



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – TELECOMUNICACIONES

PRIORIZACIÓN DE PAQUETES DE VIDEO H.264/AVC PARA REDES
INALÁMBRICAS

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
GRISSEL HUERTA MONTIEL

TUTOR
DR. VÍCTOR GARCÍA GARDUÑO
Facultad de Ingeniería - División de Ingeniería Eléctrica

MÉXICO, D. F. FEBRERO 2014



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

JURADO ASIGNADO:

Presidente: Dr. Rangel Licea Víctor
Secretario: Dr. Gomez Castellanos Javier
Vocal: Dr. García Garduño Víctor
1er Suplente: Dr. Psenicka Bohumil
2do Suplente: Dr. Moctezuma Flores Miguel

Lugar o lugares donde se realizó la tesis: San Diego State University, San Diego. CA y CIUDAD UNIVERSITARIA, CU. México, D.F.

TUTOR DE TESIS:

Dr. García Garduño Víctor

FIRMA

*Con todo mi cariño y mi amor
para las personas que hicieron todo en la vida
para que yo pudiera lograr mis sueños
por motivarme y darme la mano
cuando sentía que el camino se terminaba
a ustedes por siempre
mi corazón y mi agradecimiento
porque por ustedes soy lo que soy.*

Gracias por confiar en mí...

Mi familia

Agradecimientos

Gracias a Dios, por guiarme en el sendero correcto de la vida, cada día en el transcurso de mi camino e iluminándome en todo lo que realizo, por darme fuerzas y coraje para seguir adelante y hacer frente a los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento, gracias a él porque ha llenado mi vida de bendiciones.

A mi familia, por inculcarme valores que de una u otra forma me han servido en la vida, por apoyarme en cada decisión que tomo, y por estar a mi lado en cada momento, gracias por eso y por mucho más.

A mi asesor el Dr. Víctor García Garduño por su amabilidad, buena disposición y principalmente por abrirme las puertas a este proyecto.

Al Dr. Sunil Kumar, por su dedicación, paciencia y por el tiempo que me dedicó para que este trabajo culminara exitosamente, mi agradecimiento más sincero.

A San Diego State University (SDSU) que me permitió el uso de sus instalaciones para realizar este proyecto en el tiempo que duró la estancia de investigación.

A la Universidad Nacional Autónoma de México UNAM por la formación adquirida y por todas las oportunidades que me ha brindado.

A la Coordinación de Estudios de Posgrado (CEP) de la UNAM por el apoyo económico en forma de beca para la realización de mis estudios de maestría.

A mis sinodales, gracias por darme la oportunidad y por el tiempo que han dedicado para leer este trabajo.

A mi gran amigo el Dr. Mauricio Lara por sus consejos y experiencia de los cuales he aprendido demasiado, por su apoyo incondicional y porque siempre ha sido una fuente de inspiración.

A Gabi, por ser mi confidente, amigo y compañero; gracias a tu paciencia, comprensión y tu gran apoyo. Ahora puedo decir que esta tesis lleva mucho de ti, gracias por vivir este proyecto juntos.

A mis amigos quienes se han sumado a mi vida y son parte de mi familia, me han ayudado a crecer profesionalmente y sobretodo personalmente, que han compartido conmigo tantas experiencias, y siempre han estado a mi lado apoyándome incondicionalmente. Gracias a ustedes porque también son parte de este proyecto.

"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

- Albert Einstein

Resumen

Actualmente, debido a la alta demanda de servicios multimedia por parte de los usuarios de redes inalámbricas, se han generado una gran diversidad de estudios enfocados en mejorar la calidad del servicio (QoS) y la calidad de experiencia (QoE) del usuario, tomando en cuenta la gran pérdida de paquetes y el retardo debido a las condiciones variables del canal inalámbrico.

En esta tesis nos enfocamos sólo en la transmisión de paquetes para un servicio de *streaming* de video H.264/AVC en tiempo real; y para cuestiones de análisis, pruebas y simulaciones empleamos una red Ad hoc. El algoritmo propuesto es un administrador de paquetes de video el cual considera que los paquetes que se transmiten en tiempo real son muy susceptibles a las pérdidas y al retardo, es decir, teniendo conocimiento previo que no todos los paquetes perdidos causan el mismo impacto en el video decodificado; que un video no sincronizado puede causar *jitter*, turbulencia y problemas de latencia; y que los paquetes se deben transmitir rápidamente por el administrador de tal manera que ellos deben llegar a su destino en el menor tiempo posible. El algoritmo propuesto coloca los paquetes más importantes al principio de la cola para ser transmitidos primero; basándose en la importancia de cada paquete de video por transmitir, la cual se determina por los siguientes criterios: el timestamp de cada paquete, el número de trama dentro de un grupo de imágenes (GOP, Group of Pictures), la prioridad del paquete determinado por el codificador H.264/AVC, el retardo permitido para cada tipo de paquete y por último la prioridad del usuario; organizando todos estos campos en conjunto logramos tener mejores resultados.

En esta tesis demostramos la efectividad del esquema propuesto mediante simulaciones en ns-2.34 y posteriormente presentamos un estudio de comparación el cual demuestra una mejora del PSNR con el algoritmo propuesto en una rango de entre 0.5 dB a 1.5 dB, y del retardo logrando en algunos escenarios obtener la mitad de retardo en comparación con el esquema FIFO.

Índice General

Capítulo

Resumen	vi
1 Introducción	1
1.1 Definición del problema.....	6
1.2 Objetivo de la Tesis.....	6
1.2.1 Objetivos particulares.....	6
1.3 Organización de la Tesis.....	7
2 Introducción al estándar de video H.264/AVC en sistemas inalámbricos de video de <i>streaming</i>.	9
2.1 Características principales del códec H.264/AVC.	10
2.2 Perfiles y niveles.....	12
2.3 Codificación de video.....	13
2.4 El administrador/transmisor.....	15
2.5 Decodificación de video y cancelamiento de error.....	
2.6 Calidad de Servicio y Calidad de Experiencia en redes inalámbricas.....	19
2.6 Estimación de la distorsión.....	
2.6.1 PSNR y MSE.....	20
3 Introducción a las redes MANET	22
3.1 Introducción al estándar 802.11.....	24
3.1.1 La arquitectura de la red WLAN.....	25
3.1.2 Red Ad Hoc.....	26
3.1.3 Función de distribución controlada (DCF)	30
3.2 Protocolos de enrutamiento para redes MANET.....	33
3.2.1 Protocolo AOMDV.....	34
4 Algoritmos para la administración de paquetes	36
4.1 El administrador de paquetes en redes inalámbricas.....	38
4.2 Clasificación de los algoritmos para la administración de paquetes.....	39
4.3 Trabajos relacionados	41

5 Entorno de evaluación	45
5.1 Descripción y funcionalidades del ns-2.34.....	46
5.1.1 Arquitectura básica del ns-2.34.....	47
5.2 Implementación de sistemas de comunicaciones inalámbricos en ns-2.....	49
5.3 Esquemas de encolamiento en ns-2.....	53
5.3.1 La interfaz de la Cola.....	54
5.4 El Evalvid como interfaz entre el codec y el ns-2.34.	55
5.5 Priorización de paquetes de video H.264/AVC.....	56
5.5.1 Asignación de prioridad en el ns-2.34.	56
5.6 Importancia del <i>timestamp</i>	57
6 Esquema propuesto para la administración de paquetes de video H.264/AVC en la cola usando el ns-2.34	59
6.1 Recepción del paquete en el buffer.....	60
6.2 Encolamiento del paquete.....	61
6.3 Ordenación de paquetes en el buffer.....	61
6.3.1 Ordenamiento basado en el <i>timestamp</i> del paquete	62
6.3.2 Ordenamiento basado en la prioridad del paquete.....	63
6.3.3 Ordenamiento basado en la prioridad del usuario.....	65
7 Resultados	66
7.1 Características del escenario de simulación.	66
7.2 Resultados del PDR y retardo punto a punto en el esquema por <i>timestamp</i>	68
7.3 Desempeño del algoritmo basado en el PSNR.....	79
7.4 Resultados del PDR en el esquema por usuario.....	83
7.5 Resultados del PSNR en el esquema por usuario.....	88
Conclusiones	90
Contribuciones	93
Trabajo futuro	94
Referencias	95

Índice de Figuras

Figura

1.1	Diagrama básico que presenta la priorización y administración de paquetes de video.....	1
2.1	Línea del tiempo de los estándares de codificación de video	10
2.2	Perfiles del estándar H.264/AVC.....	12
2.3	Diagrama a bloques del codificador H.264/AVC.....	15
2.4	Diagrama a bloques del decodificador H.264/AVC.....	17
2.5	Comparación de estrategias de cancelamiento de error.....	18
3.1	Ambiente real de una red Ad Hoc.....	22
3.2	Interoperabilidad de la red Ad Hoc con otras redes inalámbricas.....	23
3.3	Evolución del protocolo IEEE 802.11.....	25
3.4	Ilustración de la arquitectura 802.11 IBSS.....	26
3.5	Red inalámbrica de sensores.....	28
3.6	Diagrama de la arquitectura de la capa MAC 802.11.....	30
3.7	Tiempos de censado antes de transmitir un paquete en el protocolo 802.11.....	31
3.8	Funcionamiento del método de acceso DCF.....	32
3.9	Zona de colisión por terminal oculta.....	32
3.10	Protocolos RTS/CTS.....	33
3.11	Enrutamiento utilizando el protocolo AOMDV.....	35
4.1	Administración de paquetes en la cola.....	36
4.2	Un típico administrador de paquetes en redes inalámbricas	38
4.3	Sistema de encolamiento básico.....	41

4.4	Sistema de encolamiento con paquetes de alta prioridad.....	43
5.1	Relación entre los lenguajes C++ y OTcl en el ns-2.34.....	47
5.2	Módulo de nodo móvil, canal inalámbrico, antenas y propagación en el ns-2.34.....	49
5.3	Funcionamiento del canal inalámbrico en el ns-2.34.....	50
5.4	Diagrama de bloques de un nodo móvil asociado a un canal inalámbrico.....	52
5.5	Diagrama a bloques que presenta la relación entre el Evalvid, el H.264/AVC y el ns-2.34.....	55
5.6	Problema en la cola cuando los paquetes no están ordenados por prioridad.....	57
5.7	Problema en la cola cuando el <i>timestamp</i> ha expirado.....	58
6.1	Paquetes ordenados por su <i>timestamp</i>	63
6.2	Paquetes ordenados por su prioridad.....	63
6.3	Paquetes ordenados por su <i>timestamp</i> y prioridad.....	64
6.4	Paquetes ordenados por su prioridad de usuario, <i>timestamp</i> y prioridad.....	65
7.1	Topología del escenario de evaluación.....	67
7.2	Resultados del PDR para una velocidad de 5 m/s, cuatro prioridades y hasta cuatro conexiones.....	69
7.3	Comparación de los resultados del PDR para una velocidad de 5m/s entre el esquema propuesto y el esquema FIFO.....	69
7.4	Comparación de resultados del retardo <i>punto a punto</i> para una velocidad de 5m/s entre el esquema propuesto y el esquema FIFO.....	71
7.5	Resultados del PDR para una velocidad de 10 m/s, cuatro prioridades y hasta cuatro conexiones.....	73
7.6	Comparación de los resultados del PDR para una velocidad de 10 m/s entre el esquema propuesto y el esquema FIFO.....	73
7.7	Comparación de resultados del retardo <i>punto a punto</i> para una velocidad de 10 m/s entre el esquema propuesto y el esquema FIFO.....	75
7.8	Resultados del PDR para una velocidad de 20 m/s, cuatro prioridades y hasta cuatro conexiones.....	77

7.9	Comparación de los resultados del PDR para una velocidad de 20 m/s entre el esquema propuesto y el esquema FIFO.....	77
7.10	Comparación de resultados del retardo <i>punto a punto</i> para una velocidad de 20 m/s entre el esquema propuesto y el esquema FIFO.....	79
7.11	Comparación de una misma trama de video en un escenario con velocidad de 5 m/s; entre el video recibido usando nuestro algoritmo y el video recibido usando el esquema FIFO.....	80
7.12	Comparación de una misma trama de video en un escenario con velocidad de 10 m/s; entre el video recibido usando nuestro algoritmo y el video recibido usando el esquema FIFO.	81
7.13	Comparación de una misma trama de video en un escenario con velocidad de 10 m/s; entre el video recibido usando nuestro algoritmo y el video recibido usando el esquema FIFO.	82
7.14	PDR obtenido en dos usuarios en el esquema por ' prioridad de usuario' con velocidades de: 5, 10 y 20 m/s.....	85
7.15	PDR obtenido en tres usuarios en el esquema por ' prioridad de usuario' con velocidades de: 5, 10 y 20 m/s.....	90
7.16	PDR obtenido en cuatro usuarios en el esquema por ' prioridad de usuario' con velocidades de: 5, 10 y 20 m/s.....	91

Índice de Tablas

Tabla

2.1	Descripción de cada valor de MOS.	19
4.1	Requisito de QoS para cada aplicación.....	37
5.1	Protocolos implementados en el ns-2.34.....	46
7.1	Características del escenario de evaluación.....	66
7.2	Resultados del PDR para una velocidad de 5 m/s, cuatro prioridades y hasta cuatro conexiones.....	68
7.3	Resultados del retardo para una velocidad de 5 m/s, cuatro prioridades y hasta cuatro conexiones.	70
7.4	Resultados del PDR para una velocidad de 10 m/s, cuatro prioridades y hasta cuatro conexiones.....	72
7.5	Resultados del retardo para una velocidad de 10 m/s, cuatro prioridades y hasta cuatro conexiones.	74
7.6	Resultados del PDR para una velocidad de 20 m/s, cuatro prioridades y hasta cuatro conexiones.....	76
7.7	Resultados del retardo para una velocidad de 20 m/s, cuatro prioridades y hasta cuatro conexiones.....	78
7.8	Resultados del PSNR para velocidades de 5, 10 y 20 m/s, cuatro prioridades y hasta cuatro conexiones.....	79
7.9	Resultados del PDR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema por prioridad de usuario.....	83
7.10	Resultados del PDR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta	84

	cuatro usuarios. Esquema FIFO.....	
7.11	Resultados del PSNR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema por prioridad de usuario.	88
7.12	Resultados del PSNR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema FIFO.....	88

ABREVIATURAS

ABR	Available Bit Rate
AODV	Adhoc On-Demand Distance Vector
AOMDV	Adhoc On-Demand Multipath Distance Vector
ARP	Address Resolution Protocol
ASO	Arbitrary Slice Ordering
AP	Access Point
AVC	Advanced Video Coding
BSS	Basic Service Set
BSSID	BSS Identification
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
CBR	Constant Bit Rate
CFP	Contention-Free Period
CMSE	Cumulative Mean Square Error
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CP	Contention Period
CRC	Cyclic Redundancy Channel
CTS	Clear to Send
DCF	Distributed Coordination Function
DCT	Discrete Cosine Transform
DDB	Drop Dependency Based
DIFS	DCF inter-frame space
DP	Data Partitionating
DRR	Deficit Round-Robin

DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DT	Drop Tail
EIFS	Extended Inter Frame Space
FDDT	Frame-Level Packet Discard with Dynamic Thresholds
FFQ	Frame-Based Fair Queuing
FGS	Fine Granularity Scalability
FMO	Flexible Macroblock Order
GOP	Group of Pictures
GPS	Generalized Processor Sharing
HOL	Head-of-Line
HRR	Hierarchical Round-Robin
IBSS	Independent BSS
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission.
ITU	Telecommunication Standardization Sector
Jitter-EDD	Jitter-Earliest-Due-Date
JVT	Joint Video Team
MANET	Mobile Ad hoc Networks
MBs	MacroBlocks
MP	Multi Path fading
MPEG-4	Moving Picture Experts Group
MSE	Mean Square Error
MVs	Motion Vector
NAL	Network Abstraction Layer

NAV	Network Allocation Vector
nrt-VBR	non-real time variable bit rate
OLSR	Optimized Link State Routing
PAN	Personal Area Network
PCF	Point Coordination Function
PDR	Packet Delivery Ratio
PIF	PCF inter-frame space
PL	Path Loss
PLR	Packet Loss Rate
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RED	Random Early Drop
RREQ	Route Request
RTP	Real-Time Transport Protocol
RTS	Request to Send
rt-VBR	real-time VBR
SCFQ	Self-Clocked Fair Queuing
SIFS	Short Inter-Frame Space
SINR	Signal to Interference and Noise Ratio
SGQ	Stop-and-Go Queuing
SH	SHadowing
TDM	Time Division Multiplexing
TTL	Time to Live
UDP	User Datagram Protocol
VC	Virtual Clock
VCL	Video Coding Layer

VoD	Video on Demand
VoIP	Voice over IP
VS	Video Streaming
WFQ	Weighted Fair Queuing
WLANs	Wireless Local Area Networks
WRR	Weighted Round-Robin

Capítulo 1

Introducción

Actualmente con el gran avance de la tecnología en redes inalámbricas se ofrecen servicios multimedia tales como: *streaming* de video (VS, *Video Streaming*), video sobre demanda (VoD, *Video on Demand*) y videoconferencia. El éxito de cada aplicación depende del manejo eficiente de los recursos limitados de la red como son: la potencia de transmisión, el ancho de banda y el tiempo. También se deben considerar las condiciones del canal inalámbrico, tal como su variabilidad en el tiempo, el tipo de contenido multimedia y el retardo permitido para que la reproducción del video proporcione una buena calidad de la experiencia (QoE, *Quality of Experience*) y a su vez una buena calidad del servicio (QoS, *Quality of Service*).



Figura 1.1 Diagrama básico que presenta la priorización y administración de paquetes de video.

Cualquier sistema inalámbrico que transmite *streaming* de video está compuesto de tres bloques principalmente (Figura 1.1):

- 1) **El servidor multimedia** que contiene una colección de secuencias de video previamente codificadas que pueden ser requeridas por los usuarios mediante dispositivos que adquieren video en tiempo real.

2) **El administrador**, el cual es el encargado de administrar y asignar los recursos de la red para la transmisión de los datos.

3) **Los usuarios** que reciben los datos y los reproducen, o los transmiten a otros usuarios en la red.

La adaptación en conjunto de estos tres bloques incrementa la eficiencia del *streaming* de video ya que se toman en cuenta diferentes parámetros como: los parámetros de codificación de macrobloques (MB) para el codificador o el servidor, los parámetros de la red y el canal tales como la administración de paquetes para su transmisión y la asignación del ancho de banda en los canales por el administrador y los parámetros para la decodificación eficiente del video utilizando esquemas de resistencia al error. El ajuste de estos parámetros refuerza el control para la asignación de recursos disponibles a través de las capas de red del modelo OSI, desde la capa de aplicación hasta la capa física, esto lleva a la implementación de esquemas más robustos denominados esquemas de capas cruzadas.

Con respecto al número de usuarios involucrados, la transmisión de video puede ser clasificada como de punto a punto, difusión y multidifusión [2].

1) La **transmisión de video de punto a punto** (*streaming de video unicast*), es de un servidor a un usuario. La videoconferencia es un caso especial de la transmisión de video de punto a punto, ya que requiere una baja latencia (un bajo retardo inicial, *buffering*). Una propiedad importante en la comunicación de punto a punto es que existe retroalimentación entre el usuario y el servidor. El servidor es capaz de adaptar su procesamiento basándose en la información que recibe del usuario con respecto a la calidad de la canal.

2) La **transmisión de video por difusión** (*streaming de video broadcast*), implica un servidor y varios usuarios. Un ejemplo clásico de esto es la radiodifusión de televisión digital terrestre o por satélite. Debido al gran número de usuarios, la retroalimentación por lo general no es factible, lo que limita la capacidad del servidor para adaptarse a las condiciones cambiantes condiciones del canal.

3) La **transmisión de video por multidifusión** (*streaming de video multicast*), también implica un servidor y varios usuarios, pero no tantos como en el anterior. Un ejemplo de la

transmisión de video de multidifusión es *Multicast streaming de video over IP* a través de Internet como el que usa el servidor de Windows Media para enviar videos a usuarios que se han suscrito con su dirección IP.

También podemos clasificar la transmisión de *streaming* de video de acuerdo a cómo se adquieren los datos. En general se consideran dos escenarios diferentes donde se utilizan los esquemas de capas cruzadas:

1) El primer escenario es el ***streaming de video en tiempo real punto a punto***. Las aplicaciones en este escenario incluyen videoconferencias y aplicaciones donde los dispositivos móviles se pueden utilizar para grabar video y transmitirlo hacia otro dispositivo en tiempo real. En estas aplicaciones, el video se debe transmitir primero sobre un enlace inalámbrico de subida a una estación base y posteriormente el video se transmite mediante un enlace de bajada al usuario receptor. El video se codifica en la marcha y el dispositivo transmisor tiene el control sobre los parámetros de codificación del video como el tamaño del paso de cuantificación de los bloques de video y los parámetros de la red como el ancho de banda. En el caso de los dispositivos móviles inalámbricos, la energía disponible en el dispositivo transmisor también es un recurso limitado que se debe considerar en el esquema de asignación de recursos.

2) El segundo escenario aplica el ***streaming de video almacenado en un servidor***. El video primero se codifica y almacena en un servidor, en este caso los parámetros de codificación de video no son controlados por el administrador a menos que sea capaz de decodificar y transcodificar el video en tiempo real. Sin embargo, en un sistema basado en paquetes el administrador se puede adaptar a las condiciones del canal y controlar los paquetes de video, eligiendo retrasar o eventualmente eliminar los paquetes de video en vez de transmitirlos. Este escenario es el que se utiliza para este proyecto de tesis.

A diferencia de los sistemas de video de descarga y reproducción, que requieren recibir todo el flujo de bits antes de comenzar la reproducción, las aplicaciones de *streaming* de video permiten a los clientes iniciar la reproducción de video sin tener que descargar todo el flujo de bits. Una vez que se inicia la reproducción de video, se puede continuar sin interrupción hasta el final del video. Con el fin de lograr la reproducción sin interrupciones, incluso en situaciones cuando el ancho de banda de la red fluctúa, un usuario debe ser capaz de amortiguar los datos que recibe tolerando los retardos. De forma ideal este retardo es fijo y no depende de la longitud del video [1].

Para lograr una reproducción continua, el intervalo entre el tiempo cuando se transmite una trama de video por el servidor y el tiempo cuando esta trama se recibe por el usuario debe ser la misma para todas las tramas. Esto significa que hay un tiempo límite para la recepción de cada trama, y todos los paquetes que corresponden a la misma trama deben estar disponibles al mismo tiempo para su visualización. Si algunos paquetes no alcanzan el tiempo límite, se los considerará como perdidos y se tendrá que utilizar alguna técnica para la ocultación de errores la cual se emplearán en la decodificación de la trama de video. Si los paquetes que no cumplieron el plazo llegan al usuario más tarde, ellos simplemente se descartarán. Este concepto de plazos para los paquetes es esencial en la transmisión de video para obtener una buena QoE.

Por lo general hay un retraso inicial antes de que un usuario inicie la reproducción del video. En el clásico *streaming* de video no interactivo, este retraso puede ser significativo, hasta varios segundos. El espectador tendrá que tolerar el retraso inicial, pero una vez que el video comience a reproducirse, este será entregado de forma continua. Esto requiere que el intervalo entre el tiempo de transmisión y recepción de una misma trama de video sea constante. Para las aplicaciones interactivas como la videoconferencia, el retraso (latencia) no puede ser grande, de lo contrario será difícil mantener una conversación entre dos o más personas. Una latencia máxima típica para aplicaciones interactivas es 150ms [2].

La mayoría de las aplicaciones de *streaming* de video utilizan el video pre-codificado el cual elimina todas las restricciones de complejidad computacional respecto a la compresión de video en el servidor y permite también el uso de diferentes técnicas de compresión de video, tales como la codificación de varios pasos. Sin embargo, la codificación de video en tiempo real puede adaptarse de manera eficiente a los cambios de las condiciones de canal, mientras que el video pre-codificado ofrece una flexibilidad limitada. Una solución a este problema es usar la codificación de video escalable para que la transmisión pueda adaptarse a los cambios de ancho de banda por adición o eliminación de las capas de mejora.

Las técnicas de codificación de video escalable permiten adaptar la tasa de bits del transmisor mediante la incorporación de la escalabilidad espacial, temporal y de calidad dentro de una misma secuencia de bits. Por ejemplo, la escalabilidad de calidad permite el uso de una tasa de bit baja para la transmisión de una capa base, posteriormente las capas de mejoramiento progresivamente aumentan la calidad del video al recibir bits adicionales. En tales esquemas la recepción completa de la capa base es vital para evitar la propagación de errores en el

decodificador. En el caso de la escalabilidad de granularidad fina (FGS, *Fine Granularity Scalability*) en MPEG-4 y la extensión escalable del estándar H.264, las capas de mejoramiento se pueden truncar a cualquier punto para lograr un avance en la calidad del video. Cada esquema de codificación de capas también permite una protección desigual del error donde diferentes niveles de protección se pueden aplicar a la capa base y a las capas de mejoramiento de los paquetes de video

El envío de los paquetes de video sobre el protocolo de internet (IP, *Internet Protocol*) es la base de la mayoría de los servicios multimedia actualmente disponibles, como por ejemplo: la televisión sobre Protocolo de Internet (IPTV, *Internet Protocol Television*), la videoconferencia y el *streaming* de video. En todos estos casos, es sumamente importante controlar la QoE que se ofrece al usuario final: primero porque la arquitectura típica de la capa de transporte puede utilizar el protocolo de transporte en tiempo real (RTP, *Real-Time Transport Protocol*), el Protocolo de Transporte por Datagrama (UDP, *User Datagram Protocol*) ó la dirección IP, lo cual no garantiza por sí mismo la prestación de servicios con una determinada QoS; y segundo porque cuando se transmite video, la relación entre la QoS de la red y QoE del usuario puede resultar complejo.

Para muchas aplicaciones, es importante que se alcance cierta QoE a pesar de la alta demanda de servicios multimedia en ambientes inalámbricos debido al incremento de usuarios y al crecimiento de aplicaciones móviles con diferentes requerimientos de QoS. Para mejorar la calidad del video recibido existen diferentes técnicas para la administración de paquetes de video en la red las cuales ofrecen grandes ventajas, sin embargo, los paquetes son susceptibles a las pérdidas debido a los diversos problemas que surgen en la red inalámbrica como los errores e irregularidades del ambiente móvil lo cual afecta el rendimiento del decodificador, esto conlleva a la generación de distorsiones en el video por lo que la QoE disminuye.

1.1 Definición del problema

En un ambiente de redes inalámbricas es muy frecuente la pérdida de paquetes debido a diferentes causas como: el rompimiento de enlaces, la congestión, la alta movilidad, capacidad del canal, no hay rutas disponibles, saturación del buffer y el control de acceso al medio. Para el caso de la transmisión de streaming de video H.264/AVC y sabiendo que tendremos por cualquier situación la pérdida de paquetes, debemos tomar en cuenta como punto principal la calidad de video entregada al usuario final.

1.2 Objetivo de la Tesis

Proponer una técnica de priorización de paquetes de video codificado en el estándar H.264/AVC dentro de una red inalámbrica, que permita aumentar la cantidad de paquetes recibidos con más alta prioridad con la finalidad de mejorar la calidad del video entregado al usuario final.

1.2.1 Objetivos particulares

1. Implementar un algoritmo para el ordenamiento de paquetes de video dentro del buffer, para poder recibir los paquetes de video más importantes.
2. Implementar un algoritmo para eliminar paquetes con un tiempo de vida expirado (TTL, *Time to Live*).
3. Estudiar el tamaño óptimo del paquete de video a ser transmitido con la finalidad de maximizar el *PDR (Packet Delivery Ratio)*.

Para implementar el modelo de administración de paquetes, se usará la herramienta de simulación de redes **ns-2.34** y el *software Evalvid* para hacer posible la transmisión de paquetes de video H.264/AVC (en un capítulo posterior se hablará un poco más del *software* libre utilizado).

1.3 Organización de la Tesis

Para la comprensión adecuada del proyecto propuesto en esta tesis decidimos organizar los capítulos como sigue:

El capítulo 2 es una breve introducción al estándar de video H.264/AVC con la finalidad de conocer sus características generales ya que es el estándar que elegimos para este proyecto. En este capítulo no se profundiza mucho en este tema debido a que nuestra tesis no está enfocada en ello, sólo es una herramienta más que utilizamos y la que nos resultó más versátil para el tipo de paquetes de video que transmitimos.

En el capítulo 3 vemos acerca de la infraestructura de las redes MANET, ya que este ambiente inalámbrico es el que utilizamos por facilidad y rápido ajuste de parámetros para el análisis de la transmisión de video en tiempo real. Este tipo de redes es con la que ha estado trabajando el grupo de multimedia y redes de San Diego State University. En este capítulo revisaremos un poco acerca del estándar 802.11 principalmente el modo independiente que es el que se aplica para redes Ad hoc utilizando la función de distribución controlada. Y por último también revisamos los protocolos de enrutamiento para redes MANET y nos enfocamos principalmente en el protocolo AOMDV que es el que utilizamos en este proyecto.

El capítulo 4 es el más importante porque de aquí partimos para la investigación y análisis de nuestro esquema propuesto para la administración de paquetes de video. Primero revisamos las diferentes clases de servicios y su retardo permitido para obtener una buena QoS. Después presentamos algunas de las características deseables para tomar en cuenta cuando se diseña un algoritmo para la administración de paquetes. Posteriormente realizamos unas clasificaciones básicas de los administradores de paquetes y cuáles de estas utilizamos para nuestro algoritmo propuesto. Una parte importante que también mencionamos en este capítulo es el estudio de los trabajos relacionados ya que es muy importante saber que investigaciones previas se han hecho para poder proponer otra solución tomando en cuenta las dificultades ya encontradas.

En el **capítulo 5** revisamos todo el entorno de evaluación, principalmente como adaptamos el software libre: ns-2.34 y el evalvid, para poder transmitir los paquetes de video en tiempo real. Elegimos escribir este capítulo para saber las características que ofrecen estos dos software para propósitos de nuestros análisis e investigaciones enfocándonos en redes inalámbricas. Un punto

muy importante es conocer la interfaz de la cola tal y como está implementada en el ns-2.34, ya que justamente este archivo es el que modificamos con nuestro algoritmo propuesto. También en este capítulo abordamos la importancia de la priorización de paquetes de video por el códec H.264/AVC y la importancia del *timestamp* ya que ambos parámetros utilizamos para el diseño de nuestro algoritmo.

En el **capítulo 6** describimos nuestro algoritmo propuesto organizado en tres partes principales: la recepción del paquete en el buffer; el encolamiento y verificación del TTL del paquete; y la parte principal que es el ordenamiento de los paquetes en el buffer. El ordenamiento de los paquetes lo hacemos tomando en cuenta tres campos: el *timestamp* del paquete, la prioridad del paquete y la importancia del usuario.

Por último escribimos el **capítulo 7** donde presentamos todos los resultados obtenidos; en gráficas y tablas mostrando el PDR y el retardo punto a punto. También mostramos las comparativas de nuestros resultados con el esquema FIFO y presentamos los resultados del algoritmo cuando implementamos el esquema de prioridad por usuario. Al final se escriben las conclusiones y el trabajo a futuro.

Capítulo 2

Introducción al estándar de video H.264/AVC en sistemas inalámbricos de *streaming* de video

El grupo de expertos de codificación de video ITU-T (*ITU Telecommunication Standardization Sector*) inicio el proyecto de Codificación de Video Avanzado (*AVC Advanced Video Coding*) con el objetivo de que este proyecto fuera mejor que el códec actual H.263, al mismo tiempo el grupo de expertos ISO/IEC (*International Organization for Standardization* y por *la International Electrotechnical Commission*) estaba trabajando en el proyecto del MPEG-4 (*Moving Picture Experts Group*), sin embargo ambos proyectos no resultaron lo que se esperaba debido a la complejidad del códec lo cual no sería muy atractivo en el mercado; por tal motivo ambos grupos unieron fuerzas para trabajar en el mismo proyecto el cual fue llamado Equipo Conjunto de Video (*JVT Joint Video Team*)[1]. Al final fue nombrado como H.264 como una recomendación de la ITU-T, este códec también es conocido en ISO como MPEG-4 parte 10 y actualmente es formalmente conocido como **H.264/AVC**.

El códec H.264/AVC, desde que se introdujo formalmente en 2003 ha sido aceptado universalmente para una gran diversidad de aplicaciones y a partir del 2010 es el códec para todos los nuevos servicios de video ya que ofrece el doble de compresión que el MPEG-2 por eso es tan atractivo para varios servicios, además de su alta calidad de video y eficiencia de almacenamiento, actualmente podemos decir que es un códec para todas las video-comunicaciones, las distribuciones, el almacenamiento y las aplicaciones de video en la red.

En la Figura 2.1 podemos ver algunos de los estándares de codificación de video más representativos a lo largo de tiempo.

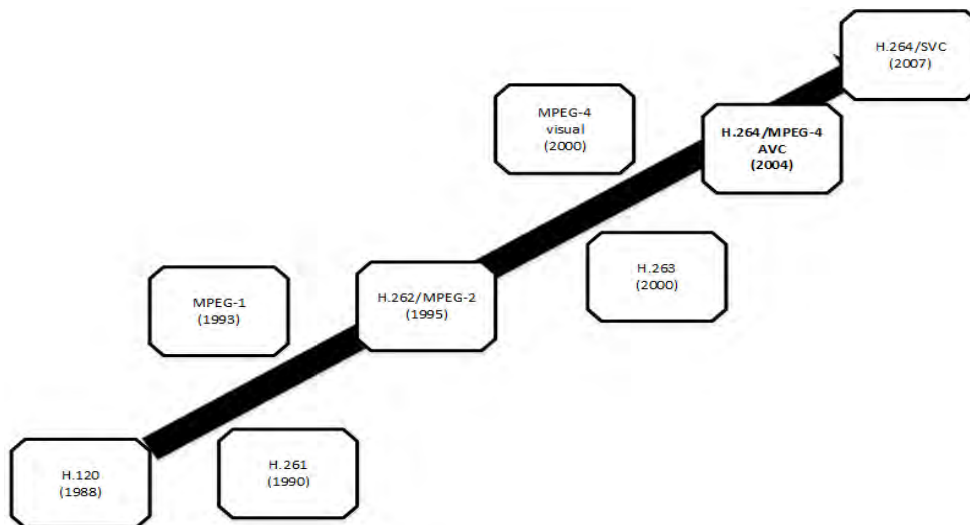


Figura 2.1 Línea del tiempo de los estándares de codificación de video [1].

2.1 Características principales del códec H.264/AVC

- Reducción de hasta un 50 por ciento en la tasa de bit en comparación con el MPEG-4: este nuevo códec logra una reducción en promedio de tasa de bit de hasta el 50 por ciento para un mismo grado de optimización de codificación. Esto significa que H.264/AVC ofrece alta calidad para todas las tasas de bits incluyendo las tasas bajas de otros estándares.
- Adaptación a limitaciones de retardos: el H.264/AVC puede operar en un modo de retardo bajo para adaptarse a aplicaciones de telecomunicaciones (ej. videoconferencias) así como también se adapta a aplicaciones con menos limitaciones de retardo tales como el video almacenado y streaming de video.
- Resistencia a errores: el H.264/AVC proporciona las herramientas necesarias para trabajar con pérdidas de paquetes y bits erróneos en un ambiente de redes inalámbricas.
- Redes compatibles: El códec tiene una funcionalidad que conceptualmente separa la capa de codificación de video (**VCL**, *Video Coding Layer*) de la capa de abstracción de red (**NAL**, *Network Abstraction Layer*). La primera capa proporciona el núcleo de alta compresión para representar el contenido de video, y la última capa se encarga de la entrega de paquetes en diferentes tipos de redes. Esto facilita la paquetización y mejora el control de prioridades de la información. La NAL puede proporcionar compresión de video en dos formatos, para el protocolo basado en *stream* como en el MPEG-2 y para el protocolo basado en paquetes como

RTP/IP y TCP/IP. Para el primer caso, los datos son proporcionados con códigos de inicio de tal forma que la capa de transporte y el decodificador puedan fácilmente identificar la estructura de la cadena de bits. Para el protocolo basado en paquetes, la estructura de bit se identifica por los encabezados del paquete[2].

- Dos esquemas de codificación de entropía: la codificación de longitud variable de contexto adaptivo (CAVLC, *Context-Adaptive Variable Length Coding*) y la codificación de aritmética variable de contexto adaptivo (CABAC, *Context-Adaptive Binary Arithmetic Coding*).
- Múltiples imágenes de referencia para la compensación de movimiento, ya que permite que los slices B y P se utilicen como referencia.
- Estimación de movimiento con tamaño de bloque variable desde 4x4 hasta 16x16 píxeles.
- Compensación de movimiento con precisión de un cuarto de píxel lo cual mejora la precisión de la predicción y la complejidad de la interpolación es más baja.
- Predicción espacial direccional en macrobloques intra-codificados para una compresión más eficiente.
- La transformada coseno discreta (DCT, *Discrete Cosine Transform*) utilizada en MPEG-2 se reemplaza en el H.264/AVC por una transformada coseno con coeficientes enteros la cual disminuye la complejidad de su implementación[x].
- Utiliza la transformada coseno con coeficientes enteros con tamaño de bloque ajustable de 8x8 y 4x4.
- Filtro de desbloqueo para remover distorsiones causadas por compensación de movimiento y cuantificación.
- Un conjunto de parámetros que son utilizados entre el codificador y decodificador para sincronización de sintaxis.
- Mayor resistencia al error a través de arreglos flexibles de MBs (*MacroBlocks*) y slices.
- Partición de datos (DP *Data Partitioning*) y paquetización de *slices* de video en tres niveles de prioridad para proporcionar protección de transmisión.
- Transmisión redundante de algunas regiones para mejorar la robustez ante la pérdida de datos.
- Sincronización y conmutación de imágenes para un streaming de video más robusto [5].

2.2 Perfiles y niveles

Debido a la diversidad en los sistemas de comunicaciones y en los servicios de multimedia ofrecidos, se requiere que los estándares de codificación soporten interoperabilidad entre los diferentes escenarios, en el caso del H.264/AVC esta característica es sumamente importante debido a que tiene que cubrir un amplio rango de servicios de video; por ejemplo, para empresas de telecomunicaciones, IPTV y video para *broadcast*. Para hacer frente a este rango diverso de aplicaciones, el estándar H.264/AVC define varios perfiles y niveles para adaptar el códec a una aplicación específica, de otra manera el códec sería extremadamente complejo y caro.

Los perfiles y niveles asignan restricciones a las cadenas de bits y a las capacidades necesarias para decodificarlos; el estándar H.264/AVC a través de los perfiles define un subconjunto de herramientas de codificación en la cadena de bits, por ejemplo, el caso donde el codificador soporta *slices* tipo B, resistencia al error y DP. Dentro de las capacidades definidas para un perfil en particular, es todavía posible tener una gran variedad de funcionalidades como: la tasa de bit, la resolución de la imagen y la velocidad de las tramas, estos son los límites impuestos por el nivel en cada cadena de bits.

La primera versión del estándar H.264/AVC especifica tres perfiles: el de base, el principal y el perfil extendido (ver Figura 2.2). Las principales aplicaciones que utilizan el perfil de base abarcan los servicios de videotelefonía y videoconferencia en comunicaciones inalámbricas; el perfil principal se utiliza para la difusión de televisión sobre IP y en el almacenamiento de video; el perfil extendido es principalmente utilizado para aplicaciones de streaming de video multimedia.

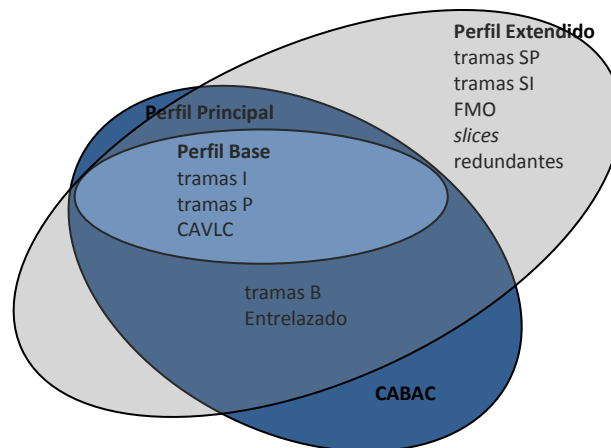


Figura 2.2 Perfiles del estándar H.264/AVC.

1) El **perfil de base**, es el perfil más simple del estándar H.264/AVC y se utiliza para transmitir video progresivo. Este perfil soporta muchas características del estándar excepto: los *slices* tipo B, la predicción basada en peso, el CABAC, el video entrelazado, los *slices* tipo SP/SI y DP. La ventaja de este perfil es la eficiencia y robustez de la codificación con baja complejidad computacional.

2) El **perfil principal** contiene todas las características del estándar H.264/AVC con la excepción de las herramientas de resistencia a errores como el ordenamiento flexible de los macrobloques (FMO, *Flexible Macroblock Order*), el ordenamiento arbitrario de *slices* (ASO, *Arbitrary Slice Ordering*), los *slices* redundantes, DP y los *slices* tipo SP/SI. Así que sólo un subconjunto de secuencias de video codificadas pueden ser decodificadas por el decodificador de perfil principal.

3) En el **perfil extendido** se incluyen todas las partes del perfil de base. Los *slices* tipo SP: es un tipo de *slice* especial codificado de tal forma que puede cambiar eficientemente de resolución entre tramas de video, es similar a la codificación de un *slice* tipo P. Los *slices* tipo SI presentan características similares al SP, pero son similares a codificar un *slice* tipo I. La partición de datos (DP); los datos codificados se colocan en particiones separadas en capas diferentes, se utiliza para reducir los errores durante la transmisión y para el empaquetamiento efectivo de los datos. Los *slices* tipo B se utilizan para predicción bidireccional, incrementan la calidad del video y la velocidad binaria. Y otra característica es que utilizan predicción basada en el peso [3][4].

2.3 Codificación de video

El H.264/AVC utiliza una etapa fundamental basada en la técnica de compensación de movimiento; el primer paso es dividir la trama de video primero en MBs, que son bloques de pixeles codificados independientemente (Intra) o codificados usando la predicción temporal de tramas previas (Inter). La inter codificación de MBs utiliza los vectores de movimiento (MVs, *Motion Vector*) que especifican la ubicación de bloques de referencia en una trama que se

transmitió previamente. Los bloques Intra y los bloques residuales se codifican después de la predicción de movimiento de los bloques Inter, posteriormente se cuantifican los coeficientes y se realiza la codificación de entropía para completar el proceso de codificación. En la Figura 2.3 se presentan el conjunto de parámetros utilizados para la codificación de fuente donde se muestran los diferentes modos aplicados a los MBs (tales como Intra, Inter o *Skip*).

Para mejorar la eficiencia de la codificación, los MBs dentro de una trama se pueden codificar aplicando la predicción espacial como la técnica de codificación de predicción intra utilizada en el estándar H.264/AVC. Se puede introducir una mayor robustez al error codificando un grupo de MBs juntos en un mismo *slice*, el cual es independientemente decodificable de otros *slices* en la misma trama de video; además un paquete de video podría consistir de un solo *slice* de una trama, el cual sería independientemente decodificable para el receptor y cuyas pérdidas no afectarían la decodificación de cualquier otro *slice* en la trama.

Un reciente análisis de las técnicas para la resistencia al error en la codificación de video se proporciona en la referencia [5], donde se toma en cuenta tanto la capacidad de los *slices* de proporcionar una resincronización así como la probabilidad de la pérdida de paquetes en ambientes inalámbricos.

Un buen análisis acerca de la importancia de considerar la longitud de paquetes en ambientes inalámbricos se encuentra en la referencia [6], los paquetes de mayor longitud son más susceptibles a errores de bits que pueden causar más grandes distorsiones; sin embargo, al dividir una trama de video en múltiples *slices* de longitudes más cortas se incrementan los datos de encabezado de los *slices* así como también se agregan encabezados de los protocolos IP/UDP/RTP.

En el caso del video pre-codificado, varias secuencias de video con diferentes tasas de bits de fuentes correspondientes a diferentes niveles de calidad de compresión pueden estar disponibles en el servidor; de esta forma, el conjunto de parámetros de codificación de fuente podrían determinar el nivel de calidad de la secuencia de video del enlace de bajada en uso.

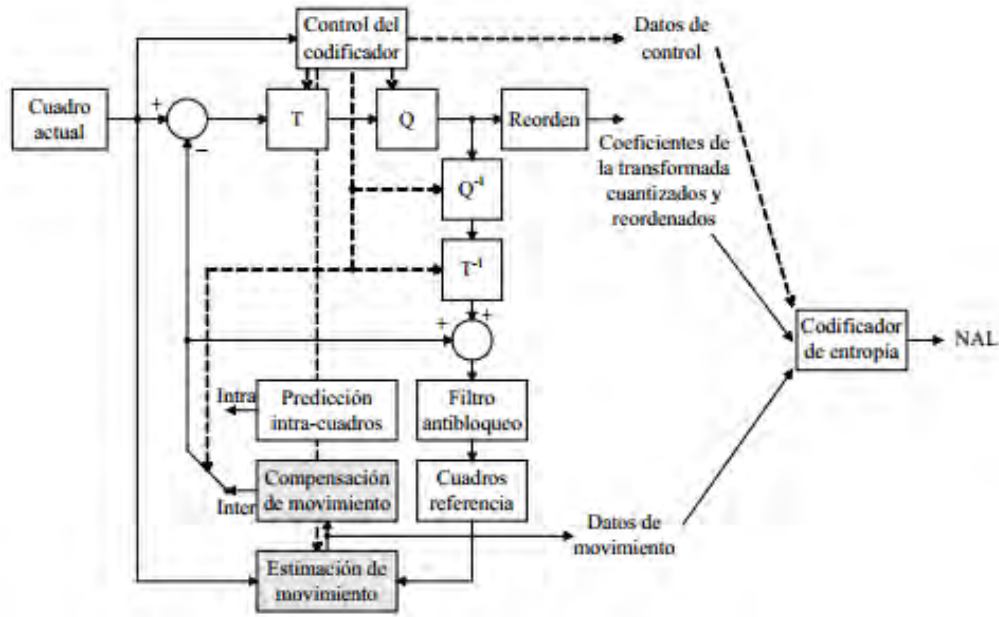


Figura 2.3 Diagrama a bloques del codificador H.264/AVC.

El H.264/AVC ofrece una gran flexibilidad para proporcionar la capacidad de conmutar entre diferentes secuencias de video pre-codificadas en puntos específicos (usando tramas SP y SI) de cada secuencia. En la codificación escalable aplicada a las secuencias de video, los parámetros de codificación de fuente pueden determinar los niveles de escalabilidad espacial, temporal, y la calidad del video transmitido, la selección de estos parámetros determinará la velocidad de transferencia de datos y también afectará la QoE después de la decodificación por parte del usuario.

2.4 El administrador/transmisor

Los paquetes de video ya codificados se envían al administrador donde se les asignarán los recursos de la red; la forma más simple de administrar sería con un esquema *round-robin* en el cual se utilice un sistema de multiplexación por división de tiempo TDM (Time Division Multiplexing) como el implementado en las redes de 2G GSM, donde el tiempo de transmisión se comparte por igual entre todos los usuarios.

Investigaciones recientes han presentado que los esquemas de administración con información del canal (*Channel-Aware Scheduling Schemes*) asignan más recursos a los usuarios que tengan las mejores condiciones del medio para la transmisión de sus datos, estos esquemas pueden presentar significativamente mejores resultados que los esquemas *round-robin* logrando una ganancia en diversidad multi-usuario. [7][8]

Para evitar favorecer a los usuarios que están más cerca a la estación base, se han estado desarrollando esquemas que son más equitativos en la asignación de recursos, los cuales toman en cuenta el promedio de *throughput* por cada usuario así como también las condiciones del canal.

Otros esquemas tales como la Regla Exponencial (*Exponential Rule*) intentan estabilizar las longitudes de la cola para cada usuario lo cual es benéfico para las aplicaciones que son sensibles al retardo tales como la transmisión de videoconferencia.[9]

En la referencia [10] se utiliza un esquema de administración basado en un gradiente, el cual asigna los recursos de tal forma que las tasas de transmisión de los usuarios son pesadas por los gradientes de un sistema. Este esquema generalmente permite el uso de un amplio rango de funciones basadas tanto en la longitud de la cola como en el *throughput*.

La mayoría de los esquemas de administración de paquetes actualmente utilizados en las redes inalámbricas no logran la mejor calidad de transmisión del video debido a que no toman en cuenta el tipo de contenido del video cuando realizan la decisión de administración, además la mayoría de estos esquemas tienen la característica de ser muy complejos; en cambio, los esquemas basados en el contenido podrían funcionar significativamente mejor en términos de la distorsión y el retardo *end-to-end* del video recibido por el usuario, asimismo utilizan más eficientemente los recursos disponibles en la red; cualquier esquema de administración de este tipo debe tener en cuenta: el método de codificación de video, las condiciones del canal, los métodos de decodificación y los métodos a la resistencia al error empleados en el decodificador para tomar las decisiones de administración lo mejor posible.

2.5 Decodificación de video y cancelamiento de error

De acuerdo a la Figura 2.4 el paso final es el decodificador del dispositivo del usuario final. El decodificador debe tomar en cuenta: los errores de los paquetes debido al canal, el retardo de los paquetes, los paquetes que se eliminaron en el buffer, y las técnicas de cancelamiento de error antes de la reproducción del video. **En un sistema de *streaming* de video, cualquier paquete que llega al decodificador después de un dado límite de tiempo para su reproducción será considerado como paquete perdido.**

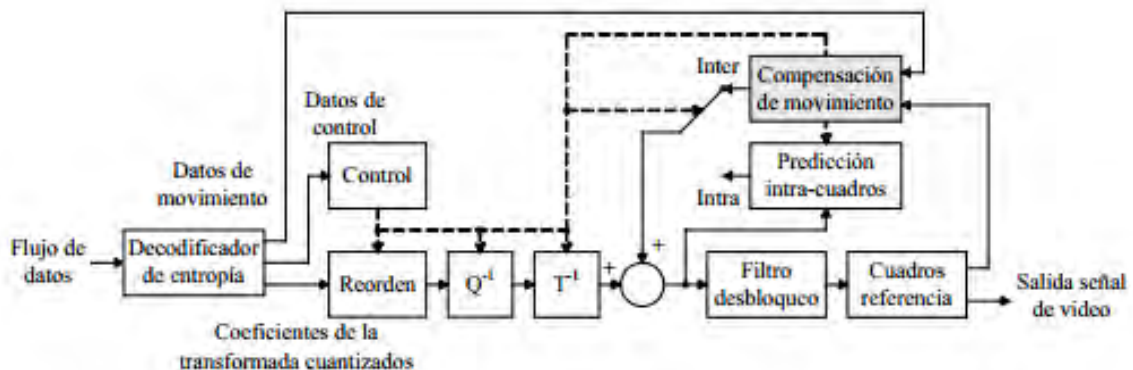


Figura 2.4 Diagrama a bloques del decodificador H.264/AVC.

Los errores en las tramas inter codificadas son comúnmente compensados usando técnicas de cancelamiento temporal que explotan correlaciones temporales en la escena [11]; asumiendo que un MB se pierde, el método más simple de cancelamiento de error temporal es copiar los píxeles de un MB que se encuentra en la misma ubicación de una trama previa; algunos métodos más complicados utilizan MVs a partir de MBs vecinos recibidos en la trama para determinar la posición de una compensación de movimiento del MB en la trama de referencia; dado que más de un candidato MVs puede estar disponible, se puede usar una técnica de relación de fronteras para determinar el mejor candidato posible [12] o volver a estimar el MVs [11]. Los errores en las tramas intra se pueden cancelar usando técnicas de cancelamiento espacial [13]; el método más simple y más efectivo de cancelación se basa en esquemas ponderados de promedio de píxeles donde el peso depende de la distancia de los píxeles cancelados.

En general, las estrategias de cancelamiento de error empleadas por el decodificador intentan explotar la redundancia que es común en la mayoría de las secuencias de video, otro recurso efectivo actualmente utilizado son los esquemas de cancelación de error híbrido espacio-temporal [14]. Una amplia revisión de esquemas de cancelación de error y sus beneficios se pueden encontrar en la referencia [15].

La Figura 2.5 presenta la diferencia en la calidad del video decodificado usando el esquema más simple de copiado de MB en comparación con una estrategia de cancelación de error más compleja como la técnica del límite coincidente para un paquete perdido.

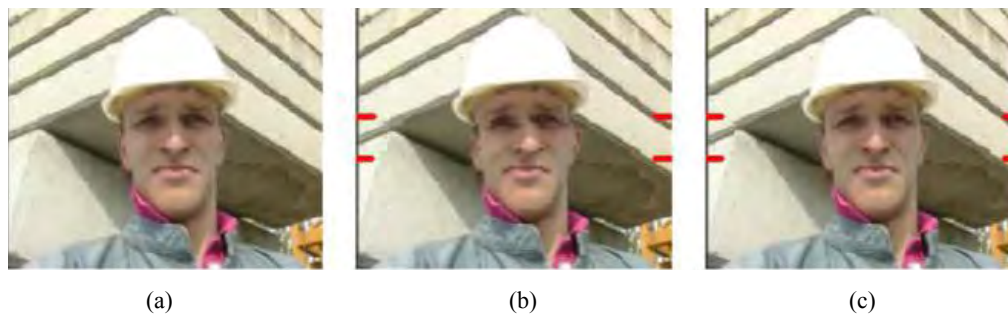


Figura 2.5 Comparación de estrategias de cancelación de error. (a) Trama decodificada sin pérdida de paquetes (b) trama decodificada utilizando una técnica simple de cancelación de error (c) trama decodificada utilizando una técnica compleja de cancelación de error [15].

2.6 Calidad de Servicio y Calidad de Experiencia en redes inalámbricas.

La calidad de servicio (QoS) se define en [ITU-T E.800] como el efecto colectivo del desempeño de la red, que determina el grado de satisfacción de un usuario del servicio. Para mejorar el funcionamiento global del sistema se implementan mecanismos de QoS que contribuyen a mejorar la experiencia del usuario final. La QoS depende de las políticas implementadas en la red como: la asignación de parámetros de configuración, los mecanismos de encolamiento, protocolos de enrutamiento y acceso al medio.

La QoS es una medida objetiva determinada por diferentes parámetros como: ancho de banda, latencia, *jitter* y paquetes perdidos. Las condiciones ideales para la transmisión de video son un gran ancho de banda y una baja latencia (retardo punto a punto de no más de 200 ms).

La calidad de experiencia (QoE) se define en [ITU-T P.10/G.100] como el nivel aceptación general de una aplicación o servicio por el usuario final, tal como la percibe subjetivamente. Incluye los efectos completos del sistema punto a punto (usuario, terminales, red, infraestructura de servicios, etc.) y está influenciada por las expectativas del usuario, el tipo de servicio proporcionado y el contexto de la aplicación; por lo que es una medida subjetiva y puede diferir de un usuario a otro.

Una de las medidas subjetivas más comunes es el MOS (*Mean Opinion Score*) presentado en la Tabla 2.1, el cual proporciona una indicación numérica de la calidad percibida del video por el usuario, en un rango del 1 al 5, siendo 1 la peor calidad y 5 la mejor.

MOS	Descripción
5	Perfecta. Como una conversación cara a cara
4	Bueno. Algunas imperfecciones se pueden percibir, pero el video sigue siendo claro. Este valor es el esperado en servicios de telefonía móvil.
3	Regular. El video en algunos momentos tiende a ser molesto.
2	Muy molesto. Difícil de visualizarse.
1	Imposible visualizarse.

Tabla 2.1 Descripción de cada valor de MOS.

2.7 Estimación de la distorsión

En los esquemas basados en el contenido del video, la forma de dar priorización a determinados paquetes es de acuerdo a la distorsión de la secuencia de video decodificado en comparación con la misma secuencia del video original. La evaluación objetiva de la distorsión de video y la calidad del video es un área de investigación en curso del procesamiento digital de video, en la referencia [16] se pueden encontrar excelentes análisis acerca de las métricas para la calidad del video.

2.7.1 PSNR Y MSE

La **Relación Señal Pico a Ruido** o **PSNR** (*Peak Signal-to-Noise Ratio*) es un término utilizado en ingeniería para definir la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Debido a que muchas señales tienen un gran rango dinámico, el PSNR se expresa generalmente en escala logarítmica, utilizando como unidad el decibelio.

El uso más habitual del PSNR es como medida cuantitativa de la calidad de la reconstrucción en el ámbito de la compresión de imágenes. La señal en este caso son los datos originales, y el ruido es el error introducido por la compresión o por la pérdida de datos durante la transmisión en la red inalámbrica. Normalmente un PSNR alto indica que el video reconstruido es de más alta calidad.

Para definir más fácilmente el PSNR se hace indispensable la formulación del error cuadrático medio **MSE** (*Mean Square Error*), que para dos imágenes monocromáticas I y K de tamaño $M \times N$ se define como:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2 \quad (2.1)$$

Donde M y N son el ancho y la altura de la imagen en pixeles, respectivamente. I y K son los valores de luminancia de la imagen original y reconstruida.

de esta forma el PSNR se define como:

$$PSNR = 10 * \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (2.2)$$

Donde MAX_I^2 denota el máximo valor de error que puede tomar un píxel en la imagen, para una representación de luminancia de 8 bits por pixel este parámetro toma un valor de 255. Cuando MAX_I^2 se representan usando B bits por muestra, toma el siguiente valor:

$$MAX_I^2 = 2^B - 1 \quad (2.3)$$

Para una imagen en formato RGB, la definición del PSNR es la misma, pero el MSE se calcula como la media aritmética de los MSEs de los tres colores (R,G y B).

Los valores típicos del PSNR para una buena calidad de video están entre 30 y 40 dB, sin embargo el rango de valores del PSNR cambia de acuerdo al tipo de video y su configuración de codificación. Cuando las dos imágenes son idénticas, el valor del MSE es cero, y para este valor el PSNR es indefinido.

El comité MPEG emplea un valor de umbral informal de 0.5 dB en el incremento del PSNR para decidir si se incluye una determinada mejora en un algoritmo de codificación, ya que se considera que este aumento del PSNR es apreciable visualmente. Para la transmisión de video en redes inalámbricas los valores aceptables de PSNR son considerados alrededor de 20 dB a 25 dB.

Capítulo 3

Introducción a las redes MANET

Las redes MANET (*Mobile Ad hoc Networks*) son redes con tecnología ad hoc cuyo propósito es proporcionar flexibilidad y autonomía aprovechando los principios de auto-organización. En la actualidad la formación de redes ad hoc incluye escenarios tales como los que se representan en la Figura 3.1.

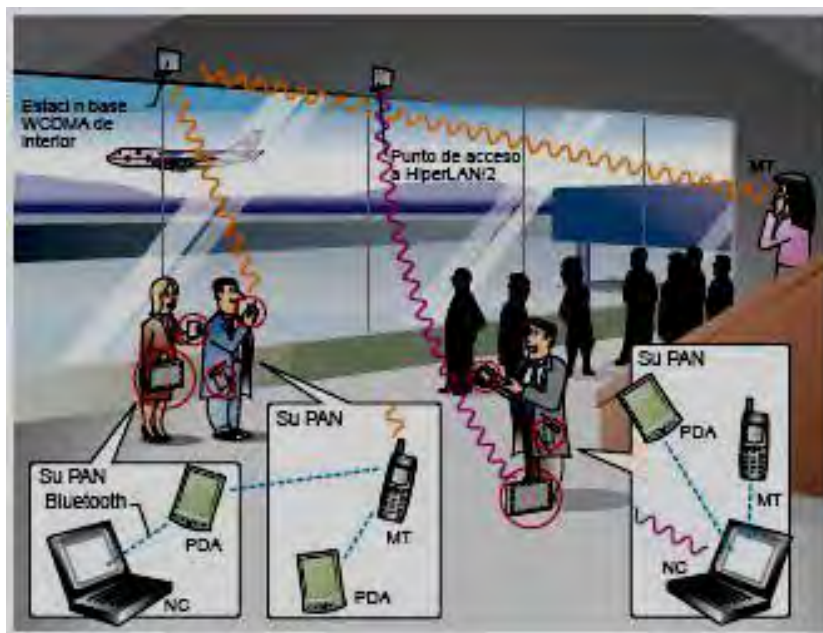


Figura 3.1 Ambiente real de una red Ad hoc.

Una red móvil ad hoc es una red formada sin ninguna administración central, es decir, no hay un nodo central, sino que consta de nodos móviles que usan una interfaz inalámbrica para enviar paquetes de datos; todos los nodos están en igualdad de condiciones. La conexión es establecida por la duración de una sesión; los nodos descubren otros nodos cercanos o dentro del rango de transmisión para formar la red, aunque también pueden buscar nodos que están fuera del área de alcance conectándose con otros nodos intermedios que estén conectados a la red y estén a su

alcance, por lo tanto las conexiones son posibles por múltiples nodos [17]. Las redes ad hoc presentan frecuentemente cambios en su topología debido a su movilidad por lo que la utilización de protocolos de enrutamiento se vuelve más complejo.

En lugares donde las personas llevan dispositivos que puedan conectarse en red ad hoc, los dispositivos pueden tanto interconectarse entre sí como conectarse a un punto de información local por ejemplo, para consultar actualizaciones de salidas de vuelos, cambios de puerta, etc. Los dispositivos ad hoc pueden también retransmitir tráfico entre dispositivos que estén fuera de su alcance; por lo tanto, el escenario del aeropuerto contiene una mezcla de tecnologías de comunicaciones con saltos de radios simples y múltiples. La Figura 3.2 representa este aspecto de interoperabilidad en lo que se relaciona con la formación de redes ad hoc.

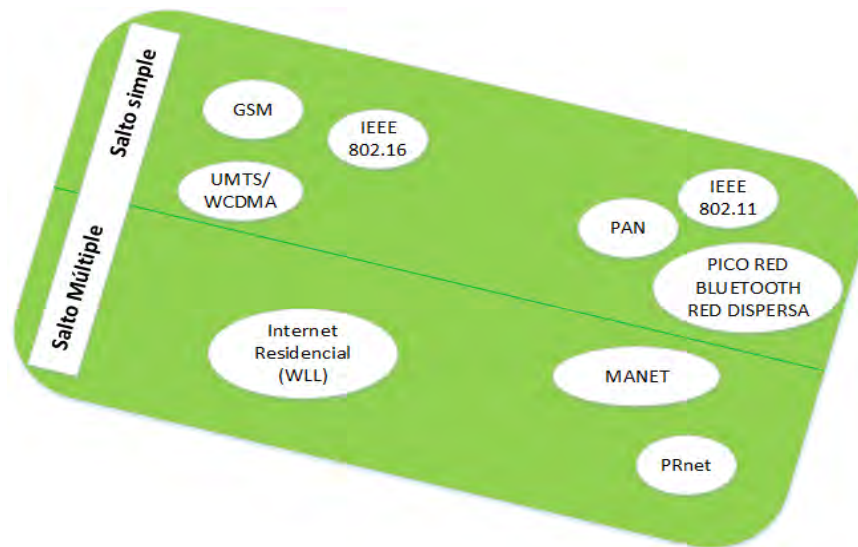


Figura 3.2. Interoperabilidad de la red Ad hoc con otras redes inalámbricas.

3.1 Introducción al estándar 802.11

Las redes inalámbricas de área local (WLANs *Wireless Local Area Networks*) o popularmente conocidas como WiFi se han desplegado enormemente en diferentes ambientes como empresas, casas y redes públicas ya que actualmente permiten altas tasas de transmisión de datos.

La primera generación de dispositivos 802.11 introducidos al mercado en los 90's proporcionaron un servicio de mejor-esfuerzo (*Best Effort*) con una tasa de transmisión de hasta 2Mbps. Actualmente el estado de arte de los dispositivos que soportan la tecnología 802.11 pueden utilizar aplicaciones de multimedia incluyendo VoIP (*Voice over IP*) y video con una seguridad más robusta, también pueden alcanzar tasas de transmisión de hasta 54Mbps y ser utilizados en ambientes de alta movilidad. Actualmente, los dispositivos basados en IEEE 802.11a y 802.11g proporcionan tasas de transmisión de hasta 54 Mbps, sin embargo el nuevo estándar IEEE 802.11n soporta hasta 600Mbps [18](ver Figura 3.3).

El rango de transmisión de un dispositivo WiFi típico es de hasta 300m en espacios abiertos sin obstáculos, donde varía dependiendo de la potencia de transmisión y las condiciones variables del ambiente inalámbrico. Los dispositivos bajo el estándar 802.11 operan sin licencia en las bandas a 2.4 y 5 GHz.

Actualmente muchas de las computadoras, tabletas, teléfonos inteligentes, cámaras digitales entre otros dispositivos utilizan interfaces WLAN para comunicarse, la aplicación más típica del 802.11 es el acceso a internet desde los dispositivos portables en diferentes ambientes inalámbricos, incluyendo ambientes al aire libre, empresas, en casa y zonas de alta demanda de tráfico como en aeropuertos donde muchos puntos de acceso (AP, *Access Point*) son utilizados para proporcionar servicio de internet en una área determinada. También el 802.11 se puede utilizar en comunicaciones punto-a-punto (*peer-to-peer*) para compartir archivos entre computadoras, o también es posible utilizar dispositivos de juego conectados inalámbricamente, y muchas más aplicaciones. También este estándar puede ser utilizado en ambientes ad hoc.

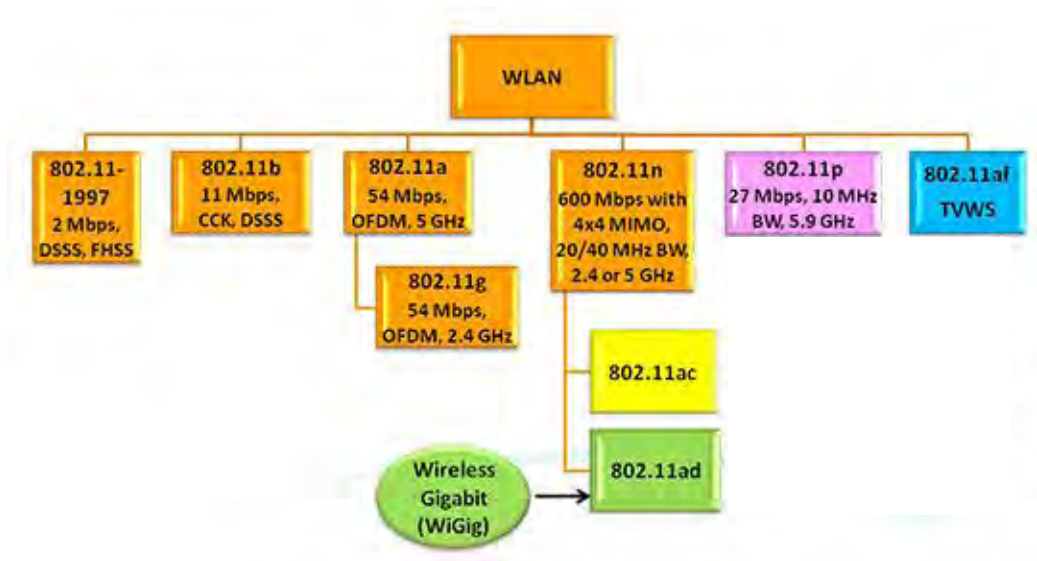


Figura 3.3 Evolución del protocolo IEEE 802.11.

3.1.1 La arquitectura de la red WLAN

La arquitectura básica de las WLAN se llama conjunto de servicio básico (BSS *basic service set*). Hay dos tipos de BSS: en modo infraestructura BSS y modo independiente BSS (IBSS *Independent BSS*). El primero se compone de un punto de acceso (AP), el cual trabaja como interfaz entre el enlace inalámbrico y varias estaciones asociadas al AP. El segundo se compone de varias estaciones donde cada una se comunica directamente con la otra, este tipo de BSS también se conoce como modo ad hoc. Una BSS se identifica mediante su correspondiente identificador BSS (BSSID *BSS identification*). El BSSID es la dirección MAC del AP en el caso del modo infraestructura, mientras que en el caso IBSS este se elige aleatoriamente por la estación[18].

El termino estación frecuentemente se puede remplazar por otros términos comúnmente utilizados como terminal inalámbrica, equipo final, nodo móvil entre otros.

3.1.2 Redes Ad hoc

Tal y como se presenta en la Figura 3.4, un par de estaciones en modo IBSS se comunican directamente punto a punto. La arquitectura IBSS asume que todas las estaciones se encuentran dentro del rango de transmisión, y por lo tanto, cualquier par de estaciones se puede comunicar directamente una con la otra. En realidad, dos estaciones pueden no estar dentro de su rango de comunicaciones, y tienen que depender de otra estación intermedia para enviar sus paquetes. El protocolo 802.11 no soporta transmisiones multisalto porque está fuera de su alcance, para ello se requiere de protocolos de enrutamiento [21][22].

A pesar de que una red en modo IBSS no involucra infraestructura, se puede conectar fácilmente con otra red externa como la de internet. Por ejemplo, si una estación en modo IBSS se conecta con el internet, y esta estación corre una funcionalidad de compartimiento de conexión a internet, el cual emplea un protocolo de enrutamiento, la estación puede trabajar como un *gateway* entre el 802.11 IBSS y el internet sin violar el concepto de sin infraestructura.

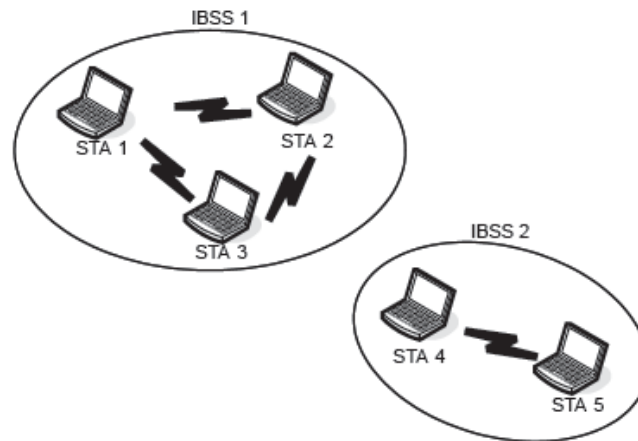


Figura 3.4 Ilustración de la arquitectura 802.11 IBSS.

Las tecnologías de radio, de acceso, y de enrutamiento completamente descentralizadas y habilitadas por: Bluetooth, el modo ad hoc IEEE 802.11, el modo sin estación PRnet, la red móvil ad hoc (MANET), y los conceptos tales como la red de área personal (PAN, *Personal Area Network*) o la comunicación PAN a PAN, se adaptan al dominio ad hoc [17].

Entre muchas de las aplicaciones que se puede tener para las redes ad hoc mencionaremos las siguientes:

1. **Aplicaciones Militares**, donde una configuración de red descentralizada es una ventaja operativa o incluso una necesidad.

Ejemplos:

- Operaciones de rescates en zonas remotas.

2. **En el sector comercial**, los equipos para informática inalámbrica móvil no han estado disponibles a un precio atractivo para los grandes mercados. A medida que aumenta la capacidad de dispositivos móviles de manera uniforme, también se espera que crezca la necesidad de formación de redes ilimitadas. Las redes comerciales ad hoc se podrían usar en situaciones allí donde no se dispone de ninguna infraestructura (fija o celular).

Ejemplos:

- Cuando la cobertura local debe ser desplegada rápidamente en un sitio de construcción.
- Acceso inalámbrico público en zonas urbanas.

3. **Al nivel local**, las redes ad hoc que enlazan dispositivos portátiles podrían ser usadas para difundir y compartir información entre los participantes de una conferencia.
4. **Aplicaciones de redes domesticas**, donde los dispositivos pueden comunicarse directamente para intercambiar información, tal como audio/video, alarmas, y actualizaciones de configuración.
5. **Aplicaciones de mayor alcance** en este contexto son las redes más o menos autónomas de robots domésticos interconectados que limpian, lavan los platos, cortan el césped, realizan vigilancia de seguridad, y otras labores parecidas.

Hay personas que han propuesto incluso redes de saltos múltiples ad hoc denominadas redes sensoras Figura 3.5

Ejemplo:

- Monitorización medio-ambiental donde las redes se podrían utilizar para predecir la contaminación del agua o para alertar con tiempo de la aproximación de un tsunami.

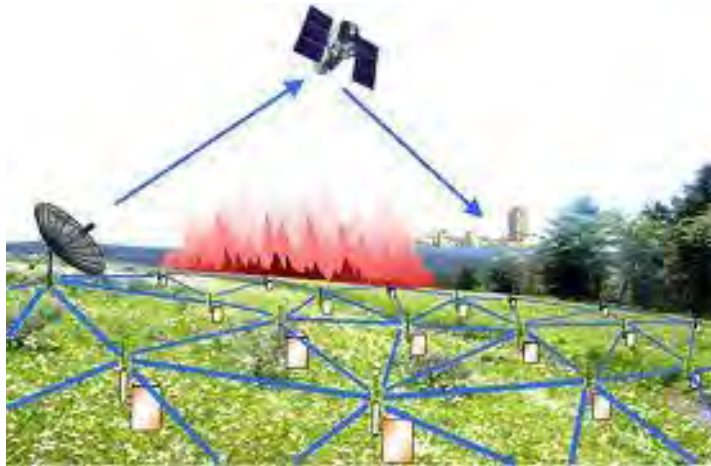


Figura 3.5 Red inalámbrica de sensores.

6. **Las redes ad hoc de corto alcance** pueden simplificar la intercomunicación entre diversos dispositivos móviles.

Ejemplo:

- Entre celular y una PDA

El sistema Bluetooth es tal vez la tecnología más prometedora en el contexto de formación de redes de área personal.

Las redes ad hoc tienen muchas ventajas por su modo sin infraestructura, sin embargo también presentan deficiencias debido a las siguientes razones.

1. **Seguridad.** La seguridad es un motivo de preocupación en una red ad hoc.

- ¿Cómo puede estar seguro un usuario de que nadie está curioseando en el tráfico mediante un nodo de reenvío?

-
- ¿Es el usuario del otro extremo realmente la persona que dice ser?

Los servicios ad hoc no implican muchos problemas nuevos. Los requisitos relativos a la autenticación, la confidencialidad, y la integridad son los mismos que para otras muchas redes de comunicaciones públicas.

En una red inalámbrica ad hoc, la confianza es un problema fundamental. No podemos confiar en el medio, la única elección que nos queda es usar la criptografía, lo que nos fuerza a confiar en las claves criptográficas usadas.

2. **Enrutamiento de redes ad hoc.** En las redes móviles ad hoc el enrutamiento de paquetes entre cualquier par de nodos llega a convertirse en una tarea comprometida, porque los nodos se pueden mover de manera aleatoria dentro de la red.

Un camino que se consideraba óptimo en un punto dado del tiempo podría no funcionar en absoluto unos pocos momentos después.

Para escenarios bajo techo el cierre de una puerta podría ser causa de que se interrumpiese un camino.

3. **Funciones de movilidad.** En una red ad hoc, pueden moverse los nodos del centro de la red o más bien, la totalidad de la red está basada en la idea del dispositivo que sirve al mismo tiempo tanto de enrutadores como de anfitriones.

En una red ad hoc, la movilidad es gestionada directamente por el algoritmo de enrutamiento.

3.1.3 Función de distribución controlada (DCF)

En el protocolo IEEE 802.11 existen dos funciones básicas para acceso al medio: la **Función de Coordinación Puntual** (PCF *Point Coordination Function*) y la **Función de Coordinación Distribuida** (DCF *Distributed Coordination Function*), las cuales se pueden observar en la Figura 3.6.



Figura 3.6 Diagrama de la arquitectura de la capa MAC 802.11.

El método de acceso al medio fundamental del protocolo 802.11 es el DCF [19], también conocido como Acceso Múltiple por Censado de Portadora con evasión de colisiones (CSMA/CA *carrier sense multiple access with collision avoidance*).

El DCF se puede implementar en todos los nodos para configuraciones de red en modo ad hoc ó en modo infraestructura. Mientras el DCF es el responsable para servicios de datos asíncronos, el PCF se desarrolló para servicios de tiempo limitado. El PCF se utiliza durante el periodo libre de contención (CFP *contention-free period*), mientras que el DCF se utiliza durante el periodo de contención (CP *contention period*).

El protocolo 802.11 utiliza tres diferentes brechas entre paquetes llamados **espacios inter-tramas** los cuales sirven para el control de acceso al medio, el espacio inter-trama corto (**SIFS** *short inter-frame space*), el espacio inter-trama PCF (**PIF**, *PCF inter-frame space*), y el espacio inter-trama DCF (**DIFS**, *DCF inter-frame space*) tal y como se muestra en la Figura 3.7.

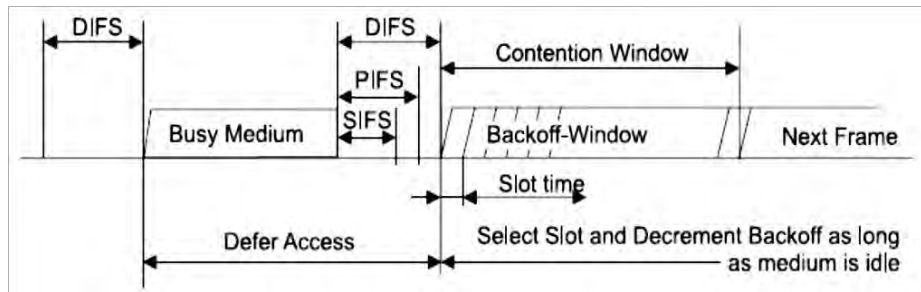


Figura 3.7 Tiempos de censado antes de transmitir un paquete en el protocolo 802.11.

En el modo DCF, los nodos contendrán unos con otros usando CSMA/CA para obtener acceso al medio. Antes de enviar una trama, la estación tiene que censar el canal, si el medio está en estado inactivo (*idle*) por al menos una duración DIFS entonces la estación puede enviar la trama. Si el canal censado está ocupado entonces la estación inicia el procedimiento de *back off*. En el caso de una transmisión exitosa, el procedimiento de *back off* se iniciará después de que un ACK se recibe.

Existen dos esquemas, uno llamado método de acceso básico (dos vías de conexión) y el esquema RTS/CTS (cuatro vías de conexión).

1) En el **primer esquema**, antes de iniciar una transmisión, la estación base escucha el canal para determinar si alguna otra estación está transmitiendo. Si el medio está inactivo durante un intervalo de tiempo que excede el DIFS, la estación continúa con su transmisión. Las estaciones activas guardan esta información en una variable local llamada NAV (*Network Allocation Vector*), de manera que el NAV informa cuál es el periodo de tiempo durante el cual el canal estará ocupado.

Después de cada recepción de paquete correcta, sin errores ni colisiones, el receptor espera un intervalo de tiempo SIFS, menor que el DIFS, para transmitir una trama de confirmación (ACK). Los errores son detectados mediante un algoritmo CRC (*Cyclic Redundancy Check*). Las colisiones entre transmisiones ocurren cuando dos o más estaciones comienzan a transmitir al mismo tiempo (ver Figura 3.8). Si no se recibe confirmación, se supone que la trama de datos se ha perdido y se procede a su retransmisión. Después de la detección de una trama errónea, el canal debe permanecer inactivo por lo menos durante un tiempo llamado espacio inter-trama extendido (EIFS, *Extended InterFrame Space*), antes de que las estaciones activen el algoritmo de *backoff* para reanudar sus transmisiones.

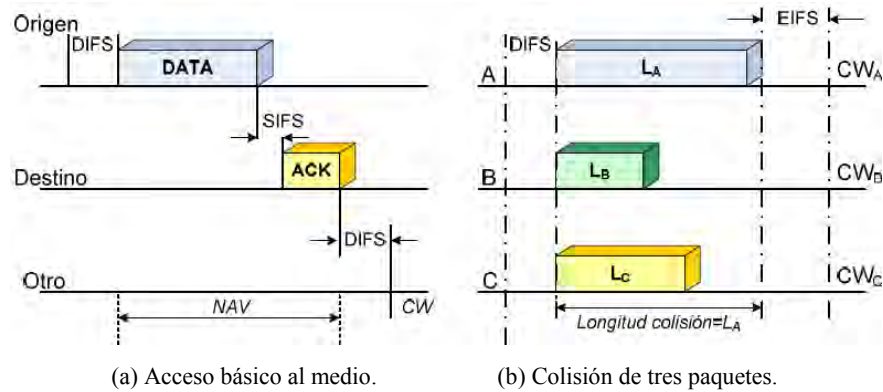


Figura 3.8 Funcionamiento del método de acceso DCF.

2) En el **segundo esquema** los nodos usan RTS/CTS antes de enviar datos y así pueden evitar el problema de la terminal oculta representado en la Figura 3.9.

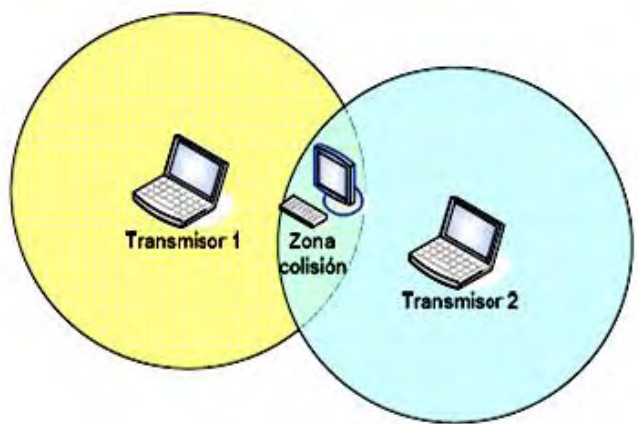


Figura 3.9 Zona de colisión por terminal oculta.

Al inicio de la transmisión se envía una solicitud para enviar (RTS, *Request to Send*), después de que el canal esta en modo inactivo al menos por un DIFS, cada paquete tiene un campo en el cual se especifica el tiempo que se requiere para su transmisión. Los demás nodos que escuchan el paquete RTS activan su vector de ubicación NAV para esta duración tal y como se presenta en la Figura 3.10. Esta duración especifica el intervalo de tiempo durante el cual el canal estará ocupado. Después que se recibe el RTS, el receptor espera una tiempo SIFS y entonces transmite el paquete listo para enviar (CTS *Clear to Send*). Cada nodo que recibe el paquete CTS activa su NAV de acuerdo a la duración contenida en el campo del paquete.

Con este mecanismo RTS/CTS se les informa a todos los nodos que se encuentran dentro del área de cobertura del transmisor y receptor acerca de la duración de transmisión del paquete que está siendo enviado para que no sea interrumpido. La principal desventaja del mecanismo RTS/CTS es el aumento de encabezados lo cual produce un desperdicio de ancho de banda y alto retardo[y].

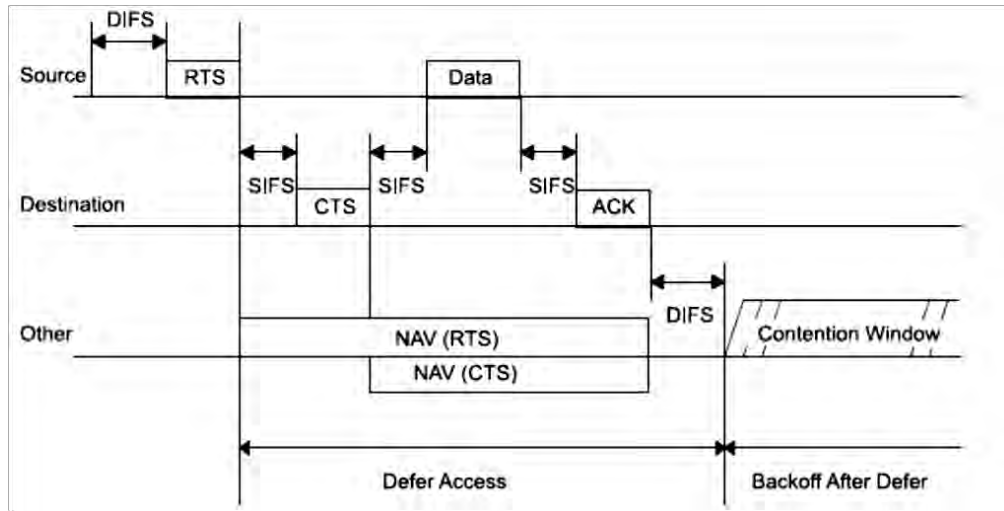


Figura 3.10 Protocolo RTS/CTS.

3.2 Protocolos de enrutamiento para redes MANET

Debido a que las redes móviles ad hoc (MANET) usualmente están limitadas en su ancho de banda, además de que el canal es muy propenso a errores por su variabilidad en el tiempo a consecuencia de la movilidad de los nodos, a la pérdida del enlace y al rompimiento de las rutas. Todas estas características representan un desafío para los esquemas de enrutamiento, especialmente en video *streaming* ya que este tipo de datos es muy sensible al retardo.

Por ejemplo:

- 1) Los **protocolos de enrutamiento proactivos** como el de vector distancia (**DSDV**, por su siglas en inglés *Destination Sequenced Distance Vector*) y el de estado de enlace (**OLSR**, por su

siglas en inglés *Optimized Link State Routing*) generan muchos encabezados en las redes MANET porque estos encuentran trayectorias para todas las fuentes y destinos, además de que periódicamente intercambian información de la topología de la red, inclusive cuando no hay datos transmitiéndose estos protocolos continúan en constante comunicación.

2) Por el contrario, los **protocolos de enrutamiento reactivos** como el de vector distancia sobre demanda para redes ad hoc (**AODV**, *Adhoc On-demand Distance Vector*) y el enrutamiento por fuente dinámica (**DSR**, *Dynamic Source Routing*) producen menos encabezados debido a que encuentran una ruta cuando se requiere[20].

A pesar que se han propuesto muchos protocolos de enrutamiento sobre demanda para redes MANET continúan los problemas de rompimiento de rutas cuando hay ambientes de alta movilidad, para esto se aplican dos aproximaciones para tratar de resolver el problema:

1) **Frecuentes actualizaciones** del enlace a lo largo de la ruta para avisar al transmisor y receptor acerca de la mala calidad del enlace.

2) **Habilitación de diversas rutas** descubriendo múltiples rutas entre el mismo par de fuente y destino.

3.2.1 Protocolo AOMDV

El protocolo de enrutamiento sobre demanda para multitr trayectorias en redes ad hoc (**AOMDV** *Adhoc On-demand multipath distance vector*) el cual es una extensión del protocolo AODV, descubre y mantiene más de una ruta entre un par de fuente y destino. Este protocolo logra un menor retardo punto-a-punto del paquete y más alto *throughput* que el protocolo AODV usando un esquema de rutas alternativas entre la fuente y el destino cuando la ruta principal falla. Sin embargo, cuando hay alta movilidad muchas rutas se rompen lo cual fuerza a la fuente a conmutar entre diferentes rutas disponibles o redescubrir nuevas rutas, esto incrementa la pérdida y retardo en los paquetes.

El protocolo AOMDV comparte varias características con el protocolo AODV, ambos se basan en el concepto del algoritmo vector distancia y utilizando enrutamiento de salto-por salto. Sin embargo AOMDV también encuentra rutas sobre demanda usando un procedimiento para el descubrimiento de rutas. La principal diferencia entre ambos protocolos se basa en el número de rutas alternas encontradas cuando un enlace se rompe.

En AOMDV, la propagación de paquetes RREQ (*Route Request*) de la fuente al destino establece múltiples rutas hacia los nodos intermedios en ambas direcciones (ver Figura 3.11). AOMDV también les proporciona rutas alternativas a los nodos intermedios así como se van encontrando nuevas trayectorias.

La base del protocolo AOMDV es asegurar que las múltiples trayectorias descubiertas estén libre de lazos y rotas, y que el algoritmo para descubrir rutas sea eficiente. Las reglas para la actualización de rutas se aplican localmente para cada nodo, esto juega un papel importante para no crear lazos infinitos ni encontrar trayectorias rotas[21][22].

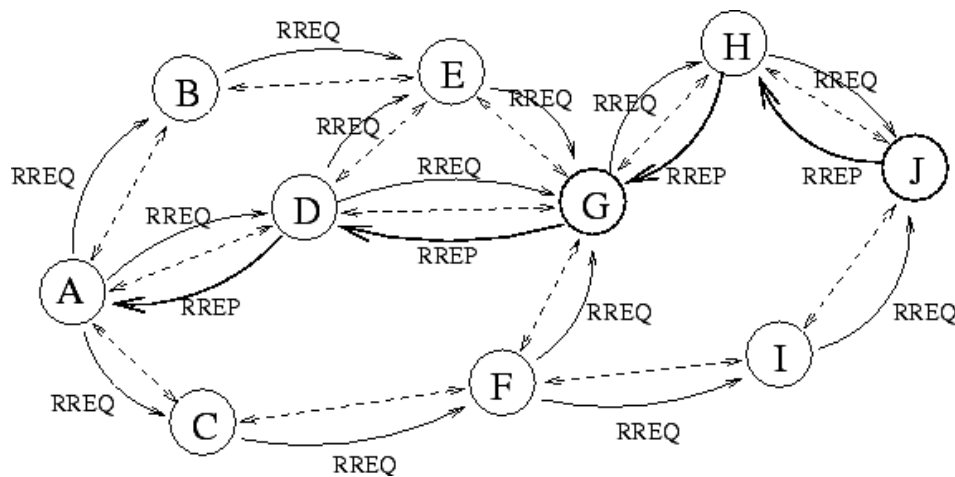


Figura 3.11 Enrutamiento utilizando el protocolo AOMDV.

Capítulo 4

Algoritmos para la administración de paquetes

En años recientes se han realizado una gran cantidad de trabajos para mejorar la calidad de video sobre redes inalámbricas [23]. Existen redes inalámbricas que proporcionan recursos dinámicamente de acuerdo a los requisitos de retardo, de ancho de banda y del tipo de aplicaciones multimedia.

Una de las claves para la entrega de datos multimedia sobre redes inalámbricas es tomar en cuenta las características dinámicas tanto del canal inalámbrico como de la fuente de datos. Para lograr este objetivo, se han investigado extensivamente la administración de paquetes y el manejo del buffer para maximizar la calidad de las aplicaciones multimedia.

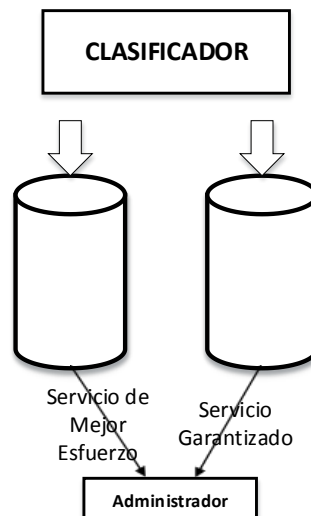


Figura 4.1 Administración de paquetes en la cola.

Los algoritmos de administración de paquetes son muy importantes para proveer servicios con calidad garantizada (ver Figura 4.1) ya que consideran parámetros como: el retardo, las

fluctuaciones (*jitter*), la pérdida de paquetes ó el *throughput*. El diseño de los algoritmos de administración de paquetes para redes móviles es un poco difícil debido a la alta variabilidad de las tasas de error y capacidad de los enlaces, y a los frecuentes cambios de conectividad por la movilidad de los usuarios.

Para muchas aplicaciones es importante que se obtenga cierta QoS a pesar de la eminente demanda por parte de los usuarios, por ejemplo, el internet está continuamente evolucionando y es muy utilizado por un gran número de aplicaciones móviles con diferentes QoS, por lo cual es necesario que la infraestructura de la red sea capaz de soportar distintos niveles de servicio. La Tabla 4.1 presenta algunas clases de servicios con sus respectivos requisitos de calidad como: nrt-VBR (*non-real time variable bit rate*), ABR (*available bit rate*), UBR (*unspecified bit rate*), CBR (*constant bit rate*) y rt-VBR (*real-time VBR*) [24].

Clase	Aplicación	Ancho de banda (b/s)	Retardo permitido (ms)	Tasa de pérdidas
CBR	Voz	32 k - 2 M	30 - 60	10^{-2}
nrt-VBR	Video digital	1 M - 10 M	200	10^{-6}
rt-VBR	Videoconferencia	128 k - 6 M	40 - 90	10^{-3}
UBR	Transferencia de archivos	1 M - 10 M	grande	10^{-8}
ABR	Buscador web	1 M - 10 M	grande	10^{-8}

Tabla 4.1 Requisitos de QoS para cada aplicación.

Tal y como se ilustra en la figura 4.2, la principal función del algoritmo para la administración de paquetes en el buffer es seleccionar los paquetes que vayan en la cabeza del buffer (*HOL Head-of-Line*) para ser los primeros en transmitirse. El proceso de selección de paquetes no es fácil debido a que no sólo se basa en el análisis de los requisitos de QoS de cada aplicación sino también se debe considerar que cada nodo puede transmitir paquetes de diferentes tipos de fuentes.

El administrador de paquetes se refiere al proceso de decisión utilizado para elegir qué paquetes se deben eliminar. El mecanismo de encolamiento del buffer se refiere a cualquier disciplina utilizada para regular la ocupación de una determinada cola.

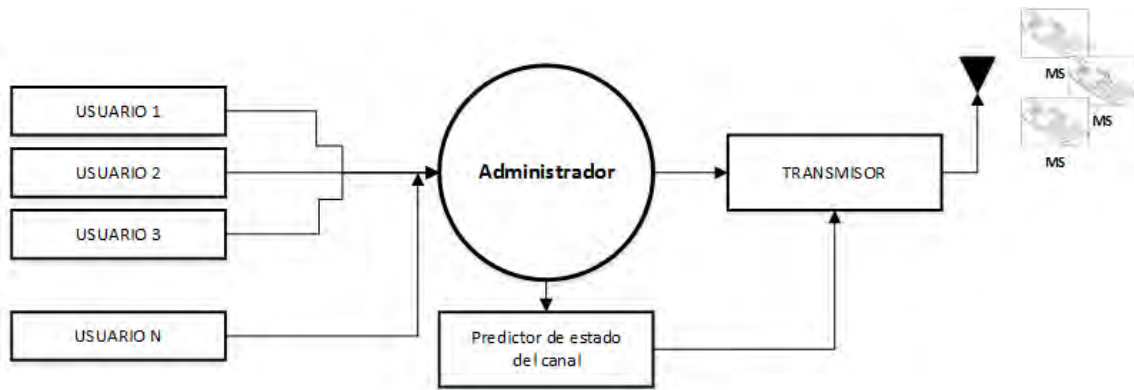


Figura 4.2 Un típico administrador de paquetes en redes inalámbricas.

4.1 El administrador de paquetes en redes inalámbricas

Los enlaces de las redes inalámbricas presentan particularidades que hacen más difícil el funcionamiento de los administradores de paquetes ya que están sujetos a la atenuación de la señal, a las interferencias, a la movilidad de los usuarios, al ruido y a la capacidad del canal. Un paquete de video sólo se recibe exitosamente si durante todo el tiempo de transmisión el enlace entre el transmisor y el receptor permanece en buenas condiciones y además que el paquete llegue dentro del umbral de retardo permitido.

A continuación se presentan algunas características que son deseables tomar en cuenta cuando se diseña un algoritmo para la administración de paquetes:

- 1) **Utilización eficiente del enlace.** Esto implica que el administrador no debe enviar paquetes cuando se ha detectado que el enlace está en malas condiciones porque sólo se desperdiciarán recursos.
- 2) **Umbral de retardo.** El algoritmo debe establecer cierto umbral de retardo para proporcionar garantía de servicio principalmente para aplicaciones sensibles al retardo.
- 3) **Equidad.** Esto significa que el algoritmo debe redistribuir equitativamente los recursos disponibles de la red entre todas sus aplicaciones principalmente cuando el canal es propenso a errores.

-
- 4) **Throughput.** El algoritmo debe garantizar cierto nivel de *throughput* dependiendo las condiciones del canal inalámbrico.
 - 5) **Complejidad de la Implementación.** Es muy importante que el algoritmo sea lo menos complejo posible para que pueda tomar decisiones rápidamente en situaciones de gran movilidad de los usuarios y en redes de alta velocidad.
 - 6) **Degradación del servicio gradual.** Cuando una transmisión que está en un enlace en buenas condiciones sufre una degradación debido a algún problema en el canal inalámbrico, esta degradación debe de ser lo menos brusca posible.
 - 7) **Aislamiento.** El algoritmo debe aislar aplicaciones de acuerdo a sus requisitos de QoS.
 - 8) **Separar la relación retardo-ancho de banda.** Muchas aplicaciones como la transferencia de archivos en internet tienen un amplio ancho de banda sin embargo pueden tolerar grandes retardos. No siempre el retardo y el ancho de banda están directamente relacionados.
 - 9) **Escalabilidad.** El algoritmo debe operar eficientemente sin importar el crecimiento de la red.

4.2 Clasificación de los algoritmos para la administración de paquetes

Los algoritmos para la administración de paquetes se pueden clasificar en **conservadores** (*work-conserving*) y **no conservadores** (*non-work-conserving*).

- 1) Un **administrador conservador** nunca se encuentra en modo inactivo si hay un paquete en espera de ser transmitido. Por mencionar algunos ejemplos de este tipo de administradores son: el GPS (*Generalized Processor Sharing*), el WFQ (*Weighted Fair Queuing*), el VC (Virtual Clock), el WRR (*Weighted Round-Robin*), el SCFQ (*Self-Clocked Fair Queuing*) y el DRR (*Deficit Round-Robin*).

2) Por el contrario, un **administrador no conservador** puede estar inactivo inclusive si hay un paquete en espera esto es debido a que puede estar esperando que otro paquete de mayor prioridad llegue al buffer. Algunos ejemplos de este tipo de administradores son: el HRR (*Hierarchical Round-Robin*), el SGQ (*Stop-and-Go Queuing*) y el Jitter-EDD (*Jitter-Earliest-Due-Date*).

Los administradores de tipo no conservadores por lo general presentan mayor retardo en los paquetes en comparación con los administradores conservadores, pero pueden ser mucho más eficientes para aplicaciones donde el *jitter* es mucho más importante que el retardo.

3) También hay **administradores clasificados por su *timestamp***, en este tipo de administradores cada vez que llega un nuevo paquete a la cola se le asigna un determinado tiempo antes de ser colocado dentro de esta. Los paquetes HOL son ordenados en orden ascendente de acuerdo a *timestamp*. La ventaja de estos algoritmos es que proporcionan mayor garantía de QoS que los algoritmos *round-robin*.

4) Existen también otro tipo de **administradores** donde los paquetes son **ordenados por su prioridad**, en este esquema, cada paquete tiene diferente nivel de prioridad dependiendo del tipo de aplicación o tipo de información. Algunos ejemplos son: el VC, el WFQ y el Jitter-EDD.

5) Otro tipo importante de **administradores** son los **basados en el tipo de trama**, es decir, en este tipo de algoritmos el tiempo se divide en tramas de tamaño fijo o variable. Para cada transmisión se reserva una parte de la trama para el envío de sus paquetes. Por ejemplo los algoritmos HRR y SGQ utilizan tramas de tamaño fijo, y los algoritmos WRR, DRR y FFQ (*Frame-Based Fair Queuing*) utilizan tramas de tamaño variable. Los administradores de tamaño de trama fijo son no conservadores porque pueden permanecer inactivos si una transmisión agota la cantidad de paquetes para enviar durante la porción de tiempo reservada.

Para esta tesis proponemos un algoritmo que mezcla las características del administrador por *timestamp* y del administrador por prioridad tal y como se mencionaron en los puntos 3 y 4 de arriba.

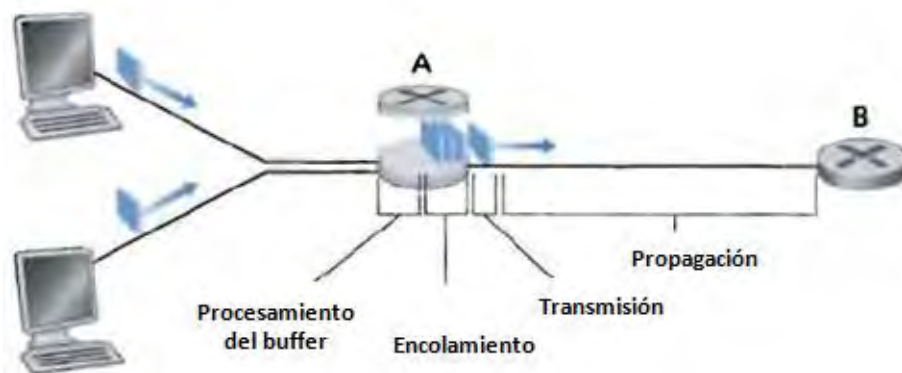
4.3 Trabajos relacionados

Los mecanismos de encolamiento más ampliamente utilizados se conocen como DT (*Drop Tail*) y RED (*Random Early Drop*).

1) **Drop Tail (DT):** En este esquema cada paquete es tratado con igual importancia. Una vez que la cola esté llena, los paquetes que están llegando se eliminan hasta que la cola se libera. El problema que presenta este esquema es que existen periodos alternativos entre una cola llena y una cola vacía lo cual causa pérdida de paquetes en ráfagas, lo cual no es deseable en aplicaciones de video (ver Figura 4.3).

2) **Random Early Detection (RED):** En este método el buffer detecta que se está originando congestión en la red y se adapta para eliminar paquetes de acuerdo a probabilidades estadísticas.

Si el buffer esta casi vacío todos los paquetes entrantes se aceptan. Así como la cola va creciendo, la probabilidad de eliminar paquetes entrantes también crece. Cuando el buffer está lleno, la probabilidad crece a 1 y todos los paquetes entrantes se eliminan. Este método es más justo que el DT, en el sentido que no tiene problemas de pérdida en ráfagas. El problema de este método es cuando existe una sola cola.



Retardo en el buffer del router A

Figura 4.3 Sistema de encolamiento básico.

Los métodos básicos de encolamiento discutidos arriba pueden mejorar el *throughput* del video que se transmite. Sin embargo la QoE del usuario no está directamente relacionada con el *throughput* obtenido. Como se discute en las secciones previas, esto principalmente depende del tipo de trama (I, P o B) de donde el paquete obtiene su prioridad. En diversas investigaciones se han propuesto algoritmos para el manejo del buffer en redes inalámbricas.

Debido a la alta demanda de servicios multimedia y a la alta calidad de servicio requerida por los usuarios es necesario un esquema de administración del buffer más eficiente e inteligente donde los paquetes pueden ser reordenados para su envío. En esta sección se discutirá más acerca de los diferentes esquemas propuestos.

Seong [25] propone un esquema de streaming de video el cual permite a las aplicaciones asignar diferentes prioridades y usar un mecanismo para el control de congestión de multi-colas dentro del enrutador para eliminar efectivamente los paquetes de menor importancia cuando el buffer está lleno.

En [26] el autor propone un esquema de administración del buffer llamado FDDT (*Frame-Level Packet Discard with Dynamic Thresholds*), en el cual los paquetes son ordenados y eliminados de acuerdo a las siguientes condiciones:

- 1) El primer paquete de tipo B o P se descarta cuando el tamaño del buffer alcanza cierto nivel de umbral.
- 2) Un paquete tipo I solo se descarta cuando el búfer está completamente lleno.
- 3) Si el primer paquete de la trama se elimina, entonces por consecuencia se eliminarán todos los paquetes subsecuentes.
- 4) Todos los paquetes entrantes se eliminan cuando el buffer está lleno. El esquema FDDT obtuvo resultados con buena calidad de video y solo usando un pequeño incremento en complejidad computacional.

Shujie Wei [27] propuso un esquema activo para la administración de la cola basado en la estimación del ancho de banda para multimedia *streaming* en tiempo real sobre redes MANET.

El modelo GM [28] se utilizó para predecir el ancho de banda de la red para una determinada longitud de cola, la cual fue ajustada dinámicamente.

En [29], se propuso un esquema llamado DDB (*Drop Dependency Based*) donde la información básica de la prioridad del paquete se proporcionar en el encabezado del paquete y el buffer utiliza esta información para la administración de paquetes. Una estrategia optimizada opera sobre los paquetes HOL más antiguos en el buffer de un grupo de paquetes. En este esquema se elimina el paquete HOL con más baja prioridad con esto se consigue una importante mejora en la calidad del video. Este esquema fue extendido en [30] para lograr una óptima combinación de administración y estrategia de eliminar paquetes.

También existen otras implementaciones basadas en enrutadores inteligentes tal como Redes Activas [31]. En el cual los enrutadores aplican una muy importante regla de descartamiento inteligente de paquetes basada en su conocimiento previo de la transmisión en progreso. Sin embargo hay que tomar en cuenta que la implementación de complejidad computacional en los enrutadores puede dificultar el envío de paquetes al siguiente nodo. Pero esta característica puede ser de gran ventaja en redes MANET donde la información el video de la comunicación actual se puede transmitir en el encabezado durante la sesión establecida.

En [32] el autor introduce un nuevo termino llamado justicia de servicio el cual significa que se realiza una asignación justa de los recursos disponibles de la red para lograr la calidad esperada, con esto se logra que más recursos sean asignados para paquetes con mas alto requisito de QoS. Los paquetes con más bajo requisito de QoS se eliminan mientras se da más acceso a paquetes importantes (ver Figura 4.4).



Figura 4.4 Sistema de encolamiento con paquetes de alta prioridad.

Van der Schaar [33] propone un esquema de capas cruzadas entre la capa de aplicación y la capa MAC utilizando video MPEG-4 y su escalabilidad fina granulada FGS. En este trabajo los paquetes que contienen datos multimedia son clasificados en diferentes clases, y en malas condiciones de la red solo los paquetes con mas alta clase son transmitidos. Las condiciones de la red son medidas conjuntamente combinando la información que se obtiene por la retransmisión, el numero de tramas MAC (ARQ) perdidas y la información proporcionada por el protocolo RTCP.

Los autores en [34] proponen un esquema de retransmisión selectiva para datos multimedia sobre redes inalámbricas. La idea es retransmitir solo la información importante del video para lograr alta calidad en el streaming de video. A pesar de que la retransmisión de paquetes importantes es buena idea, es mejor prevenir por lo cual se requiere la protección de información de alta prioridad y reducir la tasa de paquetes perdidos (PLR, Packet Loss Rate).

Todos los esquemas discutidos arriba, excepto [25] y [26], en su mayoría discuten la QoS del video priorizando diferentes flujos o administrando los paquetes de más alta prioridad. Pero hay muy poca investigación realizada para redes MANET considerando la prioridad relativa de los paquetes de video. Esto puede jugar un papel importante para determinar la calidad del video para el usuario. Por lo tanto, basándonos en la revisión previa de administración de paquetes en el buffer, se propone un nuevo esquema el cual se considera la prioridad relativa de los paquetes de video en aplicaciones de tiempo real sobre redes MANET.

Capítulo 5

Entorno de evaluación

El funcionamiento de los algoritmos propuestos en esta tesis para la priorización de paquetes de video H.264/AVC en el buffer de una red inalámbrica ad hoc han sido evaluados a través de diversas simulaciones en el software libre **ns-2.34**.

En este capítulo se abordan aspectos relacionados con el modelado y la simulación de una red de comunicaciones inalámbricas ad hoc, como herramienta para la comprobación del funcionamiento y del rendimiento de los algoritmos implementados en el presente trabajo. La creciente disponibilidad de computadoras con alta capacidad de procesamiento hacen de las técnicas de simulación una atractiva alternativa para la validación experimental de resultados teóricos en distintas áreas de investigación. Por ello, en el área de las redes de telecomunicaciones, resulta usual recurrir a este tipo de técnicas, dada la inviabilidad económica de la implementación real de grandes redes destinadas únicamente a fines de investigación.

Mediante la simulación, es posible hacer pruebas a gran escala y lo suficientemente exactas en sus resultados para validar o desechar algunas opciones de diseño de protocolos. Todo ello hace aconsejable la utilización de simuladores de redes como herramienta de validación de las propuestas de investigación. Por lo tanto, los experimentos del presente trabajo se han llevado a cabo mediante simulación en el ns-2.34.

La plataforma de simulación empleada en esta tesis está basada en ns-2 versión 2.34 con algunas mejoras adicionales, y actualizaciones proporcionadas por otros autores.

5.1 Descripción y funcionalidades del ns-2.34

El Network Simulator 2 (mejor conocido como ns-2) fue desarrollado por el *Information Sciences Institute de la University of Southern California*. Es una herramienta versátil para la investigación en redes de comunicaciones que puede ser configurada para simular un amplio rango de tecnologías de redes y distintos protocolos de comunicación:

- Tecnologías de red tanto cableadas como inalámbricas.
- Protocolos de transporte tales como TCP y UDP.
- Generación de comportamientos de tráfico FTP, Telnet, Web, CBR y VBR.
- Simulación de mecanismos de gestión de colas en *routers* tipo Drop Tail y RED.
- Soporte para diversos tipos de algoritmos de enrutamiento como AODV y AOMDV.
- Modelos de movilidad para simulación de redes inalámbricas que hacen posible por ejemplo la simulación de redes ad hoc.

En la Tabla 5.1 se muestran algunos de los protocolos de comunicación y los modelos de tecnologías de comunicación incluidos en la versión de ns-2.34 utilizada en esta tesis. Además de los módulos ya incluidos en la distribución oficial, están disponibles otros adicionales como código contribuido [35], cuyo contenido es mantenido por usuarios y puede ser utilizado junto con la distribución del oficial.

Capa	Protocolos y módulos
Aplicación	Ping, telnet, FTP, multicast FTP, HTTP, webcache
Transporte	TCP (distintas variaciones), UDP, SCTP, XCP, TFRC, RAP, RTP Multicast: PGM, SRM, RLM, PLM
Red	Unicast: IP, MobileIP, vector de distancia y estado de enlace, IPinIP, enrutamiento en origen, Nixvector. Multicast: SRM, centralizado MANET: AODV, AOMDV, DSR, OLSR, DSDV, TORA, IMEP
Enlace	ARP, HDLC, GAF MPLS, LDP, Diffserv Administración de colas: DropTail, RED, RIO, WFQ, SRR, Semantic Packet Queue, REM, Priority, VQ. MAC: CSMA, 802.11, 802.11.1 (WPAN), satellite Aloha
Física	Modelos propagacion radio: TwoRay, Shadowing, Nakagami, OmniAntennas, EnergyModel, Satellite Repeater

Tabla 5.1 Protocolos implementados en el ns-2.34.

5.1.1 Arquitectura básica del ns-2.34

El ns-2.34 es un simulador de eventos discretos que opera a nivel de paquete, desde el nivel de enlace hacia capas superiores y permite simular redes cableadas e inalámbricas. Está disponible para casi todas las plataformas basadas en UNIX (FreeBSD, Linux, Sun Solaris) y también puede ejecutarse en Windows (con el emulador cygwin).

El ns-2.34 también es un simulador orientado a objetos, escrito a la vez en C++ y en lenguaje OTcl, el cual es una evolución del Tcl que permite definir clases y crear objetos de esas clases, la parte de OTcl se emplea como interfaz hacia el usuario. La estructura del ns-2.34 se basa en una jerarquía de clases en C++ (llamada también jerarquía compilada) y una jerarquía similar de clases dentro del intérprete OTcl (llamada también jerarquía interpretada), ambas están estrechamente relacionadas tal y como lo muestra la Figura 5.1. Desde el punto de vista del usuario, existe una correspondencia exacta uno a uno entre una clase en la jerarquía interpretada y en la jerarquía compilada.

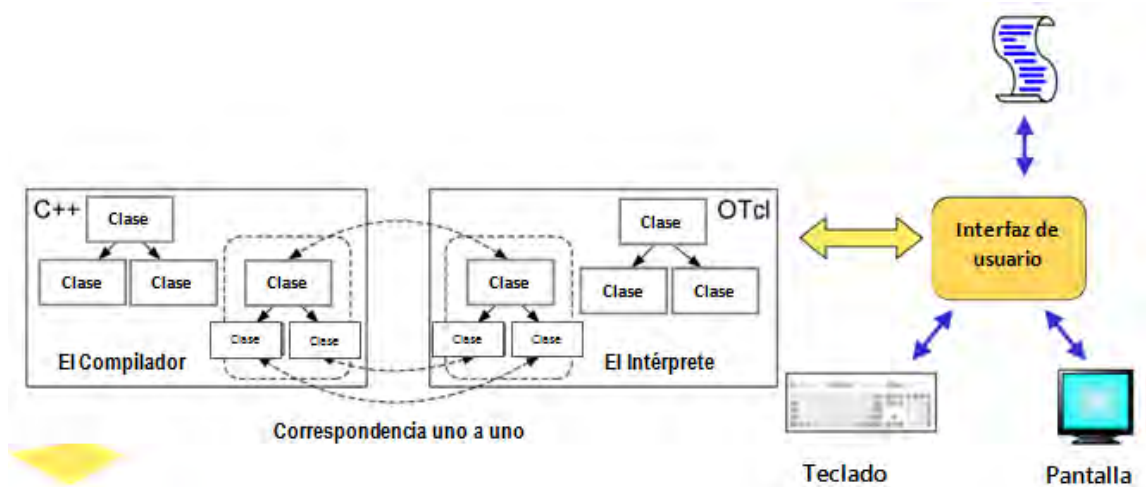


Figura 5.1 Relación entre los lenguajes C++ y OTcl en el ns-2.34 [36].

El fundamento de esta jerarquía es la clase TclObject. Los usuarios crean nuevos objetos a través del intérprete; dichos objetos son instanciados dentro del intérprete y se reflejan en la creación de un objeto equivalente en la jerarquía compilada. Además, existen algunas jerarquías dentro del código C++ y de los scripts de OTcl que no están enlazadas, es decir, no tienen correspondencia en el otro lenguaje. OTcl se comporta como una interfaz hacia el usuario para tareas de configuración, con soporte de objetos, mientras que Tcl funciona como un enlace entre C++ y OTcl.

La razón de esta implementación basada en diversos lenguajes se debe a la doble orientación de la herramienta [35]. Por un lado, está orientado a la simulación detallada de protocolos, lo cual requiere un lenguaje de programación de sistemas que maneje eficientemente grandes conjuntos de datos, bytes, encabezados, etc. Para estas tareas, el tiempo de ejecución es más importante que el tiempo de preparación. Por otro lado, en muchos casos la investigación de sistemas y redes de comunicación consiste en la simulación de escenarios en los que varían ligeramente la configuración o los valores de ciertos parámetros. En estos casos, es más importante el tiempo de iteración, es decir, el que se emplea en cambiar el modelo y volver a lanzar las simulaciones.

Lo ideal en este caso es poder lanzar simulaciones desde un script en el que automáticamente se cambian los parámetros de los escenarios de acuerdo a los aspectos que se quieren evaluar. Con los dos lenguajes de ns-2.34 se llega a un compromiso entre ambos aspectos: C++ es más rápido en ejecución pero de modificación más lenta, mientras que OTcl puede ser modificado interactivamente y controlado a través de scripts, aunque su ejecución sea comparativamente lenta. La unión entre ambos se realiza a través de tclcl [35].

Los elementos que forman el simulador ns-2.34 pueden dividirse en tres categorías:

- La librería de componentes de red (Network Component).
- La librería de interconexión de objetos (tcl / otcl / tclcl).
- El administrador de eventos (scheduler).

Los componentes de red constituyen los elementos que forman la red, tal como nodos, enlaces o colas. Su interconexión está definida en las librerías de interconexión donde se define el camino que seguirá un paquete en la simulación. Estos componentes de red pueden ser simples, creados directamente a partir de su clase de C++, o compuestos, formados por la interconexión de múltiples componentes simples. En general, todos los componentes de red en ns-2.34 se crean, se conectan y se configuran desde el código OTcl.

El administrador de eventos se encarga de gestionar el orden en el que se ejecutan los eventos de la simulación. Un evento en ns-2.34 viene definido por un instante de tiempo, un identificador único, un puntero al siguiente evento que debe ejecutarse, y un *handler* o manipulador, que apunta al objeto que deberá emprender el evento en el instante de tiempo indicado. Los eventos

se colocan en una cola ordenada por tiempo y se ejecutan uno a uno bajo la gestión del administrador de eventos.

Todos los componentes de red son una sub-clase de la clase *Handler* (administrador de eventos), puesto que requieren ejecutar eventos como la entrega o recepción de paquetes, etc.

5.2 Implementación de sistemas de comunicaciones inalámbricos en ns-2.34

La simulación de redes inalámbricas de nodos móviles se lleva a cabo gracias al modelo inalámbrico del ns-2.34, originalmente importado como una extensión de movilidad desarrollada por *CMU's Monarch Group*. El modelo CMU original permitía simulaciones de redes WLAN así como redes ad hoc multi-salto. Posteriormente, fueron introducidas extensiones para poder simular redes cableadas e inalámbricas combinadas. Además, fue incorporada una extensión para *MobileIP*.

La implementación de la parte inalámbrica de la red en ns-2.34 se compone principalmente de los tres bloques que se muestran en la Figura 5.2: nodo móvil, canal inalámbrico, antenas y propagación. Esta implementación sigue un enfoque por capas en la que los módulos están interconectados para el intercambio de paquetes.

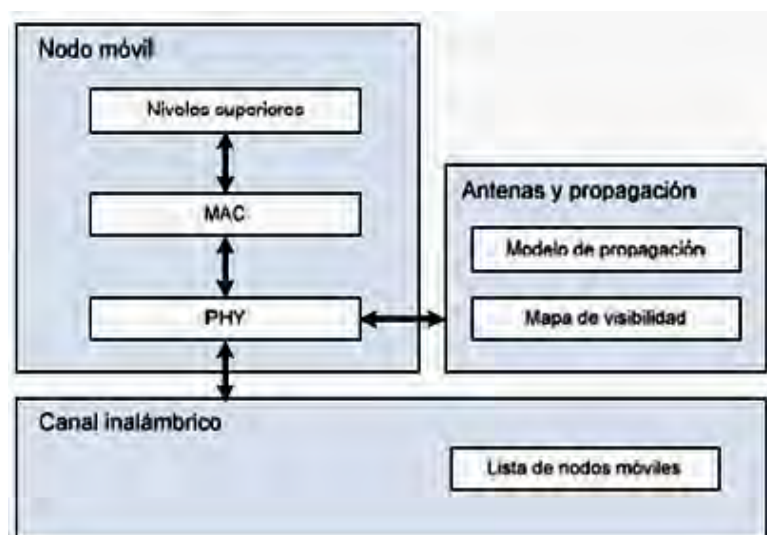


Figura 5.2 Módulo de Nodo móvil, Canal inalámbrico, Antenas y propagación en el ns-2.34.

1) **Canal inalámbrico:** este bloque interconecta todos los nodos móviles que operan en el mismo canal, como se ilustra en la Figura 5.3.

Cuando un nodo móvil transmite un paquete a través del canal inalámbrico, este bloque crea una copia del paquete en la interfaz PHY de cada nodo móvil, con un retardo de propagación proporcional a su distancia al transmisor. Cada nodo a continuación, es programado para comenzar a recibir la copia del paquete en un instante diferente, teniendo en cuenta los retrasos experimentados por los diferentes nodos a diferentes distancias desde el transmisor.

En recepción, cada nodo será el responsable de determinar si recibe correctamente el paquete o no. La decisión de la correcta recepción viene marcada por la relación señal a ruido e interferencia (SINR, *Signal to Interference and Noise Ratio*) con la que es recibido el paquete. A la hora de entregar los paquetes, se ordenan en función de la distancia al nodo transmisor. Así, el nodo más cercano será el primero en recibir el paquete a través de su interfaz física, simulando, de este modo, el retardo producido por la propagación real del paquete.

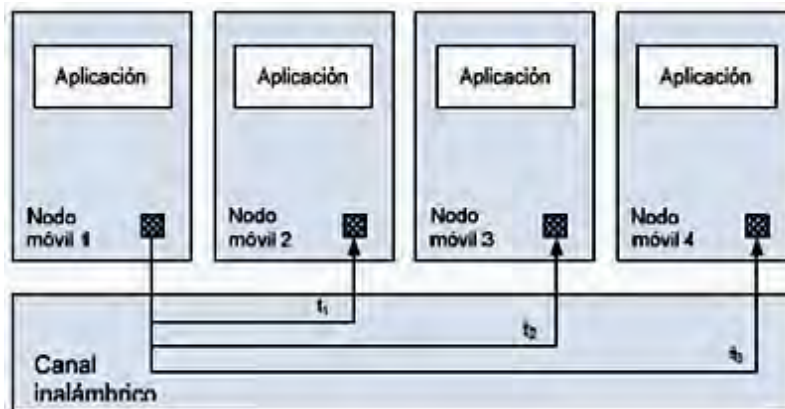


Figura 5.3 Funcionamiento del canal inalámbrico en el ns-2.34.

2) **Nodo móvil:** este bloque implementa las diferentes capas de comunicación disponibles en un nodo inalámbrico, junto con funciones de movilidad para actualizar la posición de los nodos cada vez que se recibe un nuevo paquete.

Los módulos más importantes del nodo móvil en esta tesis son PHY y MAC. La Figura 5.4 muestra el diagrama de bloques de un nodo móvil asociado a un canal radio. El objeto que implementa el nodo móvil se deriva del objeto nodo genérico, añadiendo principalmente la

funcionalidad de movilidad y permitiendo su asociación al canal radio. Los distintos componentes de red que implementa la clase nodo móvil (*class mobilenode*) son:

- PHY (interfaz física): encargada de recibir los paquetes procedentes del canal y pasárselos a la capa MAC en caso de recibirlos correctamente. En el presente trabajo se ha empleado la interfaz física del estándar 802.11a.
- MAC (control de acceso al medio): implementa el protocolo MAC de IEEE 802.11a para el caso de la función de coordinación distribuida (DCF).
- IFQ: cola de salida de los mensajes.
- LL: capa de enlace asociada al bloque ARP (*Address Resolution Protocol*). Se encarga de la búsqueda de la dirección MAC del nodo destino a partir de su dirección IP. Esta capa es la encargada de pasar en sentido descendente los paquetes a la cola IFQ y, en sentido ascendente, los pasa directamente a la capa de ruteo.
- Agente de ruteo: encargado de dar soporte al sistema para que puedan establecerse comunicaciones multi-salto entre distintos nodos. El agente de ruteo proporciona las funcionalidades necesarias para llevar a cabo el enrutamiento de paquetes de datos a cualquier nodo de la red. Actualmente, la versión de ns-2 implementa los protocolos DSDV, DSR, TORA y AODV, protocolos generalmente utilizados en redes MANET.

En la Figura 5.4 también pueden observarse los componentes que la clase nodo móvil hereda de la clase nodo genérico. Estos componentes consisten en un objeto de entrada, un clasificador de direcciones, y un clasificador de puertos a los que se asocian agentes y aplicaciones. Los objetos Agente y Aplicación han de conectarse a los nodos para la generación, procesado y recepción del tráfico de datos. Mientras que los agentes representan puntos terminales en la red, las aplicaciones son objetos que generan la información y se encuentran por encima de los agentes en la pila de protocolos de comunicación.

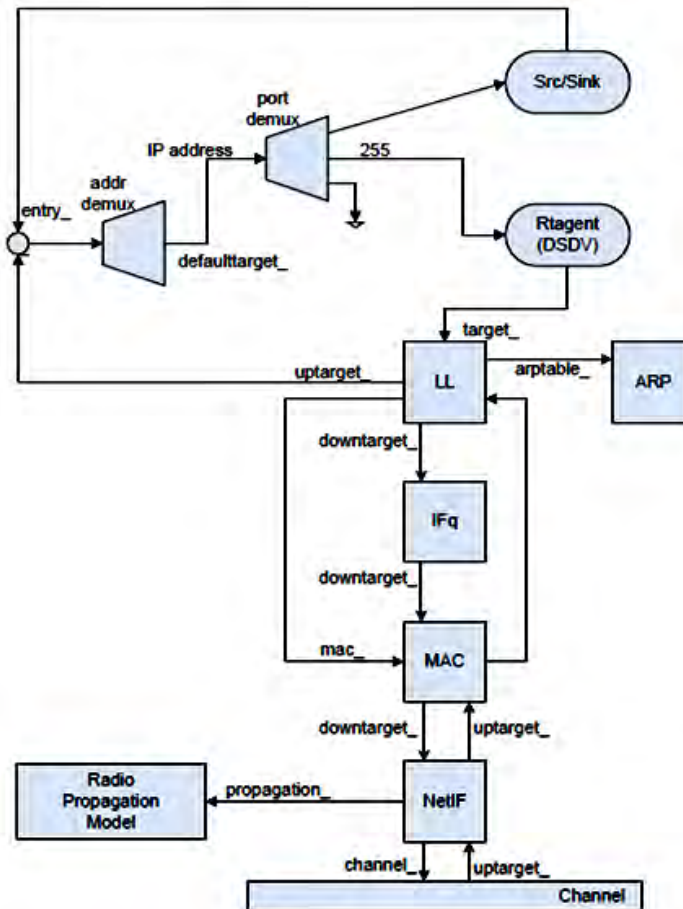


Figura 5.4 Diagrama de bloques de un nodo móvil asociado a un canal inalámbrico.

3) **Antenas y propagación:** La capa PHY hace uso del módulo de antenas y propagación para determinar el nivel de potencia recibida de cada paquete, que calcula en función del modelo de propagación considerado. El nivel de detalle del modelo de propagación radio es un aspecto importante debido a que puede afectar considerablemente a los resultados y conclusiones del estudio. En general, los modelos de propagación de radio para investigaciones a nivel de sistema deben reflejar adecuadamente los efectos de la pérdida básica de propagación (*PL Path Loss*), desvanecimiento lento (*SH SHadowing*) y desvanecimiento por multitrayecto (*MP Multi Path fading*) [148]. La pérdida básica de propagación representa la media local de la potencia de la señal recibida con respecto a la potencia de transmisión como una función de la distancia entre el emisor y el receptor. El desvanecimiento lento modela el efecto de los obstáculos próximos al receptor en la potencia media de la señal, a una determinada distancia. El desvanecimiento multitrayecto es consecuencia de la recepción de varias replicas de la señal transmitida en el receptor. Teniendo en cuenta la potencia de transmisión (P_t), y los efectos de propagación, la

potencia de la señal recibida (P_r), puede calcularse en dB, usando la siguiente ecuación, que considera ganancias unitarias de antena y con pérdidas despreciables de circuito:

$$P_r = P_t - PL - SH - MP \quad (5.2)$$

Los distintos modelos de propagación radio proporcionan expresiones para calcular valores de PL, SH y MP en función de distintos parámetros.

5.3 Esquemas de encolamiento en ns-2.34

Las colas representan ubicaciones donde los paquetes pueden ser almacenados o eliminados hasta que esté disponible una ruta para un específico destino o el canal esté disponible para enviar datos. La administración de paquetes se refiere al proceso de decisión utilizado para elegir que paquetes deben ser servidos o eliminados. Los algoritmos para el manejo del buffer se utilizan para regular la ocupación de una cola.

El ns-2.34 soporta los siguientes tipos de encolamiento; FIFO (*Drop Tail*), RED manejo del buffer, CBQ (incluyendo una prioridad y administración *round-robin*), y variantes de Fair Queuing como: el FQ, el SFQ, y el DRR[35].

El ns-2.34 también implementa la **PriQueue** como una cola que le da prioridad a los paquetes del protocolo de enrutamiento, insertándolos en la cabeza de la cola. La PriQueue contiene un filtro el cual extrae los paquetes con una dirección de destino específica, lo cual asegura que los paquetes enrutados alcancen el destino en el más mínimo tiempo.

El ns-2.34 incorpora el esquema de encolamiento DropTail/FIFO para el manejo del buffer en caso de desbordamiento, el cual es el más utilizado en los enrutadores para internet. Esta lógica se implementa en la siguiente arquitectura del ns-2.34[35].

5.3.1 La interfaz de la Cola

La IFQ es la cola que implementa el mecanismo FIFO/DropTail, como se presenta en la Figura 5.4 la cual actúa como una interfaz entre la capa de enlace y la capa MAC. Los paquetes que vienen desde la capa de enlace se encolan en la IFQ y cuando la capa MAC avisa que el canal está disponible entonces los paquetes son transmitidos [35].

La interfaz de la cola solo acepta dos niveles de prioridad :

- 1) A los **paquetes de control** se les asigna la más alta prioridad, es decir que cuando un paquete de control es recibido es colocado inmediatamente en la cabeza de la cola.
- 2) Todos los **paquetes de datos** en nuestro caso los paquetes de video se les asigna más baja prioridad en comparación con los paquetes de control.

Los paquetes de video dentro de la cola son tratados con la misma prioridad. Justamente la idea de esta tesis es clasificar los paquetes de video en cuatro prioridades, siendo los paquetes de más alta prioridad los que se coloquen al inicio de la cola para ser transmitidos lo más pronto posible.

5.4 El Evalvid como interfaz entre el codec y el ns-2.34

La operación en general del Evalvid es como se presenta en la Figura 5.5. En los siguientes pasos se describirá secuencialmente como están relacionados el Evalvid, el H.264/AVC y el ns-2 para fines de simulación de transmisión de video en tiempo real [37].

1. Codificar el video original con formato YUV a formato H.264/AVC usando el JM.
2. Obtener el archivo de traza del H.64/AVC para su análisis. Este archivo contiene el ID del paquete, el tiempo de envío (*timestamp*), el tamaño del paquete y su prioridad. Toda esta información será utilizada por el ns-2.
3. Durante la simulación, el ns-2 genera otros dos archivos de traza, uno del transmisor y otro del receptor. El archivo de traza del transmisor contiene el ID del paquete, el tiempo de envío (*timestamp*), el tamaño del paquete y su prioridad. El archivo de traza del receptor contiene la misma información, la única diferencia es que este contiene el tiempo de recibo del paquete.
4. Calcular la tasa de pérdidas de paquetes, el retardo, el *jitter* y el *throughput*.
5. Reconstruir el video recibido con todos sus errores.
6. Decodificar el video H.264/AVC y volverlo a formato YUV de nuevo usando el JM.
7. Comparar la calidad de ambos video el original y el recibido en términos del PSNR.

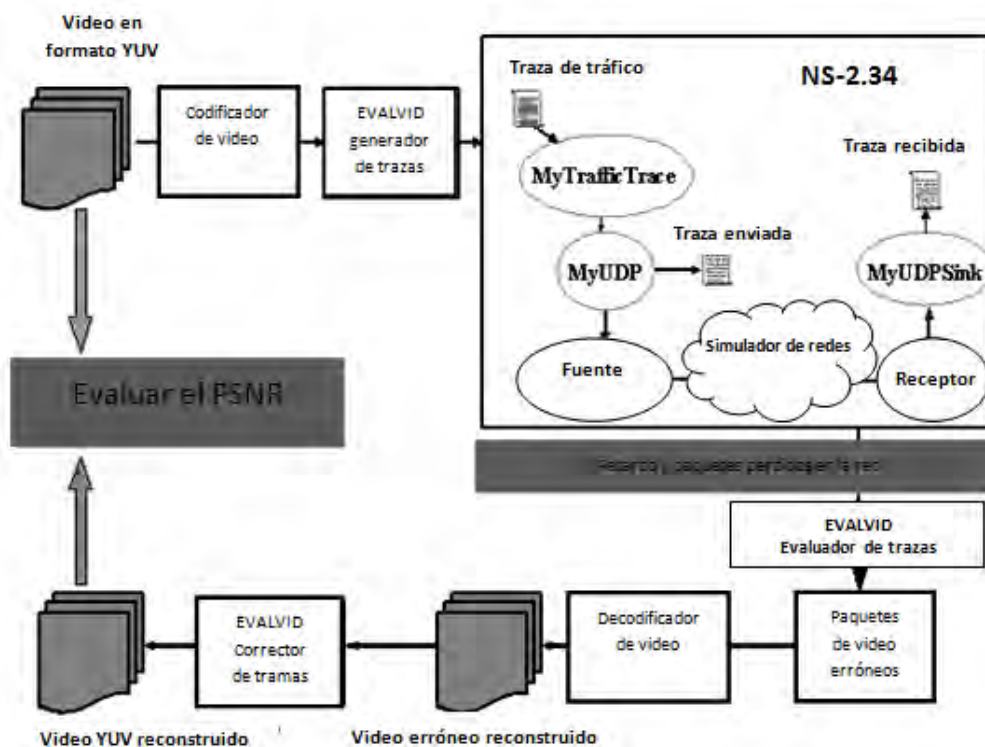


Figura 5.5 Diagrama a bloques que presenta la relación entre el Evalvid, el H.264/AVC y ns-2.34.

5.5 Priorización de paquetes de video H.264/AVC

En esta tesis se utilizan los paquetes de video H.264/AVC que están priorizados para ser transmitidos sobre la red MANET, donde básicamente los paquetes de más alta prioridad serán los primeros en transmitirse cuando haya acceso al medio que los paquetes de baja prioridad.

Para fines de simplicidad y de análisis, cada paquete contiene un sólo *slice*, por lo tanto la prioridad del *slice* es justamente la prioridad asignada al paquete que será transmitido.

La prioridad de los slices de video se asigna en base al error medio cuadrado acumulativo (CMSE *Cumulative Mean Square Error*) aportado por su pérdida, es decir que de acuerdo al impacto que tiene la pérdida de cada uno de los slices sobre el video en general, es como se le asigna la prioridad. Para este trabajo son asignadas cuatro prioridades, siendo la prioridad más importante la 1 y así sucesivamente hasta llegar a la prioridad menos importante que es la 4.

En la mayoría de los casos se cumple que los paquetes de prioridad 1 son por lo regular *slices* tipo I, sin embargo también podemos tener slices tipo I pero con prioridad 2. Esto es de acuerdo a los resultados obtenidos del CMSE del video decodificado.

Cada archivo de video H.264/AVC es repartido equitativamente entre las 4 prioridades, por lo tanto tenemos la misma cantidad de slices con prioridad: P1, P2, P3 y P4.

5.5.1 Asignación de prioridad en el ns-2.34

En la capa de aplicación los paquetes de video H.264/AVC se adjuntan al agente en ns-2.34 por medio de Evalvid [41] (ver Figura 5.5), el cual actúa como traductor entre el codificador y el ns-2. Con esta herramienta los paquetes se generan en base al archivo de traza del H.264/AVC, como se explicó previamente se le asigna prioridad a los slices de acuerdo al CMSE.

Recordando que un paquete está formado de un *slice*, la capa de aplicación crea un paquete de video y lo pasa hacia las capas más bajas, donde los paquetes son servidos así como ellos van llegando a la cola. El concepto general de este esquema es que los paquetes se transmiten en tiempo real de tal manera que los paquetes son muy susceptibles a las pérdidas y al retardo como

parte de las características de la red móvil inalámbrica, por tal motivo los paquetes se deben transmitir rápidamente por el administrador de tal manera que ellos deben llegar a su destino en el menor tiempo posible. Esta administración de paquetes en el buffer es la manera convencional que esta implementada, se conoce como FIFO , en el protocolo 802.11 MAC en la función de coordinación distribuida DCF.

Sin embargo, esto no es suficiente para la transmisión de datos multimedia porque debido a las condiciones variables de la red en ciertos periodos de tiempo en que no se pueda transmitir los paquetes estos tendrán que esperar en la cola. Esto puede causar que paquetes de alta importancia se retrasen almacenados en la cola y de este modo muchos paquetes importantes no pueden alcanzar el decodificador a tiempo [43] (ver Figura 5.7), lo cual guiará a una ineficiente reproducción del video que se manifestará en una mala QoE.

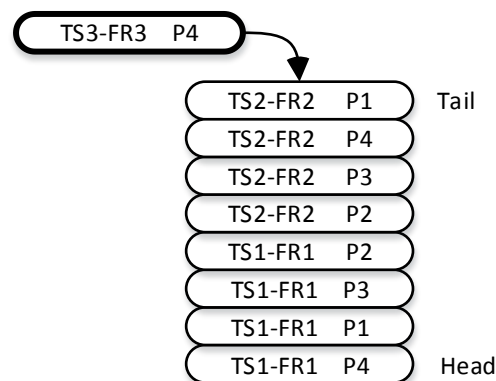


Figura 5.6 Problema en la cola cuando los paquetes no están ordenados por su prioridad.

5.6 Importancia del *timestamp*

En general, el *timestamp* es el tiempo cuando un evento ha sido grabado por una computadora. **En el caso especial de la codificación de video, todos los slices de una misma trama tienen el mismo *timestamp* ya que indica el instante de tiempo que le pertenece a una trama en todo el video, por lo que el *timestamp* es utilizado para colocar los paquetes de video entrantes al buffer en el tiempo correcto.**

En aplicaciones de telecomunicaciones la sincronización del video es de suma importancia para su adecuada reproducción, especialmente en aplicaciones de tiempo real un video no sincronizado puede causar *jitter*, turbulencia y problemas de latencia [44], por esta razón se requiere tener control del *timestamp* para presentar cada imagen en el tiempo correcto y con la duración adecuada. Se debe asegurar que el paquete que esté retardado se tire antes de ser transmitido sobre la red y de esta manera evitar que se desperdicien recursos de la red.

Para ser más claro el problema veremos un ejemplo, en la Figura 5.6 el paquete con prioridad P1, número de trama FR1 y *timestamp* TS1, supongamos que ya ha excedido el límite del tiempo de vida TTL asignado para este paquete, sin embargo los demás paquetes almacenados en la cola todavía tienen tiempo de vida para permanecer en la red, a pesar de que el paquete tiene prioridad P1 y es de suma importancia para la decodificación del video en aplicaciones multimedia no es tolerable que el paquete se reciba con retardo. Este es un buen motivo por el cual se propone checar el TTL de cada paquete en cada salto que da dentro de la red y si el límite del tiempo de vida del paquete ha excedido entonces este se tendrá que eliminar.

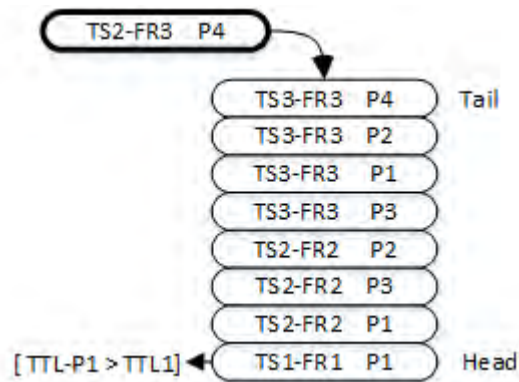


Figura 5.7 Problema en la cola cuando el *timestamp* ha expirado.

Capítulo 6

Esquema propuesto para la administración de paquetes de video H.264/AVC en la cola usando el ns-2.34

De acuerdo a la revisión que se realizó en el capítulo anterior y todas las condiciones y problemas presentados, en esta tesis se propone un conjunto de algoritmos para mejorar la calidad del video recibido sobre redes MANET.

El principal objetivo de este trabajo es implementar una nueva lógica en el algoritmo de administración de paquetes del buffer con la finalidad de obtener una mejor calidad del video decodificado en comparación con el algoritmo de encolamiento FIFO/DropTail. El algoritmo toma en cuenta el tiempo de vida del paquete y la prioridad del paquete asignada desde la capa de aplicación.

En este capítulo se presenta un mecanismo de administración de paquetes basado en la importancia de cada paquete de video por transmitir. La importancia de los paquetes de *streaming* de video se determina por los siguientes criterios: el *timestamp* de cada paquete, el número de trama dentro de un grupo de imágenes (GOP, *Group of Pictures*), la prioridad del paquete determinado por el codificador H.264/AVC y el retardo permitido para cada tipo de paquete, organizando todos estos campos en conjunto logramos tener mejores resultados. De esta forma, los paquetes más importantes de un *streaming* de video se colocan al principio de la cola para ser transmitidos primero que los paquetes menos importantes;

En esta tesis demostramos la efectividad de nuestro esquema propuesto mediante simulaciones en ns-2 y posteriormente presentamos un estudio de comparación. En las siguientes secciones

explicaremos cada una de las partes que conforman el algoritmo propuesto para la administración de paquetes de video en el buffer para obtener una mejor QoE del video recibido.

Antes de iniciar con la descripción del algoritmo propuesto es necesario definir el significado de las variables utilizadas:

TTL_paquete --> Este se calcula tomando el tiempo actual de la simulación menos el timestamp_paquete.

timestamp_paquete --> Este lo obtenemos a partir del time_tx más el timestamp_codec.

time_tx --> tiempo en que el nodo transmisor recibe el paquete por la capa de aplicación.

timestamp_codec --> timestamp del paquete asignado por el codec.

TTL --> Es el tiempo predefinido en la simulación que se asigna de acuerdo a la prioridad del paquete.

longCola --> Longitud actual de la cola medida en cantidad de paquetes.

prio_paquete_actual --> Es la prioridad del paquete nuevo que va a ser ordenado en la cola.

prio_paquete --> Es la prioridad del paquete que está siendo apuntado en la cola como referencia para el ordenamiento.

6.1 Recepción del paquete en el buffer

En las simulaciones realizadas en ns-2.34 se predefinen los valores del TTL de cada paquete de acuerdo a su prioridad. De esta forma, justo cuando el paquete se recibe en el buffer antes de ser puesto en la cola, se revisa si su tiempo de vida ha excedido; en dado caso, el paquete se descarta inmediatamente antes de ser puesto en la cola. Si eliminamos el paquete durante su travesía en un nodo intermedio, entonces hacemos un poco más eficiente el uso de los recursos disponibles en la red.

De acuerdo al capítulo anterior, en esta tesis se considera cuatro diferentes prioridades para los paquetes de video, y también asignamos diferentes TTL para cada prioridad. De esta forma tenemos que para cada prioridad P1, P2, P3 y P4 su tiempo de vida asignado es TTL1, TTL2, TTL3 y TTL4 respectivamente, donde $TTL1 > TTL2 > TTL3 > TTL4$. Con esto aseguramos que los paquetes de alta prioridad (P1 y P2) tengan más tiempo de vida en la red.

La lógica implementada es la siguiente:

```
if (TTL_paquete > TTL[j])  donde, j=1,2,3,4  indica la prioridad del
paquete
    eliminar paquete;
```

6.2 Encolamiento del paquete

En situaciones donde el canal no está disponible para transmitir paquetes, éstos permanecen en la cola en espera de ser transmitidos; durante este tiempo de almacenamiento puede expirar el TTL de muchos paquetes. Para evitar que paquetes con TTL expirado se transmitan, primero se revisa el TTL de todos los paquetes en el buffer, y se eliminan sólo aquellos que han expirado; posteriormente el paquete entrante se encola. La idea general de esta parte del algoritmo se muestra a continuación.

```
for (i=1; i<=long_cola; i++)
{
    if (TTL_paquete_P[j] > TTL[j])  donde, j=1,2,3,4  indica la prioridad del paquete
        eliminar paquete;
}
```

El paquete nuevo se encola, y el buffer queda en espera de recibir un nuevo paquete.

6.3 Ordenamiento de paquetes en el buffer

Esta es la parte más importante de esta tesis, ya que de acuerdo a las características del paquete de video previamente mencionadas, se realizará el adecuado ordenamiento de paquetes en la cola, de tal manera que los paquetes más importantes sean los que se envíen primero cuando la red esté disponible para su transmisión. En este proyecto utilizamos tres tipos de ordenamiento por: timestamp, prioridad y por usuario.

6.3.1 Ordenamiento basado en *el timestamp* del paquete

Tal y como habíamos discutido en el capítulo anterior, dada la importancia de mantener en orden las tramas del video para su adecuada reproducción y al mismo tiempo tener en cuenta el tiempo de vida de cada paquete, en este algoritmo implementado consideramos como primer campo a tomar en cuenta para el ordenamiento de paquetes dentro de la cola el *timestamp*.

La elección de este primer campo como el principal a tomar en cuenta se debió a que muchos paquetes que ya tenían más tiempo en la red se localizaban al final del buffer; entonces, cuando había acceso al canal no se enviaban tan inmediatamente y cuando finalmente llegaban a la cabeza de la cola para ser transmitidos ya su tiempo de vida había expirado. Por esta razón se propone ordenar primero los paquetes de acuerdo a su *timestamp*, para asegurar que los paquetes más viejos sean transmitidos primero y no esperen tanto tiempo almacenados en el buffer hasta que se expiren.

La lógica implementada es la siguiente:

```
for (i=0; i<=long_cola; i++)
{
    if (timestamp_paquete_actual == timestamp_paquete[i])
    {
        if (timestamp_paquete_actual == timestamp_paquete_head)
            Insertar el paquete después del timestamp_paquete_head

        else if (timestamp_paquete_actual == timestamp_paquete_tail)
            Insertar el paquete como tail

        else
            Insertar paquete después de timestamp_paquete[i])
    }
    if (timestamp_paquete_actual < timestamp_paquete[i])
    {
        Insertar paquete antes de timestamp_paquete[i])
    }
    if (timestamp_paquete_actual > timestamp_paquete[i])
    {
        Insertar paquete después de timestamp_paquete[i])
    }
}
```

Con este algoritmo, al tomar en cuenta *el timestamp* aseguramos un control en el orden de las tramas para la reproducción del video (ver Figura 6.1).

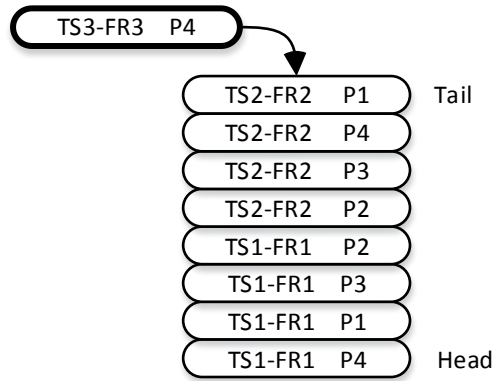


Figura 6.1 Paquetes ordenados por su *timestamp*.

6.3.2 Ordenamiento basado en la prioridad del paquete

De acuerdo a las características del archivo de traza del video obtenido del codificador, los slices que pertenecen a la misma trama por ende tienen el mismo *timestamp* pero diferente prioridad, de tal forma que una vez que la cola esté ordenada por el *timestamp*, ordenamos los paquetes que tienen el mismo *timestamp* basados en su prioridad (ver Figura 6.2).

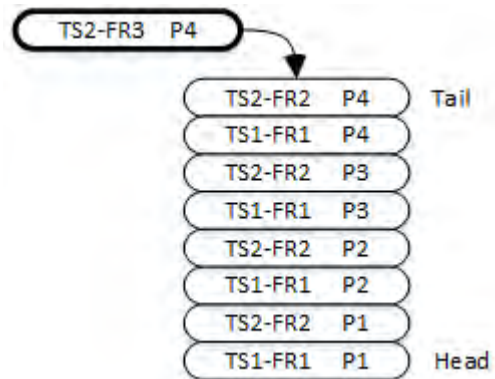


Figura 6.2 Paquetes ordenados por su prioridad.

Antes de ordenar los paquetes por su prioridad nos localizamos en la ubicación del buffer justo donde el *timestamp* del paquete actual es igual al *timestamp* del paquete encolado. La lógica implementada es la siguiente:

```

for (i=0; i<=4; i++)           donde i=1,2,3,4  indica la prioridad del paquete
{
    if(prio_paquete_actual == prio_paquete[i])
    {
        if(prio_paquete_actual == prio_paquete_head)
            Insertar el paquete después del prio_paquete_head

        else if (prio_paquete_actual == prio_paquete_tail)
            Insertar el paquete como tail
        else
            Insertar paquete después de prio_paquete[i])
    }
    if(prio_paquete_actual < prio_paquete[i])
    {
        Insertar paquete antes de prio_paquete[i])
    }
    if(prio_paquete_actual > prio_paquete[i])
    {
        Insertar paquete después de prio_paquete[i])
    }
}

```

También con este algoritmo, al tomar en cuenta la prioridad del paquete aseguramos que cuando haya acceso al canal los paquetes de mayor prioridad sean los primeros transmitidos (ver Figura 6.3).

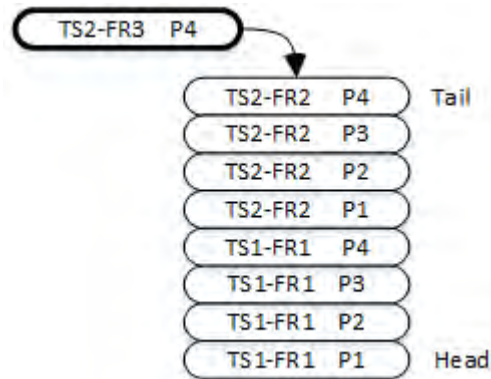


Figura 6.3 Cola cuando los paquetes están ordenados por su *timestamp* y prioridad.

6.3.3 Ordenamiento basado en la prioridad del usuario

Pensando en un sistema de comunicaciones inalámbricas más real, en esta tesis también se propone, aparte de los algoritmos de ordenamiento previamente explicados, tomar en cuenta también la prioridad de usuario, con la finalidad de que los paquetes de video del usuario más importante sean los primeros que se transmitan.

Para cuestiones de simulación, solo trabajamos con 4 usuarios como máximo, donde el usuario 1 es el de mayor importancia y el usuario 4 es el de menor importancia.

Al final obtenemos un algoritmo que en su conjunto es más robusto considerando el ordenamiento por usuario como el campo más importante, posteriormente el ordenamiento por *timestamp* y finalmente el ordenamiento por prioridad del paquete (ver Figura 6.4).

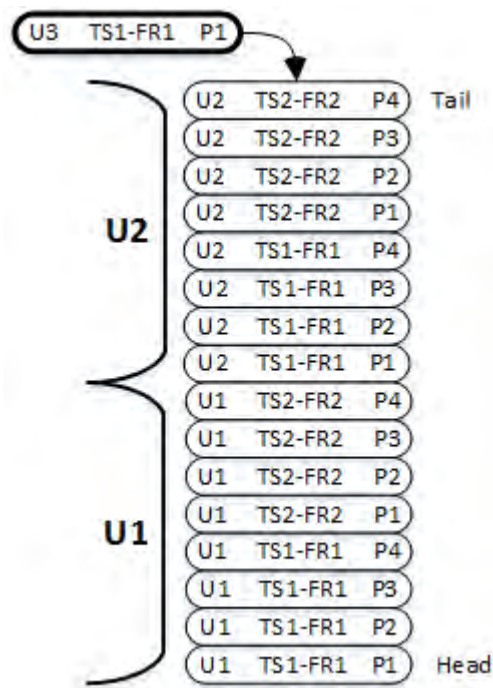


Figura 6.4 Cola cuando los paquetes están ordenados por su prioridad de usuario, *timestamp* y prioridad de paquete.

Los resultados de los algoritmos mencionados arriba se presentan en el siguiente capítulo.

Capítulo 7

Resultados

7.1 Características del escenario de simulación

La Tabla 7.1 presenta las características de escenario de evaluación en el simulador de redes ns-2.34. Para obtener los resultados que se presentan en las siguientes secciones utilizamos los siguientes parámetros:

Parámetro	Valor
Área	1000 x 1000 m
Número de nodos	25
Tiempo de simulación	30 s
Velocidad	5, 10 y 20 m/s
Número de usuarios	1, 2, 3 y 4
Protocolo MAC	802.11
Velocidad de transmisión	2 Mbps
Protocolo de enrutamiento	AOMDV
TTL	P1=0.45s, P2=0.4s, P3=0.3s y P4=0.25 s
Longitud de la cola	120 paquetes
Tipo de tráfico	paquetes de video H.264/AVC
Longitud del paquete	900 bytes
Velocidad de datos	256 Kbps
Vídeo	Foreman

Tabla 7.1 Características del escenario de evaluación.

La siguiente figura presenta la topología de la red manet utilizada para nuestras simulaciones con los parámetros de configuración mencionados arriba.

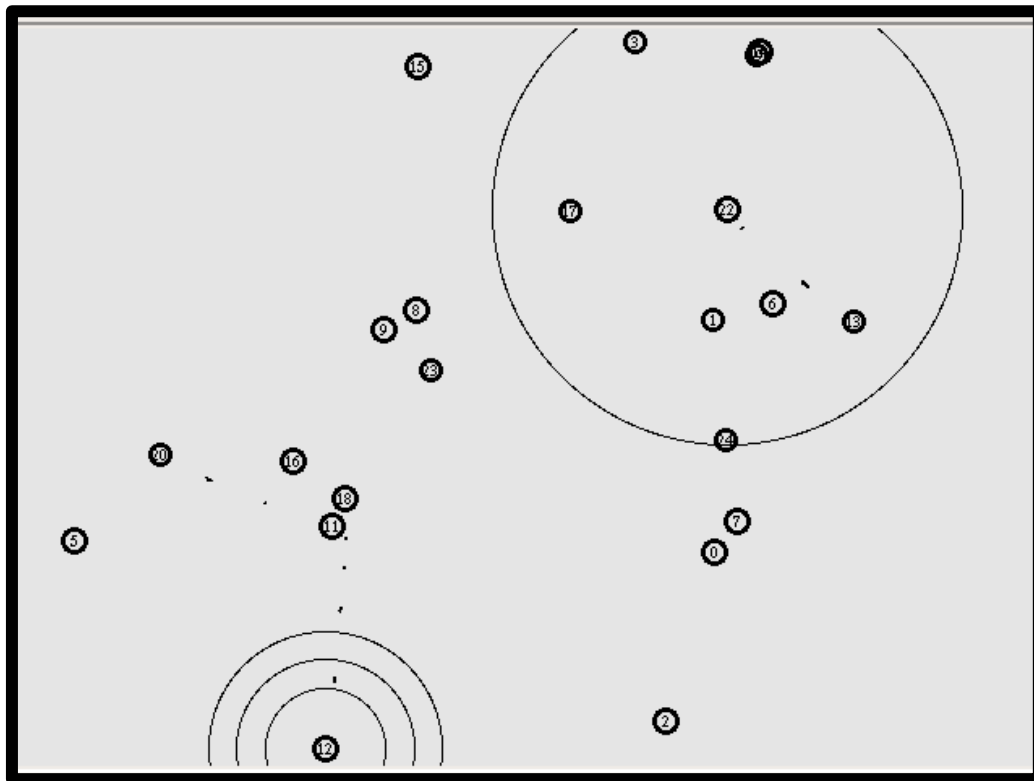


Tabla 7.1 Topología del escenario de evaluación.

En esta topología de la red para una capacidad de 2 Mbits/s sólo usamos 4 nodos transmisores y 4 nodos receptores dentro de un ambiente de 25 nodos en total, por lo que en cualquier momento debido a la movilidad de los nodos, cualquier nodo puede actuar como nodo intermedio y retransmitir los paquetes de video hacia un destino en particular.

7.2 Resultados del PDR y retardo punto a punto

En la tabla 7.2 y la figura 7.2 se muestran los resultados de las simulaciones para nuestro esquema propuesto y para el esquema FIFO; en ambos esquemas se utilizan los mismos escenarios de simulación con los mismos parámetros presentados en la Tabla 7.1, con una velocidad máxima de los nodos de 5 m/s; el único cambio realizado en estas simulaciones es la cantidad de usuarios variando de 1 a 4 (hasta 4 conexiones). Como se puede observar hay una gran mejora en el porcentaje de PDR obtenido, principalmente en el escenario más crítico con cuatro usuarios, donde se puede ver claramente que el PDR para paquetes con prioridad 1 es de 74% en comparación con los paquetes con prioridad 4 (los de menor importancia) en el cual se obtiene un PDR de 51%.

Esquema propuesto				Esquema FIFO			
Velocidad	Núm. Conexiones	Prioridad	% PDR	Velocidad	Núm. Conexiones	Prioridad	% PDR
5	1	1	99	5	1	1	99
		2	99			2	99
		3	99			3	99
		4	99			4	99
	2	1	93		2	1	93
		2	92			2	94
		3	89			3	94
		4	83			4	93
	3	1	89		3	1	84
		2	87			2	85
		3	83			3	85
		4	77			4	83
	4	1	74		4	1	65
		2	69			2	66
		3	61			3	67
		4	51			4	63

Tabla 7.2 Resultados del PDR para una velocidad de 5m/s, cuatro prioridades y hasta cuatro conexiones.

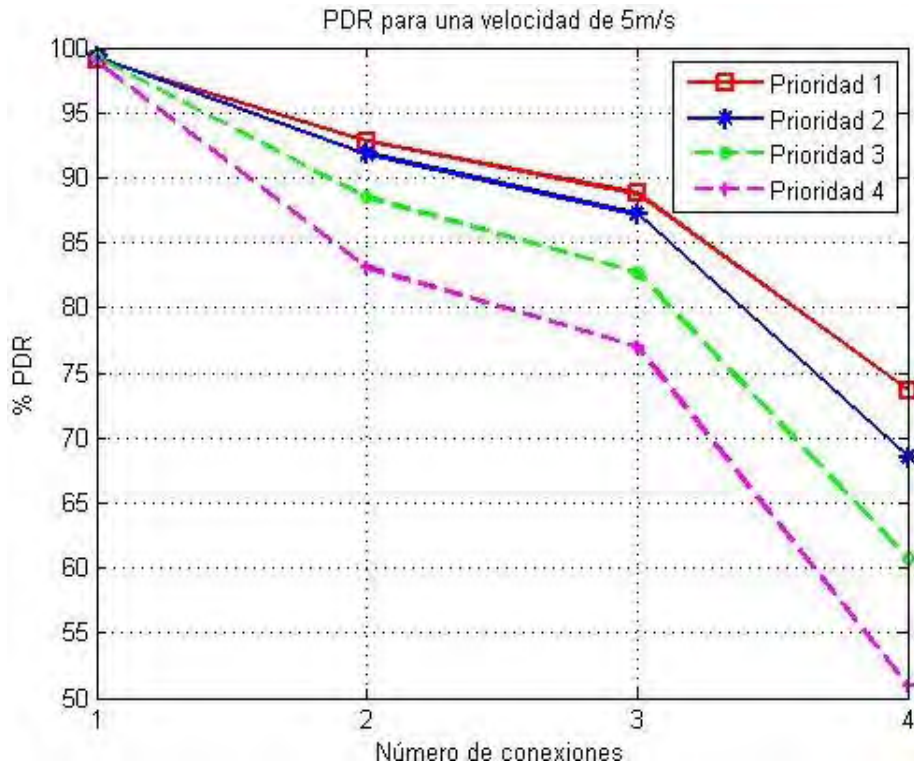


Figura 7.2 Resultados del PDR para una velocidad de 5 m/s, cuatro prioridades y hasta cuatro conexiones.

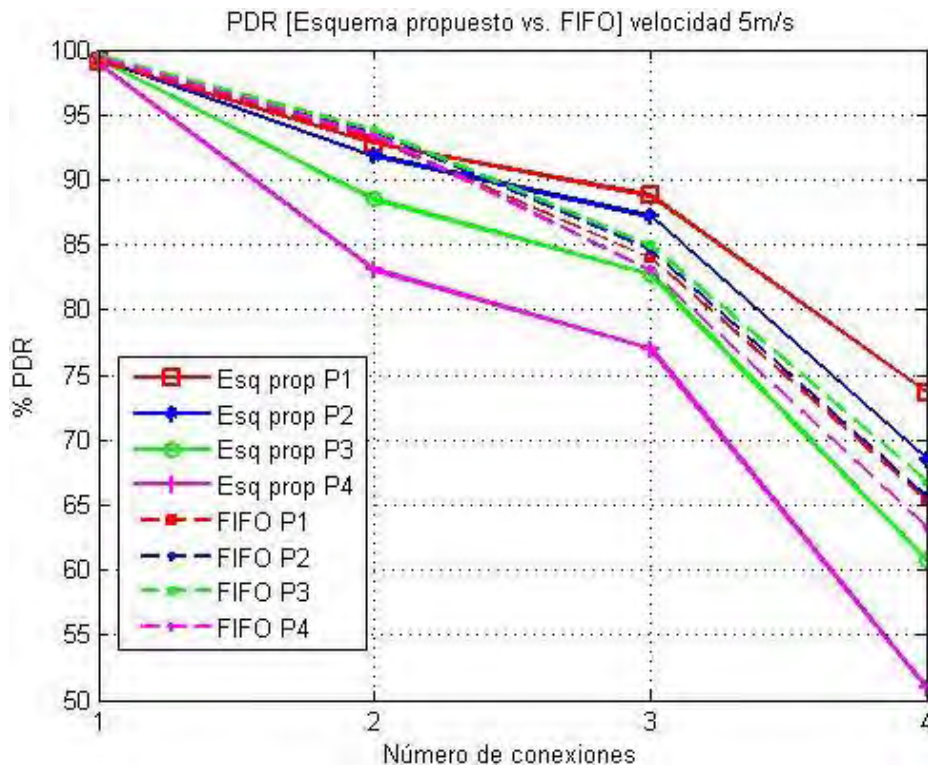


Figura 7.3 Comparación de los resultados del PDR para una velocidad de 5m/s entre el esquema propuesto y el esquema FIFO.

En la Figura 7.3 podemos observar y comparar los resultados obtenidos del esquema propuesto en comparación con los resultados del esquema FIFO; en ambos esquemas se utilizan los mismos escenarios de simulación con los mismos parámetros presentados en la Tabla 7.1, con una velocidad máxima de los nodos de 5 m/s, sólo se cambia la cantidad de usuarios (ó conexiones) de 1 a 4. Para el escenario de dos conexiones se puede observar que se obtienen mas paquetes recibidos en el esquema FIFO, sin embargo para los escenarios con más usuarios, tres y cuatro, con nuestro esquema propuesto obtenemos mejores resultados para paquetes con prioridad 1 y 2, casi 10 puntos porcentuales más para paquetes con prioridad 1 en comparación con el esquema FIFO.

De acuerdo a la Tabla 7.3 y Figura 7.4 se presentan las graficas de retardo *punto a punto*, donde se puede observar que con nuestro esquema tenemos una gran mejora en el retardo promedio por paquete. La diferencia más notoria es para los paquetes con prioridad 1 en el caso más crítico con 4 usuarios, donde la diferencia en tiempo promedio de retardo entre el esquema propuesto y el esquema FIFO es de casi la mitad de tiempo. Este resultado obtenido se verá claramente reflejado en la calidad de cada trama de video.

Esquema propuesto				Esquema FIFO			
Velocidad	Núm. Conexiones	Prioridad	Retardo (ms)	Velocidad	Núm. Conexiones	Prioridad	Retardo (ms)
5	1	1	6	5	1	1	7
		2	5			2	5
		3	4			3	4
		4	7			4	8
	2	1	27		2	1	19
		2	26			2	19
		3	27			3	21
		4	36			4	30
	3	1	43		3	1	52
		2	44			2	51
		3	44			3	50
		4	55			4	62
	4	1	76		4	1	141
		2	74			2	123
		3	73			3	119
		4	87			4	130

Tabla 7.3 Resultados del retardo para una velocidad de 5 m/s, cuatro prioridades y hasta cuatro conexiones.

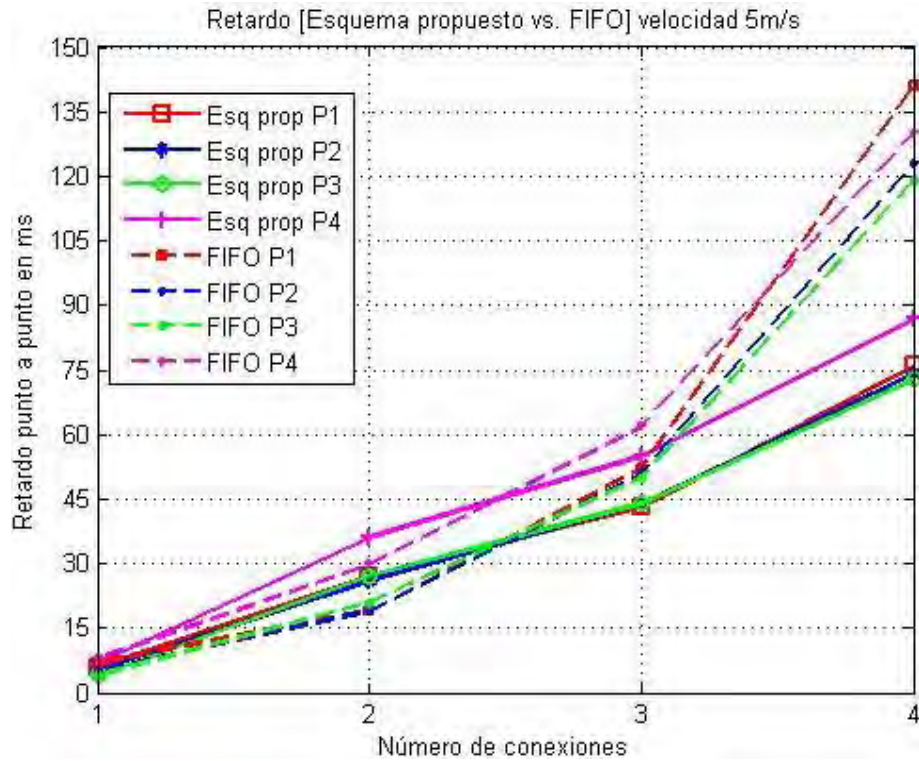


Figura 7.4 Comparación de resultados del retardo *punto a punto* para una velocidad de 5m/s entre el esquema propuesto y el esquema FIFO.

Ahora analizamos los mismos escenarios pero en este caso la velocidad es de 10 m/s. En la tabla 7.4 y la figura 7.5 se muestran los resultados de las simulaciones. Como se puede observar, también hay una gran mejora en el porcentaje de PDR obtenido, principalmente en el escenario más crítico con cuatro usuarios donde se puede ver que el PDR para paquetes con prioridad 1 es de 68% en comparación con los paquetes con prioridad 4 (los de menor importancia) en el cual se obtiene un PDR de 53%.

Esquema propuesto				Esquema FIFO			
Velocidad	Núm. Conexiones	Prioridad	% PDR	Velocidad	Núm. Conexiones	Prioridad	% PDR
10	1	1	95	10	1	1	95
		2	95			2	96
		3	95			3	96
		4	93			4	94
	2	1	91		2	1	87
		2	90			2	88
		3	89			3	88
		4	85			4	85
	3	1	80		3	1	75
		2	77			2	75
		3	73			3	73
		4	66			4	73
	4	1	68		4	1	59
		2	65			2	59
		3	60			3	60
		4	53			4	59

Tabla 7.4 Resultados del PDR para una velocidad de 10 m/s, cuatro prioridades y hasta cuatro conexiones.

Para los mismos escenarios, en la Figura 7.6 podemos observar y comparar los resultados obtenidos del esquema propuesto en comparación con los resultados del esquema FIFO. Para todos los escenarios se puede observar que se obtienen mas paquetes recibidos de alta prioridad en nuestro esquema. Para los escenarios con más usuarios, tres y cuatro, obtenemos mejores resultados para paquetes con prioridad 1 y 2, casi 10 puntos porcentuales más para paquetes con prioridad 1 en comparación con el esquema FIFO.

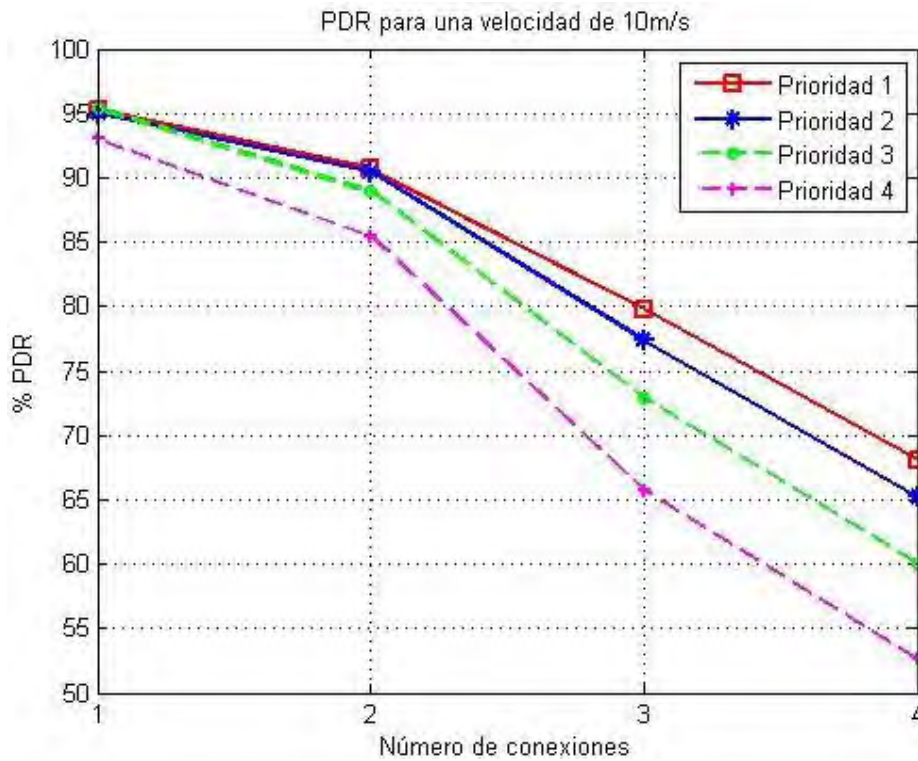


Figura 7.5 Resultados del PDR para una velocidad de 10 m/s, cuatro prioridades y hasta cuatro conexiones.

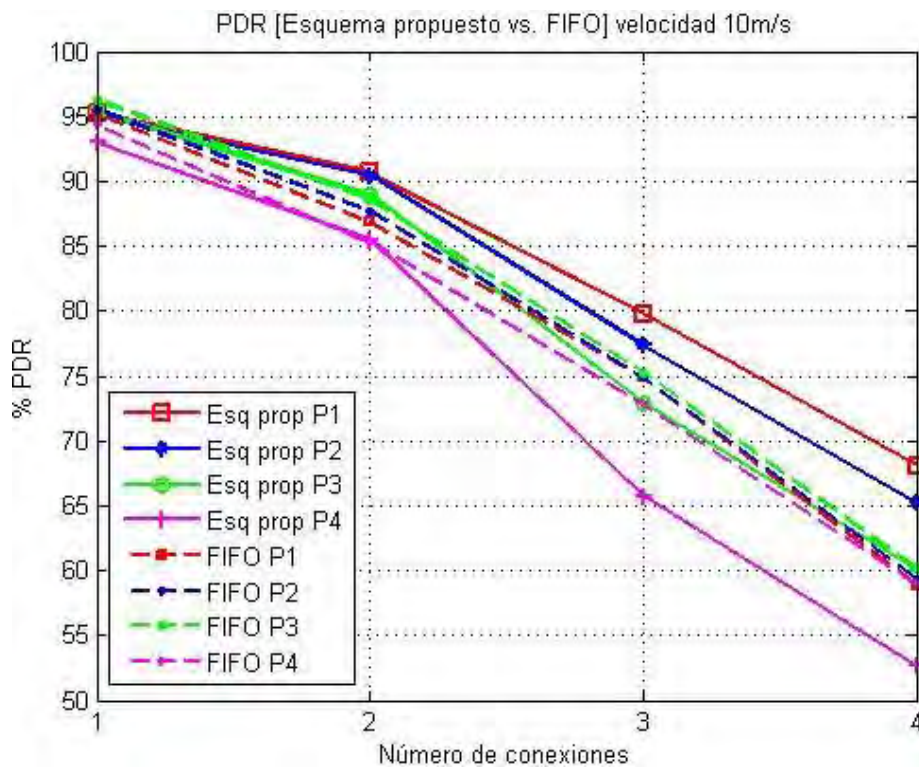


Figura 7.6 Comparación de los resultados del PDR para una velocidad de 10 m/s entre el esquema propuesto y el esquema FIFO.

De acuerdo a la Tabla 7.5 y Figura 7.7 se presentan las graficas de retardo *punto a punto*, donde se puede observar que con nuestro esquema tenemos una gran mejora en el tiempo promedio de retardo para todos los paquetes, la diferencia más notoria es para los paquetes con prioridad 1 en el caso más crítico con 4 usuarios, donde la diferencia en tiempo promedio de retardo entre nuestro esquema y el esquema FIFO es de aproximadamente 80 ms. Este resultado obtenido se verá claramente reflejado en la calidad del video.

Esquema propuesto				Esquema FIFO			
Velocidad	Núm. Conexiones	Prioridad	Retardo (ms)	Velocidad	Núm. Conexiones	Prioridad	Retardo (ms)
10	1	1	13	10	1	1	7
		2	10			2	5
		3	8			3	4
		4	12			4	8
	2	1	23		2	1	24
		2	22			2	23
		3	24			3	25
		4	31			4	32
	3	1	65		3	1	58
		2	60			2	58
		3	58			3	57
		4	71			4	70
	4	1	114		4	1	195
		2	105			2	165
		3	102			3	147
		4	118			4	177

Tabla 7.5 Resultados del retardo para una velocidad de 10m/s, cuatro prioridades y hasta cuatro conexiones.

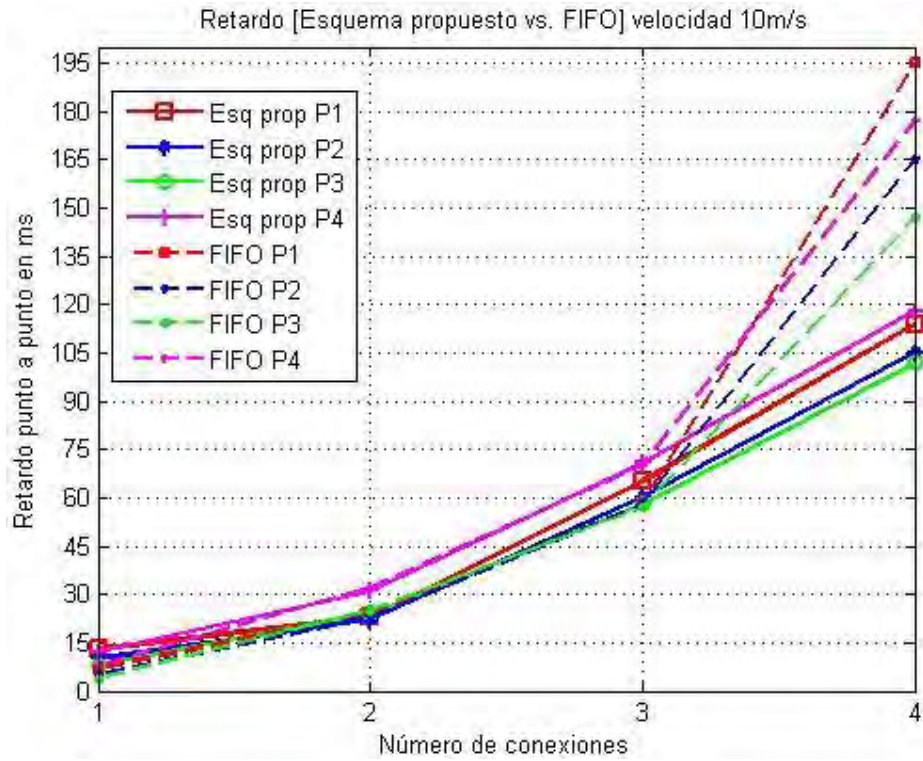


Figura 7.7 Comparación de resultados del retardo *punto a punto* para una velocidad de 10 m/s entre el esquema propuesto y el esquema FIFO.

En la tabla 7.6 y la figura 7.8 se muestran los resultados de las simulaciones para una velocidad de 20 m/s. Como se puede observar hay una gran mejora en el porcentaje de PDR obtenido, principalmente en el escenario más crítico con cuatro usuarios donde se puede ver claramente que el PDR para paquetes con prioridad 1 es de 61% en comparación con los paquetes con prioridad 4 (los de menor importancia) en el cual se obtiene un PDR de 50%.

Esquema propuesto				Esquema FIFO			
Velocidad	Núm. Conexiones	Prioridad	% PDR	Velocidad	Núm. Conexiones	Prioridad	% PDR
20	1	1	90	20	1	1	91
		2	88			2	90
		3	88			3	90
		4	87			4	88
	2	1	79		2	1	75
		2	77			2	75
		3	75			3	76
		4	71			4	74
	3	1	68		3	1	65
		2	68			2	66
		3	63			3	66
		4	60			4	64
	4	1	61		4	1	50
		2	59			2	52
		3	55			3	51
		4	50			4	51

Tabla 7.6 Resultados del PDR para una velocidad de 20 m/s, cuatro prioridades y hasta cuatro conexiones.

Para el mismo escenario, en la Figura 7.9 podemos observar y comparar los resultados obtenidos del esquema propuesto en comparación con los resultados del esquema FIFO. Para todos los escenarios se puede observar que se obtienen más paquetes recibidos en nuestro esquema. Para los escenarios con más usuarios, tres y cuatro, obtenemos mejores resultados para paquetes con prioridad 1 y 2, casi 10 puntos porcentuales más para paquetes con prioridad 1 en comparación con el esquema FIFO.

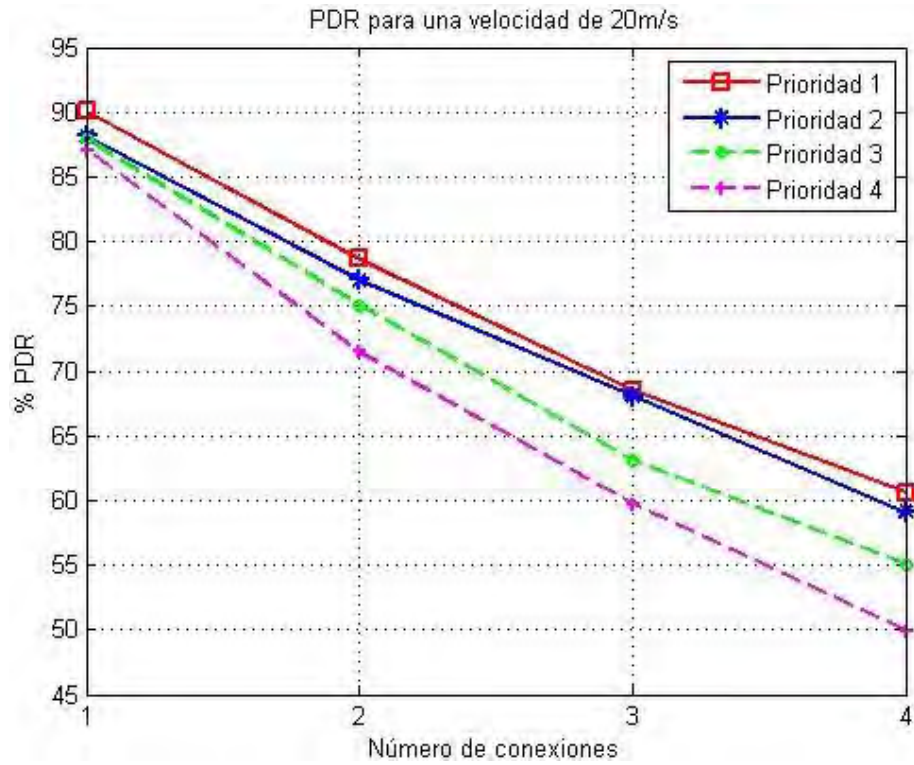


Figura 7.8 Resultados del PDR para una velocidad de 20 m/s, cuatro prioridades y hasta cuatro conexiones.

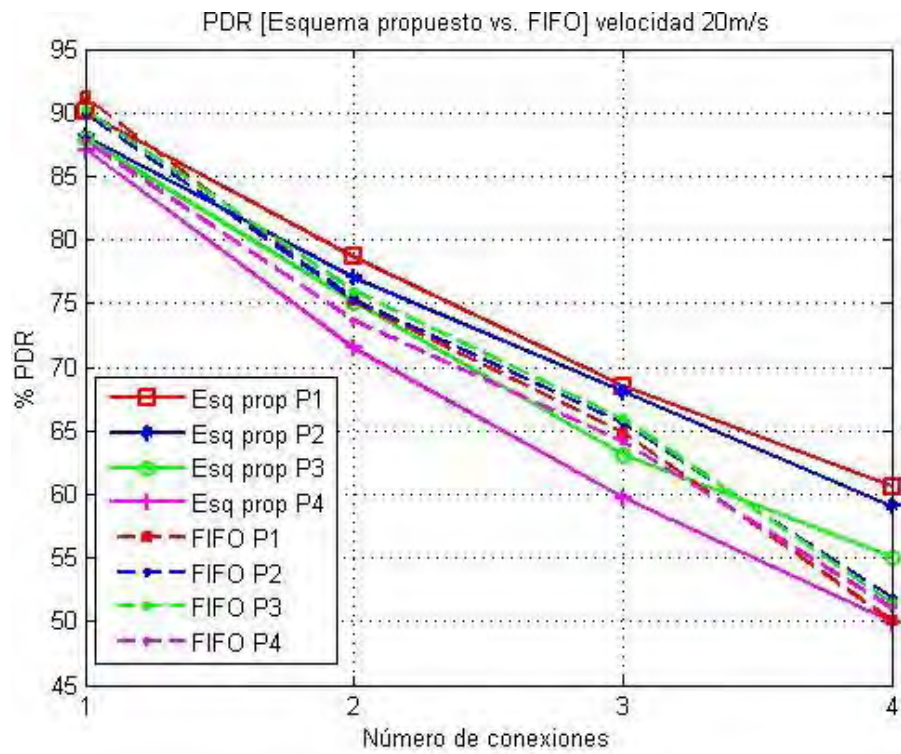


Figura 7.9 Comparación de los resultados del PDR para una velocidad de 20 m/s entre el esquema propuesto y el esquema FIFO.

De acuerdo a la Tabla 7.7 y a la Figura 7.10 se presentan las graficas de retardo *punto a punto* donde se puede observar que el retardo promedio por paquete obtenido con nuestro esquema y el esquema FIFO es muy similar. Sin embargo hay que tomar en cuenta que obtenemos más paquetes con alta prioridad manteniendo el mismo nivel de retardo.

Esquema propuesto				Esquema FIFO			
Velocidad	Núm. Conexiones	Prioridad	% PDR	Velocidad	Núm. Conexiones	Prioridad	% PDR
20	1	1	19	20	1	1	14
		2	15			2	11
		3	12			3	10
		4	17			4	15
	2	1	51		2	1	50
		2	41			2	41
		3	40			3	42
		4	48			4	49
	3	1	76		3	1	78
		2	65			2	68
		3	66			3	69
		4	71			4	75
	4	1	127		4	1	134
		2	107			2	115
		3	109			3	111
		4	113			4	119

Tabla 7.7 Resultados del retardo para una velocidad de 20 m/s, cuatro prioridades y hasta cuatro conexiones.

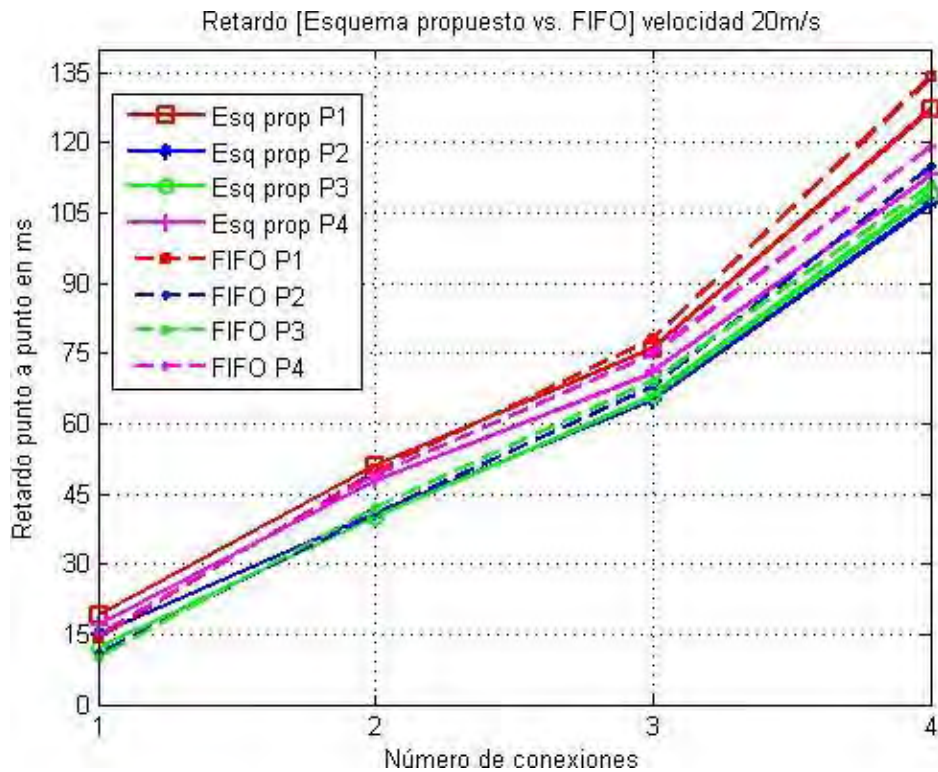


Figura 7.10 Comparación de resultados del retardo *punto a punto* para una velocidad de 20 m/s entre el esquema propuesto y el esquema FIFO.

7.3 Desempeño del algoritmo basado en el PSNR

Con los resultados del PDR presentados en la sección previa, calculamos primero el porcentaje de paquetes perdidos para cada tipo de prioridad en cada simulación y así obtenemos los valores de PSNR. En la Tabla 7.8 se puede observar en los resultados una mejora en un rango de 0.5 a 1.5 dB en nuestro esquema en comparación con el esquema FIFO.

Vel.	PSNR (Esquema Propuesto)			PSNR (Esquema FIFO)		
	2 Usuarios	3 Usuarios	4 Usuarios	2 Usuarios	3 Usuarios	4 Usuarios
5	28.40	26.03	22.19	28.75	25.23	21.25
10	27.22	23.78	21.22	26.49	22.92	20.21
20	24.00	22.36	20.08	22.92	21.29	18.48

Tabla 7.8 Resultados del PSNR para velocidades de 5, 10 20 m/s, cuatro prioridades y hasta cuatro conexiones.

Como se esperaba, con nuestro algoritmo rescatamos mas paquetes de prioridad alta lo cual impactará en la calidad del video que el usuario percibe; es decir, a pesar de que tenemos casi la misma cantidad de paquetes perdidos en promedio para todo el video, no es lo mismo perder para cada trama cierta cantidad de paquetes con prioridad 1 y/o 2 que perder paquetes de prioridad 3 y/o 4. Esto se puede apreciar en las siguientes Figuras 7.11 7.12 y 7.13 así como en la Tabla 7.8 del PSNR

VELOCIDAD 5



a) Video original

b) Video recibido
con nuestro algoritmo implementado

c) Video recibido
usando el esquema FIFO

Figura 7.11 Comparación de una misma trama de video en un escenario con velocidad de 5 m/s; entre el video recibido usando nuestro algoritmo y el video recibido usando el esquema FIFO.

En las Figuras 7.11, 7.12 y 7.13 podemos observar diferentes tramas del video para las diferentes velocidades de movilidad : 5, 10 y 20 m/s respectivamente. Si observamos, podemos comparar el impacto de la pérdida de paquetes con diferente prioridad para la misma trama. Visiblemente podemos observar que la calidad del video es ligeramente mejor si aplicamos el algoritmo propuesto.

VELOCIDAD 10



a) Video original

b) Video recibido con nuestro algoritmo implementado

c) Video recibido usando el esquema FIFO

Figura 7.12 Comparación de una misma trama de video en un escenario con velocidad de 10 m/s; entre el video recibido usando nuestro algoritmo y el video recibido usando el esquema FIFO.

VELOCIDAD 20



a) Video original

b) Video recibido
con nuestro algoritmo implementado

c) Video recibido
usando el esquema FIFO

Figura 7.13 Comparación de una misma trama de video en un escenario con velocidad de 20 m/s; entre el video recibido usando nuestro algoritmo y el video recibido usando el esquema FIFO.

Si comparamos las Figuras (7.11, 7.12 y 7.13) del video recibido utilizando nuestro algoritmo, podemos comprobar que si se logra rescatar la mayor cantidad de paquetes de prioridad alta en el video, se contribuye visiblemente a la mejora de la calidad percibida por el usuario.

7.4 Resultados del PDR en el esquema por prioridad de usuario

A continuación se presentan los resultados del PDR obtenido para cada usuario, tanto en el esquema propuesto como en el esquema FIFO.

Porcentaje de PDR (Esquema por prioridad de usuario)											
Vel	Prio	2 Usuarios		3 Usuarios			4 Usuarios				
		<i>U1</i>	<i>U2</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>	
5	P1	97	93	96	85	87	82	74	79	71	
	P2	96	94	94	83	85	77	70	76	58	
	P3	94	92	89	79	82	68	62	68	45	
	P4	89	89	80	71	76	54	50	62	34	
		2 Usuarios		3 Usuarios			4 Usuarios				
		<i>U1</i>	<i>U2</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>	
10	P1	93	88	85	82	74	75	66	69	66	
	P2	90	87	81	78	70	69	65	62	60	
	P3	87	85	74	71	66	61	59	52	49	
	P4	84	81	64	62	60	48	48	44	39	
		2 Usuarios		3 Usuarios			4 Usuarios				
		<i>U1</i>	<i>U2</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>	
20	P1	84	75	71	67	71	71	66	57	68	
	P2	80	73	70	64	65	65	62	56	58	
	P3	76	72	65	58	59	58	56	51	45	
	P4	70	68	58	49	55	49	44	44	34	

Tabla 7.9 Resultados del PDR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema por prioridad de usuario.

Porcentaje de PDR FIFO												
Vel	Prio	2 Usuarios		3 Usuarios			4 Usuarios					
		<i>U1</i>	<i>U2</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>		
5	P1	95	91	88	78	86	69	70	75	47		
	P2	95	92	88	79	87	70	70	76	48		
	P3	95	93	88	79	88	71	71	76	49		
	P4	94	93	86	77	86	67	65	75	46		
		2 Usuarios		3 Usuarios			4 Usuarios					
		<i>U1</i>	<i>U2</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>		
10	P1	87	87	77	78	70	66	62	60	47		
	P2	89	86	78	77	70	66	60	63	48		
	P3	89	88	78	78	70	68	61	63	49		
	P4	87	83	76	74	69	64	59	65	48		
		2 Usuarios		3 Usuarios			4 Usuarios					
		<i>U1</i>	<i>U2</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>		
20	P1	79	71	68	63	64	62	49	48	41		
	P2	78	73	67	64	66	62	50	51	44		
	P3	79	73	67	64	67	63	50	50	43		
	P4	75	72	66	59	67	60	46	52	45		

Tabla 7.10 Resultados del PDR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema FIFO.

De acuerdo a los resultados presentados en las tablas 7.9 y 7.10; y en las Figuras 7.14, 7.15 y 7.16, se nota una mejora en general para todos los usuarios en cuanto a los paquetes de alta prioridad recibidos, y esto es por la administración de paquetes en el buffer de cada nodo. Por ejemplo, en el caso del escenario de sólo 2 usuarios para una velocidad de 20 m/s, se obtuvo los siguientes resultados:

Esquema FIFO:

Usuario 1=79% de paquetes recibidos de prioridad 1, Usuario 2=71% de paquetes recibidos con prioridad 1.

Usuario 1=78% de paquetes recibidos de prioridad 2, Usuario 2=73% de paquetes recibidos con prioridad 2.

Esquema por prioridad de usuario:

Usuario 1=84% de paquetes recibidos de prioridad 1, Usuario 2=75% de paquetes recibidos con prioridad 1.

Usuario 1=80% de paquetes recibidos de prioridad 2, Usuario 2=73% de paquetes recibidos con prioridad 2.

Con estos resultados podemos observar la mejora obtenida en PSNR de acuerdo a la Tabla 7.10 y 7.11.

Esquema FIFO:

Usuario 1: 23.43dB Usuario 2: 22.7 dB

Esquema por prioridad de usuario:

Usuario 1: 24.5 dB Usuario 2: 22.86 dB

Notamos que el Usuario 2 permanece casi con el mismo valor, sin embargo en el Usuario 1 si se puede notar una mejora de 1dB aproximadamente.

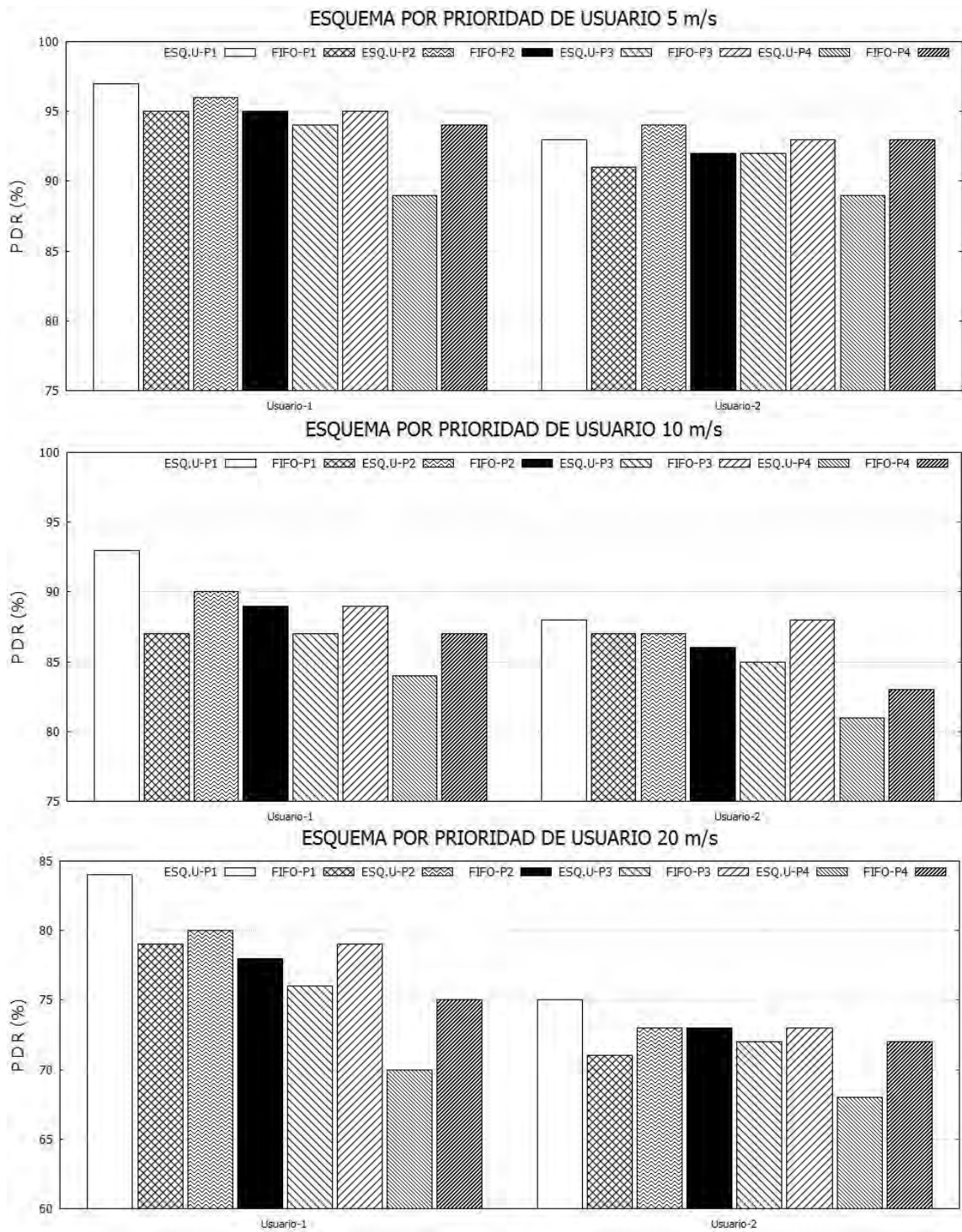


Figura 7.14 PDR obtenido en dos usuarios en el esquema por 'prioridad de usuario' con velocidades de: 5, 10 y 20 m/s.

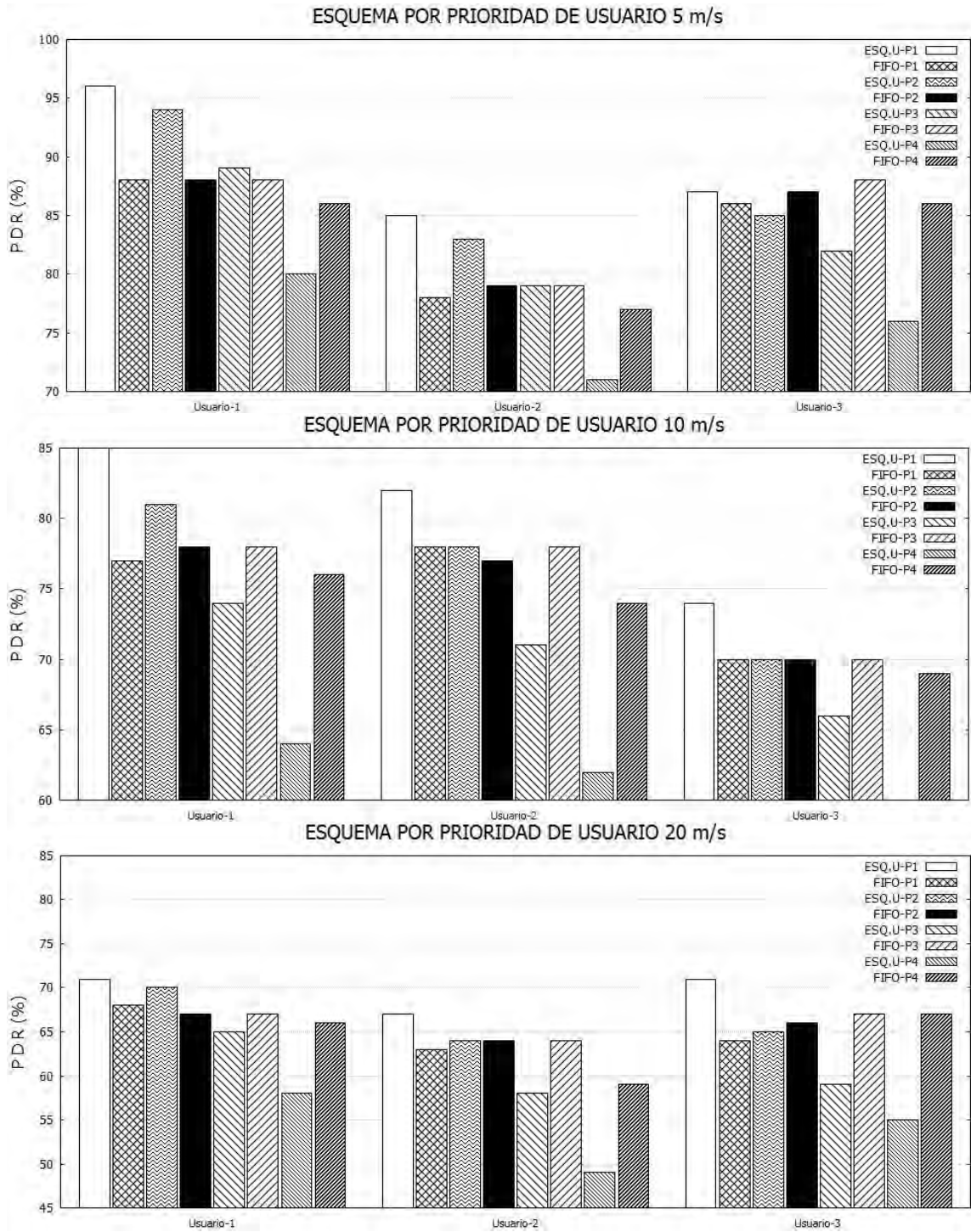


Figura 7.15 PDR obtenido en tres usuarios en el esquema por 'prioridad de usuario' con velocidades de: 5, 10 y 20 m/s.

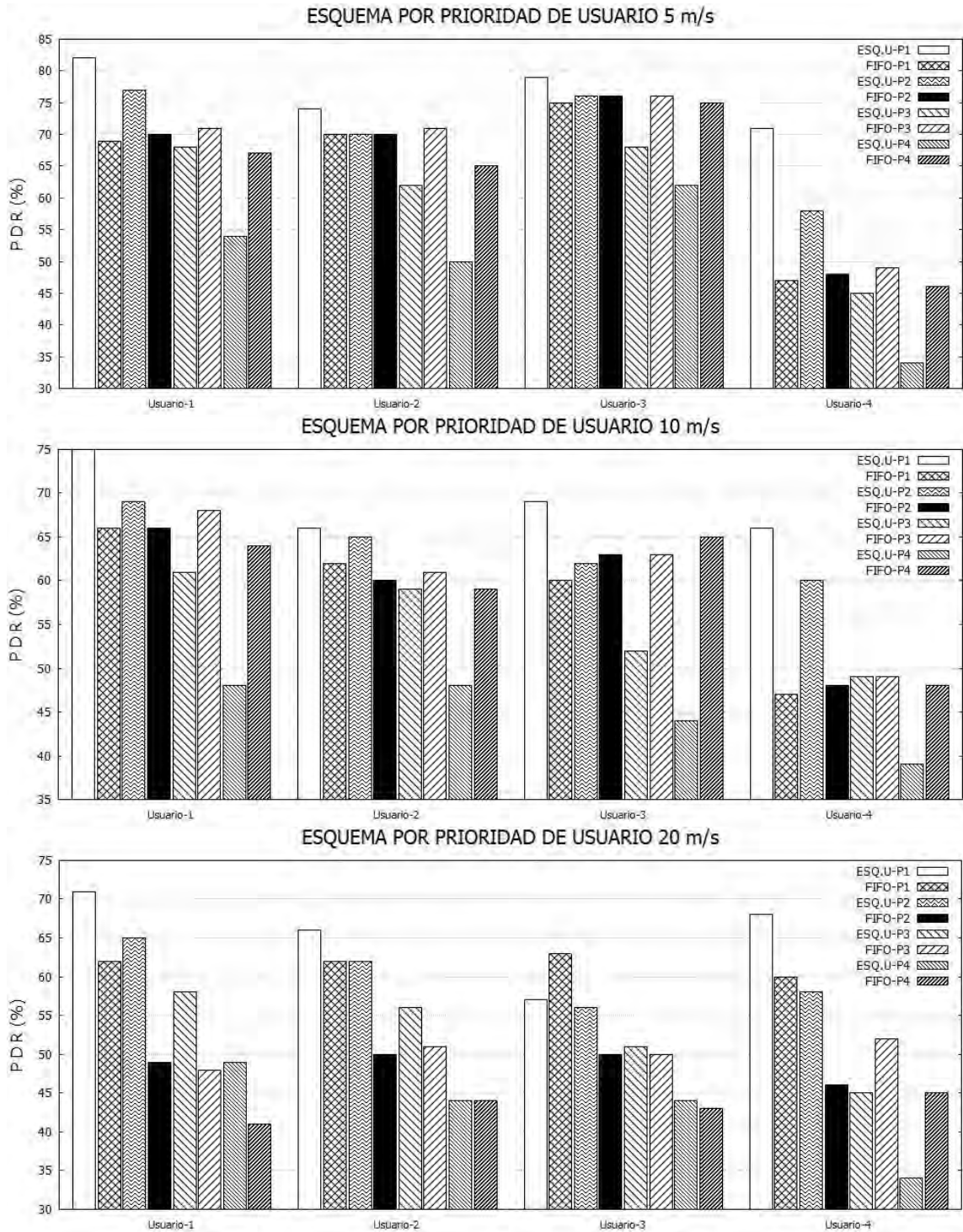


Figura 7.16 PDR obtenido en cuatro usuarios en el esquema por 'prioridad de usuario' con velocidades de: 5, 10 y 20 m/s.

7.5 Resultados del PDR en el esquema por prioridad de usuario

PSNR (Esquema Propuesto)											
Vel	2 Usuarios			3 Usuarios			4 Usuarios				
	<i>U1</i>	<i>U2</i>		<i>U1</i>	<i>U2</i>	<i>U3</i>		<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>
5	30.0	28.6		29.0	25.0	26.0		23.16	21.86	23.00	21.00
10	27.85	27.0.6		25.08	23.68	22.27		22.35	20.70	21.25	20.43
20	24.50	22.86		22.64	21.55	21.09		21.46	20.76	19.51	21.05

Tabla 7.11 Resultados del PSNR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema FIFO.

PSNR (FIFO)											
Vel	2 Usuarios			3 Usuarios			4 Usuarios				
	<i>U1</i>	<i>U2</i>		<i>U1</i>	<i>U2</i>	<i>U3</i>		<i>U1</i>	<i>U2</i>	<i>U3</i>	<i>U4</i>
5	29.27	27.56		26.51	23.41	25.97		21.51	21.62	23.21	17.54
10	26.23	25.94		23.35	23.58	21.80		21.08	20.11	20.31	17.60
20	23.43	22.7		21.7	20.66	20.62		20.56	18.3	18.17	17.41

Tabla 7.12 Resultados del PSNR para velocidades de 5, 10 y 20 m/s, cuatro prioridades, hasta cuatro usuarios. Esquema FIFO.

Las Tablas 7.11 y 7.12 presentan los resultados del PSNR obtenidos de la evaluación de todos los escenarios: de 2 a 4 conexiones; y de 5, 10 y 20 m/s.

En el esquema por 'prioridad de usuario' y en el esquema FIFO. En algunos casos se nota una mejora del PSNR como en el caso del Usuario 1, velocidad de 5 m/s y 3 conexiones (3 usuarios); en este caso la mejora del PSNR es de aproximadamente 2.5 dB (de 26.51 a 29 dB). También podemos observar en este mismo escenario de valuación, que en el caso del esquema FIFO el Usuario 3 tiene un PSNR de 25.97 dB y el Usuario 2 tiene un PSNR de 23.41, una diferencia entre ambos de 2.6 dB; presentando mejores resultados el Usuario 3 que el Usuario 2. Utilizando nuestro esquema la diferencia entre estos dos usuarios disminuye a 1dB: El Usuario 3 tiene un PSNR de 26 dB (prácticamente se mantiene el valor) y el Usuario 2 tiene un PSNR de 25 dB. De acuerdo con estos resultados, podemos observar que al implementar el algoritmo de 'prioridad por usuario' logramos ayudar un poquito al Usuario 2 respecto al Usuario 3. Como es una red manet donde se está evaluando nuestro algoritmo, no podemos observar mayor mejoras, debido a que no tenemos control de acceso al medio.

Conclusiones

De acuerdo al Capítulo 2, a pesar de que esta tesis no está enfocada primordialmente en la codificación de video, consideramos de gran utilidad saber a groso modo cuales son las características del estándar H.264/AVC, principalmente cuáles son las ventajas que ofrece para ser utilizado en la transmisión de streaming de video en redes; todo esto se utiliza para el diseño de nuestro escenario de evaluación. Es bueno mencionar que para esta tesis utilizamos el perfil extendido para la codificación de nuestro video de prueba 'Foreman'. A pesar de que este perfil nos ofrece muchas características y una gran versatilidad para la codificación de video, por fines de simplicidad y siguiendo una línea de investigación previa, sólo codificamos *slices* tipo I, P y B. Con estos tres tipos de *slices* es posible ver la diferencia de impacto que tiene la pérdida de cada tipo.

En cuestiones de análisis se decidió utilizar las métricas de PSNR para evaluar los resultados obtenidos. Estas métricas son muy comunes y fáciles en la generación de resultados.

En el capítulo 3 elegimos hacer un breve estudio de las redes MANET, debido a que fue la red inalámbrica que elegimos para nuestros escenarios. Este tipo de red es muy fácil de implementar en el ns-2.34 ya que los protocolos de capa MAC y enrutamiento se encuentran disponibles en las carpetas del software, y la red MANET es muy fácil de escalar y hacer cambios rápidos en la configuración para fines de diferentes análisis.

El capítulo más importante es el 4, ya que de aquí es donde partimos para la propuesta de nuestro algoritmo al hacer un breve estudio del estado del arte de los diferentes algoritmos para la administración de paquetes. Como ya se había plateado anteriormente en el grupo de trabajo, se propuso utilizar los paquetes de video priorizados en cuatro tipos por la capa de aplicación (P1,P2,P3 y P4) para la administración de estos en el buffer, sin embargo, los resultados obtenidos nos demostraban que el TTL de muchos paquetes se expiraba, y al analizar la causa nos dimos cuenta que muchos paquetes que ya tenían tiempo de vida en la red eran colocados al final de la cola, por lo que antes de ser el turno para enviarse su TTL había expirado. Así que se decidió organizar primero los paquetes de acuerdo a su TTL ayudándonos con el *timestamp* de cada paquete.

La decisión del entorno de evaluación utilizado en este proyecto fue principalmente para seguir la secuencia de mejora en esta área de investigación: Administración de paquetes, que el grupo

de multimedia y redes inalámbricas de la Universidad de San Diego (SDSU *San Diego State University*) ha seguido en los últimos años. Además como ya habíamos comentado, la red MANET es fácil de implementar, cambiar protocolos y escalar. Sin embargo debido a la forma de implementación del algoritmo, este puede ser utilizado en diferentes tipos de redes inalámbricas (WiMax, LTE) e inclusive redes alámbricas.

También elegimos seguir trabajando con el software ns-2.34 debido a que es un software libre y nos permite crear nuestro entorno de evaluación incluyendo archivos nuevos como es el caso del protocolo de enrutamiento AOMDV. En SDSU el ns-2.34 es el único simulador que utilizan para la evaluación de los algoritmos de redes. Sería una buena opción poder implementar el mismo algoritmo en otro simulador como lo es el OPNET para probar el desempeño.

El uso del Evalvid nos permitió hacer una relación directa entre el códec y el ns-2.34 para poder simular la transmisión de video en tiempo real, respetando todas las características del video especificadas por el códec, como lo son velocidad de datos, tamaño y número del *slice* y del GOP, y lo más importante: utilizando la priorización de los *slices* y su *timestamp*.

Nuestro primer objetivo fue elegir una adecuada administración de paquetes en el buffer. La mejor elección fue ordenarlos primero por su *timestamp* por dos razones: debido a la sensibilidad que tiene el streaming de video al retardo y para tener un control de la secuencia de tramas, ya que el número de trama está directamente ligado a su *timestamp*. Después considerando que una trama está compuesta de diferentes *slices*, y como ya se había mencionado que para fines de simplicidad, cada paquete está formado de un solo *slice*, por lo tanto cada trama está formada de paquetes de diferentes prioridades, así que posteriormente ordenamos los paquetes por su prioridad. Al final, pensando en ofrecer una QoS a usuarios preferentes, aplicamos una última ordenación que es por usuario. Los resultados obtenidos presentados en el capítulo 7 nos muestran mejoras de PDR, retardo y PSNR del escenario de evaluación. Estos resultados nos indican que en promedio obtenemos en total de cada simulación más paquetes de alta prioridad y con menor retardo con nuestro esquema propuesto en comparación con la FIFO, y esto prácticamente se debe a que se transmiten primero los paquetes más viejos dentro del buffer (ordenación por *timestamp*) por lo que mejoramos que haya menos paquetes expirados por TTL.

Así también con la ordenación por prioridad y con la diferencia de TTL para cada paquete aseguramos que cuando haya oportunidad de poder transmitir paquetes, los primeros sean los de alta prioridad, esto es parte de nuestro segundo objetivo, y como bien se había previsto, el checar

el TTL en el buffer de cada nodo nos permite eliminar paquetes que ya se expiraron en la cola antes de que sean transmitidos y así no gasten recursos de la red.

Estudiamos el tamaño del paquete, y después de varias simulaciones, el tamaño más óptimo para las condiciones de nuestros escenarios de simulación fue de 900 bytes como máximo; siempre debe haber un compromiso para la cantidad de datos transmitidos entre la longitud de los paquetes y la longitud de los encabezados.

Los resultados presentados en esta tesis fueron obtenidos después de varios análisis y simulaciones con paquetes de video H.264/AVC. Al inicio de este proyecto se había pensado en la administración de paquetes sólo por su priorización, sin embargo los resultados obtenidos no eran muy buenos debido a que muchos paquetes gastaban mucho tiempo en la cola en espera de ser transmitidos por lo cual se les expiraba su TTL correspondiente. Se perdían muchos paquetes con prioridad 2 y 3. Además de que los paquetes que lograban llegar al nodo receptor provenían de GOP distintos y había muchos problemas para el control de las tramas de video.

La opción propuesta de ordenar los paquetes primero por su *timestamp* y después por su prioridad solucionó ambos problemas. De acuerdo a los resultados obtenidos, se mejoró dentro del rango esperado la calidad del video recibido por el usuario: en medidas de PSNR obtuvimos una mejora de 0.5 a 1.5 dB en comparación con el desempeño del esquema FIFO; y midiendo el retardo, en algunos escenarios se logró reducirlo hasta la mitad.

Si analizamos los resultados obtenidos en nuestro esquema por prioridad de usuario, notamos una mejora en el PSNR en general para todos los usuarios, esto debido a que obtenemos mayor cantidad de paquetes de prioridad alta para cada usuario. Sin embargo, el objetivo de este algoritmo era tener una gran incremento del PSNR del usuario 1 en comparación con los demás usuarios, sin embargo la red manet no es el mejor escenario para evaluar este tipo de algoritmo debido a que no se tiene un control del acceso al medio, lo cual significa que a pesar de que en la administración del buffer, ponemos en la cabeza de la cola al usuario 1, que es el más importante, si no tenemos un control de acceso al medio que le permita a este usuario transmitir primero que los demás usuarios, no habrá mucha mejora. Sería ideal que este algoritmo sea complementado con otro algoritmo de acceso al medio (como puede ser el PCF del estándar 802.11 u otro), o otro entorno de evaluación cuyas características se ajusten a la idea de prioridad por usuario.

Contribuciones

- Los algoritmos propuestos en esta tesis para la administración de paquetes de video H.264/AVC es una parte de un trabajo en conjunto con el grupo de multimedia y redes inalámbricas en *San Diego State University*; ayuda a mejorar la calidad de video recibido y formará parte de una serie de algoritmos que se están elaborando para un esquema de capas cruzadas más robusto el cual proporcionará una mejor QoS y QoE al usuario final.
- Se desarrolló un algoritmo cuyos resultados presentados prueban que se tiene una mejora en el retardo punto a punto; en algunos escenarios reduciéndolo a casi la mitad y un incremento en el PSNR de entre 0.5 a 1.5 dB.
- Principalmente con el trabajo realizado en esta tesis, se busca iniciar una línea de investigación más efectiva a la solución de la calidad de video recibida en un ambiente de redes inalámbricas; esperando que se generen más ideas que complementen este algoritmo para hacerlo más robusto.

Trabajo Futuro

Debido a la versatilidad de nuestro algoritmo, hay mucho trabajo a futuro que se puede seguir, entre el cual podemos mencionar lo siguiente:

- Probar el desempeño de nuestro algoritmo en diferentes tipos de redes inalámbricas, con la finalidad de ver los pros y contras que puede ofrecer en diferentes ambientes.
- También es posible probar la eficiencia del algoritmo en redes alámbricas.
- Realizar más simulaciones con algunas modificaciones en los escenarios presentados previamente tales como: cambiando la capacidad del canal, la velocidad de datos del video, el tamaño del buffer y aumentar la cantidad de usuarios.
- Si se pensara en hacer más robusto el algoritmo, se podría proponer un esquema de capas cruzadas donde se tome en cuenta el retardo en tiempo real como medida de la congestión actual en la red, se implemente un control de acceso al medio y se encuentre la mejor ruta para paquetes de video de alta prioridad.
- Tomar en cuenta las pérdidas de paquetes en ráfagas y complementar el sistema con alguna técnica de cancelación de error.
- El algoritmo por prioridad de usuario es importante que se evalúe en un ambiente donde se controle el acceso al medio para poder aprovechar sus ventajas, y posteriormente realizar los ajustes necesarios.

Referencias

- [1] Haohong Wang, Lisimachos P. Kondi, Ajay Luthra, Song Ci, "4G Wireless Video Communications", Ed. WILEY, 2009.
- [2] Iain E. Richardson, "The H.264 Advanced Video Compression Standard", Ed. WILEY, segunda edición, 2010.
- [3] Schwarz H., Heinrich Hertz, Marpe, D., Wiegand, T., "Overview of the H.264/AVC video coding standard", Circuits and Systems for Video Technology, IEEE Transactions on (Volume:13 , Issue: 7), July 2003.
- [4] Wiegand T. Sullivan G.J. , Bjontegaard, G., Luthra, A., " Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", Circuits and Systems for Video Technology, IEEE Transactions on (Volume:17 , Issue: 9), September 2007.
- [5] Iain E. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia" Ed. WILEY, segunda edición, 2003.
- [6] Stockhammer T, Hannuksela MM, Wiegand T, "H.264/AVC in wireless environments". IEEE Transactions on Circuits and Systems for Video Technology, 2003; 13(7): 657–673.
- [7] Sem Borst, " User-level performance of channel-aware scheduling algorithms in wireless data networks ", Journal IEEE/ACM Transactions on Networking (YON), June 2005; Volume 13 Issue 3, Pages 636-647
- [8] Song, G., Ye Li ; Cimini, L.J. ; Zheng, H. "Joint channel-aware and queue-aware data scheduling in multiple shared wireless channels", Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE (Volume:3).
- [9] Sanjay Shakkottai, Alexander L. Stolyar, "Scheduling for Multiple Flows Sharing a Time-Varying Channel: The Exponential Rule", Bell Labs, Lucent Technologies.
- [10] Alexander L. Stolyar, " On the Asymptotic Optimality of the Gradient Scheduling Algorithm for Multiuser Throughput Allocation " . Bell Laboratories, Lucent Technologies, Volume 53 Issue 1, January-February 2005.
- [11] Hong MC, Kondi L, Scwab H, Katsaggelos AK., "Video error concealment techniques". Signal Processing: Image Communications 1999; 14(6–8): 437–492.
- [12] Wang Y-K, Hannuksela MM, Varsa V, Hourunranta A, Gabbouj M., "The error concealment feature in the H.26L test model". In Proceedings of the International Conference on Image Processing 2002; 2: 729–732.
- [13]. Sun H, Kwok W., "Concealment of damaged block transform coded images using projection onto convex sets". IEEE Transactions on Image Processing 1995; 4(4): 470–477.
- [14] Belfiore S, Grangetto M, Magli E, Olmo G., "Spatio-temporal video error concealment with perceptually optimized mode selection". In Proceedings of the IEEE International Conference on Multimedia and Expo 2003; 169–172.
- [15] Wang Y, Zhu Q-F, "Error control and concealment for video communication: A review", Proceedings of the IEEE 1998; 86(5): 974–997.

-
- [16] http://compression.ru/video/quality_measure/info_en.html#start
- [17] Armando Mercado, Rafael Gil Berríos Figueroa, Paul Chan Ye, “Redes inalámbricas ad hoc”. Resume.
- [18] Byeong Gi Lee, Sunghyun Choi, "Broadband Wireless Access and Local Networks: Mobile WiMAX and WiFi", Ed. ARTECH HOUSE, 2008.
- [19] Fallah, Y.P., "Enhanced controlled-access and contention-based algorithms for IEEE 802.11e wireless LAN", Wireless Networks, Communications and Mobile Computing, International Conference, 2005.
- [20] Francisco Alberto Rodríguez Mayol, Javier Gozalvez Sempere “Técnicas de Reputación para Redes de Comunicaciones Inalámbricas Multi-Salto”, Tesis para el grado de Doctor. Universidad Miguel Hernández de Elche Departamento de Ciencias de Materiales, Óptica y Tecnología Electrónica
- [21] Sunil Kumar, Siddharth Khimsara, Kashyap Kambhatla, "Robust On-Demand Multipath Routing with Dynamic Path Upgrade for Delay-Sensitive Data over Ad Hoc Networks", Wireless Networks, Communications and Mobile Computing, International Conference, 2005. Hindawi Publishing Corporation, Journal of Computer Networks and Communications, Volume 2013, Article ID 791097.
- [22] Mahesh K. Marina, Samir R. Das, "Ad hoc on-demand multipath distance vector routing", WIRELESS COMMUNICATIONS AND MOBILE COMPUTING, 2006.
- [23] S. K. Sivagurunathan and A. Mohammed. (2006, Apr.). "Multimedia over Wireless Networks [Online]". Available: <http://www.cs.ou.edu/~netlab/TR/MultimediaBookchap-TNRL-06-105.pdf>.
- [24] Hossam Fattah, Cyril Leung, "An Overview of Scheduling Algorithms in Wireless Multimedia Networks". IEEE Wireless Communications, October 2002.
- [25] K. Seong-ryong *et al.*, “Multi-layer active queue management and congestion control for scalable video de streaming,” in *24th Int. Conf. Distributed Computing Syst.*, Tokyo, JP, 2004, pp. 768-777.
- [26] M. Ito and Y. Bai, “A packet discard scheme for loss control in IP networks with MPEG video traffic,” in *8th Int. Conf. Commun. Syst.*, Singapore, 2002, pp. 497-503. 56.
- [27] W. Shujie *et al.*, “An adaptive queue management mechanism for video de streaming over mobile ad hoc networks,” in *5th Int. Conf. Wireless Commun. Networking and Mobile Computing*, Beijing, CN, 2009, pp. 1-4.
- [28] J. Deng, “Introduction to grey system theory,” *J. Grey Syst.*, vol. 1, no. 1, pp.1-24, 1989.
- [29] G. Liebl *et al.*, “Radio link buffer management and scheduling for wireless video de streaming,” in *Telecommun. Syst.*, vol. 30, no. 1-3, pp. 255-277, Nov. 2005.
- [30] G. Liebl *et al.*, “Joint buffer management and scheduling for wireless video de streaming,” in *Proc. 4th Int. Conf. on Networking*, Reunion, FR, 2005, pp. 882-891.
- [31] D. Tennenhouse *et al.*, “A survey of active network research,” in *IEEE Commun. Mag.*, vol. 35, no. 1, pp. 80-86., Jan. 1997.

-
- [32] M. Ito and Y. Bai, "User-oriented fair buffer management for MPEG video streams," in *Proc. 17th Int. Conf. on Advanced Networking and Applicat.*, Xi'an, CN, 2003, pp. 241-247.
- [33] M. Van der Schaar *et al.*, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Sel. Areas Commun.*, vols. 21, no.10, pp. 1752-1763, Dec. 2003.
- [34] B. Zheng and M. Atiquzzaman, "A novel scheme for de *streaming* multimedia to personal wireless handheld devices," *IEEE Trans. on Consum. Electron.*, vol. 49, no. 1, pp. 32-40, Feb. 2003.
- [35] K. Fall and K. Varadhan. (2011, Nov. 4). *The ns Manual (formerly ns Notes and Documentation)* [Online]. Available: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.
- [36] Textbook: T. Issariyakul and E. Hossain, "*Introduction to Network Simulator NS2*", Springer 2008.
- [37] <http://www2.tkn.tu-berlin.de/research/evalvid/EvalVid/example.html>