendo así la probabilidad de congestionar la red.

Evolución de la ventana de congestión

La evolución de la ventana de congestión (*cwnd*) puede analizarse como una función que depende del tiempo t , es decir, $cwnd(t)$. La evolución tiene dos etapas bien definidas. Asuma que el umbral de *slow-start* (*ssthresh*) se define como γ y $t_\gamma = RTT \times \log_2(\gamma)$ es el instante de tiempo donde la *cwnd* alcanza el valor del *ssthresh*, es decir, $cwnd(t_\gamma) = \gamma$; así la evolución de la *cwnd* como función de t , de t_γ y del RTT, se define como se muestra en la ecuación 5.16.

$$cwnd(t) = \begin{cases} 2^{t/RTT}, & 0 < t < t_\gamma, \quad \text{Fase de Slow – Start} \\ \frac{t - t_\gamma}{RTT} + \gamma, & t > t_\gamma, \quad \text{Fase de Congestion Avoidance.} \end{cases} \quad (5.16)$$

Por consiguiente, el ancho de banda disponible en el instante t se calcula como:

$$B(t) = \frac{cwnd(t)}{RTT}. \quad (5.17)$$

Para eliminar la dependencia de la tasa de crecimiento de $cwnd(t)$ con el RTT, se propone una modificación de la escala de tiempo, introduciendo un valor RTT normalizado y definido como:

$$\rho = \frac{RTT}{RTT_0}. \quad (5.18)$$

Donde RTT_0 es el tiempo de referencia de una conexión TCP más rápida. Para eliminar la dependencia de la $cwnd(t)$, el instante de tiempo t , de la ecuación 5.16, es escalado por el valor RTT normalizado, ρ . La ventana de congestión resultante, $cwnd(\rho \times t)$, es compensada con el peso $\rho[cwnd(\rho \times t)]$. Como resultado de este proceso de escalamiento, el instante de tiempo en el que $cwnd(t)$ alcanza el valor escalado $\rho\gamma$ ahora depende de RTT_0 y es definido como $t_{\gamma,0} = RTT_0 \times \log_2(\gamma)$. De este modo, la evolución de la ventana de congestión para TCPH, es decir, $cwnd^H(t)$, se define como

$$cwnd^H(t) = \begin{cases} \rho 2^{\rho t / RTT}, & 0 < t < t_{\gamma,0} \\ \rho \left[\rho \frac{t - t_{\gamma,0}}{RTT} + \gamma \right], & t > t_{\gamma,0}. \end{cases} \quad (5.19)$$

Con la ventana de congestión de TCPH, $cwnd^H(t)$, se puede obtener un ancho de banda disponible en el instante t , es decir, $B^H(t)$, que ahora es independiente del valor RTT, tal y como se muestra en la ecuación 5.20.

$$B^H(t) = \begin{cases} \frac{2^{t/RTT_0}}{RTT_0}, & 0 < t < t_{\gamma,0} \\ \frac{1}{RTT_0} \left[\frac{t - t_{\gamma,0}}{RTT_0} + \gamma \right], & t \geq t_{\gamma,0}. \end{cases} \quad (5.20)$$

Para concluir, la evolución de la ventana de congestión de TCPH, es decir, $cwnd^H(i)$, se define como se muestra en las ecuaciones 5.21.

$$cwnd_{i+1}^H = \begin{cases} cwnd_i^H + 2^\rho - 1, & \text{Fase de Slow - Start} \\ cwnd_i^H + \rho^2 / cwnd_i^H, & \text{Fase de Congestion Avoidance.} \end{cases} \quad (5.21)$$

Mecanismo SACK

El mecanismo de control de congestión propuesto por TCPH da como resultado un valor grande para la $cwnd$; así que es probable que se ocasionen pérdidas de paquetes en la misma ventana de transmisión. Por lo tanto, TCPH utiliza la opción de acuse selectivo (SACK) para notificar a la entidad TCP *server-side* la recepción de los paquetes de datos [138]. Por ejemplo, si no se utiliza la opción SACK, cuando la entidad TCP *client-side* recibe los paquetes 1, 2 y 3 y luego el paquete 5, solo acusará haber recibido hasta el paquete 3 y descartará todos los paquetes siguientes hasta que reciba el paquete 4. Esto provoca una retransmisión masiva de paquetes que incluso pudieron haber sido recibidos por la entidad TCP *client-side*. Cuando se emplea el mecanismo SACK, la entidad TCP *client-side* acusará los paquetes a medida que se reciben, de manera que la entidad TCP *server-side* es capaz de determinar qué paquetes se han perdido y únicamente retransmitir dichos paquetes. Por ejemplo, si la entidad TCP *client-side* recibe los paquetes 1, 2 y 3 y luego recibe los paquetes 5 y 6, se acusan todos los paquetes recibidos de manera la entidad TCP *server-side* solo tiene que retransmitir el paquete 4 perdido en lugar de tener que volver a transmitir los paquetes 4, 5 y 6. Es así como la entidad TCP *server-side* identifica y retransmite varios paquetes perdidos en un RTT.

Inicialización del umbral de Slow-Start

El valor inicial del umbral $ssthresh$ es crítico para el funcionamiento de TCPH; por lo tanto, este protocolo requiere que el umbral $ssthresh$ se aproxime de manera eficiente al ancho de banda disponible en la red. Para realizar esta aproximación, TCPH utiliza los mensajes iniciales que se emplean para el establecimiento de la conexión (SYN-SYN ACK-ACK). Con estas transmisiones se obtiene la aproximación inicial del RTT. Una vez que el proceso de conexión ha inicializado, el ancho de banda disponible se estima utilizando la estimación de mínimos cuadrados con empleando tres ACKs [139]. Derivado de esta estimación, TCPH es capaz de ajustar óptimamente el valor inicial del $ssthresh$.

Distribución de transmisiones

El ancho de banda disponible para TCPH está definido por $B^H(t) = cwnd^H(t)/RTT$; por lo tanto, para alcanzar un *throughput* alto en conexiones con un mayor RTT, es necesario utilizar una *cwnd* grande. Esto trae como consecuencia un alto número de transmisiones en ráfaga, lo cual es un factor determinante que puede ocasionar congestión en la red. Las transmisiones en ráfaga que se generan al incrementar artificialmente el tamaño de la *cwnd* se soluciona distribuyendo las transmisiones de los paquetes de datos en un RTT. Esta técnica es comúnmente denominada como *packet spreading*. TCPH puede utilizar dos variantes muy conocidas de esta estrategia, *uniform spacing* y *progressive spacing*.¹

5.3.5 TCP Freeze

TCP Freeze (TCPF) [140] es una variante que lidia con los problemas asociados a la movilidad y a la desconexión temporal de una sesión de transferencia de datos. TCPF tiene por objetivo la recuperación rápida de una conexión para reducir su impacto en la tasa de transmisión de datos. Se sabe que los mecanismos de TCP pueden desperdiciar el ancho de banda disponible en la red cada que se presenta un evento de pérdida de conectividad, incluso si los dispositivos recuperan la conectividad inmediatamente. Empleando TCPF, la entidad *client-side* monitoriza la potencia de la señal en la antena de transmisión del dispositivo y es capaz de predecir una inminente desconexión inalámbrica a medida que desvanece la potencia de la señal. Ante este evento, la entidad TCPF *client-side* adjunta a un ACK información en donde notifica que su *buffer* de recepción se ha llenado, evento denominado como *zero-reception window size* (ZWA). Esta notificación se realiza con el objetivo de que la entidad TCPF *server-side* opere en **modo persistente**.

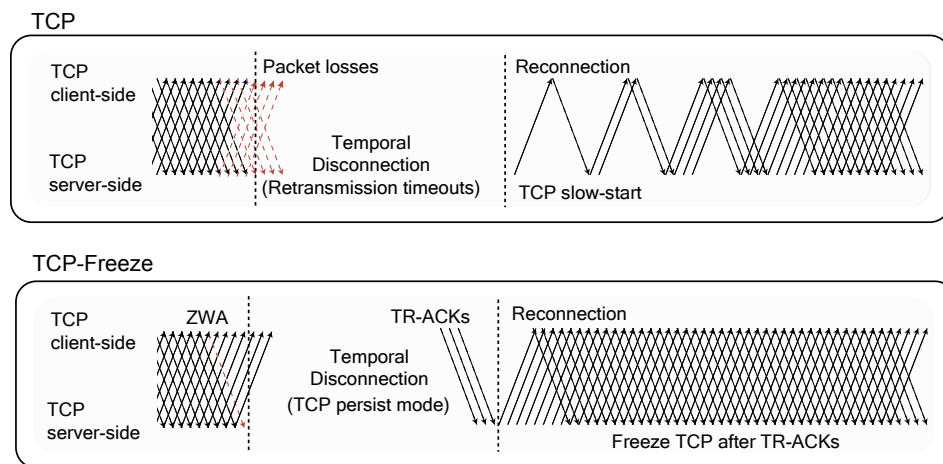


Figura 5.5: TCP Freeze evita que se ejecute la fase de *slow-start* después de un evento de reconexión.

Cuando la entidad TCPF *server-side* entra en **modo persistente** mantiene los valores de los temporizadores de retransmisión (*Timeout*) y la ventana de congestión (*cwnd*). De manera que, al salir de este modo persistente, la fase de *slow-start* se evita. Por lo tanto, TCPF es de mucha utilidad en escenarios en donde existe movilidad y se dan frecuentes pérdidas de conectividad. La entidad TCPF *client-side* debe notificar a la entidad TCPF *server-side* la inminente desconexión para garantizar que al menos un ZWA podrá ser recibido por la entidad TCPF *server-side*. Si la entidad TCPF *server-side* recibe un ZWA antes de que ocurra la desconexión, entrará en modo persistente prematuramente, provocando que la transmisión de datos se detenga antes de que ocurra la desconexión del enlace de comunicaciones. Por otro lado, si la entidad TCPF *server-side* no recibe a tiempo el ZWA, múltiples paquetes se perderán durante la pérdida de conectividad. Por lo tanto, un tiempo razonable para enviar el ZWA es el RTT, así que la entidad TCPF *client-side* debe estimar con precisión el tiempo de

¹Los detalles de operación de estas y otras técnicas de *packet spreading* se encuentran en [137]

desconexión para garantizar que el ZWA llegará a la entidad TCPF *server-side*. Tan pronto como la conexión es restablecida, la entidad TCPF *client-side* debe enviar tres copias del ACK del último segmento de datos recibido, este mecanismo es definido como *Triplicate Reconnection ACKs* (TR-ACKs). Con el envío de TR-ACKs se evita el tiempo de *backoff* exponencial después de recibir un ZWA. La Figura 5.5 muestra como TCPF evita la etapa de *slow-start* después de presentarse un evento de desconexión y de las pérdidas asociadas.

5.4 Evaluación de desempeño

En esta sección se evalúa el desempeño de las variantes de TCP denominadas como TCP New-Reno (TCPNR), TCP Westwood+ (TCPW), TCP Veno (TCPVN), TCP Vegas (TCPV), TCP Hybla (TCPH) y TCP Freeze (TCPF) en iWMNs. Para la realización de esta evaluación se han seleccionado diferentes topologías de red que nos permiten comprender el desempeño general de los diferentes protocolos de la capa de transporte en iWMNs. A través de simulaciones computacionales podemos comprender las interacciones existentes entre la infraestructura de red y los protocolos analizados, así como sus efectos en el desempeño de ambos. El análisis de los resultados nos permitirá identificar las características de la red y de los protocolos que pueden ser mejoradas para optimizar el desempeño de TCP en entornos inalámbricos altamente dinámicos, tal y como sucede en iWMNs. Además, estos resultados sirven como preámbulo para la implementación y análisis de desempeño de la técnica de codificación de red en iWMNs.

5.4.1 Análisis de desempeño de TCP en iWMN en topología de cadena

Descripción del escenario general

Una iWMNs tiene red inalámbrica de distribución denominada comúnmente como el *backbone* de la red. Asuma que R es la región de cobertura de la iWMN y que dicha área es de forma cuadrada y cubre una superficie de $A [m^2]$. El número de usuarios móviles que utilizan la red se considera variable, sin embargo se puede asumir que se distribuyen y se mueven uniformemente a través de la región de cobertura R ; de forma tal que el número de usuarios N por unidad de superficie A , se mantiene constante a través del tiempo, es decir $P[N_A(t) = c; \forall A] \simeq 1$. La región de cobertura R del *backbone* puede dividirse en n subregiones cuadradas de cobertura, las cuales son consideradas disjuntas y exhaustivas. La i -ésima subregión se define como r_i , de modo tal que las n subregiones satisfacen simultáneamente las siguientes condiciones:

$$\bigcup_{i=0}^n r_i = R \quad (5.22)$$

$$r_i \cap r_j = 0, \quad \forall i \neq j \quad (5.23)$$

Cada subregión r_i tiene un punto de acceso (*Mesh Access Point* - MAP) que ofrece conectividad a los usuarios (*STAs*) localizados en dicha subregión, tal como se ilustra en la Figura 5.6. Los MAPs se organizan en forma de rejilla en la región de cobertura R y cada MAP tiene una superficie de cobertura de $a [m^2]$, donde $a = A/n [m^2]$. Por lo tanto, con muy alta probabilidad, en cada subregión r_i se proporciona conectividad inalámbrica a $N_{A/n}(t)$ usuarios móviles. Cada MAP debe conectarse con al menos una *Mesh Station* para poder establecer una ruta de comunicación *multihop* hacia un *Mesh Portal* (MP). De este modo, el tráfico generado por las *STAs* puede ser encaminado hacia el MP, el cual ofrece conectividad con otras arquitecturas de red, p. ej., Internet. A través de k *Mesh STAs* intermedios es como cada paquete de datos puede ser encaminado desde un MAP hasta el MP, esto quiere decir que la ruta $STA \rightarrow MAP \rightarrow Mesh\ STA \rightarrow MP \rightarrow Internet$ puede ser analizada como una topología en cadena con K saltos.

Descripción del escenario ideal

Con el objetivo de simplificar las variables involucradas en el escenario de simulación podemos asumir que el protocolo de encaminamiento utilizado en la iWMN seleccionará la mejor trayectoria de comunicaciones entre el nodo fuente y el destino. Esta ruta puede tener el menor número de saltos o bien niveles mínimos de

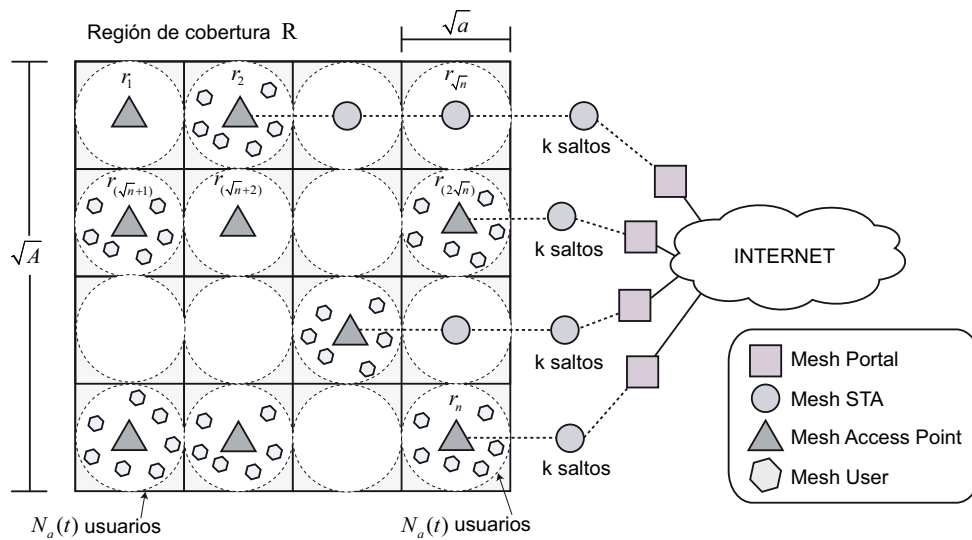


Figura 5.6: Escenario general de evaluación.

congestión. En un escenario ideal, la única contienda por el medio en la ruta seleccionada se tendrá que dar únicamente entre los usuarios móviles y el MAP, así como entre las *Mesh STA* que componen la trayectoria de comunicaciones. Es así como cada ruta se conformará al menos por un MAP, k *Mesh STAs*, y un MP conectado de manera cableada a Internet.

El primer escenario de evaluación está constituido por una red en cadena basada en la tecnología IEEE 802.11. La evaluación de desempeño se realizará con el simulador de eventos discretos *ns2* (ver. 2.35). Este primer escenario de evaluación toma como punto de referencia los escenarios propuestos por [110, 27, 126, 128]. Es importante señalar que en el simulador *ns2* se considera que cada dispositivo inalámbrico tiene asociados tres umbrales numéricos que definen tres regiones espaciales [118], estos son los siguientes.

- El umbral de detección de portadora (*Carrier Sense Threshold - CStresh*) define la energía mínima de detección que permite asumir si el medio está ocupado o vacío; a este valor se le asocia el alcance de detección de portadora (R_{cs}), este límite define la distancia máxima sobre la cual un nodo puede detectar la actividad en el medio.
- El umbral de recepción (*Receive Threshold - RXThresh*) define un umbral numérico que determina el punto en el que se puede realizar la recepción correcta de un paquete de datos, asociado a este valor se define el alcance de transmisión (R_{tx}), el cual se define como la distancia máxima sobre la cual se puede llevar a cabo una transmisión exitosa entre un par de nodos.
- El umbral de captura (*Capture Threshold - CPTresh*) determina un nivel mínimo de la calidad de la señal recibida. Este valor utiliza la relación SIR (*Signal-to-Interference Ratio*) para validar si un paquete recibido se considera como un paquete válido o bien como una colisión ocasionada por interferencia, a este umbral numérico se le asocia un límite máximo que se denomina el alcance de interferencia (R_{ir}).

Para el primer escenario de evaluación se configuran los siguientes parámetros. El alcance máximo de transmisión R_{tx} se establece en 250 [m], mientras que los alcances de interferencia R_{ir} y de detección de portadora R_{cs} se fijan en 550 [m]. La distancia entre los nodos adyacentes en la trayectoria de comunicación es de 200 [m]. De esta forma, cada nodo de la trayectoria puede detectar la actividad a dos nodos de distancia y disminuir con esto los efectos de la terminal oculta. El escenario y los detalles de su configuración se sintetizan en la Figura 5.7.

En este escenario se utiliza el modelo físico *Two Ray Ground* para modelar la potencia de transmisión entre un par de nodos. Para evaluar el desempeño de la red creamos dos categorías de flujos de datos que representan un efecto más realista del flujo de tráfico que puede existir en iWMNs. El primero se denomina como *flujo*

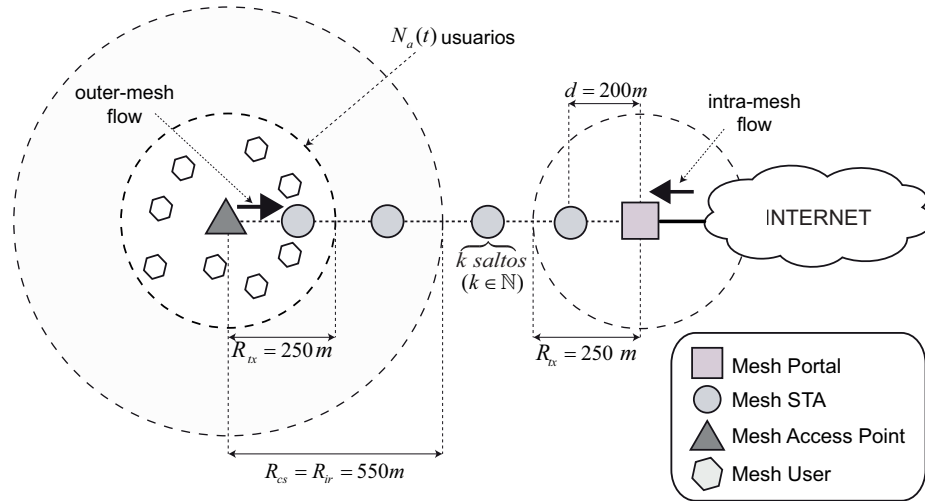


Figura 5.7: Escenario de simulación para una topología en cadena.

intermalla (*intra-mesh flow*), el cual consiste en un flujo TCP que va desde el portal de la malla hacia una STA, el segundo se denomina como *flujo intramalla* (*outer-mesh flow*), el cual consiste en un flujo TCP que va desde un STA hasta el *Mesh Portal*. Las variantes de TCP se utilizan con las configuraciones que se establecen por defecto en el simulador.

Varios parámetros del escenario de simulación son modificados para cada ronda de evaluación. El número de usuarios móviles $N_{A/n}(t)$ se incrementa gradualmente, y, simultáneamente, se definen un total de $N_{A/n}(t)/2$ flujos intermalla y su complemento como flujos intramalla. En la capa de aplicación se utilizan fuentes de tráfico persistentes para inducir un efecto de saturación en la red. En la capa de red se emplea una modificación al protocolo AODV especialmente diseñada para iWMNs [3]; esta variante de AODV contempla la configuración jerárquica de los nodos que componen la red. La movilidad de los nodos se define con el modelo *Random Waypoint* con una velocidad promedio de 1 [m/s]. Cada paquete de datos está constituido por 1460 bytes y cada nodo es capaz de almacenar hasta 50 de estos paquetes en su respectiva fila de espera. El número de saltos inalámbricos en la trayectoria, $h = k + 2$, se incrementa gradualmente en cada escenario de simulación, desde $k = 1$ hasta $k = 10$. El tiempo de simulación es de 2000s, de los cuales solo los últimos 1500 segundos son considerados para cálculos estadísticos con el fin de descartar los estados transitorios de la simulación. Cada escenario se simula 50 veces.

Finalmente, se selecciona un modelo de error de dos estados para la generación de errores de transmisión en el medio inalámbrico, tomando como referencia el modelo propuesto en [131]. De tal manera que se establecen únicamente dos posibles estados del enlace, *bueno* o *malo*. Esta característica de los enlaces se modela empleando una variable aleatoria de *Bernoulli*. Con este modelo podemos inducir, de manera controlada, errores de transmisión en la trayectoria de comunicaciones, lo cual nos permite generar un escenario con mayor complejidad para el desempeño de los protocolos de transporte. Cada transmisión entre un par de nodos tiene así una probabilidad de error por paquete P_e , de forma tal que la probabilidad de éxito de transmisión es $1 - P_e$ y, por lo tanto, la probabilidad de que un paquete de datos llegue exitosamente al MP es $(1 - P_e)^{k+2}$. La variación tanto de k , de $N_u(t)$, así como de P_e , nos permite modificar las condiciones de la red y de este modo analizar de manera más exhaustiva el efecto de degradación de las prestaciones de los protocolos de transporte. El MP es conectado a la red cableada a través de un enlace *full duplex* con un ancho de banda de 100Mbps y con un retardo de 5ms, este enlace cableado brinda un desempeño ideal, es decir, sin pérdidas. En la Tabla 5.1 se sintetizan todos los parámetros de simulación utilizados en los diferentes escenarios.

Tabla 5.1: Parámetros de simulación para la evaluación de variantes de TCP en iWMNs.

Parámetro	Valor
Escenario	Cadena de nodos y <i>butterfly</i>
$N_a(t)$ (Número de STAs)	1 - 20 STAs
P_e (Packet Error Rate)	0.0 - 0.1
k (Número de Mesh STAs en la ruta)	1 - 25 Mesh STA
R_{tx} (Alcance de transmisión)	250 [m]
R_{ir} (Alcance de interferencia)	550 [m]
R_{cs} (Alcance de detección de portadora)	550 [m]
Agente de tráfico	FTP persistente
Tamaño del paquete de datos	1460 bytes
Protocolo de transporte	TCP Westwood+ (TCPW+) TCP Venó (TCPVN) TCP Hybla (TCPH) TCP New Reno (TCPNR) TCP Vegas (TCPV) TCP Freeze (TCPF) (NICTA Module [140])
Protocolo de ruteo	AODV adaptado para iWMNs [3]
MAC	CSMA/CA + 802.11b + RTS/CTS
Patrón de movilidad	Random Waypoint (Velocidad promedio 1 m/s)
Tasa Máxima de transmisión	11 Mb/s, (Auto Rate Fallback disabled)
Tiempo de simulación	2000 s

El caso ideal ($N_a(t) = 1$, $k = 1$ y $P_e = \text{variable}$)

Este escenario es considerado como el caso ideal al tener la mínima contienda por el medio, esto se debe a que solo hay una STA y el número de Mesh STAs entre el MAP y el MP es el mínimo, es decir, $k = 1$. En este escenario se analiza primeramente los efectos adversos que induce P_e en el *throughput*. Esta medida de desempeño representa la cantidad total de información que es entregada de manera efectiva al nodo destino por unidad de tiempo.

Throughput

En la Figura 5.8 se aprecia la degradación del *throughput* como función del PER. La gráfica muestra que la eficiencia de las variantes de TCP se degrada rápidamente a medida que el PER aumenta de 0–10%. En esta gráfica podemos notar que el *throughput* promedio de la red alcanza su valor máximo cuando $P_e = 0.0$. En este punto la mayoría de los protocolos de transporte que son evaluados ofrecen un rendimiento similar, es decir alcanzan una tasa de entrega promedio cercana a 3.9 Mb/s, que representa una utilización del 35.5% del ancho de banda configurado para este escenario, es decir, 11 Mb/s. La variante de TCP que ofrece el peor desempeño en este escenario es TCPF, ya que este protocolo alcanza una capacidad promedio de 3.1 Mb/s, lo que representa una del 21% con respecto al que se obtiene con las otras variantes de TCP.

Para visualizar de mejor manera las diferencias de desempeño que existen entre las variantes de TCP, en la Figura 5.9 se contrasta el *throughput* promedio que se obtuvo por dichos protocolos. En esta imagen se revisa con mayor claridad aquellos escenarios en donde las diferencias de desempeño son más evidentes, es decir, cuando P_e varía entre el 1% y el 5%. De estos resultados podemos notar que TCPW+ es el protocolo que brinda el mejor desempeño. Hay que notar que TCPNR ofrece también uno de los mejores resultados del conjunto de simulaciones, y solo es superado por TCPW+. Esto indica que TCPNR compite de manera eficiente con sus contrapartes, principalmente con aquellas que fueron diseñadas para redes inalámbricas. A medida que P_e

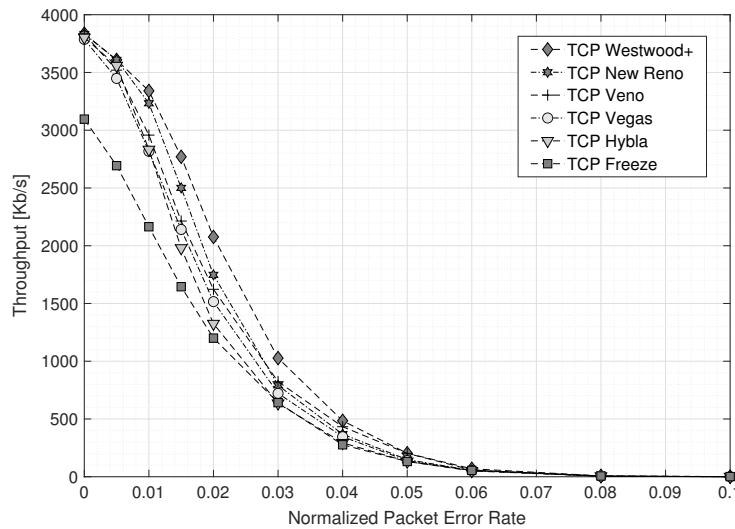


Figura 5.8: Capacidad de la red como función de la probabilidad de error de paquete $0 \leq P_e \leq 0.1$

alcanza valores superiores al 5% el desempeño de la red se reduce hasta un punto en el que la capacidad es menor a 100 Kb/s, esto sin importar la elección del protocolo de transporte que se utilice en el escenario de simulación.

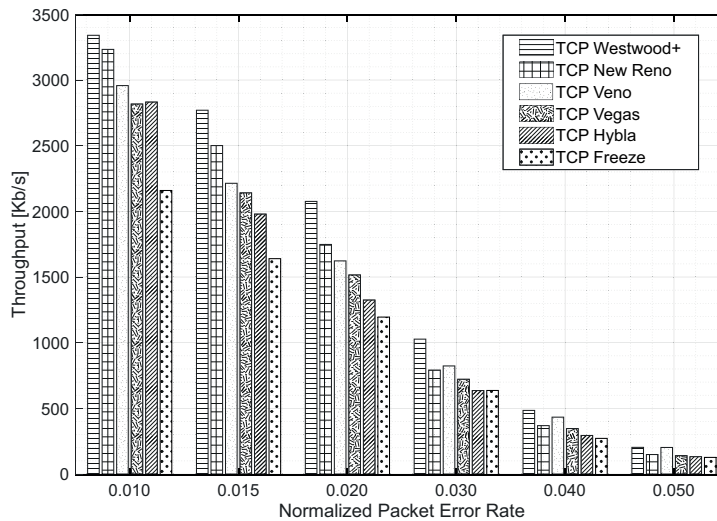


Figura 5.9: Capacidad de la red como función de la probabilidad de error $0.01 \leq P_e \leq 0.05$

A continuación, se realiza un análisis más detallado en dos escenarios con diferentes probabilidades de error ($P_e = 0.00$ y $P_e = 0.015$) debido a las singularidades que presentan ambos casos caso. Por un lado, cuando $P_e = 0.00$ cada protocolo lidió únicamente con el retardo de acceso al medio y el efecto de la terminal oculta en la red; de este modo podemos establecer la cota máxima de desempeño. Por otro lado, cuando $P_e = 0.015$, la diferencia en el desempeño de las diferentes variantes de TCP se ve maximizada; esto nos permite analizar las características de los protocolos que causan estas diferencias.

En la Figura 5.10 se sintetizan en diagramas de cajas (*boxplots*) las muestras de la capacidad de la red por

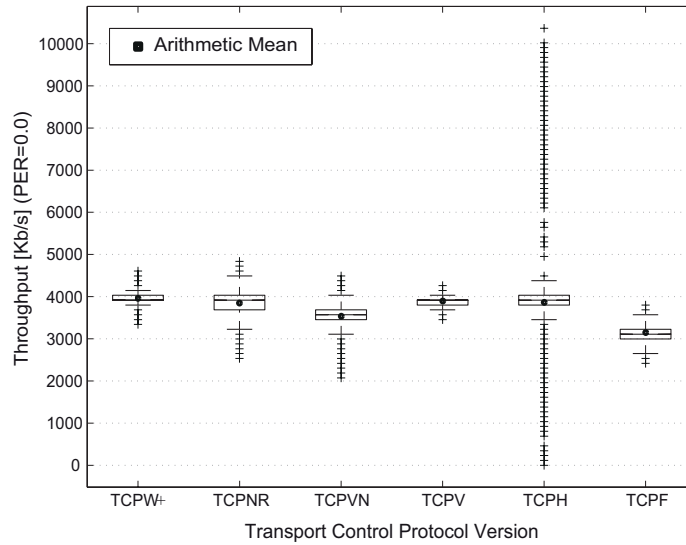


Figura 5.10: Capacidad de la red por segundo durante el tiempo de simulación y con $P_e = 0.0$

segundo durante el tiempo de simulación (estas muestras no incluyen el periodo transitorio de evaluación). Se puede notar que el *throughput* que brindan TCPW+ y TCPV tiene menor variabilidad (diferencia entre el primer y el tercer cuartil, así como entre los límites máximos y mínimos); es notorio que sus máximos y mínimos no superan al valor promedio por más de 100 Kb/s. Este resultado permite aseverar que la tasa de transmisión por segundo que alcanzan estos protocolos es estable y tiene oscilaciones menores. Entonces, TCPNR, TCPVN y TCPF tienen mayor variabilidad y en estos casos el *throughput* puede alcanzar valores máximos y mínimos que llegan a superar el valor promedio hasta en 500 Kb/s, lo cual nos muestra que estas variantes de TCP, en un escenario donde no se inducen pérdidas, tienen una mayor variabilidad y el *throughput* por segundo puede llegar a reducirse significativamente; por ejemplo, TCPNR tiene un valor mínimo cercano a 3.2 Mb/s y el primer cuartil por debajo del valor promedio. Finalmente, TCPH es el protocolo que presenta una mayor cantidad de valores atípicos que llegan van desde 0Kb/s hasta 10Mb/s, con una diferencia entre el valor máximo y mínimo muy similar a TCPVN y TCPF.

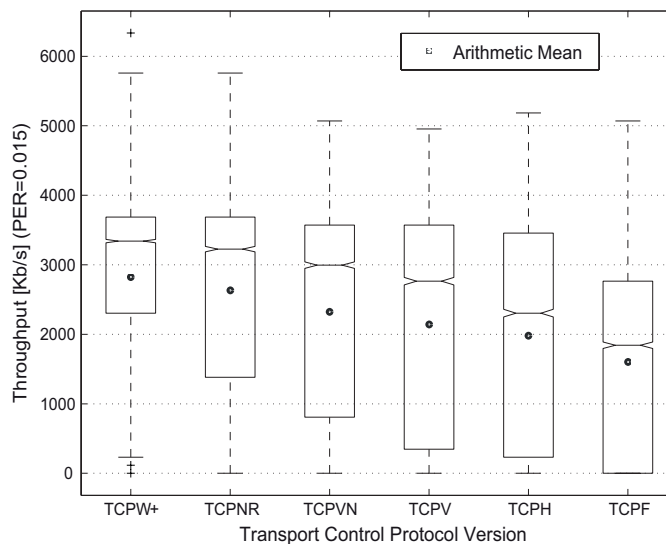


Figura 5.11: Capacidad de la red por segundo durante el tiempo de simulación y con $P_e = 0.015$

En la Figura 5.11 se muestra en diagramas *boxplots* la capacidad promedio de la red utilizando $P_e = 0.015$. En esta gráfica se aprecia que la variabilidad del *throughput* por segundo incrementa, y es marcada la diferencia en el desempeño entre las diferentes versiones de TCP. En este caso la variabilidad de TCPW+ resulta ser menor que la de las otras variantes, y, además, la media aritmética es superior en este protocolo. TCPNR ofrece un mejor desempeño incluso que aquellas variantes de TCP diseñadas para redes inalámbricas al tener un valor promedio similar al de TCPW+ y una variabilidad menor a la de los otros protocolos. De la gráfica se puede observar que los valores máximos de TCPNR y TCPW+ pueden alcanzar hasta un 50% de la capacidad máxima del estándar IEEE 802.11b. Tanto TCPVN, TCPV, TCPH y TCPF tienen el primer cuartil por debajo de 1 Mb/s, y en el caso de TCPF el primer cuartil es incluso 0, esto muestra que estas variantes de TCP tienen un desempeño por debajo del que ofrece TCPW+ y TCPNR. Sin embargo, es importante enfatizar que todas las variantes de TCP alcanzan un *throughput* promedio que, en el mejor de los casos alcanza únicamente el 27% de la capacidad de la red y en el peor un 13%.

Ventana de congestión

En la Figura 5.12 se grafican las muestras de la ventana de congestión (*cwnd*) cuando $P_e = 0.00$. El valor de la ventana de congestión tiene muy baja variabilidad y opera cercanamente en un valor de 20 MSSs para aquellos protocolos que han alcanzado el mejor desempeño en este escenario, es decir, TCPW+, TCPNR, TCPVN. Por otro lado, TCPV opera con valores de la *cwnd* menores, sin embargo, el *throughput* que otorga este protocolo es muy similar al que se obtiene con TCPW+ y TCPNR. El caso de TCPH resulta sumamente atípico, porque la ventana de congestión en ambos casos presenta una mayor variabilidad y un valor muy por encima del promedio. Esto puede explicar el bajo desempeño de TCPF, ya que a pesar de que la ventana de congestión es muy grande, la cantidad de datos que pueden ser enviados a la red están limitados por el ancho de banda del medio. Además, habrá de señalarse que la implementación de TCPF es parte de un módulo externo para *ns2*, este aporte es distribuido diseñado por NICTA (*National Information and Communications Technology Research Center of Australia*) y su funcionamiento ha sido avalado y publicado en [140].

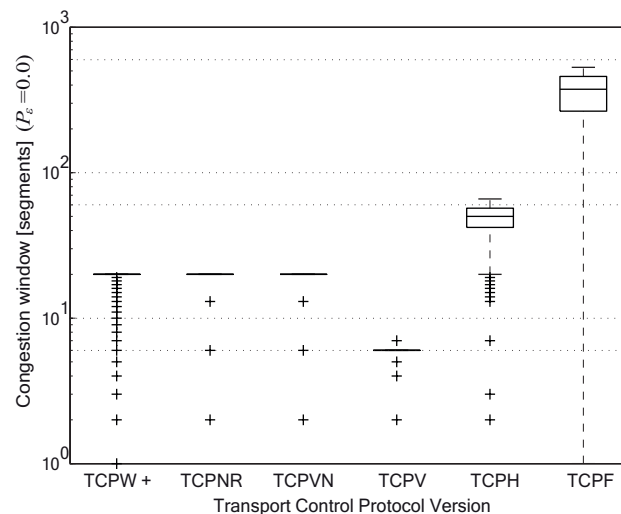


Figura 5.12: Valores de la ventana de congestión (*cwnd*) cuando $P_e = 0.0$.

En la Figura 5.13 se aprecia los valores de la ventana de congestión cuando $P_e = 0.015$. En este escenario, los protocolos que alcanzan el mejor *throughput* son TCPW+ y TCPNR, y se aprecia que la ventana de congestión de ambos protocolos alcanza valores máximos y atípicos superiores al promedio, demostrando que la estimación del BDP es muchos más eficiente en estos protocolos. El caso de TCPF es peculiar, pues teniendo estadísticamente la ventana de congestión valores muy similares a los que tienen TCPW+ y TCPNR, TCPF

alcanza el menor *throughput* en esta serie de simulaciones. Contrastando, TCPVN, TCPV y TCPH operan con una ventana de congestión menor y por esa razón ofrecen un menor *throughput* que TCPW+ y TCPNR.

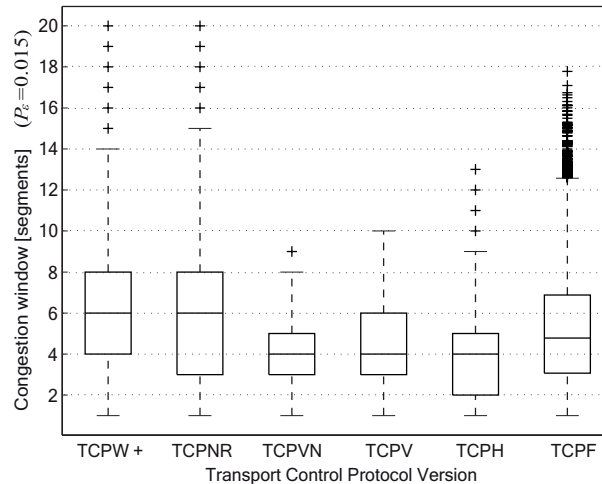


Figura 5.13: Valores de la ventana de congestión (*cwnd*) cuando $P_e = 0.015$.

Round Trip Time (RTT)

Uno de los parámetros más importantes para el funcionamiento eficiente de los protocolos de transporte, es la estimación del RTT. En la Figura 5.14 se aprecia la estimación del RTT para cada una de las variantes de TCP cuando se configura $P_e = 0.00$. En la gráfica se puede apreciar que los protocolos TCPW+, TCPNR y TCPVN estiman el valor promedio del RTT cercano a 100ms. Esta estimación se realiza con una variabilidad menor en los casos de TCPNR y TCPVN, mientras que TCPW+ presenta una mayor variabilidad. Por su parte, TCPV y TCPF estiman el RTT en un valor muy inferior y cercano a 25ms y 10ms, respectivamente. Ambas estimaciones presentan muy baja variabilidad. Sin embargo, TCPF tiene un RTT mínimo y paradójicamente alcanza el menor *throughput* promedio del grupo de simulaciones. Finalmente, TCPH muestra una estimación con alta variabilidad y con una mayor cantidad de valores atípicos de la distribución. El valor promedio de la estimación del RTT es aproximadamente 300 ms y con valores máximos que pueden alcanzar hasta los 500ms.

En el escenario en el que $P_e = 0.015$, la estimación del RTT tiene variaciones similares para la mayoría de los protocolos. TCPW+ y TCPNR son las variantes que ofrecen el mejor *throughput* cuando $P_e = 0.015$ y la estimación del RTT en ambas variantes es muy similar. La variabilidad de la estimación del RTT en este caso es superior al 50% del valor promedio. Por otro lado, TCPF estima un RTT muy cercano a 0ms y con muy baja variabilidad, lo cual sigue manifestando un comportamiento singular de este protocolo. No obstante, la estimación mínima del RTT, TCPF es el protocolo que alcanza el menor *throughput* de esta serie de simulaciones. Por otro, lado las variantes restantes estiman el RTT promedio cercano a 25ms pero con una variabilidad superior al 50% y con un número considerable de muestras atípicas. Se puede apreciar que la estimación del RTT tiene un efecto significativo en el *throughput* que alcanza cada uno de los protocolos analizados.

Escenario con variaciones en el número de saltos ($N_a(t) = 1$, $k = \text{Variable}$ y $P_e = 0.0$ y $P_e = 0.015$)

En este escenario de evaluación se analizan los efectos en el desempeño de la red que ocasiona el incremento del número de *Mesh STAs* ubicados entre la *STA* y el *Mesh Portal*. La cadena de nodos que se forma tiene $k + 2$ saltos entre el usuario de la red y el acceso cableado a Internet, esto quiere decir que cada flujo TCP tiene que

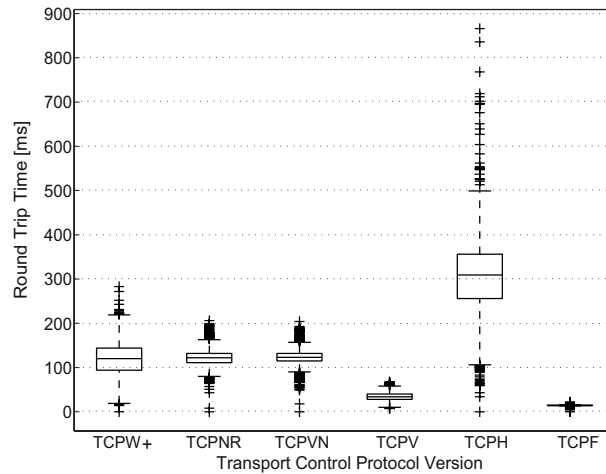


Figura 5.14: Valores de la estimación RTT cuando $P_{\epsilon} = 0.0$.

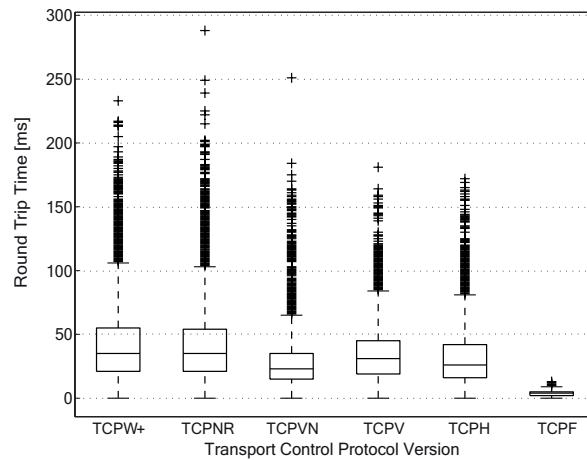


Figura 5.15: Valores de la estimación RTT cuando $P_{\epsilon} = 0.015$.

atravesar por un mayor número de enlaces inalámbricos. Primeramente, se miden los efectos en la capacidad de la red en un esquema en el que la probabilidad de error es $P_{\epsilon} = 0.00$. En la Figura 5.16 se muestra el *throughput* promedio que ofrecen las variantes de TCP. En la gráfica se aprecia la tendencia decreciente de este parámetro de desempeño a medida que incrementa k . 20% en todos los casos analizados.

Si el número de saltos en la trayectoria incrementa, el *throughput* promedio disminuye de manera significativa; en el caso ideal, cuando $k = 1$, la mayoría de los protocolos, a excepción de TCPF, alcanzan un *throughput* promedio cercano a 4Mb/s. Sin embargo, cuando k incrementa desde 1 hasta 8, la capacidad del sistema se degrada rápidamente, reduciendo el *throughput* promedio del caso ideal en intervalos que van desde el 26.4% al 42.2%. Es de advertir que cuando el número de saltos se incrementa hasta alcanzar $k = 20$, la degradación del *throughput* promedio es solo de un 60%. En este escenario, tal como sucedió en el caso anterior, TCPW+ y TCPNR son los protocolos que presentan el mejor desempeño, mientras que por otra parte TCPF ofrece el menor de todos. La diferencia de desempeño entre TCPW+ y TCPF es cercana al 30%.

Para el segundo escenario de evaluación se ha seleccionado una probabilidad de error $P_{\epsilon} = 0.015$; dado que esta es la probabilidad de error en la que son más evidentes tanto la degradación de las prestaciones de la red como la diferencia de desempeño entre las variantes de TCP. Se puede apreciar en la Figura 5.17 que TCPW+ ofrece el mejor desempeño en este escenario; además, es notorio que la capacidad de la red presenta una importante tasa de degradación. Obsérvese que cuando $k > 6$, es decir, cuando hay una distancia

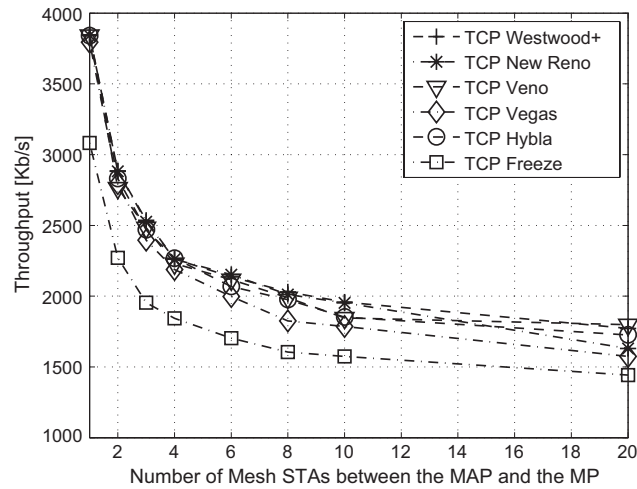


Figura 5.16: *Throughput* promedio como función de k cuando $P_e = 0.0$.

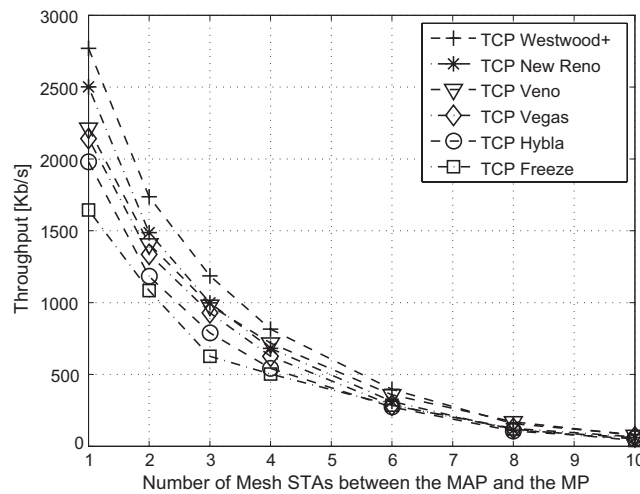


Figura 5.17: *Throughput* promedio como función de k cuando $P_e = 0.015$.

entre el usuario móvil y el MP que oscila entre 1400 y 1800 metros, el *throughput* promedio es menor al 5% de la capacidad máxima de la red (11Mb/s para este este escenario). Cuando k se configura con valores entre 1 y 5, TCPW+ continúa ofreciendo el mejor desempeño, pues el *throughput* promedio que alcanza este protocolo es hasta un 25% superior al que ofrecen las otras variantes. En general, el desempeño de cualquiera de los protocolos de transporte resulta ser menor al 20% de la capacidad máxima de la red cuando $k \geq 2$, lo cual representa una muy baja tasa de desempeño. Véase también que la tasa de pérdidas, cuando $k = 1$, es aproximadamente del 5% de la totalidad de paquetes enviados; lo cual representa una enorme cantidad de pérdidas ocasionadas por los errores acumulados en la trayectoria. A medida que k incrementa hasta alcanzar un valor máximo (10), la tasa de pérdidas alcanza un valor cercano al 14% y, consecuentemente, se obtiene la grave degradación de desempeño.

Escenario con variaciones en el número de usuarios ($N_a(t) = \text{Variable}$, $k = 3$ y $P_e = 0.015$)

En este escenario de evaluación se **analizan los efectos que tiene el incremento de flujos TCP en competencia en el desempeño de la red**. Para este fin se emplea un escenario en donde se configuran los siguientes parámetros:

- El número de *Mesh STA*, k , es tres; de manera tal que el tráfico de datos tiene que pasar por cinco enlaces inalámbricos entre la fuente y el destino (refiérase a la Figura 5.7 para más detalles)
- La probabilidad de error en los enlaces inalámbricos, P_e , se configura con el valor 0.015, escenario en donde se presentan diferencias significativas de desempeño entre las diferentes variantes de TCP (véase la Figura 5.17)
- El número de usuarios móviles es variable y se incrementa en cada evaluación desde 2 hasta 20, de esta forma se explora el efecto que tiene el incremento de tráfico en la red en el desempeño de TCP

En este escenario se evalúan **dos parámetros de desempeño significativos**: el *aggregated throughput*, es decir, la cantidad total de datos que se entregan al destino por unidad de tiempo, y, el índice de equidad de Jain (*Jain's Fair Index - JFI*) [141]. El JFI es una medida de desempeño que determina de forma cuantitativa la equidad con la que se reparte un recurso entre varias conexiones TCP. Con este parámetro de desempeño se validará la manera en la que las diferentes variantes de TCP distribuyen la capacidad de la red entre las STAs.

Lema 3

El *Jain's Fair Index* (JFI) se define algebraicamente como:

$$JFI(x_1, x_2, \dots, x_i) = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2}. \quad (5.24)$$

Donde n representa el número de conexiones TCP y el *throughput* de la i -ésimo conexión se define como x_i . El JFI tiene propiedades que le brindan una gran utilidad a esta medida de desempeño, pues, por un lado, el JFI es independiente de la cantidad de conexiones, y, por otro lado, no depende de la escala ni del tipo de recurso que se quiere distribuir. El JFI está acotado entre 0 y 1, siendo 1 el caso ideal que indica que el recurso es asignado con equidad entre las conexiones TCP, y, $JFI = 1/n$ cuando se da el peor de los casos, es decir, la captura de los recursos por parte de una conexión TCP.

Aggregated throughput

El *aggregated throughput* se puede definir como la cantidad total de datos que se entregan a todos las entidades TCP *client-side* de la red por unidad de tiempo. Con esta medida de desempeño se puede analizar la eficiencia que tiene la red y las diferentes variantes de TCP para entregar correctamente los datos a los respectivos destinos. En la Figura 5.18 se sintetizan las muestras del *aggregated throughput* obtenidas por cada variante de TCP durante las 50 simulaciones. En la gráfica se puede apreciar que para todas las variantes de TCP el *aggregated throughput* incrementa en medida que lo hace el número de flujos TCP en competencia. Esto se puede explicar por dos razones; la primera, y la más notable, el incremento de la cantidad de tráfico en la red y, la segunda, la distribución de los errores de transmisión P_e entre los diversos flujos TCP, lo cual se traduce en una menor cantidad de paquetes perdidos para cada flujo TCP y en correspondencia en una *cwnd* que reduce su tamaño menos frecuentemente. De los resultados presentados en la gráfica también se puede apreciar como el incremento del *aggregated throughput* es marginal cuando se pasa de 10 a 20 flujos TCP en competencia, esto nos permite inferir que, aun incrementando el número de usuarios, la red alcanzará un límite máximo que depende no solo de las variantes de TCP sino también de la probabilidad de error, de la contienda por el medio entre los usuarios de la red MAP y de los mecanismos MAC de la capa de enlace. Para las variantes de TCP (excluyendo TCPW+) se muestra una clara tendencia creciente y con mínima dispersión del *aggregated throughput*, en algunos casos este parámetro puede llegar a incrementar por encima de los 1 Mb/s con respecto al escenario con un solo usuario (refiérase la Figura 5.17). TCPW+, por otro lado, presenta una gran dispersión para este parámetro y, en algunos casos, no muestra un incremento significativo. Es notable que TCPW+ ofrece un desempeño menos predecible que el alcanzado por las otras variantes analizadas y es superado por las otras variantes de TCP. Finalmente, se puede comprobar en la gráfica que TCPF alcanza el menor *aggregated throughput*, no obstante, lográndolo con una menor dispersión que la que presenta TCPW+.

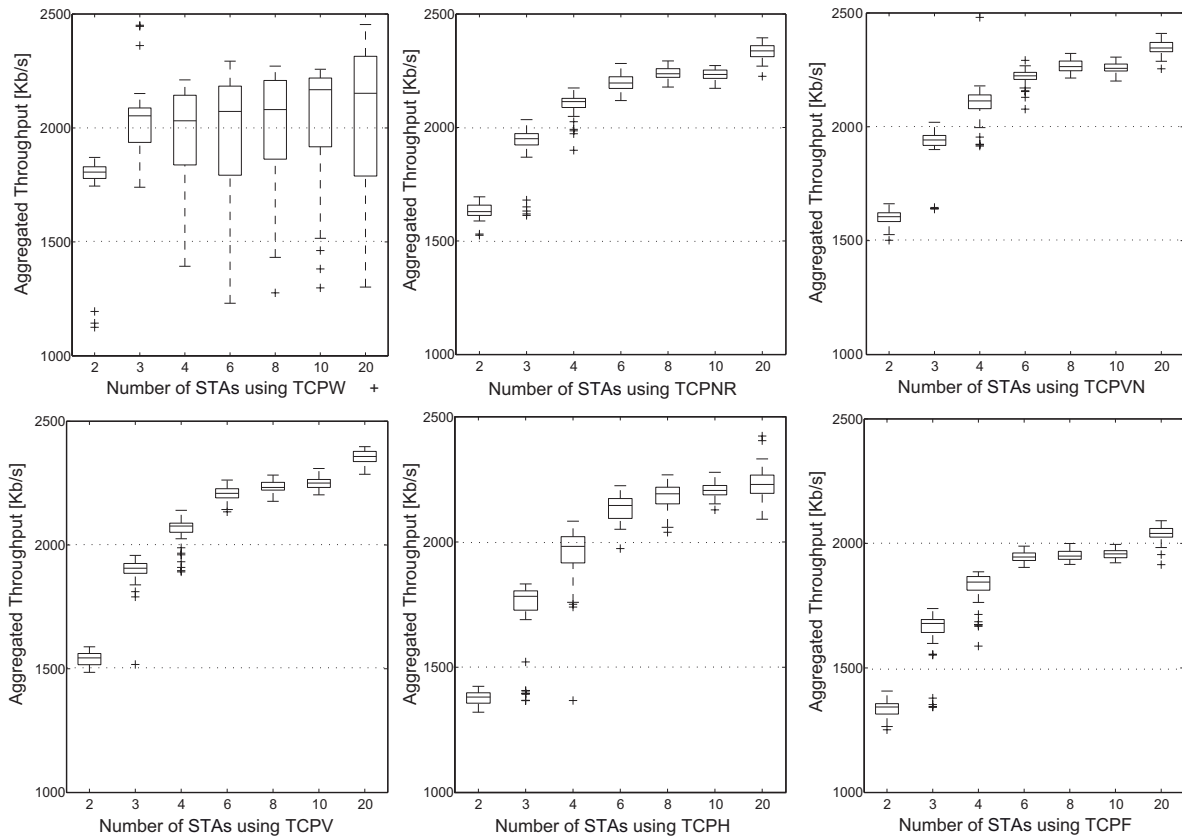


Figura 5.18: *Aggregated throughput* para las diferentes variantes de TCP cuando $N_a(t) = 2, 3, 4, 6, 8, 10, 20$

Jain's Fair Index (JFI)

Uno de los parámetros más importantes para la evaluación de desempeño de los protocolos de la capa de transporte es la **medición de la equidad**. Pues, cuando existen diversos flujos de datos en competencia, puede existir una mala coordinación entre los protocolos de la capa acceso al medio y los protocolos de la capa transporte, ocasionando el efecto de captura del medio. Este fenómeno se presenta cuando un flujo de datos gana constantemente el acceso al medio y reduce la probabilidad de que otros flujos puedan acceder a él, disminuyendo así el *throughput* que pueden alcanzar dichos flujos. Para **evaluar la equidad con la que se distribuye la capacidad de la red se utiliza el índice de equidad JFI**. Este parámetro se calculó empleando el *throughput promedio* obtenido por cada flujo TCP durante todo el tiempo de simulación. En la Figura 5.19 se resumen las muestras del índice de equidad JFI. En estos resultados resalta el desempeño del protocolo TCPW+, pues conduce a una inequitativa distribución de la capacidad de la red, llegando a alcanzar valores cercanos al caso de captura del medio (esto es cuando $JFI = 1/n$). Además, se puede apreciar que el mal desempeño de TCPW+ se da incluso cuando el número de STAs es muy bajo; en estos casos (dos y tres usuarios) se alcanzan valores del JFI muy cercanos a el caso de captura de los recursos. En el caso de TCPNR, TCPVN, TCPV y TCPF, la distribución del JFI se mantiene por encima de 0.8 aun cuando el número de usuarios incrementa. Demostrando la eficiencia de estas variantes de TCP para distribuir la capacidad de la red entre las STAs. Destaca TCPNR en esta evaluación, porque a pesar de no ser un protocolo diseñado para operar en redes inalámbricas, sigue ofreciendo resultados que superan a los ofrecidos por aquellas variantes diseñadas precisamente para operar en redes inalámbricas. A la vez, TCPH presenta un comportamiento decreciente del JFI a medida que incrementa el número de usuarios en la red, confirmando la ineficacia de este protocolo para distribuir equitativamente la capacidad.

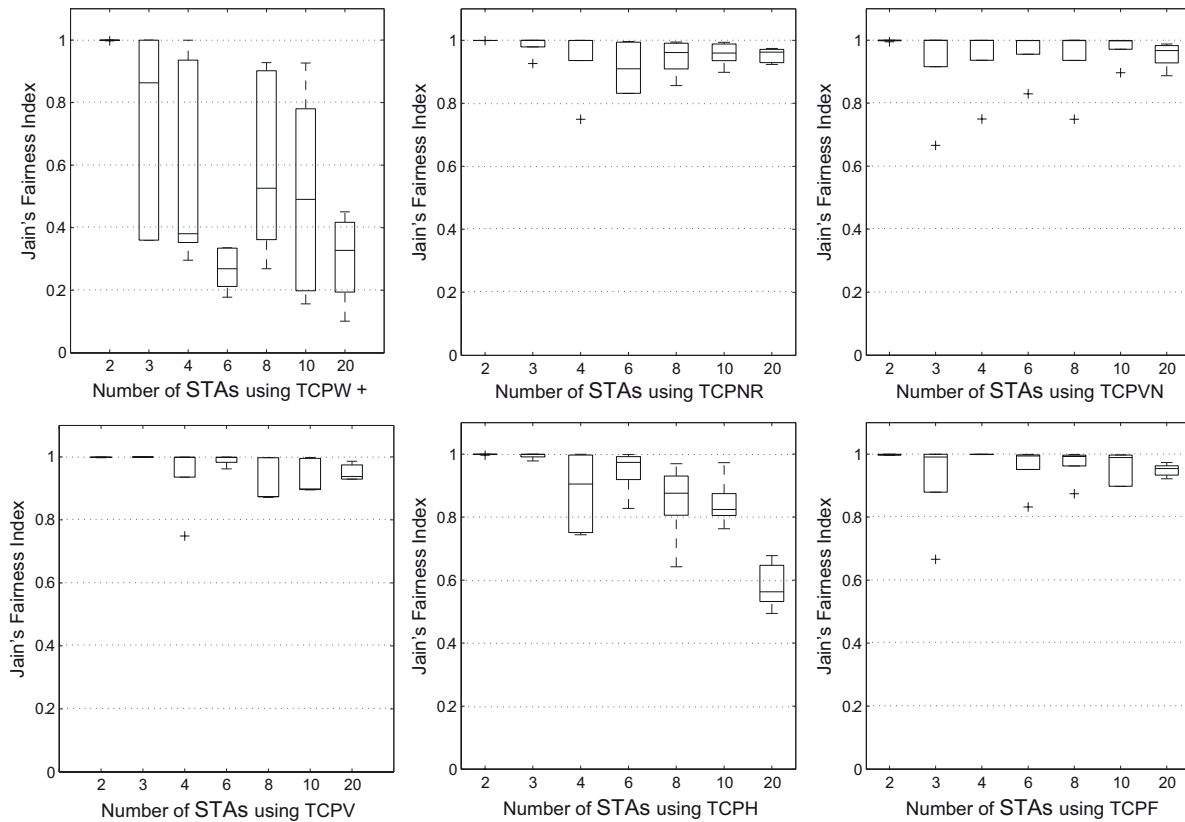


Figura 5.19: Índice JFI para las diferentes variantes de TCP en un escenario con $N_a(t) = 2, 3, 4, 6, 8, 10, 20$.

5.4.2 Análisis de desempeño de TCP en iWMN en topología *butterfly*

En esta evaluación se analiza un problema distintivo de la arquitectura de las iWMNs, la equidad. En un escenario típico de una iWMN existen *STAs* cercanas al MP y otras más alejadas. Esto quiere decir que el tráfico de los usuarios que se encuentran más alejados debe ser encaminado por un mayor número de *Mesh STAs*, que el tráfico que proviene de *STAs* más cercanas al MP. Esta diferencia provoca que los usuarios más cercanos al MP obtengan mayor capacidad y, muy probablemente, capturen los recursos de la red, afectando el desempeño del resto de los usuarios. En esta subsección se verifica como la arquitectura y los protocolos de la capa de transporte juegan un rol fundamental en la equidad con la que se distribuyen los recursos de la red.

Descripción del escenario

Para evaluar el desempeño de los protocolos de transporte se utiliza una topología como la que se muestra en la Figura 5.20. En este escenario se utilizan dos MAP para ofrecer servicios de conectividad a dos usuarios inalámbricos cada uno, es decir, el MAP1 ofrece conectividad a las *STAs* 3 y 4, mientras que el MAP2 ofrece conectividad a las *STAs* 1 y 2. La cantidad de *Mesh STAs* entre el MAP2 y el MP es uno, y se mantiene constante durante todas las simulaciones. Por otro lado, la cantidad de *Mesh STAs* entre el MAP1 y el MP incrementa desde 1 hasta 4. La probabilidad de error en los enlaces inalámbricos P_e se configura en 0.0. Las *STAs* 1 y 3 establecen una conexión TCP hacia Internet mientras que las *STAs* 2 y 4 establecen una conexión TCP desde Internet. Se evalúa en cada simulación el *aggregated* e *individual throughput*; así como el JFI, para poder determinar la equidad con la que la red y los protocolos de transporte distribuyen los recursos de la red.

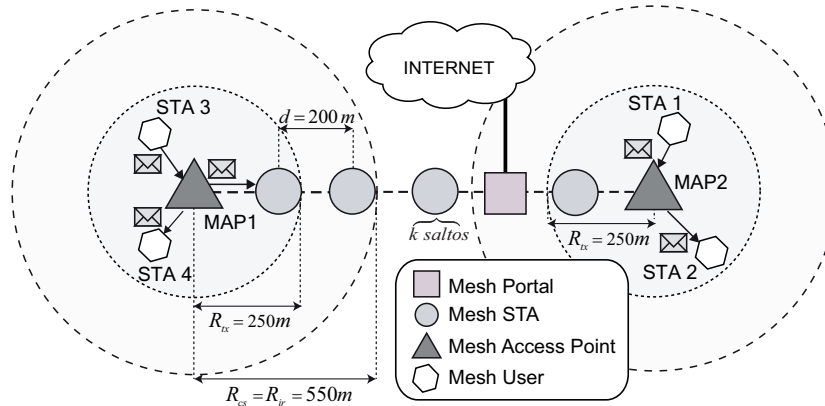


Figura 5.20: Escenario con dos MAPs ofreciendo conectividad a 4 STAs para establecer comunicaciones desde/hacia Internet.

Throughput individual y agregado

En la Figura 5.21 se muestra el *aggregated throughput* que se obtiene con cada variante de TCP y en cada uno de los diferentes escenarios de evaluación. Se puede apreciar que los protocolos TCPW+, TCPNR, TCPVN y TCPV obtienen un rendimiento muy similar, alcanzando un *aggregated throughput* cercano a los 3Mb/s en todos los escenarios evaluados. Por otro lado, TCPH y TCPF obtienen un menor rendimiento, alcanzando un *aggregated throughput* cercano a los 2.5Mb/s. Un importante resultado que se aprecia en la figura es como la arquitectura de la iWMN afecta significativamente la equidad con la que se distribuyen los recursos de la red. En la imagen se muestra claramente como a medida que incrementa el número de *Mesh STA* entre el MAP1 y el MP, el *throughput* de los usuarios conectados a dicho MAP se reduce significativamente. En la gráfica se nota que al incrementar el número de *Mesh STA* a dos, el *throughput* de las STAs 3 y 4 disminuye hasta un 50%, mientras que el *throughput* de las STAs 1 y 2 incrementa hasta un 50%. Los más agudos casos de inequidad se dan cuando el número de *Mesh STAs* entre el MAP1 y el MP incrementa a tres y cuatro; en ambos casos el *throughput* de las STAs 3 y 4 se reduce a 0Kb/s mientras que las STAs 1 y 2 obtienen todos los recursos de la red, obteniendo cada una de estas STAs un *throughput* individual de 1.5Mb/s.

En la Figura 5.22 se muestra el JFI para los cuatro flujos en competencia. Es evidente que las variantes TCPF, TCPNR y TCPW+ distribuyen de manera más equitativa los recursos de la red. Sin embargo, contrasta que las diferencias de desempeño, en términos de equidad, son mínimas entre las variantes evaluadas. También se puede notar el mal desempeño que se obtiene con TCPH. Resulta muy importante apreciar que todas las variantes son afectadas tanto por la estructura de la red como por los protocolos de acceso al medio. Por un lado, la topología de la red y la diferencia entre el número de saltos entre los MAPs y el MP influyen de manera negativa en el desempeño, porque el tráfico proveniente del MAP1 debe atravesar más enlaces inalámbricos, incrementando las etapas de contienda por el medio y, por ende, la estimación del RTT para TCP. Por otro lado, la ventana de congestión de las conexiones TCP de las STAs 3 y 4 crece con una tasa menor que la de las conexiones TCP de las STAs 1 y 2; esto propicia que las conexiones TCP de estas últimas STAs capturen el medio en competencia entre el MP y las Mesh STAs contiguas y obtengan así un mayor *throughput* (véase la Figura 5.21).

5.5 Conclusiones sobre las evaluaciones de desempeño

En este capítulo se evaluaron algunas de las variantes más populares de TCP para redes inalámbricas en escenarios representativos de las iWMNs. A través de simulaciones se evaluaron los efectos de los errores de transmisión en el medio, en el desempeño de los protocolos de transporte. Además, se ha evaluado los efectos causados por el incremento del número de saltos en las iWMNs, así como el incremento de la competencia entre usuarios de la red.

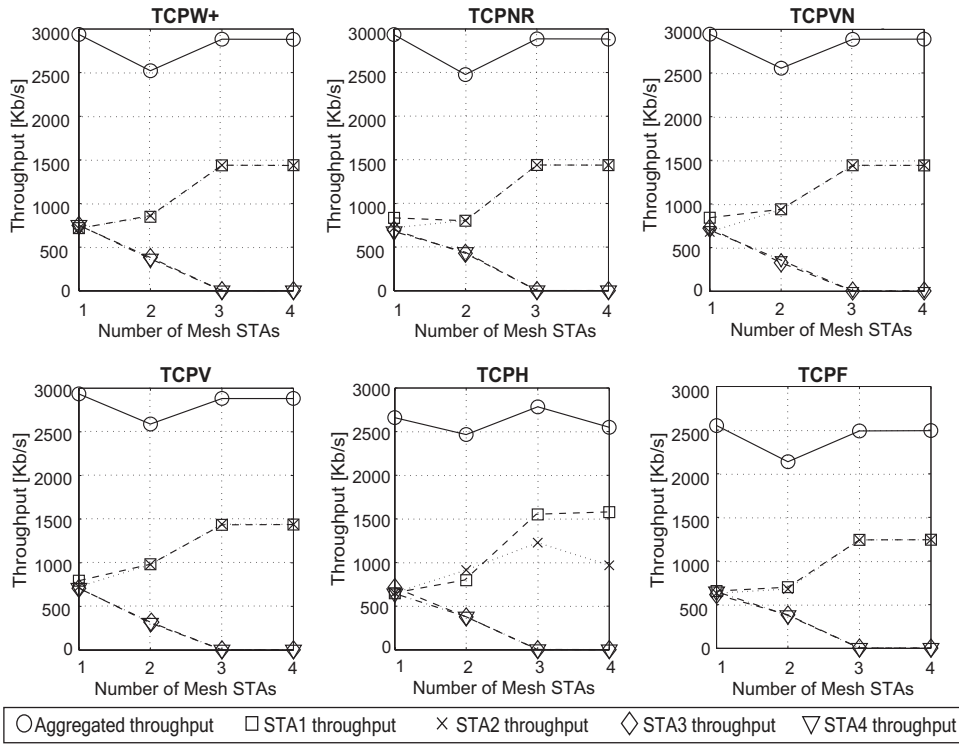


Figura 5.21: Aggregated e individual throughput como función del número de Mesh STAs entre el MAP1 y el MP.

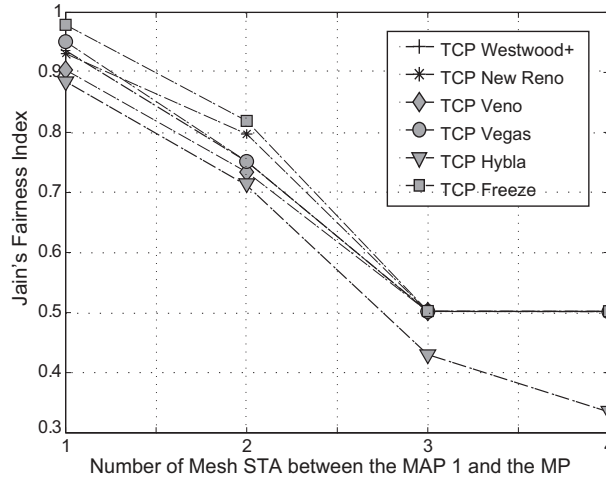


Figura 5.22: Índice JFI como función del número de Mesh Points entre el MAP 1 y el MP.

Los resultados obtenidos muestran que:

1. La probabilidad de error en el medio inalámbrico causa una degradación importante en el desempeño de la red, pues con márgenes de error superiores al 0.05 % por enlace de comunicación, la red reduce sus prestaciones, y las variantes de TCP ofrecen un *throughput* promedio por debajo del 5% de la máxima capacidad ofrecida por el estándar 802.11b (11 Mb/s). En términos de *throughput*, TCPW+ ofrece un mejor desempeño, superando a las otras variantes de TCP en márgenes que van del 20% al 48%.
2. El número de saltos en una topología simplificada de una iWMN reduce la capacidad de la red, hasta

en un 50%, cuando el número de saltos es mayor a ocho. Este efecto decreciente se debe a la contienda por el medio y a los efectos de la terminal oculta y expuesta que existen en la trayectoria. Bajo estas condiciones, TCPW+ continúa superando a las otras variantes de TCP. Sin embargo, a pesar de la superioridad promedio de TCPW+, este protocolo falla en renglones importantes. Por un lado, TCPW+ presenta una **alta variabilidad en la capacidad de la red y un bajo índice de equidad para distribuir el recurso entre un número creciente de usuarios móviles**. El caso de TCPNR es de destacar, pues esta versión de TCP ha ofrecido un comportamiento promedio muy similar al alcanzado por TCPW+, sumándose que muestra una menor variabilidad para la capacidad promedio del sistema y una distribución del recurso más equitativa, lo cual lo posiciona por encima de opciones como TCPF y TCPH, las cuales son versiones diseñadas específicamente para redes inalámbricas.

3. Al incrementar el número de usuarios que disputan el acceso a la red encontramos que, en general, las variantes de TCP combinadas con el algoritmo de acceso CSMA/CA ofrecen un buen nivel de equidad. Resulta importante hacer notar que a medida que el número de flujos TCP se incrementa se obtiene un beneficio en la capacidad de la red, porque cuando los errores de transmisión del medio inalámbrico se distribuyen entre flujos en competencia se propicia un mejor aprovechamiento de la capacidad de la red.
4. La distancia, entendida como el número de Mesh STAs entre los MAPs y el MP, afecta seriamente el desempeño de la red. Los resultados muestran que, cuando existen varios MAPs ofreciendo conectividad en la iWMN, los usuarios asociados al MAP más cercano al MP pueden llegar a capturar los recursos de la red. Esto reduce, y en el peor de los casos anula, el *throughput* de aquellos usuarios que estén asociados al MAP más alejado. La equidad con la que se distribuyen los recursos de la red, no solo depende del protocolo de transporte utilizado, sino que también depende estrechamente de la topología de la red y de efectos tales como la contienda por el medio y los efectos de la terminal oculta y expuesta.

Los resultados de este capítulo fueron obtenidos en escenarios simples, en donde no se consideran otros elementos que pudieran afectar más intensamente el desempeño de la red y el de las variantes de TCP; por ejemplo, no se han considerado las interferencias entre trayectorias, los efectos derivados de los protocolos de encaminamiento o de la movilidad de los nodos. Esto significa que, si complementamos estos resultados con los obtenidos en [3], el desempeño de TCP será mucho menor en condiciones de mayor competencia y congestión en las iWMNs.

Considerando que una iWMN tiene una infraestructura dedicada, estática, y sin restricciones de procesamiento o de energía en los nodos que la componen, se puede afirmar que estas redes ofrecen ventajas para emplear los métodos de codificación de red. En estas circunstancias podemos afirmar que usar técnicas de codificación de red minimizaría la competencia por el medio, ocultaría las pérdidas de paquetes que afectan a TCP, y mejoraría su proceso de auto regulación. Sin embargo, y como bien se menciona en la literatura, la codificación de red no siempre trae beneficios significativos; por tal motivo se requiere un uso inteligente para aprovechar al máximo las prestaciones de infraestructura de la red y las características del tráfico en la red. Partiendo de los resultados de esta primera evaluación, podemos identificar algunas características que pueden ser optimizadas haciendo uso de la codificación oportunista de red.

1. Los puntos de acceso de la malla (MAP) centralizan el flujo de los usuarios en un comportamiento similar al de las redes WLAN. La codificación de red en escenarios con flujos *unicast* en redes WLAN ha sido abordado por [89] y se ha podido constatar que la codificación ofrece ganancias importantes para la capacidad de la red. El uso de la codificación en iWMNs puede ofrecer minimizar la contienda por el medio y el tráfico en la región de cobertura de un MAP.
2. La codificación de intra-flujos TCP combinando los paquetes ACK, tal y como se ha propuesto en [97, 99, 98], puede maximizar el transporte de flujos dentro del *backbone* de la malla. Con esto, disminuirían los efectos de la terminal oculta y expuesta en la trayectoria de datos; y se maximizaría la transmisión de información.
3. La centralización de flujos en los MPP y MAP pueden inducir un esquema de codificación soportado por el conocimiento de las propiedades del tráfico en los nodos vecinos. Tal y como se propone en [7],

cada nodo puede identificar el estado de sus nodos vecinos, utilizando un esquema de notificación, para maximizar la información que puede ser recibida con una transmisión.

4. Finalmente, la equidad con la que se distribuyen los recursos de la red puede ser beneficiada de un esquema de codificación, tal y como se sugiere en [105].

Propuesta para la codificación adaptativa en iWMNs y evaluaciones de desempeño

“Solo por que algo no haga lo que planeaste no significa que sea inservible.”

Thomas A. Edison

DERIVADO de los resultados obtenidos y documentados en el capítulo anterior, se ha observado que el desempeño de las iWMNs se degrada a medida que incrementan: el tamaño de la red, los errores en los enlaces inalámbricos, la densidad de tráfico y la contienda por el medio. Se constató, tras un análisis minucioso de los efectos que inducen diferentes variantes de TCP en el desempeño de las iWMNs, que estas no operan eficientemente en iWMNs, ya que el medio inalámbrico y la topología jerárquica de la red afectan el *throughput* y equidad con la que operan estos protocolos. Con el objetivo de mejorar el desempeño de la red y de los protocolos que en ella operan, se propone adaptar, implementar y evaluar la técnica de codificación de red inalámbrica (WNC- Wireless Network Coding) en iWMNs. La propuesta está sustentada con base en los siguientes argumentos:

- Los patrones de tráfico y la arquitectura jerárquica de las iWMNs tienen características idóneas para implementar la técnica de codificación de red, ya que estas propiedades pueden incrementar significativamente las oportunidades de codificación.
- WNC puede reducir el número de transmisiones en las trayectorias *multihop* que transportan flujos de datos TCP y sus respectivos ACKs; de este modo, se puede reducir el número de transmisiones inalámbricas en las trayectorias e incrementar la capacidad de las iWMNs [99, 97, 107].
- Los Mesh APs (MAPs) concentran una gran cantidad de flujos TCP generados por los usuarios móviles. Se ha demostrado que al aplicar WNC en redes inalámbricas WLAN con flujos convergentes se puede disminuir la contienda por el medio [89], este beneficio puede obtenerse en los MAPs de una iWMN.
- La equidad con la que se distribuyen los recursos de la red entre varios flujos TCP puede mejorarse utilizando WNC, tal como se documenta en [142]. Esto puede beneficiar la equidad con la que operan las iWMNs.
- La topología que se forma alrededor de los *Mesh Access Points* (MAPs) facilita la implementación y operación de WNC, ya que se pueden satisfacer los requerimientos necesarios para codificar a la vez de incrementar las oportunidades de codificación.

En este capítulo evaluamos los beneficios que se obtienen al utilizar WNC en los MAPs; nodos donde se concentra la mayor parte del tráfico de la red y donde se generan las mayores oportunidades de codificación. Se subraya que el presente análisis puede aplicarse a subsecciones de nodos en las trayectorias *multihop* que se forman en iWMNs. El objetivo principal de este análisis es identificar los efectos que tienen el mecanismo de

acceso aleatorio al medio y la densidad de flujos TCP en la generación de oportunidades de codificación en los MAPs. Además, se propone configurar un esquema de codificación adaptativa que incorpore tiempos de espera para los paquetes nativos con el objetivo de formar una oportunidad de codificación –esquema inspirado en las propuestas [108, 107, 10]. A diferencia de las soluciones existentes en la literatura, en esta propuesta se plantea que el tiempo de espera de los paquetes nativos se ajuste de manera dinámica. Para esto, se propone emplear promedios móviles ponderados –un método utilizado por TCP para estimar el RTT. El cálculo de esta estimación es de baja complejidad, no consume recursos de los dispositivos, y es una solución rápida para ajustar dinámicamente el tiempo de espera de los paquetes nativos.

6.1 Oportunidades de codificación en iWMNs

Los flujos de datos en una iWMNs son transportados a través de **cadenas de nodos inalámbricos**; comúnmente cada una de estas cadenas comienza en el usuario móvil (STA) y termina en un Mesh Portal (MP). La formación de estas trayectorias *multihop* es determinada por los protocolos de encaminamiento que operan en la red. Dada la existencia de varios flujos de datos en una iWMN, se requerirá de varias cadenas de nodos para transportarlos. La Figura 6.1 muestra cadenas de nodos en una iWMN; en este ejemplo las cadenas pueden intersectarse y crear estructuras en las cuales se pueden aprovechar diferentes variaciones de la técnica WNC.

En la imagen se muestran dos estructuras que pueden beneficiarse de la implementación de WNC. En la primera, los MAPs brindan conectividad a las STAs, formando una estructura con propiedades muy similares a las descritas en el escenario de la Sección 3.5, es decir, cada MAP reexpide paquetes de datos que viajan en direcciones opuestas (paquetes que van desde el MAP hacia la Mesh STA más cercana y aquellos que van desde el MAP hacia las STAs). Esta característica puede ser aprovechada por WNC para reducir la contienda por el medio e incrementar la capacidad de la red. En la segunda estructura, por otro lado, existen Mesh STA donde convergen varios flujos, es decir, donde se intersecan varias cadenas de nodos, estas estructuras son similares a la descrita en la Sección 3.4. En este tipo de estructuras se puede implementar y obtener los beneficios de la técnica denominada como Codificación Oportunista (*Opportunistic Network Coding*), ya que esta estructura satisface los requisitos necesarios para la operación de esta técnica.

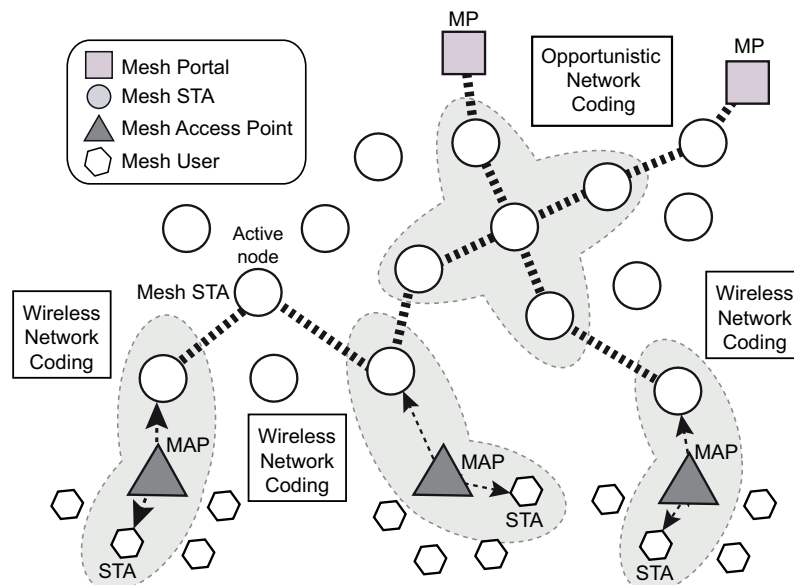


Figura 6.1: Estructuras internas de una iWMNs y la aplicación de la técnica de codificación de red.

6.2 Codificación en los puntos de acceso (MAPs)

Un MAP debe reexpedir múltiples paquetes de datos que comúnmente fluyen en direcciones opuestas; es decir, un flujo va desde los usuarios móviles al MP más cercano (STAs→MP) y viceversa (MP→STAs), véase la Figura 6.1. Para implementar la técnica de codificación de red en un MAP, se debe rediseñar la arquitectura del dispositivo. Primero, son necesarias dos **filas de codificación**, S_1 y S_2 , en donde almacenen los paquetes nativos antes de ser codificados. Dado que un MAP actúa como un dispositivo de encaminamiento —es decir, se encarga de determinar el salto siguiente (*nexthop*) para cada paquete de datos— entonces cada paquete nativo que se recibe puede ser fácilmente clasificado y almacenado en su respectiva fila de codificación. Al recibir paquetes nativos, el **clasificador** es responsable de almacenarlos en su fila respectiva (S_1 para los paquetes nativos con dirección Mesh STA→STAs, y S_2 para los paquetes nativos con dirección STAs→Mesh STA); si las filas de codificación están llenas y se recibe un paquete nativo, se libera espacio enviando a las filas de transmisión de EDCA el paquete nativo que lleva mayor tiempo almacenado en la fila de codificación; así se pueden almacenar los paquetes nativos que se reciben cuando las filas de codificación están llenas. La Figura 6.2 muestra el esquema propuesto para la implementación de WNC en un MAP.

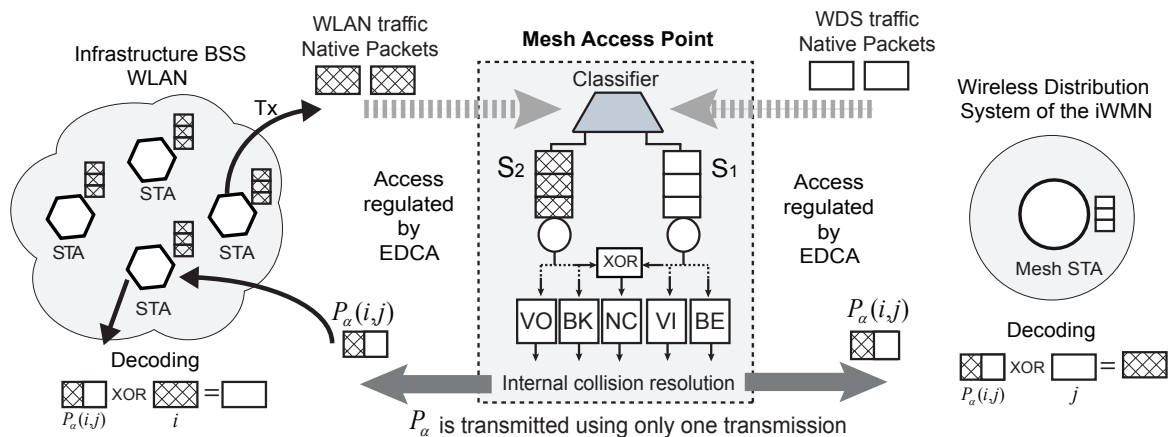


Figura 6.2: Arquitectura de un Mesh Access Point para implementar la técnica WNC.

El MAP debe **verificar constantemente si existe una oportunidad de codificación**; es decir, que se haya almacenado al menos un paquete nativo en cada una de las filas S_1 y S_2 . Al existir una oportunidad de codificar, el MAP puede realizar la operación XOR con un paquete nativo de cada fila, dando como resultado el **paquete codificado** $P_\alpha(i, j) = S_1(P_i) \oplus S_2(P_j)$. En este instante, el paquete P_α está listo para ser transmitido en el medio inalámbrico, compartido por las STAs, el MAP y la Mesh STA. Para incrementar la probabilidad de transmisión de los paquetes codificados, se propone una nueva categoría para EDCA denominada como *Network Coding* (NC). Esta nueva categoría de acceso tiene configurados sus parámetros de operación (CW_{\min} , CW_{\max} y AIFS_N) para priorizar su acceso al medio por encima de las otras categorías. Por otro lado, las categorías definidas por EDCA, *voice* (VO), *video* (VI), *background* (BK) y *best-effort* (BE), siguen operando para la transmisión de paquetes nativos.

Cuando la categoría NC obtiene el acceso al medio, se transmite el paquete codificado P_α ; esta transmisión es recibida simultáneamente por la STA destino y por el Mesh STA —gracias a la propiedad de difusión del medio inalámbrico—. Para simplificar el modelo propuesto se asume que las STAs son capaces de **escuchar de manera oportunista** las transmisiones de otras STAs; esto significa que pueden almacenar copias de paquetes nativos que serán utilizados durante el proceso de decodificación —los problemas de seguridad de este modelo tienen oportunidad de ser corregidos en capas superiores de la pila de protocolos—. Así, una vez ocurrido lo anterior, se puede asumir que todas las STAs tienen una copia de los paquetes almacenados en S_2 y la Mesh STA tiene una copia de todos los paquetes almacenados en S_1 . Si se omite esta consideración, el MAP debe codificar de manera selectiva; es decir, codificar únicamente con base en flujos y no en paquetes, lo

cual reducirá significativamente las oportunidades de codificación. Cuando las STAs y la Mesh STA reciben el paquete codificado $P_\alpha(i, j)$, deben utilizar los paquetes nativos que han almacenado previamente para realizar el proceso de decodificación con una operación XOR. Para llevar a cabo la decodificación es necesario que los paquetes codificados P_α pueda notificar los identificadores de los paquetes nativos, i y j , que se utilizaron para construirlo, véase la Figura 6.2. Una de las ventajas más significativas de este escenario es que los paquetes codificados siempre podrán ser decodificados, debido a la escucha oportunista que realizan las STAs y a las direcciones de los flujos TCP y ACK en la red.

6.3 Retardo de codificación

La generación de oportunidades de codificación está estrechamente relacionada con la densidad de tráfico en la red. Esto quiere decir que si las densidades de tráfico en las direcciones STAs→MP y MP→STAs son desiguales, quedan reducidas las oportunidades de codificación. Esta reducción se explica con el siguiente ejemplo. Cuando se recibe un paquete nativo y se almacena en su respectiva fila de codificación puede suceder que, *i*) la otra fila de codificación tenga almacenados paquetes nativos –pudiendo construir inmediatamente un paquete codificado–, o *ii*) la fila puede estar vacía debido a una menor densidad de tráfico –aquí no se puede construir inmediatamente un paquete codificado–. Como la cantidad de paquetes que se almacenan en ambas filas de codificación depende de la densidad de tráfico en la red y si la densidad en una dirección es mayor que en la otra, entonces una de las filas de codificación se vaciará a una tasa mayor que la tasa a la que se llena, y, consecuentemente, esta fila se vaciará. Es así que, si un paquete nativo, P_i , es clasificado y almacenado en la fila de codificación S_1 y, además, estando vacía la fila de codificación S_2 , entonces puede realizarse una de tres posibles acciones (refiérase a la Figura 6.3):

1. El paquete P_i **espera durante tiempo ilimitado** en la fila de codificación S_1 hasta que el paquete P_j arribe y se almacene en la fila S_2 , conformando una oportunidad de codificación.
2. El paquete P_i **espera durante tiempo limitado** en la fila de codificación S_1 hasta que el paquete P_j arribe y se almacene en la fila S_2 , formando así una oportunidad de codificación. Si el tiempo de espera límite expira, entonces el paquete P_i se transmite en su forma nativa [108, 107, 10].
3. Si paquete P_i no encuentra una oportunidad de codificación, en esta situación **no espera** por P_j en la fila de codificación, sino que se transmite en su forma nativa (*No delayed transmission*) [7].

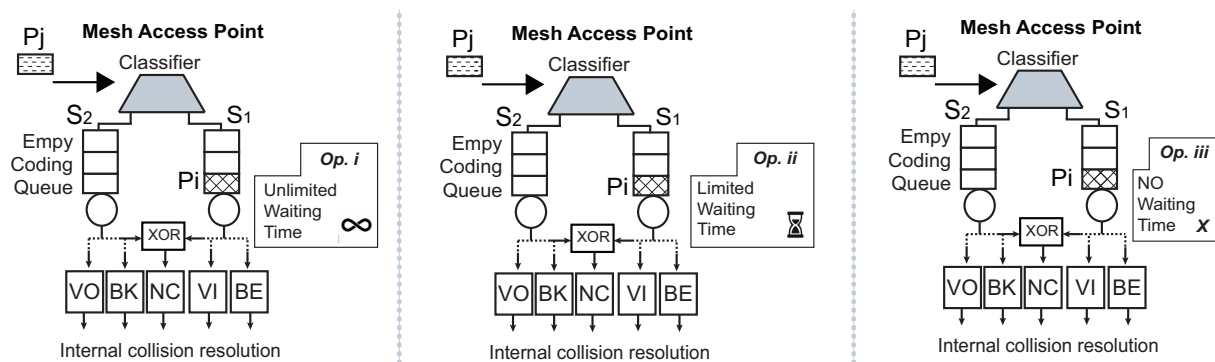


Figura 6.3: Cuando el paquete P_i se almacena en la fila S_1 y la fila S_2 está vacía, P_i puede: *i*) esperar a P_j un tiempo ilimitado, *ii*) esperar a P_j un tiempo limitado, o *iii*) no esperara a P_j y es transmitido en su forma nativa.

Es las acciones anteriores se demuestra que las primeras dos alternativas agregan un retardo a los paquetes nativos. Dicho **retardo de codificación** termina por afectar a aplicaciones sensibles al retardo e incluso llegar a afectar el desempeño de protocolos que operan con base en la estimación del RTT, tal como sucede con TCP. El retardo de codificación puede estimarse de la siguiente manera: En el instante de tiempo t_i , el paquete P_i es almacenado en S_1 y la fila de codificación S_2 está vacía. El paquete P_i debe esperar que arribe un paquete P_j y

se guarde en la fila S_2 , para poder llevar a cabo la codificación. Cuando el paquete P_j arriba en el instante de tiempo t_j , es decir, $t_i < t_j$, se puede realizar inmediatamente la codificación $P_i \oplus P_j$. El retardo de codificación será el tiempo que el paquete P_i esperó por la oportunidad de codificación, calculándose como a $t_j - t_i$. La primera estrategia, que sería esperar un tiempo ilimitado hasta crear una oportunidad de codificación, termina afectando seriamente a las aplicaciones, pues puede inducirse un mayor retardo de codificación si la tasa de arribos a las filas de codificación es desigual, provocando que $t_i \ll t_j$. La segunda alternativa, consistente en limitar el tiempo de espera de un paquete nativo, requiere establecer el **tiempo máximo de espera**, junto con la interrogante: **¿qué valor debe asignarse a ese máximo tiempo de espera?** Si el tiempo de espera máximo es pronunciado, puede llegar a afectar el desempeño de las aplicaciones, y, si el mismo es reducido, se verían reducidas las oportunidades de codificación [10]. La última alternativa, donde un paquete nativo no espera por la formación de una oportunidad de codificación, elimina el retardo de codificación, perdiendo con esto los beneficios de WNC, y en consecuencia reducir significativamente el número de oportunidades de codificación [107].

6.4 Modelo del sistema como una DTMC

El tamaño de las filas de codificación y su ocupación son factores determinantes para la creación de oportunidades de codificación [10, 107]. Dado que el medio inalámbrico es compartido por STAs, MAP y Mesh STA, la contienda por el medio y la probabilidad de transmisión de cada entidad influye directamente en la ocupación de las filas de codificación. El número de paquetes almacenados en estas filas S_1 y S_2 varía en el tiempo y puede ser modelado como un *Proceso de Markov*. Por consiguiente, el tamaño conjugado de las filas de codificación, es decir, (L_{S_1}, L_{S_2}) , es un proceso estocástico bidimensional, y sus diferentes estados y transiciones pueden modelarse como una cadena de Markov en tiempo discreto (*Discrete-time Markov Chain - DTMC*) [143]. Una condición necesaria para asumir que el proceso es markoviano –independencia entre el presente y futuro de la DTMC– es que el número de flujos de datos en el sistema debe ser suficientemente grande [144].

Asuma que M_1 y M_2 determinan capacidad máxima de las filas de codificación S_1 y S_2 . Además, $N(t)$ es el proceso de Markov que representa el número de paquetes i y j almacenados en el instante de tiempo t en S_1 y S_2 , respectivamente. Asimismo, asuma que P_γ es la probabilidad de transmisión del elemento de red γ , donde $\gamma = \{\text{MAP}, \text{STA}, \text{MP}\}$. Si se define una ventana de observación Δ_t , lo suficientemente precisa para observar únicamente un evento de transmisión en el medio, entonces la dinámica de $N(k \cdot \Delta_t)$, donde $k = 1, 2, 3, \dots, n$, puede ser representada por el diagrama de estados de la Figura 6.4. El **incremento del número de paquetes almacenados en las filas de codificación se rige por dos diferentes transmisiones en el medio**. Por un lado, cada que una STA transmite un paquete hacia el MAP, el proceso $N(t)$ pasa del estado (i, j) al estado $(i + 1, j)$ con probabilidad P_{STA} –esto significa que el paquete nativo proveniente de la STA se almacena en la fila de codificación S_2 ; en otras palabras, L_{S_1} incrementa una unidad. Por otro lado, si la Mesh STA transmite al MAP, el proceso $N(t)$ transita del estado (i, j) al estado $(i, j + 1)$ con probabilidad P_{MP} –esto significa que el paquete nativo proveniente de la Mesh STA se almacena en la fila de codificación S_2 ; en otras palabras, L_{S_2} incrementa una unidad.

Similarmente, existen **dos eventos que reducen el tamaño de las filas de codificación**. Por un lado, las transiciones entre los estados $(0, j)$ y $(0, j - 1)$, para $j > 0$ y entre los estados $(i, 0)$ y $(i - 1, 0)$, para $i > 0$, se realizan **cada que el MAP transmite un paquete nativo con probabilidad P_{MAP} –esto significa que cuando el MAP tiene la oportunidad de transmitir, lo hace aun si no existe una oportunidad de codificación; es decir, los paquetes nativos no esperan en las filas de codificación**. Por otro lado las transiciones del estado (i, j) al estado $(i - 1, j - 1)$, para $i, j > 0$, se realizan cada que el MAP realiza una transmisión codificada –esto quiere decir que **si ambas filas de codificación tienen más de un paquete almacenado, el MAP toma un paquete de cada fila para realizar el proceso de codificación**; en otras palabras L_{S_1} y L_{S_2} reducen en una unidad. Cuando la DTMC se encuentra en el estado (i, M_2) (es decir, fila S_2 llena) y una STA transmite; el paquete recibido por el MAP se almacena en la respectiva fila de codificación y el paquete con mayor tiempo de espera en S_2 pasa inmediatamente a las filas EDCA para su transmisión en forma nativa; por este motivo el proceso se mantiene en el estado (i, M_2) con probabilidad P_{STA} . Lo mismo sucede si el la DTMC se encuentra

en el estado (M_1, j) (es decir, fila S_1 llena) y la Mesh STA transmite un paquete al MAP con probabilidad P_{MP} , el proceso se mantiene en el estado (M_1, j) y el paquete con mayor tiempo de espera en la fila S_1 se transmite en su forma nativa. Las transiciones del estado $(0,0)$ a los estados $(0,1)$ y $(1,0)$ tienen mayores probabilidades, debido a que el MAP no tiene paquetes que transmitir, así que no contienda por el medio y la probabilidad de transmisión de la STA y la Mesh STA se incrementa en $P_{MAP}/2$. Para concluir, el estado (M_1, M_2) indica que las filas de codificación están llenas y por lo tanto las transmisiones de las STAs y del Mesh STA no afectan el tamaño de las filas, por esa razón el proceso se mantiene en ese estado con probabilidad $P_{MP} + P_{STA}$. Es claro que **el proceso tiene un total de $(M_1 + 1) \times (M_2 + 1)$ posibles estados, de los cuales $(M_1 \times M_2)$ son estados en los que se puede realizar el proceso de codificación**; es decir, estados en los que i y j son mayores a 0 (véase la Figura 6.4).

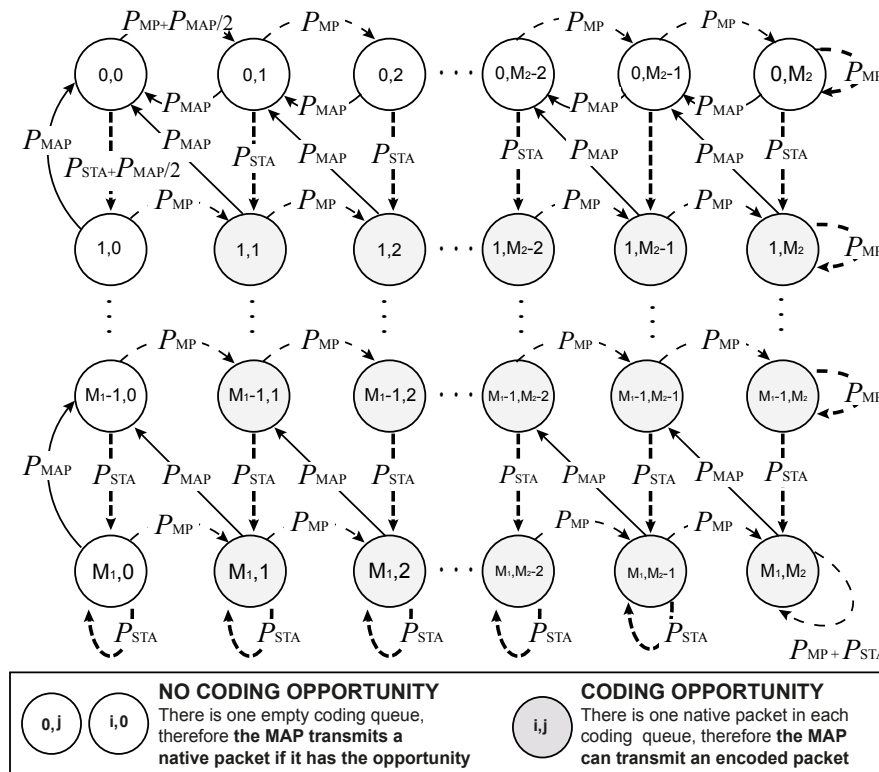


Figura 6.4: Diagrama de estados que representa la dinámica del proceso aleatorio bidimensional $N(\Delta_t)$.

Se debe asumir que el sistema propuesto en la Figura 6.2 opera en condiciones de saturación; es decir, las filas de transmisión de las STAs y de la Mesh STA siempre tienen paquetes esperando ser transmitidos hacia el MAP. Además, entre los tres elementos de la red (MAP, Mesh STA y STAs) la distancia es suficiente como para evitar el fenómeno de terminal oculta y solo puede darse una transmisión en el medio. Para concluir, las oportunidades de transmisión se distribuyen equitativamente entre los elementos de la red. Esto significa que el proceso $N(\Delta_t)$ dependerá del número máximo de paquetes que pueden ser almacenados en las filas de codificación (M_k) y de las probabilidades de transmisión de cada dispositivo.

6.4.1 Estado estacionario y transiciones de la DTMC

En esta subsección se evalúa mediante simulaciones numéricas con el software *Wolfram Mathematica 10* el estado estacionario $\pi_{i,j}$ de la DTMC [143, 145]. El objetivo de esta evaluación es capturar la formación de oportunidades de codificación y de cómo son afectadas por la probabilidad de acceso al medio, como también el tamaño máximo de las filas de codificación. Inicialmente se ha configurado un acceso equiprobable al me-

dio; por lo tanto se utilizan las siguientes probabilidades de transición: $P_{MAP} = 1/3, P_{MP} = P_{STA} = (1 - P_{MAP})/2$.

Se simularon un millón de transmisiones aleatorias en el medio y cada una de ellas representa una transición de la DTMC, tal como se detalla el modelo del sistema. En la Figura 6.5 se muestra como varía la probabilidad de estado estacionario, $\pi_{i,j}$, a medida que incrementa la capacidad máxima de las filas de codificación, es decir, $M_k = \{2, 3, 4, 9\}$. Se aprecia que cuando M_k se incrementa, entonces la probabilidad de que el sistema se encuentre en un estado donde hay una oportunidad de codificación, es decir $[(i, j); i > 0, j > 0]$, es mayor que la probabilidad de hallarse en un estado donde la codificación no puede aplicarse, es decir $[(i, j); i = 0 | j = 0]$. Esto se debe a que la probabilidad de transmisión del MAP es menor que la suma de las probabilidades de transmisión de la STA y el Mesh STA y, además, se incrementa el número de estados donde es posible realizar la codificación, es decir, $M_k^2 = \{4, 9, 16, 81\}$. En todos los casos analizados, la probabilidad de que la DTMC se encuentre en el estado (M_k, M_k) es mayor que la probabilidad de estar en cualquier otro estado. Este resultado es propiciado, igualmente, por la probabilidad de transmisión del MAP, ya que esta configuración, como ha sido mostrado, se traduce en una mayor tasa de arribos a las filas de codificación y su respectiva saturación, debido a la baja tasa de transmisión que obtiene el MAP. Por lo tanto, la probabilidad de transmisión del MAP P_{MAP} y el tamaño de las filas de codificación tienen ambos un efecto significativo en la formación de oportunidades de codificación. Se debe resaltar que en el modelo evaluado, el MAP transmite paquetes nativos de las filas de codificación incluso si no existe una oportunidad de codificarlos.

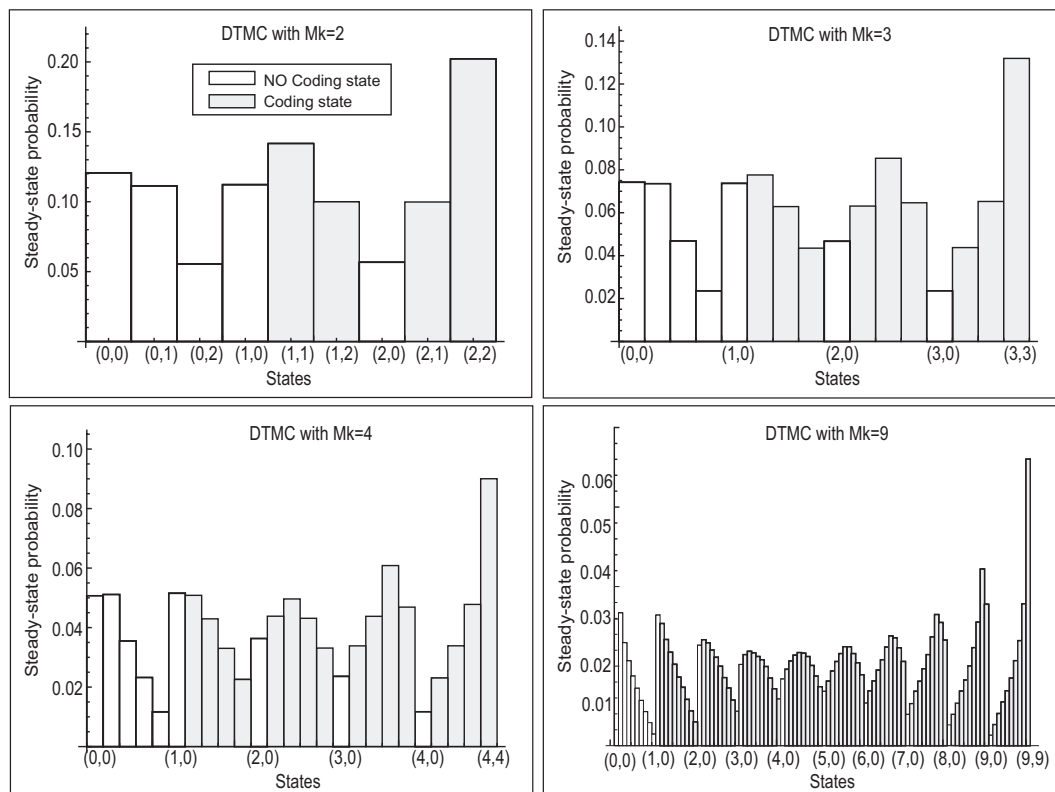


Figura 6.5: Probabilidad de estado-estable de la DTCM como función de M_k y $P_{MAP} = P_{MP} = P_{STA} = 1/3$.

En la Figura 6.6 se muestra el porcentaje normalizado de ocurrencia de las seis posibles transiciones de la DTMC. Se puede apreciar que al incrementar el valor de M_k , de 2 a 9, el porcentaje de transiciones que representan una transmisión codificada por el MAP (barra A en la imagen) incrementa del 18% al 28%. También se puede ver que el número de transmisiones de paquetes nativos sin codificar por el MAP (barras B y C en a imagen) reduce el 10% al 4% –el modelo del sistema considera que el MAP compite por el acceso al medio si existen paquetes nativos almacenados en las filas de codificación, es decir, los paquetes

nativos no esperan por otros paquetes para poder ser codificados y se transmiten en cualquier una oportunidad de hacerlo. Paralelamente, el número de paquetes que arriban al MAP que encuentran las filas de codificación saturadas (barra D en la imagen) se reduce del 24 al 7%, pues hay más localidades para el almacenamiento de paquetes nativos. También es de destacar que el número de transmisiones de paquetes nativos realizadas por la Mesh STA y la STA (barras E y F de la imagen) incrementan de un 23% a un 30%. Estos datos señalan la importancia que tiene el tamaño máximo de las filas de codificación, evidenciando que al incrementar, también lo hacen las oportunidades de codificación y la capacidad de la red.

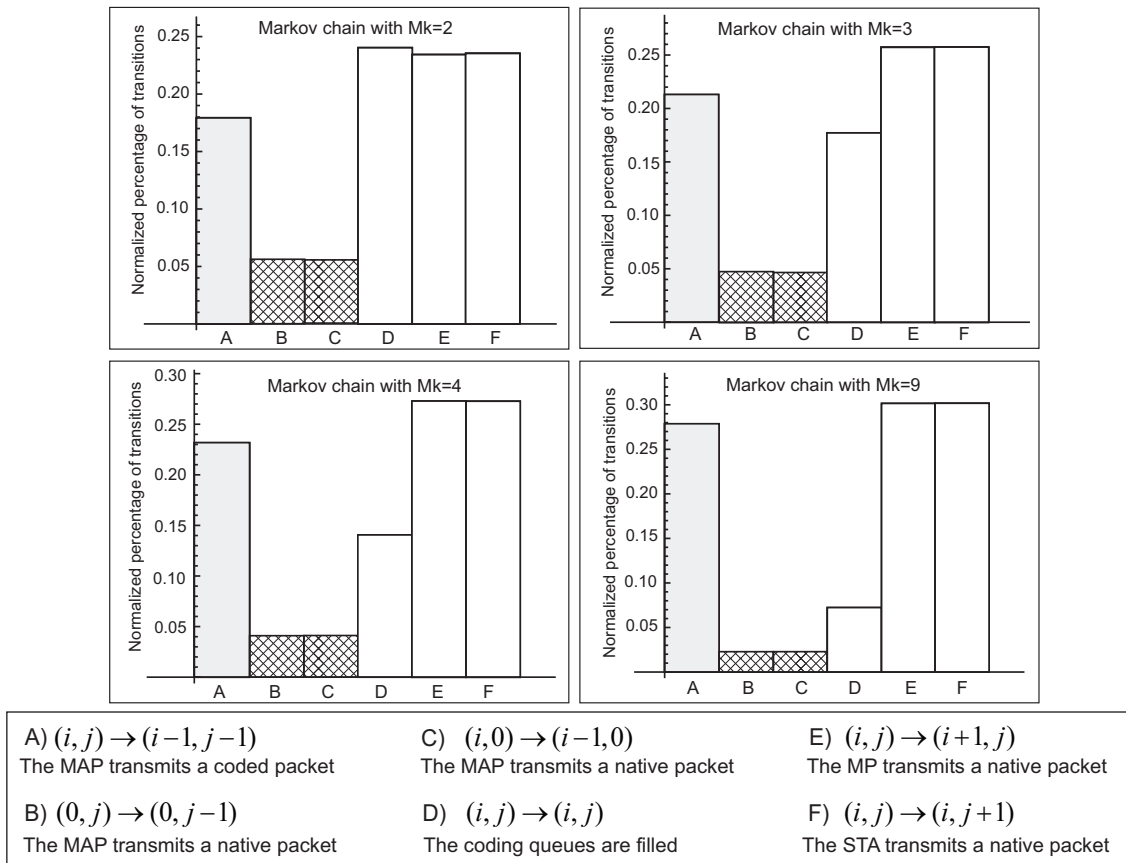


Figura 6.6: Número normalizado de transiciones de la DTMC como función de M_k y $P_{MAP} = P_{MP} = P_{STA} = 1/3$

Para valorar diferentes probabilidades de transmisión para el MAP y sus efectos en la DTMC, se evaluó el comportamiento del proceso, utilizando una distribución equitativa de las probabilidades de transmisión P_γ . La probabilidad P_{MAP} puede incrementarse para priorizar las transmisiones realizadas por el MAP o, como otra opción, reducir para incrementar la probabilidad de transmisión de las STAs y la Mesh STA. La Figura 6.7 muestra el número de ocasiones en la que la DTMC pasa por los diferentes estados del proceso como función de M_k e incrementando la probabilidad de transmisión ($P_{MAP} = 1/4, 1/3, 1/2, 2/3$), mientras que $P_{MP} = P_{STA} = (1 - P_{MAP})/2$. Se puede apreciar en la gráfica que al verse incrementada la probabilidad P_{MAP} , se reducen las oportunidades de codificación; por ello el número de ocurrencias de estados en donde las filas de codificación tienen paquetes para codificar disminuye. Esto quiere decir que **al incrementar la probabilidad de transmisión del MAP, quedan vacías las filas de codificación; ya que el MAP transmite los paquetes en las filas de codificación, sin esperar la formación de una oportunidad de codificación.** Este mismo comportamiento se presenta si se incrementa el tamaño máximo de las filas de codificación M_k . Como conclusión de estos resultados, en la Figura 6.8 se contabilizan las diferentes transiciones que realiza la DTCM. Aquí se aprecia que al incrementar P_{MAP} , el MAP transmite una menor cantidad de paquetes codificados junto con

una gran cantidad de paquetes nativos, teniendo, en este escenario, un menor impacto el tamaño máximo de las filas de codificación M_k .

6.4.2 DTMC sin transmisión de paquetes nativos

En el modelo previo se consideró como el MAP puede transmitir paquetes nativos sin esperar la formación de una oportunidad de codificación; demostrando que, si un paquete nativo se encuentra en las filas de codificación, el MAP compite por el medio tratando de transmitirlo. Por tal razón, la DTMC que se muestra en la Figura 6.4 tiene transiciones entre los estados $(0, j)$ y $(0, j-1)$, para $j > 0$, y entre los estados $(i, 0)$ y $(i-1, 0)$, para $i > 0$. Esta consideración propició una elevada tasa de transmisión de paquetes nativos, llegando, en algunos casos, a superar a la tasa de transmisión de paquetes codificados. En esta subsección se modifica el modelo y se asume lo siguiente: El MAP recibe paquetes nativos provenientes de las Mesh STA y las STAs, pasando a ser almacenados en las filas de codificación; sin embargo, si no existe una oportunidad de codificación, el MAP no compite por el medio para transmitir. El MAP solo compite por el medio si tiene un paquete codificado por transmitir. Estas consideraciones modifican la DTMC, tal como se muestra en la Figura 6.9. Cuando el proceso se encuentra en un estado $(0, j)$ o $(i, 0)$, las transiciones solo pueden darse hacia los estados $(0, j+1)$ o $(i+1, 0)$ con probabilidades $P_{MP} + P_{MAP}/2$ y $P_{STA} + P_{MAP}/2$, respectivamente.

Se simularon un millón de transiciones de la DTMC, y en la Figura 6.10 se muestran las ocurrencias de los diferentes estados como función de M_k y de la probabilidad P_{MAP} . En contraste con los resultados que se muestran en la Figura 6.7, ahora, al incrementar la probabilidad P_{MAP} , también aumenta la probabilidad de ocurrencia de que la DTMC se encuentre en estados donde puede llevarse a cabo el proceso de codificación. Además, se aprecia que al incrementar el tamaño límite M_k se reduce la probabilidad de saturar las filas de codificación. Complementando este resultado, en la Figura 6.11 se muestra el conteo de transiciones de la DTMC. Se aprecia que se eliminan las transiciones etiquetadas con B y C (transmisión de paquetes nativos). Quedando claro que al incrementar M_k y P_{MAP} se reduce la probabilidad de saturar las filas de codificación y se incrementa el número de transmisiones de paquetes codificados.

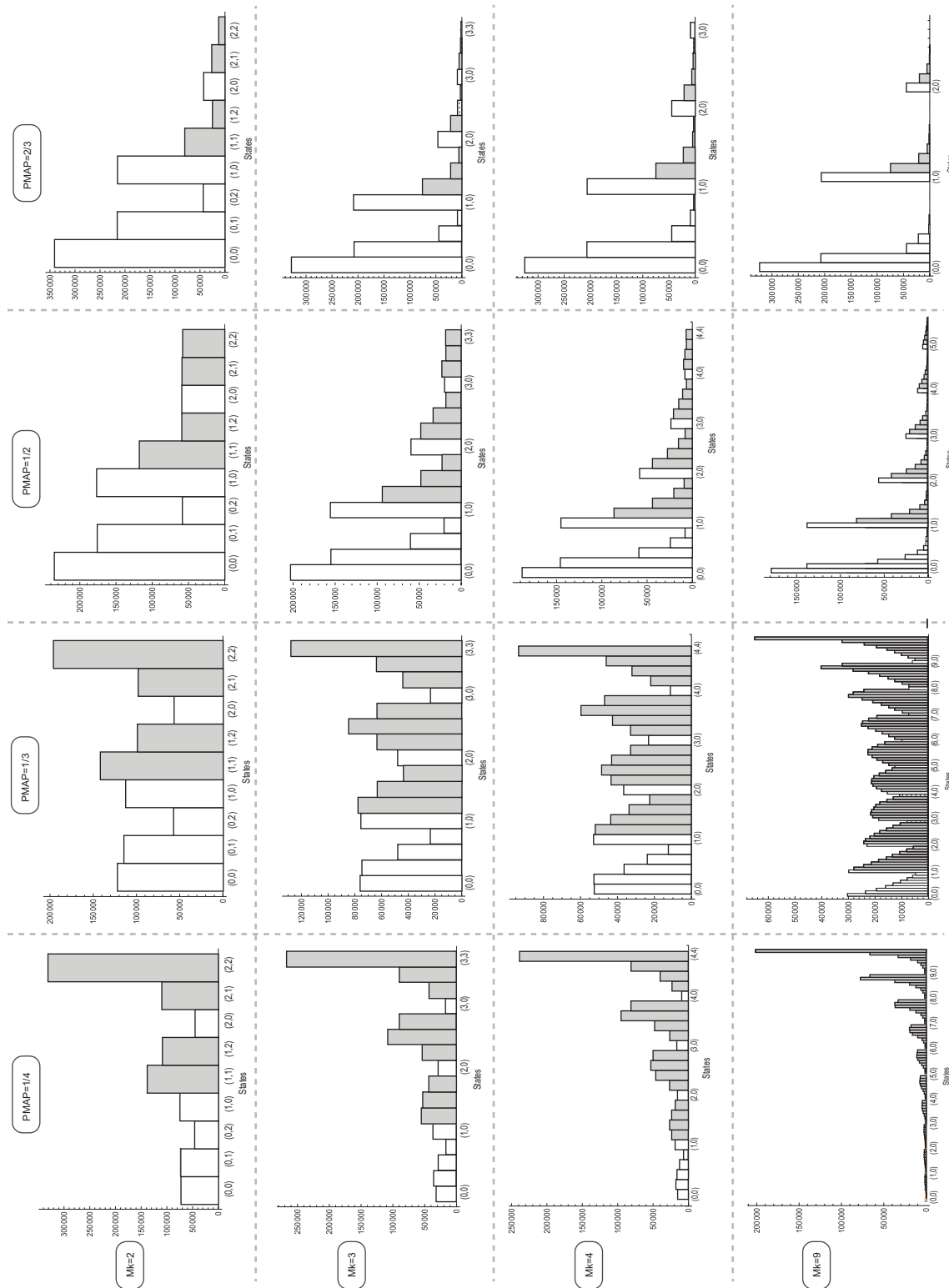


Figura 6.7: Histogramas de ocurrencias de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$ y $M_k = \{2, 3, 4, 9\}$.

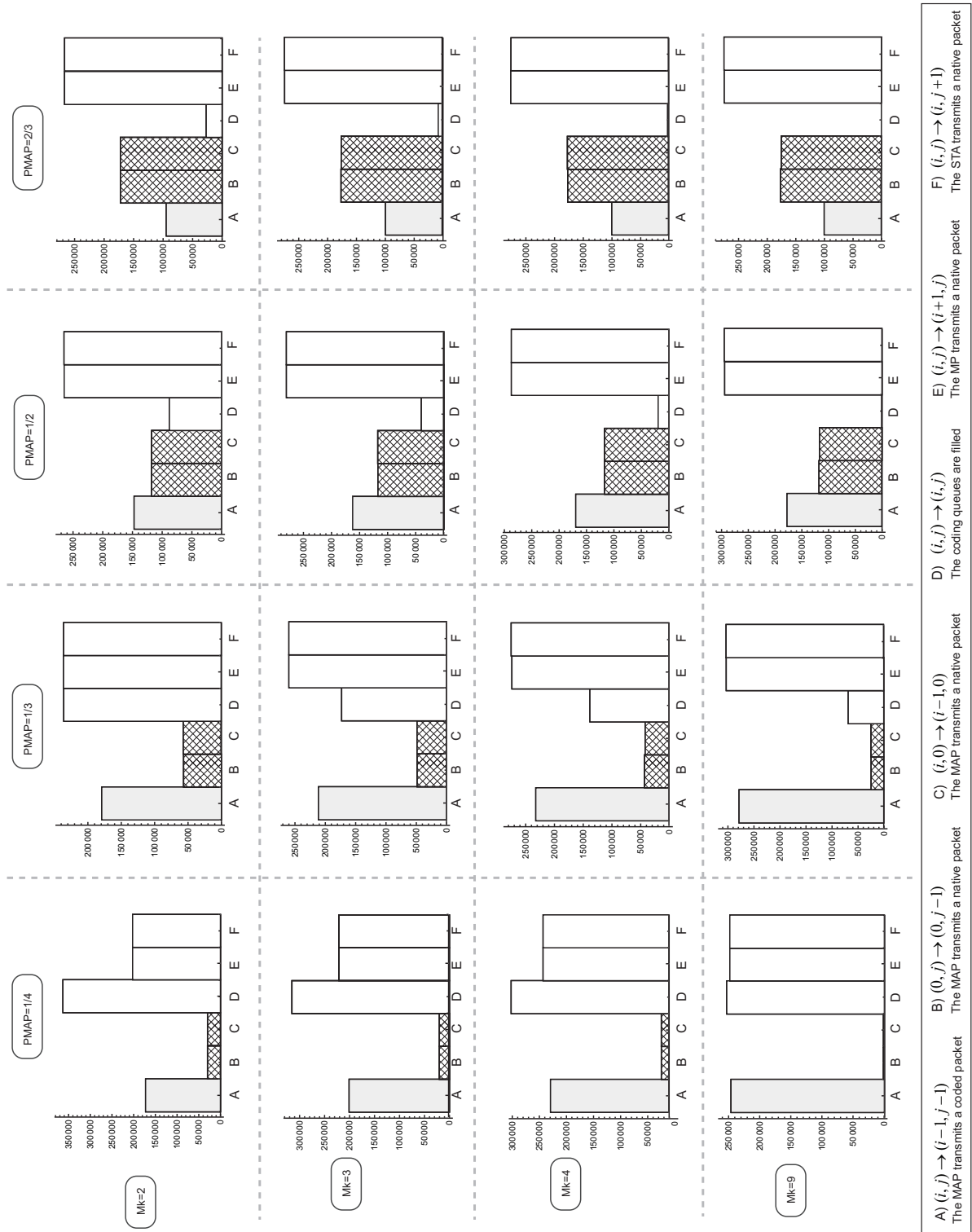


Figura 6.8: Número de transiciones de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$, $P_{STA} = P_{MP} = (1 - P_{MAP})/2$ y $M_k = \{2, 3, 4, 9\}$.

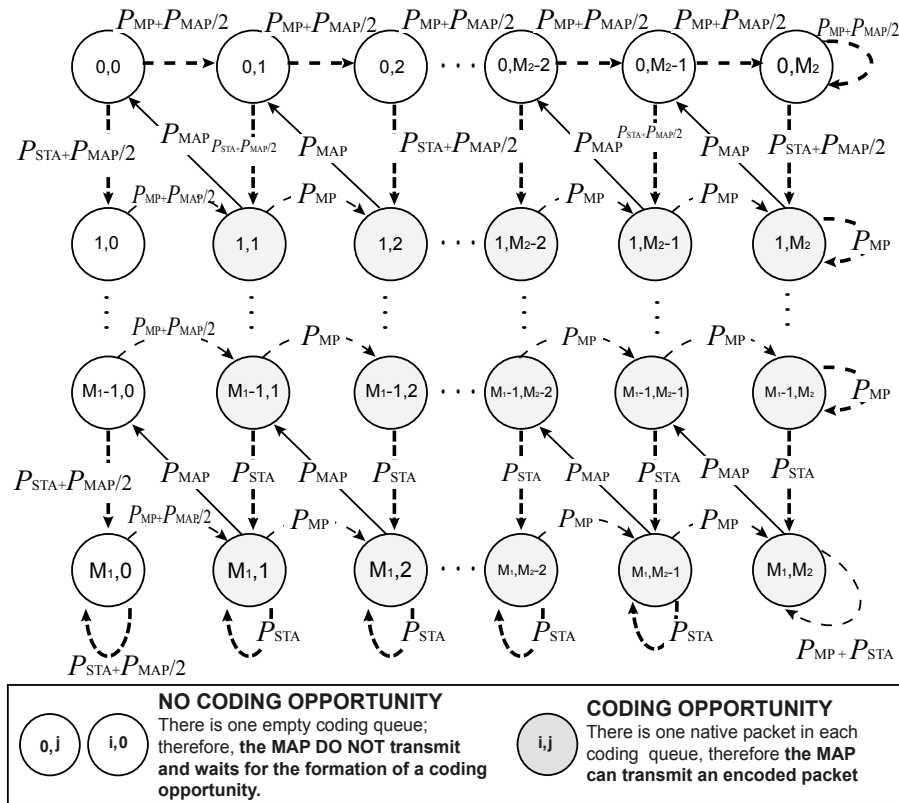


Figura 6.9: Diagrama de estados que representa la dinámica del proceso aleatorio bidimensional $N(\Delta_t)$.

6.5 Codificación de red en un MAP

Ha quedado demostrado que el algoritmo de acceso al medio y, por consiguiente, la probabilidad de transmisión del MAP, es un factor fundamental para la creación de oportunidades de codificación. El acceso al medio dictado por la enmienda IEEE 802.11s es de carácter aleatorio y no priorizado, lo cual podría impactar directamente en la formación de oportunidades de codificación. Además, con base en los anteriores resultados, se ha concluido que la estrategia de espera para formar oportunidades de codificación puede maximizar los beneficios que brinda la técnica de codificación de red. Con auxilio del simulador de redes NS-2, en esta sección se evalúa la generación de oportunidades de codificación en un MAP, así como el retardo causado por la espera de una oportunidad de codificación. Logrando de esta manera verificar el efecto que tiene este retardo sobre el desempeño de protocolos tales como TCP. Por ahora se excluye de este análisis el proceso de codificación, ya que la presente evaluación está únicamente enfocada en el proceso de formación de oportunidades de codificación.

6.5.1 Escenario de simulación

Para esta evaluación se ha modificado el modelo de nodo inalámbrico provisto por el simulador NS-2 con el propósito de incorporar un par de filas de codificación, Q1 y Q2, en un nodo inalámbrico. Estas filas son utilizadas para almacenar los paquetes nativos provenientes de la Mesh STA y de las STAs con el objetivo de generar oportunidades de codificación, la Figura 6.12 muestra los detalles de esta implementación. La fila Q1 almacena paquetes provenientes de las STAs, y la fila Q2 paquetes provenientes de la Mesh STA. El método de acceso al medio se rige por la función IEEE 802.11 - DCF, lo cual brinda aleatoriedad y competencia entre los dispositivos de la red. El MAP registra los cambios en las filas de codificación Q1 y Q2; los instantes de arribo

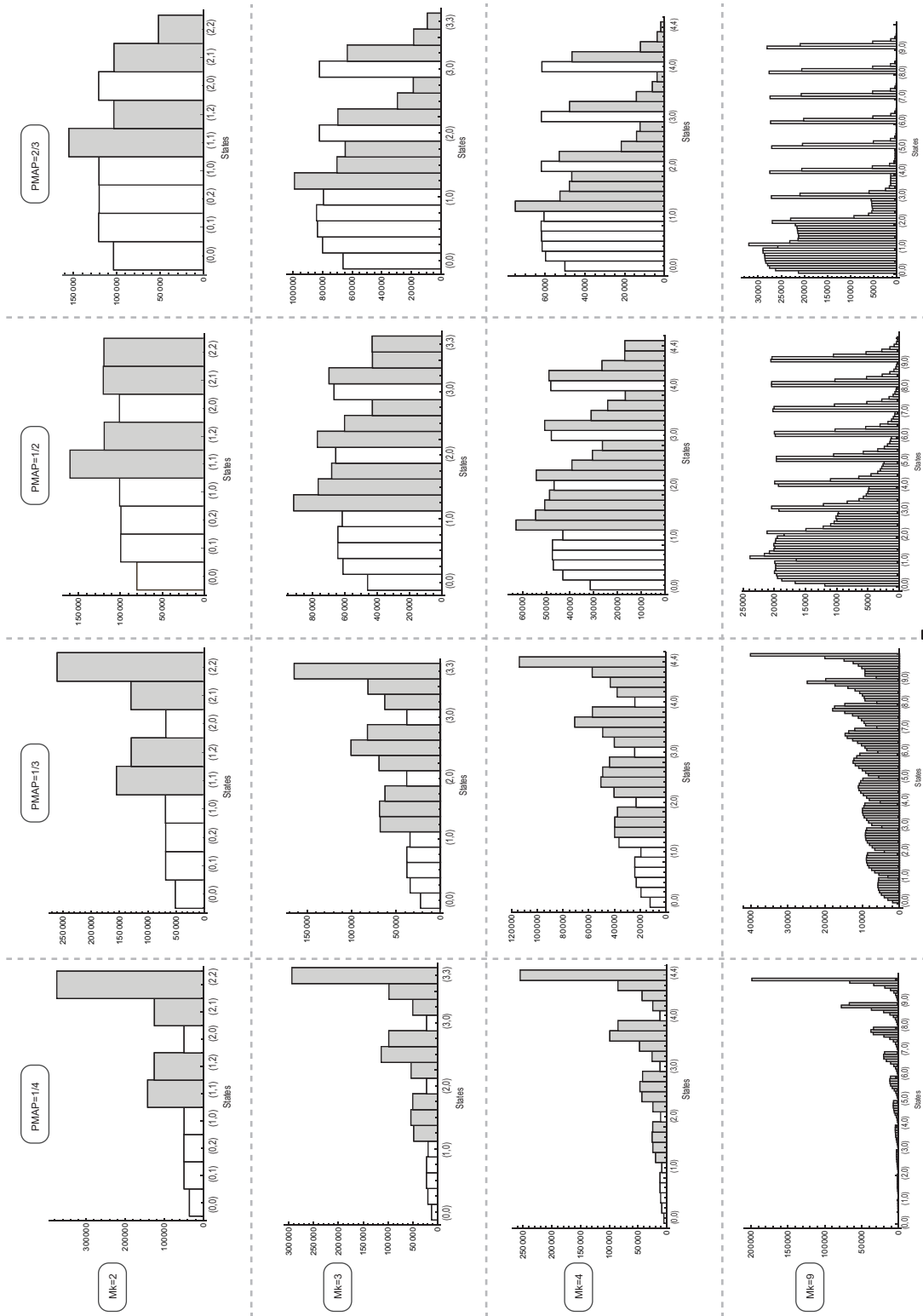


Figura 6.10: Histogramas de ocurrencias de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$ y $M_k = \{2, 3, 4, 9\}$.

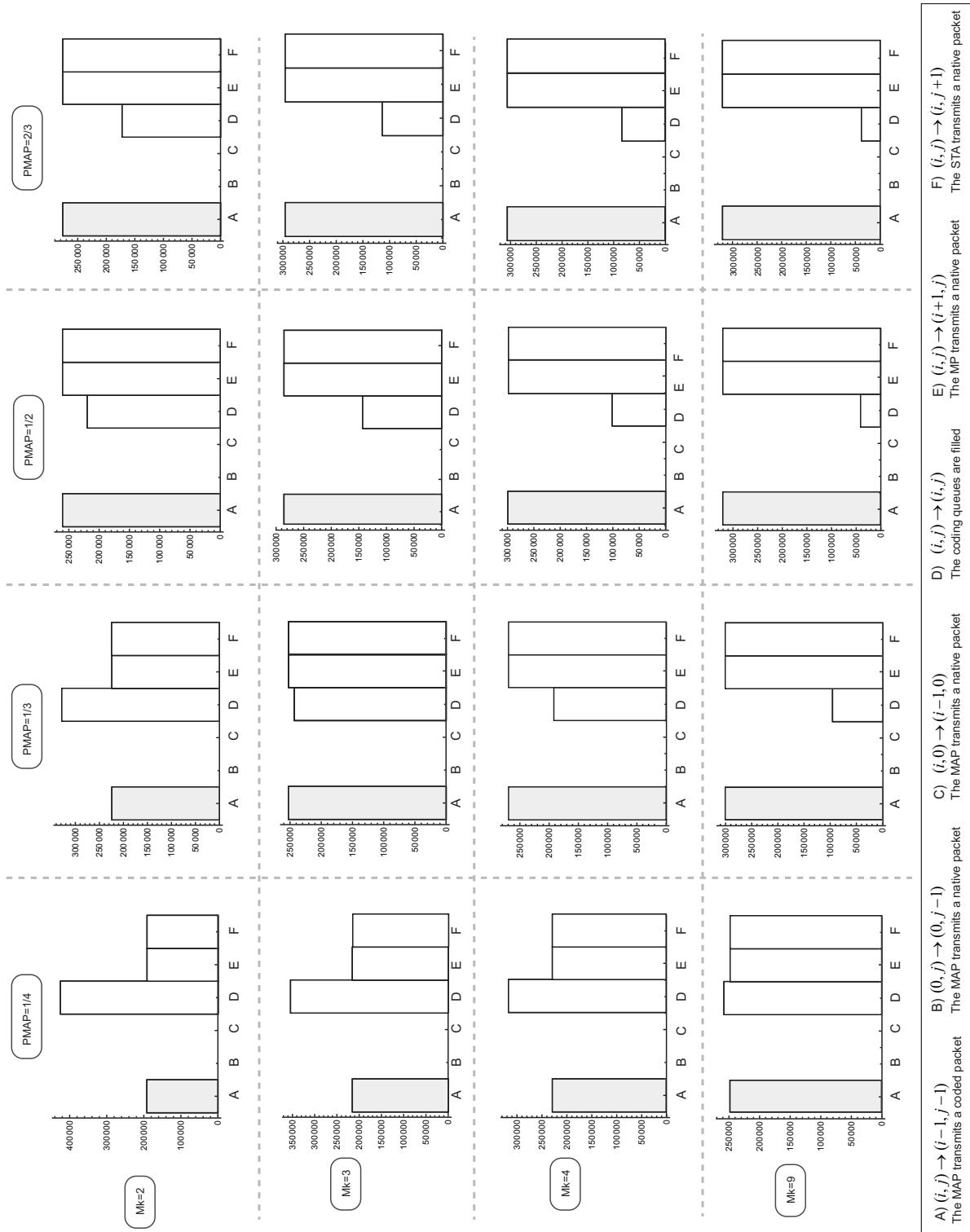


Figura 6.11: Número de transiciones de la DTMC como función de $P_{MAP} = \{1/4, 1/3, 1/2, 2/3\}$, $P_{STA} = P_{MP} = (1 - P_{MAP})/2$ y $M_k = \{2, 3, 4, 9\}$.

en cada fila de codificación son almacenados para construir una serie de tiempo. El tamaño máximo de las filas de codificación es ilimitado.

Cada una de las STAs del escenario establece una conexión TCP cuyo sentido, ya sea STA → Internet o Internet → STA, se define de manera aleatoria empleando una variable de Bernoulli simétrica. Así, el número de flujos STA→Internet es igual, en promedio, al número de flujos Internet→STA. La cantidad de STAs en la red se incrementa gradualmente para evaluar los efectos que tiene el aumento de la densidad de tráfico TCP en la generación de oportunidades de codificación. Se ha seleccionado TCP New-Reno como protocolo de transporte, sustentados principalmente por los resultados obtenidos en la evaluación previa en la que se ha demostrado que TCP New-Reno supera a muchas otras variantes de TCP diseñadas específicamente para redes inalámbricas. Los detalles generales del escenario de simulación se sintetizan en la Tabla 6.1.

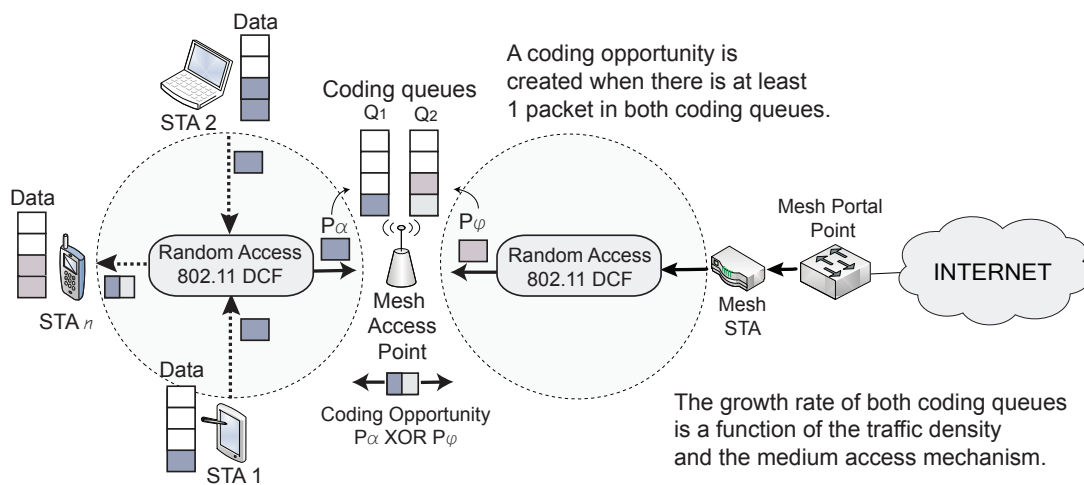


Figura 6.12: Escenario de simulación.

Tabla 6.1: Parámetros de simulación para la evaluación de WNC en iWMNs.

Parámetro	Valor
Número de STAs	1,2,4,8,16,32,64,128 STAs
R_{tx} (Alcance de transmisión)	250 [m]
R_{ir} (Alcance de interferencia)	550 [m]
R_{cs} (Alcance de detección de portadora)	550 [m]
Agente de tráfico	FTP persistente
Tamaño del paquete de datos	1460 bytes
Protocolo de transporte	TCP New Reno (TCPNR)
Protocolo de ruteo	AODV adaptado para iWMNs [3]
MAC	IEEE 802.11b DCF
Patrón de movilidad	Random Waypoint (Velocidad promedio 1 m/s)
Tasa Máxima de transmisión	11 Mb/s, (Auto Rate Fallback disabled)
Tiempo de simulación	1000 s

6.5.2 Tasa de arribos

En la Figura 6.13 se muestra como incrementa el número de paquetes almacenados en las filas de codificación Q1 y Q2 a medida que transcurre el tiempo de simulación. Las tasas de arribo a ambas filas tienen una tendencia lineal con pendientes que difieren significativamente cuando el número de estaciones es reducido, es decir, entre 1 y 4. Este resultado es originado por las pocas conexiones TCP y por ende un menor tráfico bidireccional en la red. Al añadir más STAs a la red se pueden incrementar las probabilidades de codificación de paquetes, porque, como se aprecia en la figura, ambas filas comienzan a tener una tasa de arribos muy similar. Esto confirma que la densidad de tráfico tiene un rol fundamental para poder maximizar los posibles beneficios de la técnica de codificación de red.

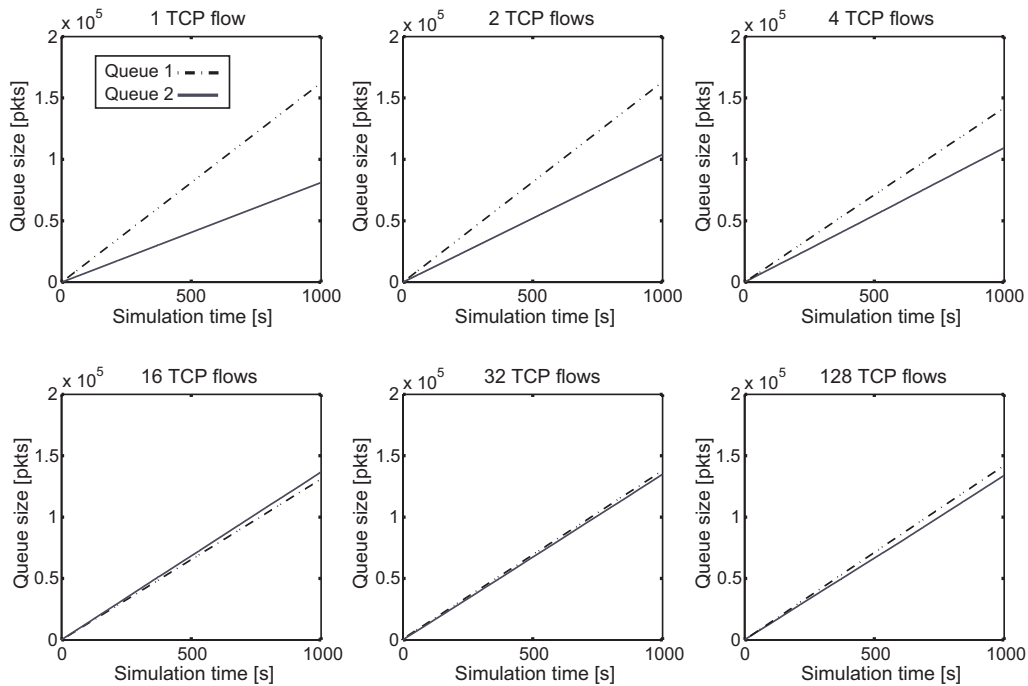


Figura 6.13: Tasa de arribos a las filas de codificación Q1 y Q2.

Cuando el número de STAs incrementa, la tasa de crecimiento de ambas filas de codificación es muy similar y alcanza un valor límite. Este efecto se aprecia con mayor claridad en la Figura 6.13, ya que son los casos en donde el número de STAs toma valores desde 16 hasta 128. Podemos inferir que el ancho de banda del medio inalámbrico junto con el algoritmo aleatorio de acceso al medio DCF limitan el acceso a un número elevado de STAs. Esta característica se traduce en una limitante para las posibles oportunidades de codificación en la red; dado que, aun cuando se incremente el tráfico TCP en la red, las oportunidades de codificación no se incrementan.

6.5.3 Oportunidades de codificación

Asuma que $\chi_\tau(Q_k)$ es una variable que contabiliza el número de paquetes recibidos en la fila de codificación Q_k durante una ventana de tiempo de duración τ segundos. En la Figura 6.14 se muestra la distribución estadística de 2000 muestras de $\chi_1(Q_k)$ a medida que incrementa el número de STAs en el escenario de simulación. Cuando el número de usuarios es reducido (es decir, 1,2,4), la cantidad de paquetes por segundo que se almacenan en ambas filas de codificación es desigual, lo cual se traduce en un menor número de oportunidades de codificación, pues ambas filas reciben un número muy distinto de paquetes por segundo. Ahora bien, cuando el número de STAs es mayor a 16 STAs, la media y el intervalo Inter cuartil (IRQ) de $\chi_\tau(Q_k)$ incrementa, y llega a ser similar en todos los escenarios de evaluación. Esto demuestra que el número de oportunidades de

codificación por segundo puede aumentar si lo hace también la densidad de tráfico en la red; sin embargo, habrá de señalarse que el límite máximo para las oportunidades de codificación queda condicionado por el ancho de banda del medio inalámbrico y el mecanismo de acceso al medio DCF.

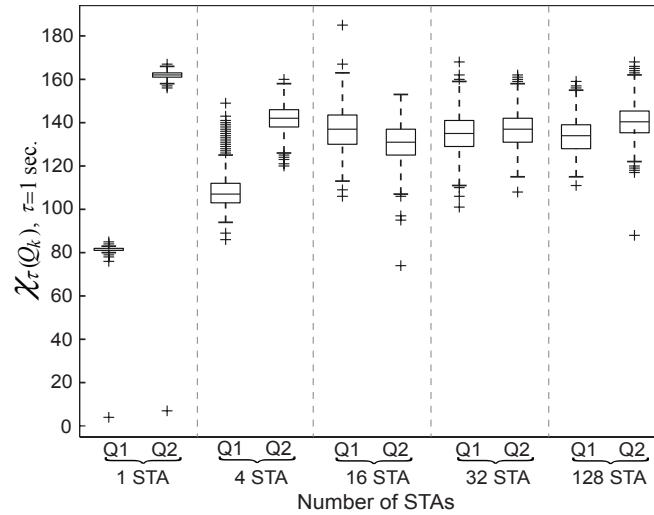


Figura 6.14: Estadísticas de $\chi_\tau(Q_k)$ con una ventana de observación $\tau = 1$ sec.

Las oportunidades de codificación en un intervalo de tiempo dependen del número de paquetes almacenados simultáneamente en ambas filas de codificación. Por consiguiente, el número de oportunidades de codificación es una función que varía en el tiempo dependiente del número mínimo de paquetes en las filas de codificación. De manera tal que el número de oportunidades de codificación (OPP_CODING) puede estimarse como:

$$\text{OPP_CODING}(t) = \begin{cases} Q1(t), & Q1(t) < Q2(t), & \text{La fila } Q1 \text{ limita las oportunidades de codificación} \\ Q2(t), & Q2(t) < Q1(t), & \text{La fila } Q2 \text{ limita las oportunidades de codificación.} \end{cases} \quad (6.1)$$

El número máximo de oportunidades de codificación (MAX_OPP_CODING) en un intervalo de tiempo se calcula cómo:

$$\text{MAX_OPP_CODING} = \max(\text{OPP_CODING}(t)). \quad (6.2)$$

En la Figura 6.15 se muestra la evolución temporal del número de paquetes almacenados en las filas de codificación, este es el resultado de una simulación numérica empleando *Wolfram Mathematica 11*. Las oportunidades de codificación en cada instante de tiempo están limitadas por la fila que almacena el menor número de paquetes; ya que estos podrían codificarse con los paquetes almacenados en la otra fila. De este modo es posible calcular el número total y el número máximo de oportunidades de codificación en una ventana de tiempo.

La Figura 6.16 muestra el número máximo de oportunidades de codificación en ventanas de observación de un segundo. Cuando el número de estaciones aumenta de 1 a 32, el número máximo de oportunidades de codificación promedio aumenta en de 80 a 130, lo cual representa un incremento superior al 40%. Sin embargo, cuando el número de STAs es mayor a 32, las oportunidades de codificación no aumentan significativamente; reafirmando así que existe un límite de desempeño establecido por el ancho de banda del medio inalámbrico y el algoritmo de acceso DCF. En la figura se observa que si se incrementan las STAs desde 32 hasta 128, el número de oportunidades de codificación se mantiene con un promedio constante, con un valor cercano a 130, y con un IRQ similar en todos los casos evaluados.

6.5.4 Número de transmisiones

Si se asume que el MAP puede codificar y transmitir todas las oportunidades de codificación que se le presentan, entonces se puede contabilizar el número total de transmisiones realizadas por el MAP, las STAs y el Mesh STA

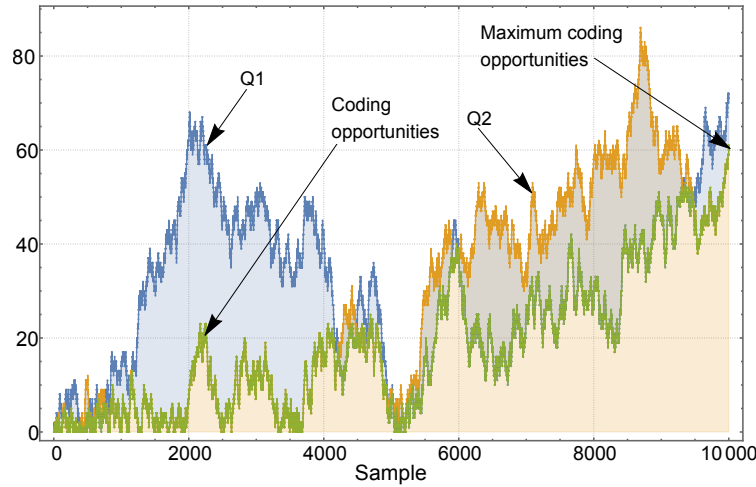


Figura 6.15: Formación de oportunidades de codificación con los paquetes de las filas de codificación Q1 y Q2.

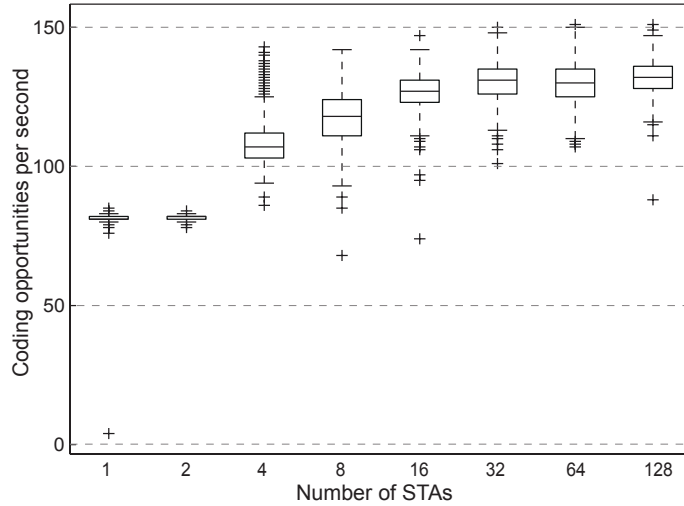


Figura 6.16: Oportunidades de codificación por segundo como función del número de STAs.

en el medio inalámbrico de la siguiente forma:

$$Tx_{STA} + Tx_{MeshSTA} + Tx_{OPP_CODING} + Tx_{NATIVE}. \quad (6.3)$$

Donde Tx_{STA} representa el número de transmisiones realizadas por las STAs y cuyos paquetes han sido almacenados en Q1; $Tx_{MeshSTA}$ representa el número de transmisiones realizadas por el Mesh STA y cuyos paquetes han sido almacenados en Q2; Tx_{OPP_CODING} representa el número de transmisiones codificadas realizadas por el MAP utilizando un paquete nativo de cada fila de codificación; finalmente, Tx_{NATIVE} representa el número de transmisiones de paquetes nativos realizadas por el MAP cuando no existe una oportunidad de codificación. El número total de transmisiones utilizadas con la técnica de codificación de red se puede comparar con el número de transmisiones utilizadas por el método de re-expedición *store-and-forward*, las cuales pueden calcularse como $2 \times (Tx_{STA} + Tx_{MeshSTA})$. La Figura 6.17 compara el número de transmisiones totales entre los nodos STAs-MAP-MP utilizando el esquema de reexpedición *store-and-forward* y el número de transmisiones totales empleando la técnica de codificación de red. Al utilizar esta última técnica se ahorran aproximadamente un 20% de las transmisiones totales. Este ahorro puede ser utilizado para la transmisión de un mayor número de

paquetes codificados, incrementando así el *throughput* de la red.

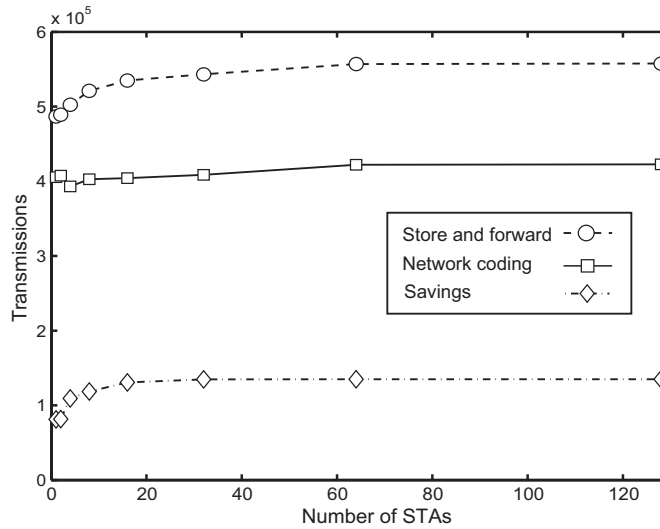


Figura 6.17: Comparación entre el número de transmisiones utilizadas con NC y Store-and-Forward.

En la Figura 6.18 se muestra la tendencia creciente del número total de oportunidades de codificación como función de las STAs en el escenario de simulación. Las oportunidades de codificación totales incrementan y alcanzan un punto máximo cuando el número de STAs es mayor 32. Si la densidad de usuarios supera dicho umbral, las oportunidades de codificación no incrementan significativamente y se mantienen constantes, alcanzando así un límite máximo. Cuando el número de usuarios aumenta desde 1 hasta 32, se nota un incremento sustancial del número de oportunidades de codificación, el cual alcanza hasta un 65 %.

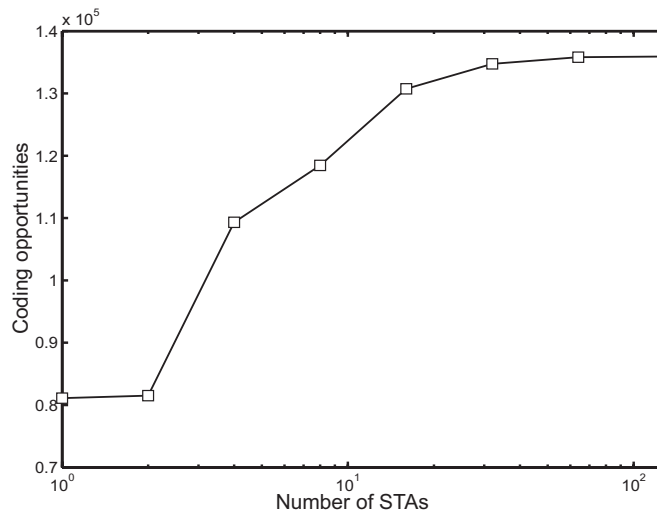


Figura 6.18: Oportunidades de codificaciones totales como función del número de STAs en la red.

6.5.5 Retardo de codificación

Tal como se analizó en la Sección 6.3, cuando un paquete se almacena en una de las filas de codificación pueden emplearse tres diferentes estrategias para utilizar la técnica de codificación de red: *i*) si no hay una oportunidad de codificación, el paquete se puede transmitir en su forma nativa (escenario caracterizado y analizado en la

Sección 6.4), *ii*) si no hay una oportunidad de codificación, el paquete debe esperar indefinidamente hasta que se forme una (escenario caracterizado y analizado en la Sección 6.4.2). Al contrastar los resultados de estos dos análisis se puede concluir que la segunda estrategia incrementa notablemente las oportunidades de codificación; además, también se muestra en este escenario que al incrementar el tamaño máximo de las filas de codificación se reducen los eventos de saturación y por lo tanto la transmisión de paquetes nativos. No obstante estos resultados positivos, existe un factor negativo que podría afectar significativamente el desempeño de las aplicaciones, este es el **retardo de codificación**.

Asuma que el proceso de codificación emplea el i -ésimo paquete de ambas filas de codificación para formar el paquete $P_\alpha = P1_{Q1} \oplus P1_{Q2}$. Si el paquete $P1$ se recibe y se almacena en su respectiva fila de codificación, y, además, la otra fila está vacía, entonces dicho paquete debe esperar el arribo del paquete $P1$ en la otra fila de codificación para que ambos puedan ser codificados. El tiempo que $P1$ espera en la fila de codificación para poder ser codificado es denominado como el retardo de codificación y denotado como $\psi_{1,1}$. La Figura 6.19 muestra los primeros instantes de simulación en los que arriban paquetes nativos a las filas de codificación $Q1$ y $Q2$ —estos datos fueron obtenidos configurando únicamente un flujo TCP en el escenario de simulación.

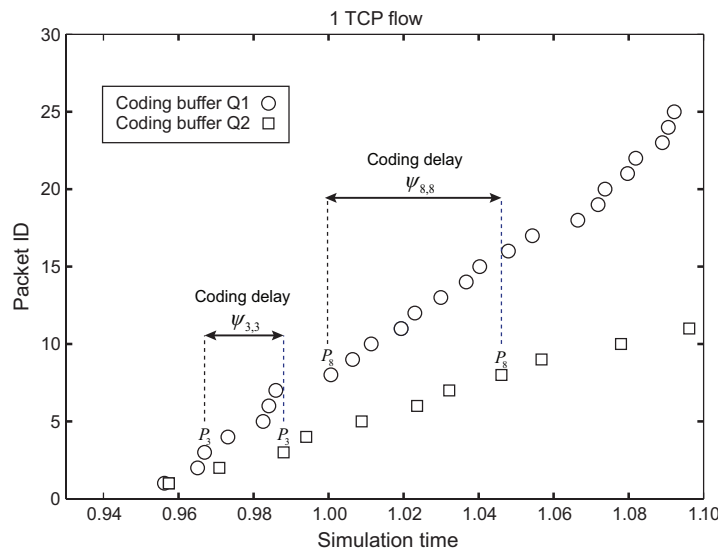


Figura 6.19: Instantes de arribos a las filas de codificación $Q1$ y $Q2$ con un flujo TCP en el escenario de simulación.

Puede observarse en la figura que las tasas de arribo para ambas filas tienen una tendencia lineal creciente cuyas pendientes difieren notoriamente. Este comportamiento se debe a que, en este escenario, el buffer $Q1$ almacena los paquetes de datos de TCP, mientras que el buffer $Q2$ almacena sus respectivos ACKs. Si se considera que el paquete codificado P_α se construye utilizando el paquete de datos P_i del buffer $Q1$ y el paquete P_i del buffer $Q2$ (es decir, $P_{iQ1} \oplus P_{iQ2}$) y es cuando se produce un retardo de codificación $\psi_{i,i}$. Esto indica que los paquetes almacenados en la fila con la mayor tasa de arribos deberían esperar por los arribos en la otra fila de codificación. En la figura se señala como el paquete P_{3Q1} experimenta un retardo de codificación $\psi_{3,3}$, puesto que este paquete debe esperar en la fila de codificación $Q1$ hasta que el paquete P_{3Q2} arribe y se almacene en su respectiva fila de codificación $Q2$, puesto que solo si ambos paquetes han sido almacenados en las filas de codificación, solo entonces pueden ser codificados. En la misma figura se muestra como el paquete P_{8Q1} experimenta un retardo de codificación $\psi_{8,8} > \psi_{3,3}$; se concluye que $\psi_{i,i} > \psi_{i-1,i-1}$. En la Figura 6.19 se aprecia como $\psi_{i,i}$ se ve incrementado a medida que transcurre el tiempo de simulación.

En la Figura 6.20 se grafican algunos instantes de arribo a las filas de codificación $Q1$ y $Q2$ en un escenario de simulación con 32 flujos TCP. A medida que se incrementa el número de STAs en el escenario de simulación encontramos dos resultados importantes: *i*) el retardo $\psi_{i,i}$ se incrementa a una tasa menor y *ii*) el número de oportunidades de codificación aumenta. Esto se debe invariablemente a que el MAP recibe un mayor número

de paquetes viajando en direcciones opuestas, incrementando de este modo la tasa de arribos en las filas y, por tanto, las oportunidades de codificación.

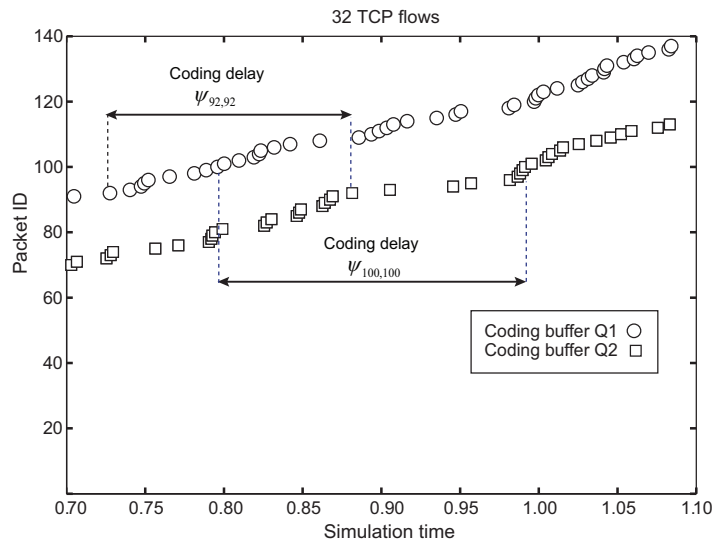


Figura 6.20: Instantes de arribos a las filas de codificación Q1 y Q2 con 32 flujos TCP en el escenario de simulación.

Para comprender los efectos que induce la densidad de tráfico en la red sobre el retardo de codificación, la Figura 6.21 muestra la evolución de las primeras 300 muestras del retardo $\psi_{i,i}$ conforme se incrementa el número de STAs en la red, mostrando que el retardo de codificación se reduce conforme aumenta la densidad de tráfico. Destaca que $\psi_{i,i}$ disminuye cuando el número de STAs incrementa entre 1 y 32. Es de subrayar que $\psi_{i,i}$ presenta un crecimiento lineal que puede alcanzar —en extremo— un valor superior a decenas de segundos. A medida que incrementa el número de STAs en la red, el retardo de codificación disminuye y puede alcanzar —en el peor de los casos— un valor cercano a unos cuantos segundos, demostrando como es que la densidad de tráfico puede provocar una reducción significativa de $\psi_{i,i}$. No obstante que $\psi_{i,i}$ decrece al aumentar el número de STAs en la red, aún conserva una tendencia ascendente a medida que transcurre el tiempo de simulación, lo cual puede afectar gravemente el desempeño de TCP, ya que este retardo distorsiona la estimación del ancho de banda que realiza este protocolo. Además, aún en aquellos escenarios donde la densidad de tráfico es alta, el retardo de codificación es superior a varias centenas de milisegundos.

En la Figura 6.22 se muestran las estadísticas del retardo de codificación como función de la densidad de tráfico en la red. Se aprecia que la mediana y el IRQ disminuyen si incrementa el número de STAs —esta reducción se acentúa cuando la cantidad de STAs aumenta desde 4 hasta 32—. No obstante, el retardo de codificación aún podría afectar significativamente al desempeño de diversos protocolos y, particularmente, al de TCP, porque a pesar de su reducción, el retardo de codificación puede llegar a alcanzar valores superiores a varios segundos, ya que oscila con una tendencia lineal creciente durante el tiempo de simulación.

Con base en los resultados obtenidos, se concluye lo siguiente. Por un lado, el acceso aleatorio, regulado por DCF, ofrece un acceso equitativo al medio entre la Mesh STA y las STAs; propiciando que las tasas de arribos a ambas filas de codificación sean similares y, en consecuencia, se incrementan las oportunidades de codificación —tal como se mostró en la Figura 6.14—. Por otro lado, el incremento de STAs en la red favorece la generación simétrica de tráfico en direcciones opuestas, es decir, STAs→MP y MP→STAs, induciendo así un acceso equiprobable al medio y, por tanto, la secuencia de transmisiones en la red favorece el proceso de codificación. Sin embargo, y a pesar del notable incremento de oportunidades de codificación, el retardo de codificación aumenta significativamente en todos los casos evaluados; lo cual puede afectar seriamente el desempeño de aquellas aplicaciones sensibles al retardo, y también a las que utilizan el protocolo de transporte TCP. Es evidente que limitar el tiempo de espera de los paquetes almacenados en las filas de codificación puede reducir el número de oportunidades de codificación pero, simultáneamente, puede disminuir y limitar el retardo de codificación, mitigando así el efecto negativo de este factor sobre el desempeño de TCP. El tiempo

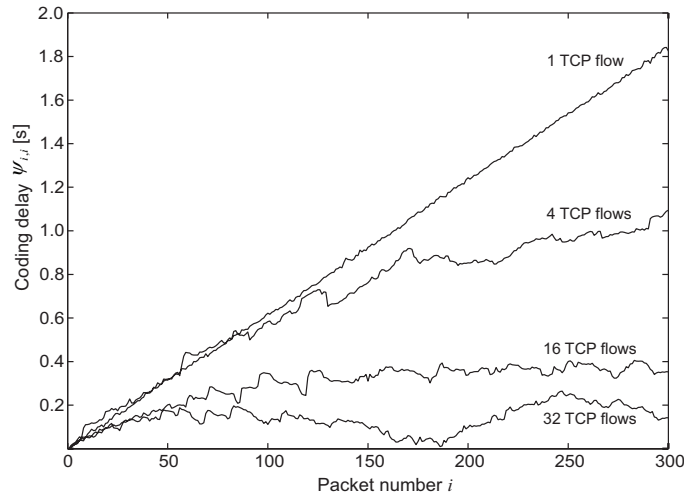


Figura 6.21: Muestras iniciales del retardo de codificación $\psi_{i,i}$ como función del número de STAs.

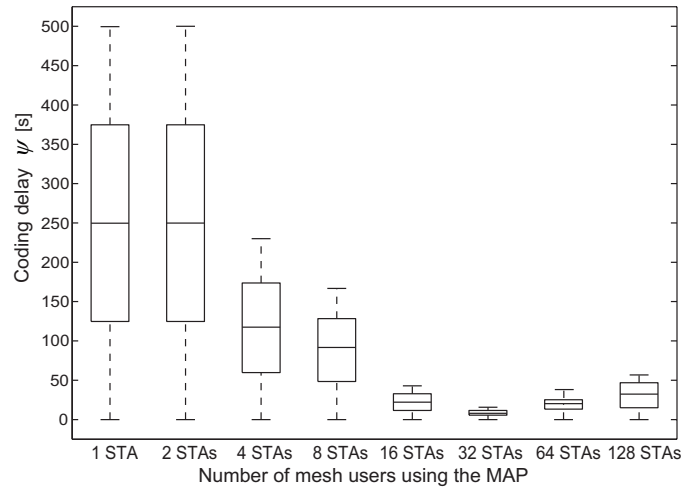


Figura 6.22: Muestras estadísticas del retardo de codificación $\psi_{i,i}$ como función del número de STAs.

máximo que puede esperarse de un paquete en las filas de codificación puede estimarse dinámicamente para ajustarse a las condiciones de tráfico en la red.

6.5.6 Ajuste del retardo de codificación

El retardo de codificación puede limitarse dinámicamente para satisfacer dos objetivos fundamentales: *i*) reducir el tiempo de espera de los paquetes nativos en las filas de codificación y *ii*) incrementar el *throughput* de la red al transmitirse paquetes codificados y nativos. Como se analizó en la Subsección 6.5.3, el número de oportunidades de codificación incrementa cuando la densidad de tráfico en la red también lo hace; no obstante, el retardo de codificación, $\psi_{i,i}$ aumenta significativamente, y este efecto podría afectar el desempeño de aplicaciones sensibles al retardo. Para resolver este inconveniente se pueden emplear diferentes estrategias. Por un lado, el protocolo de acceso al medio podría dejar de ser aleatorio y orientarse a atender eficientemente las necesidades de NC, es decir, priorizar el acceso al medio para el nodo codificador; sin embargo, esta es una modificación difícil de implementar, pues se afectaría seriamente la compatibilidad de esta solución con el estándar IEEE 802.11. Por otro lado, se puede definir un tiempo de espera máximo en el buffer de codificación para los paquetes nativos, evitando así que el retardo $\psi_{i,i}$ incremente (Véase la Sección 6.3); esta estrategia

es factible y, además, no afecta la compatibilidad con el estándar. Al establecer un tiempo de espera máximo para los paquetes que esperan las filas de codificación, se debe aceptar que el proceso de codificación no podrá realizarse con todos los paquetes nativos, esto significa que el MAP podrá transmitir paquetes tanto codificados como nativos.

La Figura 6.23 ejemplifica cómo incrementa el retardo de codificación $\psi_{i,i}$. En la Figura se ilustran los instantes de recepción en el MAP de los paquetes P_1, P_2, P_3 y P_4 , transmitidos por las STAs, y los paquetes P_1, P_2 y P_3 , transmitidos por la Mesh STA. Si se emplean los paquetes i -ésimos para realizar el proceso de codificación, es decir, $P_i \oplus P_i$, entonces el retardo de codificación $\psi_{i,i}$ incrementa significativamente —tal cómo se observó en la subsección 6.5.5—. En la Figura 6.23 se ilustra cómo el paquete P_1 —transmitido por las STAs y recibido por el MAP— debe esperar el arribo del paquete P_1 —transmitido por la Mesh STA— para poder calcular y formar el paquete codificado $P_1 \oplus P_1$, esto conlleva un retardo de codificación $\psi_{1,1}$. Dado que en este ejemplo la tasa de transmisión de la Mesh STA y las STAs es diferente, entonces el retardo de codificación $\psi_{i,i}$ incrementa paquete tras paquete, es decir, $\psi_{1,1} < \psi_{2,2} < \psi_{3,3}$. Nótese, además, que el número de transmisiones que realiza el MAP son únicamente tres en esta ventana de observación; es decir, los paquetes codificados $\psi_{1,1}$, $\psi_{2,2}$ y $\psi_{3,3}$. Estos tres paquetes codificados transportan conjuntamente la información de seis paquetes nativos; de tal modo que el paquete nativo P_4 proveniente de las STAs no se alcanza a transmitir en esta ventana de observación, ya que debe esperar el arribo del paquete P_4 proveniente de la Mesh STA para poder formar el paquete codificado $P_4 \oplus P_4$.

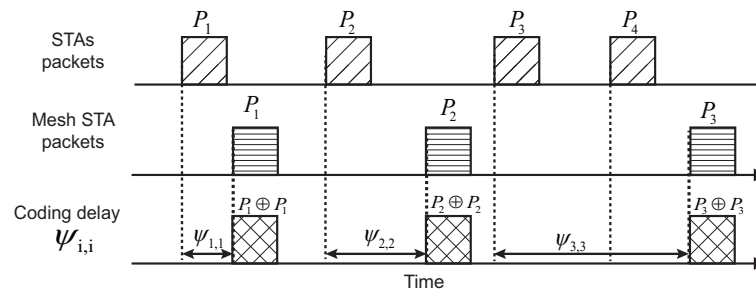


Figura 6.23: Incremento del retardo de codificación $\psi_{i,i}$.

Se puede establecer un tiempo de espera máximo para los paquetes nativos que se almacenan en las filas de codificación con el objetivo de evitar el crecimiento del retardo de codificación $\psi_{i,i}$. Si ese tiempo de espera expira, entonces el paquete nativo se debe transmitir sin ser codificado. Para implementar este método se requiere que cada paquete nativo, al ser recibido por el MAP, se etiquete con un tiempo de espera máximo, definido como $\text{MAX}_{\text{codingtime}}$. Si este tiempo expira y no se forma una oportunidad de codificación, entonces el paquete nativo debe enviarse inmediatamente a la capa MAC para su transmisión en forma nativa. En caso de detectarse una oportunidad de codificación antes de expirar $\text{MAX}_{\text{codingtime}}$, el paquete en espera será utilizado para calcular y formar un paquete codificado, y este paquete pasará inmediatamente a la capa MAC para su transmisión. Es importante señalar que al utilizar el tiempo límite $\text{MAX}_{\text{codingtime}}$, algunos paquetes en las filas de codificación no encontrarán una oportunidad de codificación y se transmitirán en su forma nativa; sin embargo, con esta técnica se reduce el retardo de codificación ψ y se potencian los beneficios de NC.

En la Figura 6.24 se ilustra el proceso propuesto. El paquete P_1 , transmitido por las STAs, se recibe en el MAP y debe esperar en las filas de codificación por el arribo del paquete P_1 proveniente de la Mesh STA. El tiempo de espera para P_1 es menor que $\text{MAX}_{\text{codingtime}}$ y, por lo tanto, se puede realizar la codificación $P_1 \oplus P_1$; lo mismo sucede con el tiempo de espera para el paquete P_2 . Por el contrario, el paquete P_3 , proveniente de las STAs, tiene un retardo de codificación $\psi_{i,i} > \text{MAX}_{\text{codingtime}}$; es así que este paquete debe ser transmitido en su forma nativa y sin codificar. Como consecuencia, el proceso de codificación ya no se ejecuta entre los i -ésimos paquetes de las filas de codificación, ya que, como se muestra en la figura, el paquete P_4 proveniente de las STAs se codifica con el paquete P_3 transmitido por la Mesh STA. De esta forma se evita que el retardo ψ crezca linealmente y, simultáneamente, se incrementa el número de transmisiones que podría realizar el MAP; siendo en este caso cuatro transmisiones durante la ventana de observación; es decir, tres transmisiones de paquetes codificados y la transmisión de un paquete nativo. Es claro que el MAP ahora puede transmitir

una mayor cantidad de información, es decir, siete paquetes (seis codificados y uno nativo) durante la misma ventana de observación.

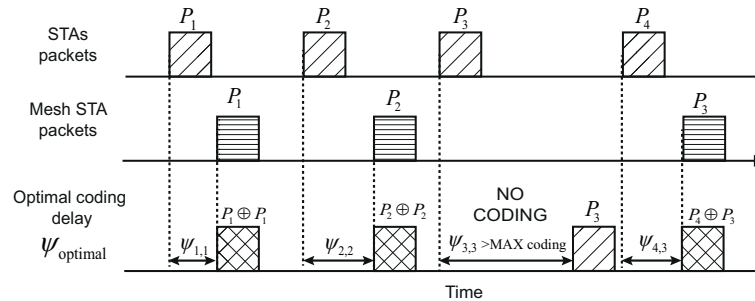


Figura 6.24: Retardo de codificación óptimo ψ_{optimal} .

6.5.7 Retardo de codificación óptimo

El **retardo de codificación óptimo**, ψ_{optimal} , se define como el tiempo mínimo que un paquete nativo debe esperar en las filas de codificación para obtener una oportunidad de ser codificado. En la Figura 6.25 se muestra cómo evoluciona este retardo óptimo en un escenario de simulación con una conexión TCP. Se aprecia en la figura que el retardo ψ_{optimal} es la **distancia temporal mínima** que existe entre los paquetes nativos almacenados en Q_1 y Q_2 ; esta consideración propicia que algunos paquetes no sean codificados y, por consiguiente, son transmitidos en su forma nativa.

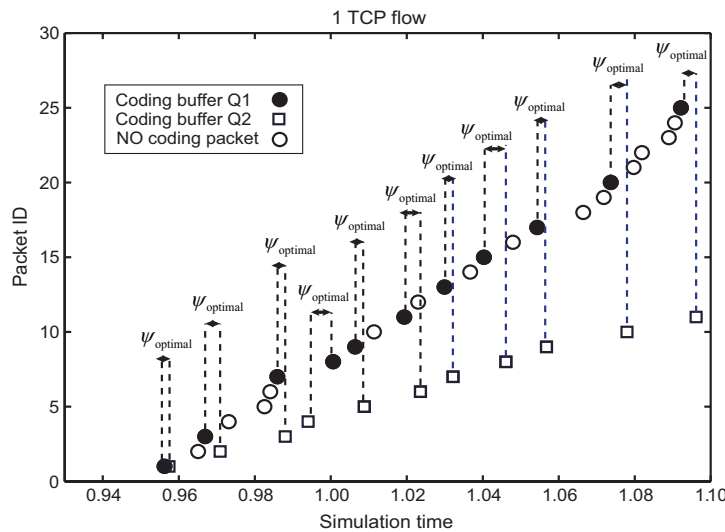


Figura 6.25: Retardo de codificación óptimo ψ_{optimal} en un escenario de simulación con una STA.

A pesar de la reducción del número de codificaciones y de la transmisión de una mayor cantidad de paquetes en forma nativa, se puede apreciar que se obtienen un par de importantes beneficios al establecer el retardo de codificación óptimo. Por un lado, se reduce el retardo de codificación y al mismo tiempo se limita su crecimiento; y, por otro lado, se incrementa la capacidad de la red al aumentar la cantidad de información que se transmite en el medio. En el ejemplo mostrado en la Figura 6.25 se nota que, si solo se transmitieran paquetes codificados, la cantidad de paquetes que se pueden transmitir serían 11. Por el contrario, si se utiliza el retardo ψ_{optimal} , se transmitirían 11 paquetes codificados y 25 paquetes en su forma nativa, incrementando de este modo el *throughput* de la red hasta en un 120%. Por otro lado, al incrementar el número de STAs

en la red —desde 1 hasta 128—, el retardo ψ_{optimal} mantiene un comportamiento similar en todos los casos analizados. En la Figura 6.26 se grafican las muestras estadísticas de esta variable. Se puede apreciar que la mediana de las muestras obtenidas en todos los casos es menor a 5 milisegundos y las muestras atípicas son solo superiores a 50 ms. A medida que incrementa el número de STAs en el escenario de simulación, el IRQ del retardo ψ_{optimal} muestra un incremento mínimo y cercano al 8%.

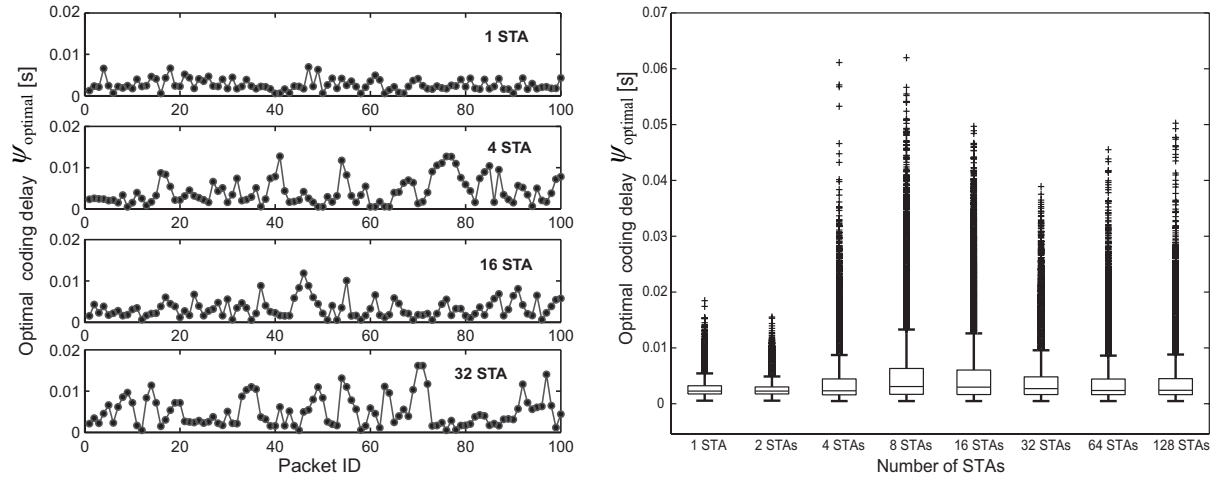


Figura 6.26: Retardo de codificación óptimo ψ_{optimal} como función del número de STAs.

6.5.8 Ganancia de codificación

La *ganancia de codificación* es una métrica definida como la **razón entre el número de paquetes codificados y el tamaño máximo de las filas de codificación por unidad de tiempo**. Esta medida de desempeño toma valores delimitados entre $[0, 1]$. Cuando la ganancia de codificación vale 0, siendo este el peor de los casos, significa que ningún paquete en las filas de codificación pudo ser codificado por unidad de tiempo; por otro lado, si la ganancia de codificación vale 1, entonces significa que todos los paquetes en la fila de codificación fueron codificados por unidad de tiempo. Si se **asume que la red opera con un mecanismo de estimación ideal que ayude a calcular el retardo óptimo ψ_{optimal}** , entonces la ganancia de codificación de la red se comportaría tal como se muestra en la Figura 6.27. Se aprecia en la figura que la ganancia de codificación depende del número de STAs en el sistema. Cuando el número de STAs incrementa y supera a 4 usuarios, se advierte que la ganancia de codificación resultante es mayor a 0.7, e incluso cuando el número de STAs es mayor a 16 usuarios, la ganancia de codificación es incluso mayor a 0.9. Este resultado muestra la importancia que tiene la estimación del retardo ψ_{optimal} , ya que con una estimación precisa se pueden incrementar el número de codificaciones y, por lo tanto, la capacidad de la red.

Es evidente que uno de los mayores retos para la técnica de codificación de red es el cálculo del valor $MAX_{\text{codingtime}}$ (tiempo máximo que cada paquete de datos puede permanecer en las filas de codificación en espera de una oportunidad para ser codificado) [10, 107]. Una solución consiste en **asignar un valor estático a $MAX_{\text{codingtime}}$** . Sin embargo, con esta solución pueden emerger dos problemas. **Primero**, si $\psi_{\text{optimal}} \gg MAX_{\text{codingtime}}$, entonces se reduce el número de oportunidades de codificación y el tiempo de espera para los paquetes nativos almacenados en las filas de codificación. **Segundo**, si $\psi_{\text{optimal}} \ll MAX_{\text{codingtime}}$, entonces se pueden incrementar las oportunidades de codificación pero se incrementa significativamente el retardo de codificación, lo cual afectaría seriamente el desempeño de TCP y de aplicaciones sensibles al retardo.

Una solución que ayudaría a resolver los problemas mencionados consiste en estimar $MAX_{\text{codingtime}} = \psi_{\text{optimal}}$; no obstante, ψ_{optimal} es un parámetro dinámico que varía con el tiempo, tal como se mostró en los resultados presentados en las figuras 6.25 y 6.26, lo cual dificulta su estimación. Para ajustar dinámicamente el valor $MAX_{\text{codingtime}}$ se pueden utilizar **métodos de estimación de promedio móvil (Moving average Estimation**

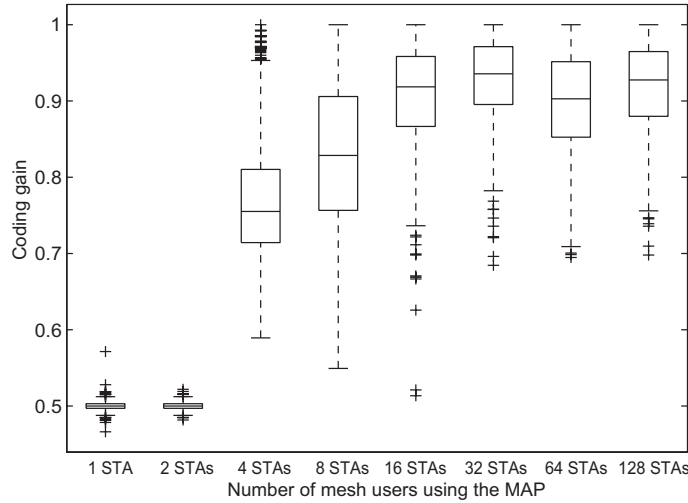


Figura 6.27: Ganancia de codificación por segundo empleando el retardo de codificación óptimo ψ_{optimal} .

Methods). Estos métodos de estimación/predicción son sencillos y demandan capacidades reducidas de procesamiento y almacenamiento. Debido a estas características, en esta subsección utilizamos y evaluamos los siguientes métodos para **estimar dinámicamente el valor $MAX_{\text{codingtime}}$** : **Simple Weighted Moving Average (SWMA)**, **Exponential Weighted Moving Average (EWMA)**, **Modified Weighted Moving Average (MWMA)**, **Gaussian Weighted Moving Average (GWMA)** y **Triangular Weighted Moving Average (TWMA)**.

6.6 Estimación adaptativa del tiempo de espera

Muchas de las técnicas que son empleadas para estimar la tendencia de una serie de tiempo están basadas en el cálculo de promedios móviles. Estas técnicas de estimación utilizan un subconjunto de n elementos de la serie de tiempo x_t para tratar de pronosticar el elemento futuro x_{t+1} [146]. Una de las mayores ventajas de estas técnicas es que tienen bajos requerimientos de procesamientos y almacenamiento, lo cual les permite ser empleadas en diversos escenarios. A continuación se detalla el funcionamiento de estas técnicas de estimación.

6.6.1 Estimación por promedios móviles

Considere una serie de datos obtenidos en distintos instantes de tiempo; dicha serie se definirá como la secuencia de muestras $x = [x_t, x_{t-1}, \dots, x_1, x_0]$. El subíndice de cada muestra es un indica el instante de tiempo en el que fue obtenida. Esto quiere decir que la muestra más reciente es x_t , mientras que la muestra más antigua es x_0 . Para tratar de estimar el valor de la muestra futura, x_{t+1} , el método de promedios móvil asume que x_{t+1} responde a una combinación lineal de n muestras previas. Es así que cada una de las n muestras previas, es decir, $x = [x_t, x_{t-1}, \dots, x_{t-n+1}]$, debe ser multiplicada por un vector de pesos $\alpha = [\alpha_t, \alpha_{t-1}, \dots, \alpha_{t-n+1}]$ La aproximación resultante, y_{t+1} , se obtiene al sumar los n elementos ponderados de la serie de tiempo x . Por lo tanto, y_{t+1} se calcula como:

$$y_{t+1} = \sum_{j=0}^{n-1} \alpha_{t-j} x_{t-j} = \alpha_t x_t + \alpha_{t-1} x_{t-1} + \dots + \alpha_{t-n} x_{t-n} \quad (6.4)$$

La estimación y_{t+1} es una aproximación del valor futuro x_{t+1} , por lo tanto, existe un error de predicción, definido como $e_{t+1} = |y_{t+1} - x_{t+1}|$. Cada que se realiza una nueva estimación, la muestra más antigua de x se descarta. El procedimiento de estimación se muestra en la Figura 6.28. En el ejemplo, el número de muestras consideradas para realizar la estimación es igual a 10; se puede apreciar que la primera estimación, y_{11} , requiere que la serie de tiempo x_t tenga las 10 primeras muestras $[x_1, x_2, \dots, x_{10}]$ para poder calcular y_{11} . En

consecuencia, la estimación y_{12} requiere de los 10 elementos más actuales de la serie, es decir, $[x_2, x_3, \dots, x_{11}]$, eliminando así el elemento más antiguo de la serie, es decir, x_1 . Se debe tener en consideración que cada estimación tiene asociado un error de predicción, en la figura se ilustra el error $e_{11} = |y_{11} - x_{11}|$.

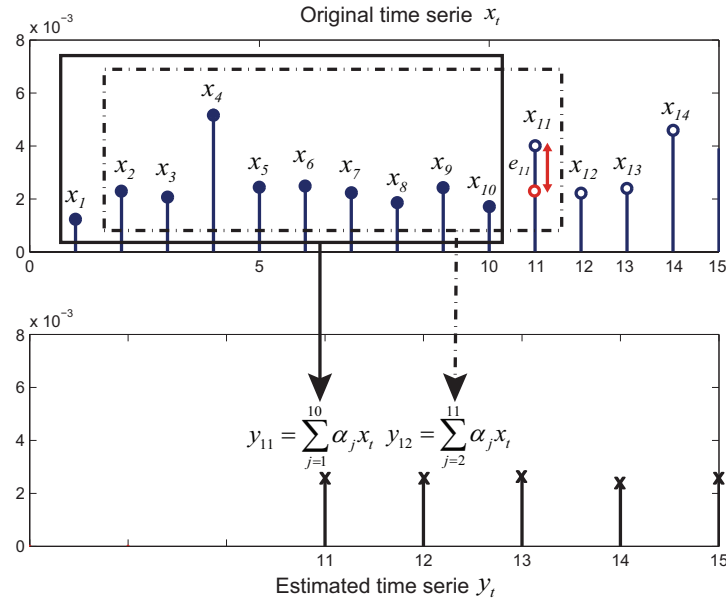


Figura 6.28: Método de estimación por promedio móvil.

Al modificar el vector de pesos α y el número de muestras n utilizadas en el proceso de estimación, se generan diferentes métodos de estimación de promedios móviles. A continuación detallamos los métodos seleccionados para calcular dinámicamente el tiempo máximo de estancia de los paquetes nativos en las filas de codificación $MAX_{\text{codingtime}}(t)$:

- **Simple Weighted Moving Average (SWMA):** Este método de estimación asigna a cada una de las n muestras un peso α_i constante, definido como $\alpha_i = \frac{1}{n}$.
- **Exponential Weighted Moving Average (EWMA):** Este método asigna un peso más grande a las muestras más recientes de la serie x para incrementar así su aporte en el cálculo del promedio y_{t+1} y un menor peso a las muestras más antiguas para reducir su influencia en el promedio. El vector de pesos α decrementa exponencialmente hacia los elementos más antiguos de la serie x . Una característica importante de este método de estimación es que los elementos más antiguos de la serie nunca dejan de ser considerados para la estimación y_{t+1} ; sin embargo, el peso asociado a estas muestras disminuye en cada estimación. La estimación y_{t+1} puede expresarse de manera simplificada como:

$$y_{t+1} = \alpha x_t + (\alpha - 1)y_t \quad (6.5)$$

- **Modified Weighted Moving Average (MWMA):** Este método asigna los pesos α tomando en consideración la edad cronológica de las muestras. Las muestras recientes de x_t reciben un mayor peso, y este peso decrementa a medida que las muestras son más antiguas. Puede verse como una variante del método exponencial con $\alpha = 1/n$.
- **Gaussian Weighted Moving Average (GWMA)** Este método de estimación asigna una distribución gaussiana a la serie de pesos α . De tal forma que se da un mayor peso a los elementos centrales de las n muestras de x , y los elementos más antiguos y más recientes tienen un menor peso α para la estimación.
- **Triangular Weighted Moving Average (TWMA)** Este método es equivalente a un promedio simple realizado en dos ocasiones. Por un lado, el método triangular calcula un promedio móvil simple sobre

las n muestras para obtener una muestra s_t . Por otro lado, el promedio resultante s_t es promediado utilizando de nueva cuenta un promedio móvil simple sobre las muestras pasadas de s_t . Es decir:

$$s_t = \sum_{j=t-n}^t \frac{1}{n} x_j; y_{t+1} = \sum_{j=t-n}^n \frac{1}{n} s_j. \quad (6.6)$$

6.6.2 Algoritmo de estimación adaptativa

La función principal del MAP dentro de una iWMN es recibir paquetes nativos y reexpedirlos hacia dos posibles dispositivos, ya sea hacia una STA o bien hacia un MP. Para explotar los beneficios que ofrece la técnica de codificación de red, se el MAP debe ser capaz de estimar dinámicamente el tiempo de espera máximo que los paquetes nativos pueden ser almacenados en las filas de codificación. Esta estrategia tiene por objetivos incrementar el número de codificaciones y al mismo tiempo reducir el retardo asociado a la formación de oportunidades de codificación. Para realizar la estimación dinámica del tiempo de espera, el MAP puede utilizar los métodos basados en promedios móviles. Para que el MAP pueda implementar cualquiera de los métodos de estimación descritos en la subsección anterior, es necesario que ejecute las siguientes etapas del algoritmo de estimación adaptativa.

Inicialización del vector de muestras

Inicialmente, el MAP debe inspeccionar cada paquete nativo que recibe para poder determinar el dispositivo hacia donde debe ser reexpedido (esta información es almacenada en el campo `packet . next` de cada paquete nativo en el simulador `ns-2`). Los paquetes nativos que inicialmente se reciben no son almacenados en las filas de codificación, por el contrario, son reexpedidos en su forma nativo y solo son almacenados los instantes de tiempo en el que arribaron al MAP. Los arribos al MAP son marcados en el simulador de eventos discretos en el campo `packet . time` de cada paquete nativo. Con esta información, el MAP almacena los instantes de arribo de los paquetes nativos con destino a las STAs en el arreglo S_1 y los instantes de arribo de los paquetes con destino a la Mesh STA son almacenados en el arreglo S_2 . EL MAP almacenará los instantes de arribo en S_1 y S_2 hasta obtener n muestras en ambos arreglos. La Figura 6.29 muestra el algoritmo de esta primera etapa. En la figura se muestra como cuando el MAP recibe paquetes nativos destinados a las STAs, los instantes de tiempo $t_{i,j}$ de estos paquetes se almacenan en el arreglo S_1 y, por otro lado, los instantes de arribo paquetes destinados al Mesh STA $t_{2,i}$ son almacenados en el arreglo S_2 . Debido a que el tráfico es variable e impredecible, es altamente probable que el número de muestras almacenadas en S_1 y en S_2 sean diferentes; esto significa que un arreglo podría tener más de n muestras. Es importante señalar que los paquetes con dirección STA \rightarrow Mesh STA pueden requerir de un retardo de codificación óptimo diferente al que requieren los paquetes que van en dirección Mesh STA \rightarrow STA, consecuentemente, el proceso de estimación del retardo de codificación óptimo se realiza para ambas filas de codificación.

Calculo de las muestras iniciales del retardo óptimo de codificación

Cuando ambos arreglos S_1 y S_2 han almacenado al menos n instantes de arribo, entonces el MAP puede calcular con estos datos las primeras n muestras del retardo óptimo de codificación. Para lograrlo el MAP deberá definir las muestras del retardo óptimo calculando la distancia temporal mínima entre el tiempo de arribo del paquete i de arreglo S_1 y cualquier tiempo de arribo del arreglo S_2 . El MAP debe tomar cada una de las n muestras almacenadas en S_1 y estimar la distancia temporal con todas las muestras almacenadas en S_2 , si el resultado del cálculo $S_1[k1] - S_2[k2]$ es negativo, entonces esto significa que la muestra $S_1[k1]$ debería esperar por la muestra $S_2[k2]$, por lo tanto, la diferencia $S_1[k1] - S_2[k2]$ es almacenada en el vector ψ ; por el contrario, si $S_1[k1] - S_2[k2]$ es positivo, esto significa que el tiempo de arribo $S_1[k1]$ es mayor que el tiempo de arribo $S_2[k2]$ y por lo tanto la diferencia $S_1[k1] - S_2[k2]$ se descarta. Las diferencias $S_1[k1] - S_2[k2]$, calculadas sobre todos los elementos de S_1 y S_2 , son almacenadas en el vector auxiliar denominado como `distance`, sobre este vector se calcula el valor mínimo y el valor resultante es un elemento del vector ψ . De esta manera se obtienen la secuencia de n muestras del retardo óptimo de codificación, es decir, $\psi = [\psi_1, \psi_2, \dots, \psi_{n-1}, \psi_n]$. Debido

Algorithm: Formation of the initial sampling vector

Description: Each time that a packet arrives to the MAP, the arrival time of the packet is stored in array S1 or S2 considering the next hop of the packet; finally, the packet is forwarded in its native form.

```

1: procedure Samplig (packet)
2: begin
3:   while i < n or j < n do
4:     if packet.next = STA then
5:       S1[i] ← packet.time
6:       i ← i+1
7:     else-if packet.next= mesh STA then
8:       S2[j] ← packet.time
9:       j ← j+1
10:    end if
11:    forward (packet)
12:  end while
13: end procedure

```

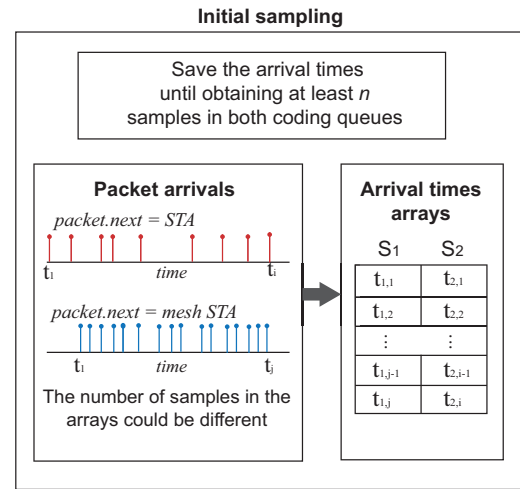


Figura 6.29: Construcción de los arreglos iniciales S_1 y S_2 con muestras de los instantes de arribo de los paquetes nativos al MAP.

a que existen paquetes transmitiéndose en sentidos opuestos, ambos requieren tiempos de espera distintos para poder obtener una oportunidad de codificación. En la Figura 6.30 se detalla el algoritmo necesario para construir el vector ψ para los paquetes que van de la Mesh STA a las STAs, un mecanismo similar debe aplicarse para los paquetes que van en dirección opuesta.

Algorithm: Computation of the initial samples of the optimal coding delay

Description: The MAP uses the arrays S1 and S2 and their lengths, i and j , to estimate the minimal temporal distance between the arrival times. Clearly, this estimation procedure should be performed twice, for packets from the mesh STA to the STAs and for packets from STAs to mesh STA.

```

1: procedure EstimationMeshtoSTA (S1, S2, i, j)
2: begin
3:    $k_1, k_2, k_3 \leftarrow 0$ 
4:   while  $k_1 < i+1$  do
5:     while  $k_2 < j+1$  do
6:        $diff \leftarrow S1[k_1] - S2[k_2]$ 
7:       if  $diff < 0$  then
8:          $distance[k_3] \leftarrow |diff|$ 
9:          $k_3 \leftarrow k_3 + 1$ 
10:      end if
11:       $k_2 \leftarrow k_2 + 1$ 
12:    end while
13:     $\psi[k_1] \leftarrow \min[distance]$ 
14:     $k_1 \leftarrow k_1 + 1$ 
15:  end while
16: end procedure

```

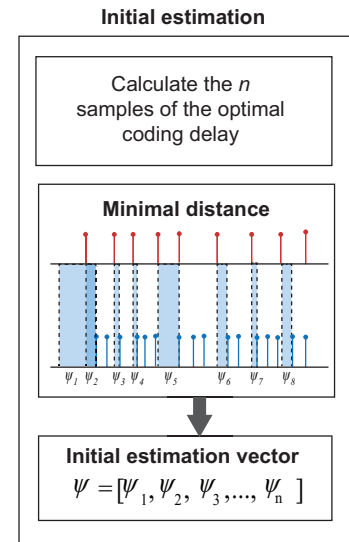


Figura 6.30: Cálculo del vector ψ donde se almacenan las muestras del retardo de codificación óptimo.

Estimación dinámica del retardo óptimo de codificación

Habiendo obtenido las n muestras del retardo óptimo de codificación ψ , el MAP puede realizar el proceso de estimación empleando cualquier método de estimación por promedios móviles para estimar el valor de ψ_{n+1} .

Con esta predicción se puede asignar a cada paquete recibido un tiempo máximo de espera en las filas de codificación. Para reducir el error de predicción e cada estimación ψ_{n+1} deberá incrementarse para aumentar la media del proceso de estimación resultante. De este modo se incrementa el tiempo máximo de estancia en las filas de codificación, definido como $MAX_{codingtime}$; en este caso proponemos utilizar $MAX_{codingtime} = 2\psi_{n+1}$. De esta forma se evita que el proceso de codificación descarte los paquetes cuyos retardos de codificación óptimos están muy por encima de la media de estimación ψ_{n+1} . Por cada nuevo par de paquetes recibidos, el MAP debe calcular una muestra nueva de ψ_{n+1} . Esta muestra es reutilizada por el proceso de estimación para calcular el nuevo valor de $MAX_{codingtime}$. El proceso propuesto se muestra en la Figura 6.31, el método de estimación utiliza estimaciones basadas en promedios móviles (*WMN - Weighted Moving Average*) para calcular cada muestra del retardo óptimo de codificación ψ y el tiempo límite de espera en las filas de codificación $MAX_{codingtime}$.

Algorithm: Forecast and sampling

Description: The MAP uses the array ψ to forecast the optimal coding delay for the next packets. Furthermore, the MAP always monitors the arrival times of packets flowing in both directions, STA \rightarrow mesh STA and mesh STA \rightarrow STA, to compute a new sample of the estimation vector and the estimation error.

```

1: procedure Forecast(  $\psi = [\psi_1, \psi_2, \psi_3, \dots, \psi_n]$  )
2: begin
3:    $\psi_{n+1} \leftarrow \text{WMA}([\psi_1, \psi_2, \psi_3, \dots, \psi_n])$ 
4:    $MAX_{coding} \leftarrow 2\psi_{n+1}$ 
5: end procedure

```

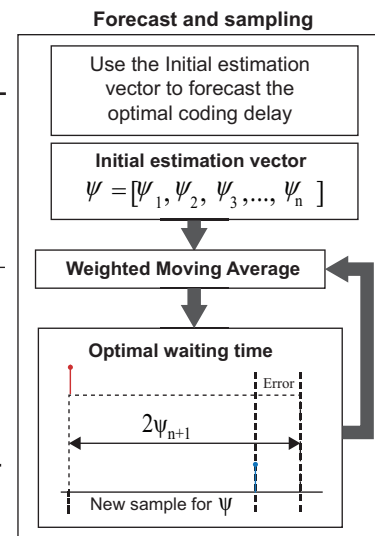


Figura 6.31: Estimación del retardo de codificación empleando mecanismos basados en promedios móviles (*WMA*).

6.6.3 Evaluación del proceso de estimación

Para evaluar el desempeño del método adaptativo de codificación de red se evalúan distintos métodos de estimación con promedio móviles para calcular el tiempo límite de espera en las filas de codificación, $MAX_{codingtime}$. Los métodos evaluados son los siguientes: **Simple Weighted Moving Average (SWMA)**, **Exponential Weighted Moving Average (EWMA)**, **Modified Weighted Moving Average (MWMA)**, **Gaussian Weighted Moving Average (GWMA)** y **Triangular Weighted Moving Average (TWMA)**. Para esta evaluación se comparan las trazas del retardo óptimo de codificación obtenidas en la Subsección 6.5.7 con los valores de estimación obtenidos con los métodos seleccionados.

Retardo óptimo de codificación

En la Figura 6.32 se compara el retardo óptimo de codificación y la estimación dinámica del $MAX_{codingtime}$. Se aprecia en la figura que las primeras 10 muestras del retardo óptimo de codificación son utilizadas para calcular el vector inicial $\psi = [\psi_1, \psi_2, \dots, \psi_9, \psi_{10}]$. Las siguientes 140 muestras son estimadas empleando diferentes métodos basados en el cálculo de promedios móviles. Es claro que empleando cualquiera de los métodos seleccionados, el cálculo del $MAX_{codingtime}$ muestra una tendencia muy similar al retardo óptimo de codificación. Se aprecia en la figura que la media de la estimación $MAX_{codingtime}$ es mayor que la media del retardo óptimo de codificación. Esta propiedad crea un margen de tolerancia para que la estimación de $MAX_{codingtime}$ pueda capturar algunos eventos en los que el retardo de codificación óptimo resulta superior a la media. Al incrementar el número de STAs en la red, se nota que la estimación de $MAX_{codingtime}$ tiene una mayor

variación, y en consecuencia, un mayor margen de error. Todos los métodos de estimación utilizados ofrecen un desempeño muy similar; la selección de cualquiera de estos métodos no marca diferencias importantes en el resultado de la estimación $MAX_{codingtime}$.

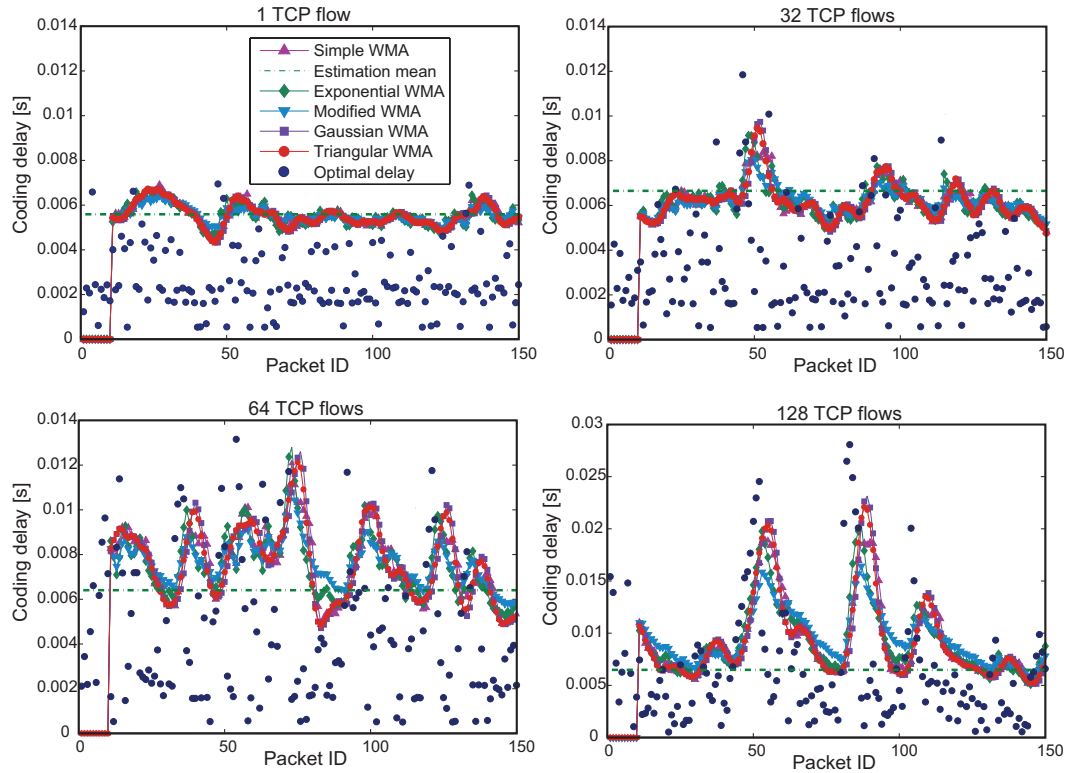


Figura 6.32: Comparación entre retardo óptimo de codificación y el valor dinámico de $MAX_{codingtime}$ obtenido mediante diferentes métodos de estimación con promedios móviles.

Además, la estimación propuesta, es decir, $MAX_{codingtime} = 2 \times \psi_{n+1}$, es útil para evitar la subestimación del retardo óptimo de codificación, ya que la estimación resultante $MAX_{codingtime}$ es ligeramente superior al retardo óptimo de codificación. Con esta propuesta se coadyuva a que los paquetes almacenados en las filas de codificación puedan formar, con alta probabilidad, una oportunidad de codificación. El uso de cualquiera de los métodos de estimación propuestos puede ayudar a reducir el retardo de codificación e incrementar la probabilidad de crear una oportunidad de codificación. Sin embargo, existe un compromiso entre el retardo de codificación y el número de codificaciones que se pueden obtener; esto es, un alto retardo de codificación puede incrementar el número de oportunidades de codificación. Sin embargo, un mayor retardo de codificación afecta seriamente a las aplicaciones sensibles al retardo, al desempeño de TCP y, en consecuencia, a la calidad de la experiencia.

Error de estimación

Cada uno de los métodos de estimación basados en promedios móviles tiene asociado un error de predicción e . En la Figura 6.33 se muestran las estadísticas del error de predicción, es decir, $e = MAX_{codingtime} - \psi_{optimal}$. Es notorio que a medida que incrementa el número de STAs en la red, el error de estimación tiene un mayor número de errores; es decir, cuando $e < 0$ significa que $MAX_{codingtime} < \psi_{optimal}$, lo cual significa que son oportunidades de codificación perdidas. Cuando el error de estimación es superior a 0 significa que el tiempo $MAX_{codingtime}$ para cada paquete en las filas de codificación es lo suficientemente grande como para generar una codificación. El caso ideal de estimación es cuando el valor de $MAX_{codingtime}$ es igual al retardo óptimo de codificación, es decir, $MAX_{codingtime} = \psi_{optimal}$. Todos los métodos de estimación analizados se comportan

de manera similar y presentan, en promedio, un número de errores de estimación y un número de sobre estimaciones muy parecido. Podemos concluir que la utilización de cualquiera de los métodos de estimación propuestos reduce el retardo de codificación e incrementa la probabilidad de codificación al elevar la media del proceso de estimación. Sin embargo siempre existe un compromiso entre el retardo de codificación y el número de codificaciones que pueden realizarse.

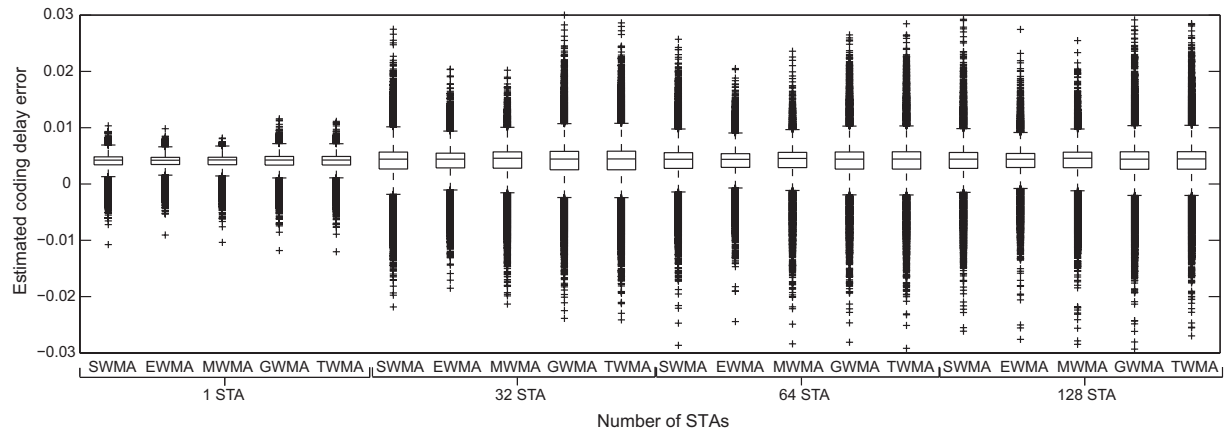


Figura 6.33: Error de estimación utilizando diferentes métodos de media móvil e incrementando el número de STAs.

Conclusiones y trabajo futuro

“La ciencia, muchacho, está hecha de errores, pero son errores que conviene cometer, porque nos conducen poco a poco a la verdad.”

Julio Verne

LAS REDES INALÁMBRICAS IEEE 802.11s (iWMNs) son una excelente alternativa para brindar acceso inalámbrico a Internet en regiones en donde el uso de otras tecnologías de comunicaciones resulta costoso e inviable, tales como en regiones rurales o comunitarias. No obstante el enorme potencial que tiene esta novedosa tecnología de red, existen evidencias que señalan diversos retos de desempeño que se deben atender para poder mejorar sus prestaciones y la calidad de la experiencia que ofrece para los usuarios. Los retos de desempeño en iWMNs son diversos y complejos, tales como la capacidad limitada de los enlaces inalámbricos, el gran número de usuarios que debe soportar la red, la contienda por el medio, la alta tasa de errores en el medio inalámbrico, así como la interacción de la red con protocolos de la pila TCP/IP.

La técnica denominada como **codificación de red** es utilizada comúnmente para atender diversos problemas de desempeño en redes inalámbricas. Esta técnica ha demostrado ser útil para incrementar la capacidad de redes inalámbricas, para reducir la contienda por el medio e incluso para reducir la probabilidad de errores en el medio inalámbrico. La técnica de codificación de red permite combinar algebraicamente paquetes de datos (denominados como paquetes nativos) y de este modo construir paquetes codificados, que, cuando son transmitidos, transportan la información de los paquetes de datos nativos que los conformaron. De este modo, la red puede transportar una mayor cantidad de información utilizando una menor cantidad de transmisiones inalámbricas; reduciendo así la contienda por el medio y la probabilidad de errores. No obstante los numerosos beneficios que ofrece la técnica de codificación de red, **en redes inalámbricas resulta difícil explotar las oportunidades de codificación, ya que las condiciones operativas de la red y el tráfico de datos varían dinámicamente**. Existen diferentes estrategias para atender esta dificultad, una de ellas consiste en incorporar tiempos de espera para los paquetes nativos y de este modo poder obtener una oportunidad de codificación. Sin embargo, definir este tiempo de espera es un reto ya que, si este tiempo de espera es muy grande, el retardo puede afectar a las aplicaciones y a la calidad de la experiencia; por otro lado, si el tiempo de espera es pequeño, se pueden perder numerosas oportunidades de codificación.

Inicialmente, en este trabajo de investigación se evaluaron los efectos que tienen algunas variantes de TCP en el desempeño de las iWMNs; encontrando que a medida que aumenta el número de usuarios y conexiones TCP, el *throughput* de la red se reduce hasta en un 50%; ya que se incrementa significativamente la contienda por el medio. También se demostró que el número de saltos en la trayectoria de comunicaciones tiene un impacto negativo en la capacidad de la red, reduciéndola hasta en un 50% a medida que incrementa el número de saltos entre los usuarios y el MP. Se encontró, además, que la tasa de errores en el medio inalámbrico puede afectar seriamente el desempeño de la red, reduciendo el *throughput* de la red hasta en un 80%. Mas aún, algunas variantes de TCP mostraron un bajo rendimiento en escenarios con competencia, ya que no pueden

distribuir equitativamente los recursos de la red entre los usuarios. Los resultados obtenidos en esta etapa demostraron que el desempeño de las iWMNs podría verse beneficiado al utilizar la técnica de codificación de red.

Para mejorar el desempeño de las iWMNs se propuso como objetivo de esta investigación adaptar y utilizar la técnica de codificación de red. La propuesta consistió en implementar la técnica de codificación en los puntos donde converge el tráfico de los usuarios y el tráfico proveniente de la infraestructura inalámbrica, es decir, en los Mesh Access Points (MAPs). Se propusieron un conjunto de modificaciones para implementar la técnica de codificación de red en un MAP. Se incorporaron filas de codificación en un MAP y se modeló su ocupación como una Cadena de Markov en Tiempo Discreto (DTMC - Discrete Time Markov Chain) con dos diferentes condiciones operativas: *i)* el MAP puede transmitir paquetes nativos y codificados, y *ii)* el MAP solo puede transmitir paquetes codificados. Se demostró con el modelo que al incrementar la probabilidad de transmisión del MAP, se incrementan las oportunidades de codificación y el mismo beneficio se obtiene al incrementar el tamaño de las filas de codificación.

Para complementar la investigación se encontró, empleando el simulador de red ns-2, que al incrementar el número de flujos TCP convergentes en un MAP (de 1 hasta 128), el número de oportunidades de codificación crece hasta en un 70%. No obstante, también se demostró que el proceso de codificación induce un retardo que puede afectar gravemente el desempeño de TCP. Para reducir este retardo de codificación se propuso asignar una estampa de tiempo dinámica que limite el tiempo máximo que un paquete nativo puede esperar en las filas de codificación. Para establecer el valor de la estampa de tiempo se propuso un esquema adaptativo de estimación que basa su funcionamiento en métodos de promedios móviles. Los promedios móviles evaluados para este fin fueron: SWMA, EWMA, MWMA, GWMA, y TWMA. Los resultados mostraron que los métodos de estimación son eficientes para asignar el tiempo máximo de espera gracias a que el error de predicción supera normalmente el retardo de codificación óptimo, lo cual se traduce en un mayor número de codificaciones en la red. Una implicación importante de la propuesta es que no todas las posibles codificaciones podrán ser realizadas, pues existen casos en los que la estimación del tiempo asignado a cada paquete en las filas de codificación resulta ser menor al retardo de codificación óptimo, lo que significa que estos paquetes son transmitidos en su forma nativa.

Para concluir este trabajo de investigación, se presentan algunas de las líneas de investigación que pueden ser consideradas como trabajo futuro:

1. La codificación de red puede ser implementada en todos los nodos que componen la trayectoria de datos entre el MAP y el MPP, lo cual podría mejorar significativamente el desempeño integral de la red. Evaluar esta implementación coadyuvaría a resaltar los posibles beneficios de la técnica de codificación de red en iWMNs.
2. Al incluir un tiempo de espera dinámico para los paquetes nativos se puede afectar el funcionamiento de cada flujo TCP en la red, evaluar estos efectos demostraría los efectos que puede tener esta aproximación en el desempeño del protocolo de transporte.
3. La estimación dinámica del tiempo de espera máximo para los paquetes nativos puede ser calculado empleando técnicas diferentes de estimación que pueden mejorar la predicción.
4. En esta investigación se empleó una heurística que establece el tiempo máximo de espera como el doble del valor estimado dinámicamente; sin embargo, este método de estimación podría mejorar el tiempo de espera si se considera la variabilidad y el error de la aproximación.

Bibliografía

- [1] R. Bruno, M. Conti, and E. Gregori, “Mesh Networks: Commodity Multihop Ad Hoc Networks,” *IEEE Communications Magazine*, vol. 43, pp. 123–131, Marzo 2005.
- [2] P. Avakul, H. Nishiyama, N. Kato, Y. Shimizu, and T. Kumagai, “Mesh router selection to maximize system throughput in dense Wireless Mesh Networks,” *IEEE 14th International Conference on High Performance Switching and Routing (HPSR’13)*, pp. 115–120, Julio 2013.
- [3] E. Carrillo and V. Ramos, “Performance Evaluation of Reactive and Proactive Routing Schemes for Infrastructure Wireless Mesh Networks,” *Eighth International Conference on Wireless and Optical Communications Networks, (WOCN’11)*, pp. 1–6, May 2011.
- [4] A. Ouni, H. Rivano, and F. Valois, “Capacity of Wireless Mesh Networks: Determining Elements and Insensible Properties,” *Wireless Communications and Networking Conference Workshops (WCNCW 2010)*, pp. 1–6, Abril 2010.
- [5] L. Barreto and S. Sargento, “TCP, XCP and RCP in Wireless Mesh Networks: An Evaluation Study,” *IEEE Symposium on Computers and Communications (ISCC’10)*, pp. 351–357, Junio 2010.
- [6] A. Islam and V. Raghunathan, “iTCP: an intelligent TCP with neural network based end-to-end congestion control for ad-hoc multi-hop wireless mesh networks,” *Springer Wireless Networks*, vol. 21, pp. 581–610, Septiembre 2015.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the Air: Practical Wireless Network Coding,” *IEEE/ACM Transactions on Networking*, vol. 16, pp. 497–510, June 2008.
- [8] J. Andrews, N. Jindal, M. Haenggi, R. Berry, M. Neely, S. Weber, S. Jafar, and A. Yener, “Rethinking Information Theory for Mobile ad Hoc Networks,” *IEEE Communications Magazine*, vol. 46, pp. 94–101, Diciembre 2008.
- [9] A. Jannesari, M. A. Sarram, and R. Sheikhpour, “A Novel Network Coding Algorithm to Improve TCP in Wireless Networks,” *Springer Wireless Personal Communications*, pp. 1–18, Septiembre 2019.
- [10] Y. Huang, M. Ghaderi, D. Towsley, and W. Gong, “TCP Performance in Coded Wireless Mesh Networks,” *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. (SECON’08)*, pp. 179–187, June 2008.
- [11] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, “The Importance of Being Opportunistic: Practical Network Coding for Wireless Environments,” *Allerton Annual Conference on Communication, Control and Computing*, Enero 2005.
- [12] “IEEE Standard 802.11: IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks— Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE standards*, 2012.
- [13] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Costa, and B. Walke, “The IEEE 802.11 universe,” *IEEE Communications Magazine*, vol. 48, pp. 62–70, January, 2010.
- [14] “IEEE 802.11s: Standard for Information Technology. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 10: Mesh Networking,” *IEEE standards*, 2012.
- [15] A. Hakiri and A. Gokhale, “Work-in-Progress: Towards Real-Time Smart City Communications using Software Defined Wireless Mesh Networking,” *2018 IEEE Real-Time Systems Symposium (RTSS)*, pp. 177–180, Dec. 2018.
- [16] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, “Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies,” *Future Internet*, vol. 11, pp. 1–35, Apr 2019.
- [17] I. F. Akyildiz and X. Wang, “A Survey on Wireless Mesh Networks,” *IEEE Communications Magazine*, vol. 43, pp. 23–30, September 2005.

- [18] G. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: The WLAN Mesh Standard," *Wireless Communications, IEEE*, vol. 17, pp. 104–111, february 2010.
- [19] X. Wang and A. O. Lim, "IEEE 802.11s Wireless Mesh Networks: Framework and Challenges," *Ad Hoc Networks Journal (Elsevier)*, vol. 6, pp. 970–984, August 2008.
- [20] E. Khorov, A. Kiryanov, A. Lyakhov, and A. Safonov, "Analytical study of link management in IEEE 802.11s mesh networks," *International Symposium on Wireless Communication Systems (ISWCS)*, pp. 786–790, August, 2012.
- [21] C. Huang and W. Liao, "Throughput and delay performance of IEEE 802.11e enhanced distributed channel access (EDCA) under saturation condition," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 136–145, January, 2007.
- [22] E. Khorov, A. Krasilov, A. Krotov, and A. Lyakhov, "Will MCCA revive wireless multihop networks?," *Computer Communications*, vol. 104, pp. 159 – 174, 2017.
- [23] F. J. Matus, L. E. Morales, and M. R. Arias, "Performance analysis for a wireless mesh network test-bed using HWMP and BATMAN-Adv routing," *IEEE 37th Central America and Panama Convention*, pp. 1–6, Nov. 2017.
- [24] R. G. Garroppo, S. Giordano, and L. Tavanti, "A joint experimental and simulation study of the IEEE 802.11s HWMP protocol and airtime link metric," *International Journal of Communication Systems*, vol. 25, no. 2, pp. 92–110, 2012.
- [25] B. Staehle, M. Bahr, D. Fu, and D. Staehle, "Intra-Mesh Congestion Control for IEEE 802.11s Wireless Mesh Networks," *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN'12)*, vol. 1, pp. 1–7, Jul. 2012.
- [26] B. Blywis, M. Guenes, F. Juraschek, and J. H. Schiller, "Trends, Advances, and Challenges in Testbed-based Wireless Mesh Network Research," *Elsevier, Mobile Networks and Applications*, pp. 315–329, August, 2010.
- [27] K. Chen, Y. Xue, S. H. Shah, and K. Nahrstedt, "Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks," *IEEE Computer Communications*, vol. 27, pp. 923–934, 2004.
- [28] E. H. N. and M. A., "Strategies and a New Technique of Load Balancing in Wireless Mesh Networks," *Elsevier 3th International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing*, vol. 797, Sept. 2018.
- [29] A. Raniwala, P. De, S. Sharma, R. Krishnan, and T. . Chiueh, "End-to-End Flow Fairness Over IEEE 802.11-Based Wireless Mesh Networks," *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 2361–2365, May 2007.
- [30] R. L. Gomes, J. Jailton, W. Moreira, and A. J. Abelem, "QoE and QoS in wireless mesh networks," *2009 IEEE Latin-American Conference on Communications*, pp. 1–6, September, 2009.
- [31] S. Marwaha, J. Indulska, and M. Portmann, "Challenges and recent advances in QoS provisioning in wireless mesh networks," *8th IEEE International Conference on Computer and Information Technology*, pp. 618–623, July, 2008.
- [32] P. A. Shah, K. M. Awan, Z. ur Rehman, K. Iqbal, F. Aadil, K. Muhammad, I. Mehmood, and S. W. Baik, "A Route Optimized Distributed IP-Based Mobility Management Protocol for Seamless Handoff across Wireless Mesh Networks," *Elsevier Mobile Networks and Applications Journal*, pp. 752–774, February, 2018.
- [33] C. Barakat, E. Altman, and W. Dabbous, "On TCP performance in a Heterogeneous Network: A survey," *IEEE Communications Magazine*, vol. 38, pp. 40–46, Jan. 2002.
- [34] Y. Tian, K. Xu, and N. Ansari, "TCP in Wireless Environments: Problems and Solutions," *IEEE Communications Magazine*, vol. 43, pp. S27–S32, March 2005.
- [35] Z. Fu, X. Meng, and S. Lu, "How Bad TCP Can Perform In Mobile Ad Hoc Networks," *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC 2002)*, pp. 298–303, 2002.

- [36] A. Al Hanbali, E. Altman, and P. Nain, "A Survey of TCP over ad hoc Networks," *IEEE Communications Surveys Tutorials*, vol. 7, pp. 22–36, First quarter 2005.
- [37] K. Fouli, V. W. C. Chook, J. Heide, M. Medard, and S. Shi, "Random Linear Network Coding (RLNC)-Based Symbol Representation," *IETF Internet Draft*, pp. 1–13, March 2018.
- [38] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, pp. 1204–1216, July 2000.
- [39] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE/ACM Transactions on Information Theory*, vol. 49, pp. 371–381, February 2003.
- [40] S. R. Katti, "Network Coded Wireless Architecture," *Thesis dissertation - Massachusetts Institute of Technology*, September 2008.
- [41] R. W. Yeung, "Information Theory and Network Coding," *Springer Information Technology: Transmission, Processing and Storage*, pp. 401–405, 2008.
- [42] T. Ho and D. Lun, "Network Coding: An Introduction," *Cambridge Press*, pp. 1–184, 2008.
- [43] J. Robinson and E. W. Knightly, "A Performance Study of Deployment Factors in Wireless Mesh Networks," in *IEEE INFOCOM - 26th IEEE International Conference on Computer Communications*, pp. 2054–2062, 2007.
- [44] P. Gupta and P. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, March 2000.
- [45] Y. Gao, D.-M. C., and J. C. S. Lui, "Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications," *ACM Proceedings of the joint international conference on measurement and modeling of computer systems*, vol. 10, pp. 39 – 50, June 2006.
- [46] S. Sampaio, P. Souto, and F. Vasques, "A review of scalability and topological stability issues in IEEE 802.11s wireless mesh networks deployments," *International Journal of Communication Systems*, vol. 29, no. 4, pp. 671–693, 2016.
- [47] W. Fu and D. P. Agrawal, "Capacity of Hybrid Wireless Mesh Networks with Random APs," *IEEE Transactions on Mobile Computing*, vol. 99, 2011.
- [48] J. Jun and M. L. Sichitiu., "The nominal capacity of wireless mesh networks," *IEEE Wireless Communications*, vol. 10, no. 5, pp. 8 – 14, 2003.
- [49] A. K. Tiwari, T. G. Basavaraju, and B. R. Chandavarkar, "Congestion adaptive routing in Wireless Mesh Networks," in *Seventh International Conference on Wireless and Optical Communications Networks - (WOCN)*, pp. 1–5, 2010.
- [50] J. A. Stine, "Exploiting smart antennas in wireless mesh networks using contention access," *IEEE Wireless Communications*, vol. 13, no. 2, pp. 38–49, 2006.
- [51] I. Nicolaescu and D. Stoica, "Smart antennas for wireless communications systems," *IEEE 20th International Conference on Applied Electromagnetics and Communications*, pp. 1–4, 2010.
- [52] R. Bhatia and L. Li, "Throughput Optimization of Wireless Mesh Networks with MIMO Links," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 2326–2330, 2007.
- [53] S. Chu and X. Wang, "MIMO-Aware Routing in Wireless Mesh Networks," in *Proceedings of IEEE INFOCOM*, pp. 1–9, 2010.
- [54] E. Chai and K. G. Shin, "M-Polar: Channel Allocation for Throughput Maximization in SDR Mesh Networks," in *Proceedings IEEE INFOCOM*, pp. 1–9, 2010.
- [55] M. Muhammad, N. M. Salleh, M. S. Zakaria, V. R. Gannapathy, M. N. Husain, I. M. Ibrahim, M. S. Johal, M. R. Ahmad, and M. Z. A. A. Aziz, "Physical and MAC Cross Layer Design for Wireless Mesh Networks," in *IEEE Asia-Pacific Conference on Applied Electromagnetics*, pp. 1–5, 2007.
- [56] Da Guo, J. Li, M. Song, and J. Song, "A Novel Cross-Layer Routing Algorithm in Wireless Mesh Network," in *2nd International Conference on Pervasive Computing and Applications*, pp. 262–266, 2007.

- [57] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang, "Security in mobile ad hoc networks: challenges and solutions," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 38–47, 2004.
- [58] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.
- [59] M. S. Siddiqui, "Security Issues in Wireless Mesh Networks," in *IEEE International Conference on Multi-media and Ubiquitous Engineering*, pp. 717–722, 2007.
- [60] P. Yi, Y. Wu, F. Zou, and N. Liu, "A Survey on Security in Wireless Mesh Networks," *IETE Technical Review*, vol. 27, no. 1, pp. 6–14, 2010.
- [61] N. Wang and H. Wang, "A Security Architecture for Wireless Mesh Network," in *International Conference on Challenges in Environmental Science and Computer Engineering*, vol. 2, pp. 263–266, 2010.
- [62] P. Yi, T. Tong, N. Liu, Y. Wu, and J. Ma, "Security in Wireless Mesh Networks: Challenges and Solutions," *IEEE Sixth International Conference on Information Technology: New Generations*, pp. 423–428, 2009.
- [63] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, 1974.
- [64] C. Barakat, E. Altman, and W. Dabbous, "On TCP performance in a Heterogeneous Network: A survey," *IEEE Communications Magazine*, vol. 38, pp. 40–46, Jan. 2002.
- [65] M. Todorovic and N. Lopez-Benitez, "Efficiency Study of TCP Protocols in Infrastructured Wireless Networks," *IARIA International Conference on Networking and Services (ICNS)*, vol. 1, p. 103, 2006.
- [66] K. Leung and V. O. K. Li, "Transmission control protocol (TCP) in wireless networks: issues, approaches, and challenges," *IEEE Communications Surveys Tutorials*, vol. 8, no. 4, pp. 64–79, 2006.
- [67] K. Xu, S. Bae, S. Le, and M. Gerla, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Elsevier Ad hoc Networks Journal*, vol. 1, pp. 107–123, Jul. 2003.
- [68] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 1, pp. 72–76, Nov 2002.
- [69] J. Lee, S.-J. Lee, W. Kim, D. Jo, T. Kwon, and Y. Choi, "Understanding interference and carrier sensing in wireless mesh networks," *IEEE Communications Magazine*, vol. 47, pp. 102–109, July 2009.
- [70] S. Xu and T. Saadawi, "Revealing TCP unfairness behavior in 802.11 based wireless multi-hop networks," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 2, pp. E83–E87, 2001.
- [71] A. Margolis, R. Vijayakumar, and S. Roy, "Modelling Throughput and Starvation in 802.11 Wireless Networks with Multiple Flows," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 5123–5127, Nov. 2007.
- [72] C. Kai and S. Liew, "Temporal Starvation in CSMA Wireless Networks," *IEEE International Conference on Communications (ICC)*, pp. 1–6, Jun. 2011.
- [73] M. Backhaus, M. Theil, M. Rossberg, and G. Schaefer, "Poster: Seamless Client Integration for Fast Roaming in Wireless Mesh Networks," *2020 IFIP Networking Conference (Networking)*, pp. 637–639, 2020.
- [74] H. Zhai, X. Chen, and Y. Fang, "Alleviating intra-flow and inter-flow contentions for reliable service in mobile ad hoc networks," *IEEE Military Communications Conference (MILCOM)*, vol. 3, pp. 1640–1646, Oct. 2004.
- [75] K. Chen, Y. Xue, and K. Nahrstedt, "On setting TCP's congestion window limit in mobile ad hoc networks," *IEEE International Conference on Communications (ICC)*, vol. 2, pp. 1080–1084, May. 2003.
- [76] P. Kyasanur, J. So, C. Chereddi, and N. Vaidya, "Multichannel Mesh Networks: Challenges and Protocols," *IEEE Wireless Communications*, vol. 13, pp. 30–36, april 2006.
- [77] S. Das, H. Pucha, D. Koutsonikolas, Y. Hu, and D. Peroulis, "DMesh: Incorporating Practical Directional Antennas in Multichannel Wireless Mesh Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 2028–2039, nov. 2006.

- [78] D. Nandiraju, L. Santhanam, N. Nandiraju, and D. P. Agrawal, "Achieving Load Balancing in Wireless Mesh Networks Through Multiple Gateways," *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 807–812, 2006.
- [79] Y. Bejerano, S.-J. Han, and A. Kumar, "Efficient load-balancing routing for wireless mesh networks," *Computer Networks*, vol. 51, no. 10, pp. 2450–2466, 2007.
- [80] I. F. Akyildiz and X. Wang, "Cross-Layer Design in Wireless Mesh Networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 2, pp. 1061–1076, 2008.
- [81] C. Lai, K.-C. Leung, and V. Li, "Enhancing Wireless TCP: A Serialized-Timer Approach," *Proceedings of the IEEE INFOCOM, 2010*, pp. 1–5, march 2010.
- [82] S. Prasanthi, S.-H. Chung, and Y.-H. Jo, "A New Loss Recovery Algorithm for Increasing the Performance of TCP Over Wireless Mesh Networks," *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pp. 229–236, march 2012.
- [83] H. Xie, A. Boukerche, and M. Almula, "A novel cross layer TCP pacing protocol for multi-hop wireless networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1428–1433, 2013.
- [84] K. V. Singh, S. Gupta, S. Verma, and M. Pandey, "Improving Performance of TCP for Wireless Network using SDN," *ACM Proceedings of the 20th International Conference on Distributed Computing and Networking*, pp. 267–276, Jan. 2019.
- [85] P. Kulkarni, M. Sooriyabandara, and L. Li, "Improving TCP Performance in Wireless Networks by classifying causes of packet losses," *Proceedings of the 2009 IEEE conference on Wireless Communications & Networking Conference (WCNC'09)*, pp. 2786–2791, 2009.
- [86] R. Bassoli, H. Marques, J. Rodriguez, K. Shum, and R. Tafazolli, "Network Coding Theory: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1950–1978, Apr. 2013.
- [87] J. Sundararajan, D. Shah, M. M'endard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proceedings of the IEEE*, vol. 99, pp. 490–512, march 2011.
- [88] J. Zhang, Y. P. Chen, and I. Marsic, "Network Coding via Opportunistic Forwarding in Wireless Mesh Networks," *Wireless Communications and Networking Conference, 2008. (WCNC'08). IEEE*, pp. 1775–1780, april 2008.
- [89] C. Chiochan and E. Hossain, "Network Coding for unicast in a WiFi hotspot: Promises, challenges, and testbed implementation," *Computer Networks*, vol. 56, pp. 2963–2980, May 2012.
- [90] Z. Ning, Q. Song, L. Guo, Z. Chen, and A. Jamalipour, "Integration of scheduling and network coding in multi-rate wireless mesh networks: Optimization models and algorithms," *Elsevier Ad Hoc Networks*, vol. 36, pp. 386–397, Jan. 2016.
- [91] Y. Hsu, N. Abedini, N. Gautam, A. Sprintson, and S. Shakkottai, "Opportunities for Network Coding: To Wait or Not to Wait," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 1876–1889, Dec. 2015.
- [92] X. Yang, W. Wang, M. Lu, J. Wang, and X. Zhang, "Fair coding for inter-session network coding in wireless mesh networks," *Wireless Communications and Mobile Computing*, vol. 16, pp. 1922–1942, Dec. 2015.
- [93] Q. Liu, G. Feng, and Y. Guo, "A Framework of Joint Scheduling and Network Coding for Real-Time Traffic with Diverse Delay Constraints," *Wireless Personal Communications*, vol. 97, pp. 4855–4876, Dec. 2017.
- [94] K. Chi, X. Jiang, and S. Horiguchi, "Network Coding Opportunity Analysis of COPE in Multihop Wireless Networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 97, pp. 2858–2863, Apr. 2008.
- [95] S. Kafaie, M. Ahmed, Y. Chen, and O. Dobre, "Performance Analysis of Network Coding with IEEE 802.11 DCF in Multi-Hop Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 17, pp. 1148–1161, May. 2018.
- [96] K. Chi, Y. Wu, Y. Zhu, and V. Leung, "A Network Coding Scheme to Improve Throughput for IEEE 802.11 WLAN," *Springer Mobile Networks and Applications*, vol. 21, pp. 425–435, Jun. 2016.

- [97] L. Scalia, F. Soldo, and M. Gerla, "PiggyCode: A MAC Layer Network Coding Scheme to Improve TCP Performance Over Wireless Networks," *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 3672–3677, november 2007.
- [98] C.-C. Chen, G. Tahasildar, J.-S. Park, Y.-T. Yu, M. Gerla, and M. Sanadidi, "CodeMP: Network Coded Multipath to Support TCP in Disruptive MANETs," *The 9th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MESS'12)*, October 2012.
- [99] C.-C. Chen, C. Chen, J.-S. Park, S. Y. Oh, M. Gerla, and M. Sanadidi, "Multiple Network Coded TCP Sessions in Disruptive Wireless Scenarios," *Military Communications Conference (MILCOM'11)*, pp. 754–759, November 2011.
- [100] C. Zhang, Y. Chen, and C. Li., "Support of TCP in Wireless Mesh with Unstable Packet Forwarding Capacity," *IEEE International Conference on Communications (ICC)*, vol. 10, pp. 5295–5299, Jun. 2015.
- [101] C. Zhang, Y. Chen, and C. Li., "TCP adaptation with network coding and opportunistic data forwarding in multi-hop wireless networks," *PeerJ Computer Science*, Oct. 2016.
- [102] A. Jannesari, M. Sarram, and R. Sheikhpour, "A Novel Network Coding Algorithm to Improve TCP in Wireless Networks," *Springer Wireless Personal Communications*, vol. 110, pp. 1199–1216, Feb. 2020.
- [103] K. Alferaidi and R. Piechocki, "TCP-MAC Cross Layer Integration for Xor Network Coding," *Springer Advances in Intelligent Systems and Computing*, vol. 857, Nov. 2018.
- [104] Y. Xu, X. Bai, P. Wu., and L. Ding, "Evaluation and Enhancement of TCP with Network Coding in Wireless Multihop Networks," *EAI International Wireless Internet Conference (WiCon)*, vol. 10, pp. 490–500, Oct. 2011.
- [105] Y. O. Lee and M. K. Singh, "Network Coding for Improving the Fairness of Long-Hop TCP flows in a Multi-Hop Wireless Network," *Technical Report - Dept. of Electrical and Computer Engineering, Texas University*, 2010.
- [106] H. Liu, J. Chen, and Y. Gu, "A New Mechanism to Incorporate Network Coding Into TCP in Multi-radio Multi-channel Wireless Mesh Networks," *Mobile Ad-hoc and Sensor Networks (MSN'11)*, pp. 256–260, december 2011.
- [107] D. Gómez, S. Hassayoun, A. Herrero, R. Aguero, and D. Ros, "Impact of Network Coding on TCP Performance in Wireless Mesh Networks," *Personal, Indoor and Mobile Radio Communications (PIMRC'12)*, September 2012.
- [108] E. Takimoto, S. Aketa, S. Saito, and K. Mouri, "Network Coding with Wait Time Insertion and Configuration for TCP Communication in Wireless Multi-hop Networks," *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*, vol. 2, Mar. 2017.
- [109] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," *IEEE INFOCOM'03*, vol. 3, pp. 1744–1753, 2003.
- [110] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over Wireless Multihop Protocols: Simulation and Experiments," *IEEE International Conference on Communications (ICC'99)*, vol. 2, no. 2, pp. 1089–1094, 1999.
- [111] A. Bakre and B. Badrinath, "I-TCP: indirect TCP for mobile hosts," *Proceedings of the 15th International Conference on Distributed Computing Systems*, pp. 136–143, may 1995.
- [112] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi, "Split TCP for Mobile Ad Hoc Networks," in *Proceedings of the IEEE Global Communications Conference, GLOBECOM 2002*, vol. 1, pp. 138–142, nov 2002.
- [113] S. Floyd and T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm," *RFC 2582*, April 1999.
- [114] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan, "Understanding Congestion Control in Multihop Wireless Mesh Networks," *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pp. 291–302, 2008.

- [115] V. Loscri, "MAC Protocols over Wireless Mesh Networks: Problems and Perspective," *Journal of Parallel Distributed Computing*, vol. 68, pp. 387–397, Mar. 2008.
- [116] J. Núñez-Martínez and J. Mangues-Bafalluy, "A Survey on Routing Protocols that really Exploit Wireless Mesh Network Features," *IEEE Journal of Communications*, vol. 5, no. 3, pp. 211–231, 2010.
- [117] M. P. A. Pirzada and J. Indulska, "Performance Comparison of Multi-Path AODV and DSR Protocols in Hybrid wireless Mesh Network," *Computer Communications*, vol. 5, no. 31, pp. 885–895, 2008.
- [118] L.-P. Tung, W.-K. Shih, T.-C. Cho, Y. Sun, and M. C. Chen, "TCP Throughput Enhancement over Wireless Mesh Networks," *IEEE Communications Magazine*, vol. 45, pp. 64–70, November 2007.
- [119] C.-W. Hsu, T.-C. Hou, and C.-S. Wu, "A Transport Layer Mechanism for Fair TCP Throughput over 802.11 Multi-Hop Wireless Mesh Networks," *IEEE Wireless Communications and Networking Conference (WCNC'10)*, pp. 1–6, april 2010.
- [120] J. Kurose and K. Ross, "Computer Networking: A Top-Down Approach," *Pearson Education*, 7 edition, 2017.
- [121] V. Jacobson, "Congestion avoidance and control," *ACM Special Interest Group on Data Communication Conference (SIGCOMM)*, vol. 18, pp. 314–329, Aug. 1988.
- [122] D.-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, June 1989.
- [123] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control." RFC 5681 (Draft Standard), Sep. 2009.
- [124] L. Kleinrock, Fouad, and A. Tobagi, "Packet Switching in Radio Channels: Part I—Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, vol. 23, pp. 1400–1416, December 1975.
- [125] K. X. S. Bae, S. Lee, and M. Gerla, "TCP Behavior Across Multihop Wireless Networks and the Wired Internet," *Proceedings of the ACM Workshop on Wireless Mobile Multimedia (WoWMoM'02)*, pp. 41–48, September 2002.
- [126] F. Zhenghua, L. Haiyun, Z. P. Chiu, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Performance," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 209–221, March 2005.
- [127] X. M. Zhang, W. B. Zhu, N. N. Li, and D. K. Sung, "TCP Congestion Window Adaptation Through Contention Detection in Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 4578–4588, November 2010.
- [128] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Proceedings of the 5th annual ACM/IEEE international conference on Mobile Computing and Networking*, pp. 219–230, 1999.
- [129] C. Lai, K.-C. Leung, and V. Li, "TCP-NCL: A Unified Solution for TCP Packet Reordering and Random Loss," *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'09)*, pp. 1093–1097, sept. 2009.
- [130] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and Y. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, 1997.
- [131] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *Wireless Networks*, vol. 8, pp. 467–479, September 2002.
- [132] A. Zanella, G. Procissi, M. Gerla, and M. Sanadidi, "TCP Westwood: Analytic model and performance evaluation," *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, pp. 1703–1707, 2001.
- [133] A. Capone, L. Fratta, and F. Martignon, "Bandwidth Estimation Schemes for TCP over Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 129–143, april-june 2004.
- [134] R. Ferorelli, L. Grieco, S. Mascolo, G. Piscitelli, and P. Camarda, "Live Internet measurements using Westwood+ TCP congestion control," *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 3, pp. 2583–2587, nov. 2002.

- [135] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 216–228, feb 2003.
- [136] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1465–1480, Oct. 1995.
- [137] C. Caini and R. Firrincieli, "TCP Hybla: A TCP Enhancement for Heterogeneous Networks," *International Journal of satellite communications and Networking*, vol. 22, pp. 547–566, april-june 2004.
- [138] V. Jacobson and R. Braden, "TCP Extensions for Long-Delay Paths." RFC 1072 (Draft Standard), Oct. 1988.
- [139] J. C. Hoe, "Improving the Start-Up Behavior of a Congestion Control Scheme for TCP," *SIGCOMM Comput. Commun. Rev.*, vol. 26, pp. 270–280, Aug. 1996.
- [140] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, vol. 3, pp. 1537–1545, 2000.
- [141] R. Jain, D.-M. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," *Digital Equipment Corporation Technical Report, TR 301*, sep 1984.
- [142] Y. O. Lee and M. K. Singh, "Network Coding for Improving the Fairness of Long-Hop TCP flows in a Multi-Hop Wireless Network," *Technical Report - Dept. of Electrical and Computer Engineering, Texas University*, 2010.
- [143] J. Le, J. Lui, and D. Chiu, "How Many Packets Can We Encode? - An Analysis of Practical Wireless Network Coding," *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM'08*, pp. 371–375, 2008.
- [144] W. Chen, K. Letaief, and Z. Cao, "Buffer-Aware Network Coding for Wireless Networks," *IEEE/ACM Transactions on networking*, vol. 20, no. 5, pp. 1389–1401, 2012.
- [145] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, Mar. 2000.
- [146] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *Elsevier International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.



Codificación adaptativa de red para sistemas inalámbricos IEEE 802.11s en modo infraestructura

En la Ciudad de México, se presentaron a las 12:30 horas del día 25 del mes de enero del año 2023 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

- DR. ENRIQUE STEVENS NAVARRO
- DR. ULISES PINEDA RICO
- DRA. GRACIELA ROMAN ALONSO
- DR. GERARDO ABEL LAGUNA SANCHEZ
- DR. VICTOR MANUEL RAMOS RAMOS

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron a la presentación de la Disertación Pública cuya denominación aparece al margen, para la obtención del grado de:

DOCTOR EN CIENCIAS (CIENCIAS Y TECNOLOGIAS DE LA INFORMACION)

DE: CARLOS ERNESTO CARRILLO ARELLANO

y de acuerdo con el artículo 78 fracción IV del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

Aprobar

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.



CARLOS ERNESTO CARRILLO ARELLANO

ALUMNO

REVISÓ

MTRA. ROSALBA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

Román Linares Romero
DR. ROMAN LINARES ROMERO

PRESIDENTE

Enrique Stevens Navarro
DR. ENRIQUE STEVENS NAVARRO

VOGAL

Ulises Pineda Rico
DR. ULISES PINEDA RICO

VOCAL

Graciela Roman Alonso
DRA. GRACIELA ROMAN ALONSO

VOCAL

Gerardo Abel Laguna Sanchez
DR. GERARDO ABEL LAGUNA SANCHEZ

SECRETARIO

Victor Manuel Ramos Ramos
DR. VICTOR MANUEL RAMOS RAMOS