VNIVERSIDAD NACIONAL AVFNOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"ADAPTACIÓN DE TCP A REDES INALÁMBRICAS 802.11G"

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA (C O M P U T A C I Ó N)

P R E S E N T A

JONATHAN EMMANUEL LÓPEZ FIGUEROA

DIRECTOR DE TESIS:
DR. JAVIER GÓMEZ CASTELLANOS

MÉXICO, D.F. 2008.

DEDICATORIAS

A mi esposa Araceli:
Por todos los momentos que hemos compartido, sueños y anhelos, secretos, risas y lágrimas, y sobre todo amor. Por dedicarme tiempo, tiempo para demostrar tu preocupación por mí, tiempo para escuchar mis problemas y ayudarme a buscarles solución, y sobre todo, tiempo para sonreír y mostrarme tú cariño. Por darme una familia llena de dicha y amor que me ha ayudado a poder crecer personal y profesionalmente. Te Amo...

A mis padres:
Ana Elizabeth Figueroa Morales y Gustavo López Bustos quienes me han heredado el tesoro más valioso que puede dársele a un hijo: amor. A quienes sin escatimar esfuerzo alguno, han sacrificado gran parte de su vida para formarme y educarme. A quienes la ilusión de su vida ha sido convertirme en persona de provecho. A quienes nunca podré pagar todos sus desvelos ni aún con las riquezas más grandes del mundo....

A mis abuelos:
Ana María Morales Carvajal y Luis Figueroa Cruz ✝
Como un testimonio de cariño y eterno agradecimiento por mi existencia, valores morales y formación profesional. Porque sin escatimar esfuerzo alguno, han sacrificado gran parte de su vida para formarme cuando mis padres no lo podían hacer y porque nunca podré pagar todos sus desvelos ni aún con las riquezas más grandes del mundo. Por lo que soy y por todo el tiempo que les robé pensando en mi...

AGRADECIMIENTOS

Al resto de mi familia y amigos:
Al término de esta etapa de mi vida, quiero expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión me alentaron a logar esta hermosa realidad y por todo el apoyo que me han dado, con quienes he podido compartir alegrías y tristezas …

A la Universidad Nacional Autónoma de México:
Por darme la grandiosa oportunidad de estudiar en esta institución y así formar parte de ella…

A mi asesor, el Dr. Javier Gómez Castellanos:
Por la confianza puesta en esta tesis para así dirigirla y apoyarme en todo lo necesario para poder terminarla…

Al Consejo Nacional de Ciencia y Tecnología:
Por el apoyo brindado a través de su programa de becas, el cual me ha permitido sustentarme durante un período en mis estudios…

A los revisores de esta tesis:
Dr. Héctor Benítez Pérez
Dr. Sergio Rajsbaum Gorodesky
Dr. Víctor Rangel Licea
Ing. Mario Rodríguez Manzanera
Por todas las sugerencias y comentarios que me ayudaron a mejorar esta tesis…

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la actualidad el ser humano tiene la necesidad de comunicarse con sus semejantes en cualquier lugar y en cualquier momento, es por esto que las diferentes formas de comunicación han ido evolucionando desde las señales de humo hasta las más novedosas comunicaciones vía satélite.

Por lo anterior se realizó un trabajo de tesis en el que se involucra algún aspecto relacionado con las comunicaciones. En la maestría enfoqué mis estudios hacia el área de Redes de Computadoras, en específico el área de Redes Inalámbricas; de aquí también el interés por hacer un trabajo donde se trate sobre comunicaciones inalámbricas.

Esta tesis lleva por título *Adaptación de TCP a Redes Inalámbricas 802.11g*; se optó por este tema debido al creciente impacto que están teniendo las redes inalámbricas en la vida del ser humano.

Actualmente no podemos pensar en como sería la vida sin los medios de comunicación impresos o aún más sin la radio o la televisión, que de algún modo son comunicaciones inalámbricas. En un futuro muy cercano tampoco podremos imaginar como será la vida sin las redes inalámbricas, ya que muchas de las tecnologías se están enfocando hacia este tipo de comunicaciones.

CAPÍTULO I
EL PROYECTO DE ADAPTACIÓN DE TCP A REDES INALÁMBRICAS 802.11G

1.1. DEFINICIÓN DEL ÁREA

En el mundo de las redes inalámbricas, el estándar IEEE (The Institute of Electrical and Electronics Engineers) 802.11 es el más usado en todo el mundo, y en particular en nuestro país, por lo que se decidió tomarlo como base para este trabajo.

Por otra parte, las redes de computadoras, ya sean alámbricas o inalámbricas no tienen importancia sin los protocolos de comunicación, ya que éstos establecen una descripción formal de los formatos que deberán presentar los mensajes para poder ser intercambiados entre diferentes equipos de cómputo. Además también definen las reglas que deben seguir dichos equipos para poder lograr esa comunicación.

El protocolo TCP (Transmission Control Protocol) forma parte de la pila de protocolos de TCP/IP (Transmission Control Protocol/Internet Protocol), la cual provee una plataforma de comunicación ínter operable entre todo tipo de hardware y software, de este modo, en la actualidad la mayoría de las aplicaciones de redes de computadoras requieren transmitir datos usando TCP como protocolo de transporte. Por lo mencionado anteriormente se optó por realizar una tesis donde se involucrara por una parte las redes inalámbricas y por otra el protocolo TCP.

Básicamente esta tesis tratará sobre una revisión del uso de TCP en redes inalámbricas y en que forma la movilidad de un equipo de cómputo, limitado por las capacidades del estándar IEEE (Institute of Electrical and Electronics Engineers) 802.11g, afecta al mecanismo de "*inicio lento e impedimento del congestionamiento*"

desarrollado en TCP para optimizar el uso del ancho de banda para transmisión. Para lo anterior se utilizará el simulador de redes *Network Simulator NS-2*.

Imagine una pequeña red de solo dos equipos, los mecanismos de inicio lento y control de congestionamiento de TCP trabajarán bien si esta red es cableada, ya que el ancho de banda siempre será el mismo a través de todo el enlace. Sin embargo, en redes inalámbricas todo cambia por completo, ya que el ancho de banda disponible cambia de acuerdo a la distancia entre los equipos y por la calidad de la señal recibida por los radios de estos equipos.

El mecanismo de inicio lento de TCP permite enviar datos siempre dentro del ancho de banda disponible para transmisión en relación a otras variables como la ventana de congestionamiento o el MSS (Maximum Segment Size), entre otras; por el momento olvide todas las demás variables y enfoquémonos en el ancho de banda.

Volviendo a nuestra pequeña red, suponga que el equipo A envía datos al equipo B, el equipo B es un punto de acceso que se encuentra fijo en una ambiente determinado, mientras que el equipo A es un equipo móvil que se encuentra en movimiento a diversas velocidades dentro del área de cobertura del equipo B.

Ahora imagine que el equipo A se encuentra a una distancia tal que puede enviar datos con una velocidad de hasta 54 Mbps con destino a B, el mecanismo de inicio lento poco a poco adaptará la velocidad hasta que empiece a haber congestionamiento en la red; pero que pasaría si en el momento en que TCP envía datos a 22 Mbps y el mecanismo de inicio lento sube exponencialmente a 44 Mbps, el equipo A se mueve a una zona donde el ancho de banda es de 5.5 Mbps, seguramente la mayoría de los datos se perderán y el mecanismo de disminución multiplicativa no serviría de mucho.

Es en lo anterior donde esta investigación se enfoca y trata de analizar el efecto que tiene el movimiento de los equipos, y por consiguiente la adaptación automática de la tasa de datos a diversos mecanismos de TCP que ayudan al control del congestionamiento de la red, lo que corresponde al título de este trabajo.

Cabe señalar que el simulador corre bajo un ambiente Unix o Linux, o en su defecto en el emulador para Windows Cygwin, por lo que es necesario también conocer al menos los comandos básicos para dichos sistemas.

1.2. OBJETIVOS DEL PROYECTO

Los objetivos del proyecto son: Hacer las modificaciones necesarias de las capas 1 y 2 del modelo OSI que maneja el simulador NS – 2 y así poder simular redes inalámbricas 802.11G. Una vez funcionando el simulador hacer una serie de simulaciones de ambientes inalámbricos usando TCP como protocolo de transporte, para a continuación analizar como afecta la movilidad de un equipo inalámbrico al comportamiento de los mecanismos de control del congestionamiento de la red de TCP, tomando como base los resultados de las simulaciones.

Basándose en la hipótesis de que TCP, al haber sido concebido para redes cableadas y al agregado de datos de la misma arquitectura de una red inalámbrica, lo cual causará una baja eficiencia en TCP, como objetivo secundario restará, el realizar algún algoritmo que complemente a TCP y resuelva los problemas originados por la movilidad, o de ser posible sugerir un nuevo protocolo similar a TCP pero con un mejor control de congestionamiento para redes inalámbricas, tema que puede ser tratado aparte en un trabajo quizá de doctorado.

1.3. CONTRIBUCIÓN Y RELEVANCIA

Un mejor control del congestionamiento en la transmisión de datos se traduce en un mejor aprovechamiento de la red, es decir, más bits por segundo para el usuario y sus aplicaciones. Los resultados que se obtengan podrían servir de base para futuros estándares de redes inalámbricas, así como de protocolos de transporte, con el fin de obtener tasas de transmisión mucho más altas a las actuales.

1.4. METAS DEL PROYECTO

Son varias las metas propuestas al realizar este proyecto, cada una va de la mano con la anterior y se expresan de la siguiente forma:

- Implementar el estándar IEEE 802.11G en el simulador NS – 2.
- Realizar un estudio del control de congestión de TCP en el simulador NS – 2.
- Proponer mejoras a TCP en redes WLAN (Wireless Local Area Network).
- Proveer a la comunidad científica del área de Redes de Computadoras, de un análisis alternativo para optimizar el desempeño de las redes inalámbricas de área local.

1.5. METODOLOGÍA

Para realizar la adaptación del simulador NS – 2 con 802.11G es necesario tener conocimientos del lenguaje de programación C así como de TCL (Tool Command Languaje) ya que se modificarán los archivos mac-802_11.cc y mac-802_11.h que están escritos en C, así como los archivos ns-default.tcl y ns-mac.tcl que están escritos en TCL.

Se crearán simulaciones con ambientes inalámbricos en lenguaje TCL para poder ser simulados en el NS – 2, usando primero UDP (User Datagram Protocol) como protocolo de transporte para poder observar la diferencia de salidas con respecto a TCP, para después cambiar a TCP. Es también necesario tener conocimientos del formato que emplea el simulador en los archivos de salida que genera durante las simulaciones, ya que estos suelen ser en ocasiones muy extensos y complejos.

Una vez analizados los archivos de salida del simulador, se procederá a realizar comparaciones entre los diferentes archivos y realizar gráficas para poder interpretar de mejor forma los resultados obtenidos y así poder dar las conclusiones adecuadas.

1.6. TRABAJO RELACIONADO

Existen distintos trabajos en todo el mundo relacionados con la adaptación de TCP a redes inalámbricas, la mayoría de ellos están enfocados en redes inalámbricas 802.11b, analizando el rendimiento de TCP en este tipo de redes. Algunos otros analizan tan solo hasta el estándar 802.11 puro, mientras este trabajo se enfoca en las redes WLAN 802.11G.

Por otra parte el rendimiento de TCP puede ser analizado haciendo uso de varias herramientas, mientras en muchos trabajos se utilizan herramientas como OPNET, otros utilizan una red local con un generador de paquetes mientras se realizan capturas con el sniffer Ethereal, así estas capturas son analizadas para verificar el tráfico real a través de varios scripts generados con la utilidad GAWK. Entre otras herramientas la mas usada en los muchos trabajos relacionados a TCP en redes inalámbricas es el NS-2 por su habilidad de simular redes 802.11b.

Este trabajo difiere de otros en que el NS – 2 fue manipulado para poder simular redes 802.11G, de lo cual se han hecho pocos trabajos. Otra diferencia radica en que muchos de los estudios que se han realizado solo se enfocan en la comunicación del último salto en enlaces inalámbricos, mientras que en este trabajo se analizarán varios escenarios donde se podrá observar el rendimiento de TCP a través de todo un enlace desde que sale de un transmisor inalámbrico hasta que llega a un nodo alámbrico luego de pasar por otros equipos, lo que comúnmente sucede en la realidad.

Algunos casos de estudio se enfocan en el retardo que ocasionan otros protocolos de red de las WLAN, otros analizan el comportamiento de las distintas implementaciones existentes de TCP, como son TCP Tahoe, Reno, NewReno y Sack; otros trabajos se enfocan en implementaciones especiales no verificadas, hechas por otras personas o fabricantes. Esta tesis se enfocará en TCP puro en el cual se tienen las peores pérdidas en redes inalámbricas.

1.7. ESTRUCTURA DE LA TESIS

Esta tesis consta de siete capítulos incluido el actual, en el segundo capítulo se encontrarán algunos conceptos básicos como son la definición de redes inalámbricas y su clasificación, así como los fundamentos primordiales que se usan en las redes inalámbricas.

En el tercer capítulo el lector podrá encontrar temas relacionados al estándar IEEE 802.11, en particular al IEEE 802.11G, como son la pila de protocolos, su capa física, la capa MAC (Medium Access Control), entre otros.

El cuarto capítulo muestra las bases de TCP y del protocolo de datagrama de usuario (*UDP*) como sus cabeceras, la forma en que establecen una conexión y otros puntos más.

En el quinto capítulo se presenta un estudio del simulador de redes NS-2 incluyendo su historia, evolución, aplicaciones y la forma en que ha sido utilizado y manipulado para poder simular los escenarios necesarios en el proyecto, ya que fue necesario hacer diversas modificaciones al propio código del simulador así como al código de las simulaciones por realizar. También se hablará sobre el ambiente de desarrollo *LINUX* y *Cygwin*.

Será en el sexto capítulo en donde se presenta la implementación del estándar IEEE 802.11G en el simulador de redes NS – 2, se tratarán los distintos archivos que fueron modificados para que el simulador funcionara como se requería.

En el séptimo capítulo se da una explicación de los experimentos realizados con ayuda del simulador para posteriormente interpretar los resultados y poder dar las conclusiones y recomendaciones oportunas.

Finalmente se presentan las conclusiones generales y un pequeño resumen del trabajo realizado, analizando si es viable el objetivo del proyecto, el cual es básicamente el analizar el protocolo TCP en redes inalámbricas y en base a diferentes resultados poder dar una propuesta de algún algoritmo que complemente a TCP para poder funcionar de manera satisfactoria en dichas redes, o bien el proponer la creación de un nuevo protocolo de transporte inmune a la movilidad.

CAPÍTULO II
ANTECEDENTES DE REDES INALÁMBRICAS

2.1. DEFINICIÓN DE RED INALÁMBRICA

Con el advenimiento de nuevas tecnologías en todos los ámbitos, los patrones de trabajo se encuentran cambiando y más gente necesita acceder a redes o bien a Internet, desde cualquier lugar, un ejemplo tangible de esto se presenta al proveedor de servicios de telecomunicaciones e Internet al serle mas fácil y recomendable brindar a sus usuarios el acceso a sus servicios sin alambres que cablear a cada uno de ellos, siendo más sencilla la incorporación de un nuevo usuario a una red inalámbrica que a una totalmente alambrada.

Con los nuevos productos y tecnologías inalámbricas los usuarios podrán acceder a las redes corporativas e Internet desde su casa, de camino al trabajo o la escuela, o en la carretera sin una conexión física. En un futuro muy cercano, la velocidad de los dispositivos inalámbricos se incrementará dramáticamente debido en gran medida a las nuevas tecnologías inalámbricas y a los nuevos estándares, los cuales permitirán la interoperabilidad entre los equipos y compatibilidad entre las redes.

Con esto todos los fabricantes de equipos inalámbricos incrementarán sus ventas y al mismo tiempo se decrementarán poco a poco los precios de los productos inalámbricos como lo hemos visto en los últimos años. No obstante no hay que olvidar que un dispositivo inalámbrico se interconecta a las redes pasando por un dispositivo que sí requiere de un cableado, un ejemplo: el AP (Access Point). Siendo la excepción a la regla la comunicación inalámbrica de PC (Computadora Personal) a PC la cual requiere solamente que cada una de ellas cuente con la tarjeta correspondiente.

En las redes alámbricas los medios físicos de transmisión han sido diferentes tipos de cables. Un costo importante asociado a estas redes es el de instalar el cableado físico. Además si se modifica la disposición de las computadoras interconectadas, se puede incurrir en un costo similar al de la instalación original para cambiar el plan del cableado. Esta es una de las razones por las que han aparecido las redes inalámbricas, es decir, redes que no usan cables físicos como medio de transmisión.

Una segunda razón es la aparición de las terminales manuales y de las computadoras portátiles. Aunque la razón primordial para usar estos dispositivos es su transportabilidad, a menudo tienen que comunicarse con otras computadoras que pueden ser también computadoras portátiles, o lo que es más probable computadoras conectadas a una red por cable.

Es tiempo de definir el término red inalámbrica. Se manejará la siguiente definición: una red inalámbrica es un sistema de comunicación que permite que dos o más computadoras o dispositivos electrónicos intercambien información, recursos y/o servicios sin usar un cable físico como medio de transmisión, es decir usando ondas de radiofrecuencia.

Es importante mencionar que para no estar teniendo que hacer la distinción entre computadoras PC's o algún otro equipo electrónico, se llamará Nodo a cualquiera de los anteriores que se encuentre conectado a la red inalámbrica. La señal transmitida por un nodo solo puede ser percibida dentro de cierta área de transmisión, a la cual se llamará rango del nodo.

2.2. CLASIFICACIÓN DE LAS REDES INALÁMBRICAS

Las redes inalámbricas se pueden clasificar en tres categorías: WAN/MAN (redes de área amplia/redes de área metropolitana), LAN (red de área local) y PAN (redes de área personal).

En la primer categoría WAN/MAN están las redes que cubren desde decenas de metros hasta miles de kilómetros. En la segunda categoría LAN, están las redes que comprenden varios metros hasta decenas de metros. Y en la última y más nueva categoría PAN se encuentran las redes que comprenden desde centímetros hasta 30 metros.

En la categoría MAN/WAN existe primeramente el acceso por telefonía celular; otras tecnologías WAN/MAN inalámbricas que permiten el acceso a redes de datos o Internet son MMDS (Multichannel Multipoint Distribution Service), LMDS (Local Multipoint Distribution Service), WLL (Wireless Local Loop), enlaces de microondas terrestres, enlaces vía láser infrarrojo y comunicaciones vía satélite.

En la segunda categoría de redes locales, las inalámbricas se han vuelto muy populares hoy en día, éstas pueden proveer de acceso a redes cableadas e Internet. En esta categoría existen diversos estándares para poder transmitir de diferentes maneras y a diferentes velocidades. Por otra parte, las redes tipo PAN cubren distancias cortas y cerradas, utilizando entre otras tecnologías el Bluetooth, 802.15 y HomeRF.

Dentro de las categorías mencionadas anteriormente este trabajo se enfoca en la segunda, las redes inalámbricas de área local WLAN (*Wireless LAN*). En los diferentes estándares que existen para las WLAN se definen básicamente dos tipos de estructuras: Redes Centralizadas y Redes Distribuidas.

"En cada una estas dos topologías existe el Conjunto de Servicio Básico (*Basic Service Set, BSS,* por sus siglas en inglés), que consiste en dos o más *nodos* a veces conocidos como estaciones. Un BSS tiene dispositivos que se reconocen y trabajan en conjunto unos con otros para minimizar la cantidad de colisiones que existen dentro del dominio del BSS."[1]

[1] Reid, Neil. *802.11 (Wi-Fi) Manual de Redes Inalámbricas*. Ed. Mc Graw Hill. México 2004. P. 68.

Las Redes Inalámbricas Centralizadas son también llamadas Redes con Infraestructura o también Redes de Último Salto. En este tipo de redes "mediante un dispositivo intermedio llamado unidad de acceso portátil (PAU: *portable access unit*) se obtiene acceso a una computadora servidor conectado a una LAN por cable. Por lo regular, el campo de cobertura de la PAU es de 50 a 100 m, y en una instalación grande hay muchas de estas unidades distribuidas dentro de un sitio.

En conjunto, éstas proporcionan acceso a la LAN del sitio – y por tanto a las computadoras servidores – a través de una terminal manual, una computadora portátil o una computadora estática, todas las cuales pueden estar ubicadas en cualquier punto del sitio."[2] Generalmente las redes inalámbricas centralizadas son una extensión de una red cableada. Este tipo de redes están ilustradas en la figura 2.1.

La PAU es lo que comúnmente se conoce como Punto de Acceso (*Access Point, AP*) o Estación Base, éste dispositivo no es móvil y generalmente se encuentra ubicado en un punto central de la red inalámbrica, ya que este equipo es la base del sistema y como cuenta con antenas de transmisión ominidireccionales, el área de cobertura es una esfera donde el centro es el AP. También es posible colocar antenas direccionales al AP para hacer la función de puente entre redes y otras aplicaciones, a través de un enlace inalámbrico.

Las redes con infraestructura usan dos tipos de canal para transmisión, uno es de bajada, es decir de la estación base al nodo, y otro es de subida, del nodo a la estación base. El canal de bajada es de tipo *broadcast*, lo que significa que puede ser escuchado por todos los nodos que se encuentren en el área de cobertura del punto de acceso. El canal de subida en cambio, es del tipo de acceso múltiple, de tal modo que todos los usuarios comparten el canal, y puede ser administrado por la estación base dependiendo de varios factores.

[2] Halsall, Fred. *Comunicación de Datos, Redes de Computadores y Sistemas Abiertos*. Ed. Pearson Education. Edic. 4ª. México 1998. P. 332.

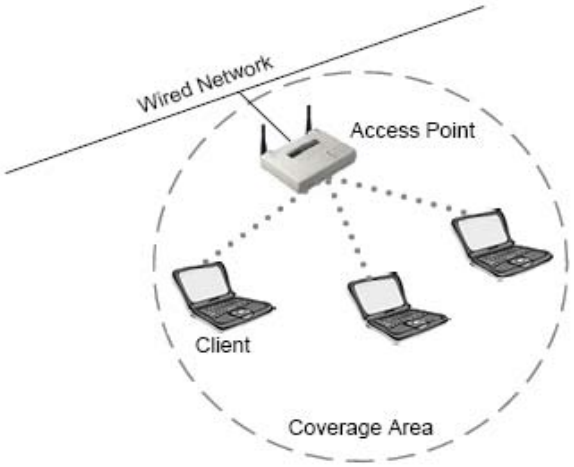

Figura 2.1 Red inalámbrica con infraestructura.

Estas redes pueden operar tanto por multiplexión por división de tiempo (TDM) como por multiplexión por división de frecuencia (FDM).

"La referencia apropiada para los clientes y AP en una red de infraestructura es Conjunto de servicio extendido (*Extended Service Set*, *ESS*, por sus siglas en inglés), debido a que este término incluye dispositivos que provienen de más de un BSS y normalmente está conectado mediante Ethernet a través de un sistema de distribución, como una LAN, a lo largo de toda una empresa. Los clientes pueden desplazarse dentro de los BSS, por lo que proporcionarán una conectividad sin problemas a los usuarios cuando están dentro de sus redes."[3]

A las Redes Inalámbricas Distribuidas también se les llama redes Ad Hoc debido a que estas redes se crean por demanda. "Normalmente están compuestas de dos o más clientes que son iguales entre ellos, por ejemplo, computadoras portátiles o PDA con tarjetas 802.11 integradas. Una red ad-hoc suele conocerse como un *conjunto de servicio básico independiente* (*Independent Basic Service Set*, *IBSS*, por sus siglas en inglés), donde la palabra *independiente* se refiere al hecho de que no existe un punto de acceso (AP) dentro de este conjunto de servicio. Las redes ad-hoc tienden a ser temporales."[4]

[3] Reid, Neil. Op. Cit. P. 69.
[4] Ibídem. P. 68.

La figura 2.2 muestra una red ad-hoc, con ejemplos de computadoras portátiles y una computadora de escritorio con una tarjeta de interfaz inalámbrica. Cabe destacar que comúnmente a la tarjeta de red se le conoce como NIC (*Network Interface Card*, de sus siglas en inglés), por lo que así se le llamará en lo sucesivo.

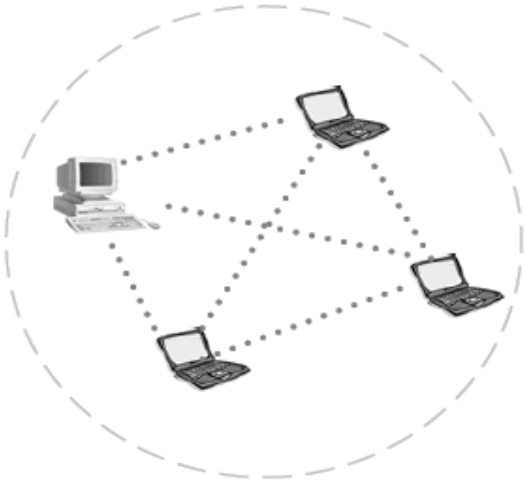

Figura 2.2 Red inalámbrica Ad-Hoc.

Como una red Ad-Hoc no tiene equipos de infraestructura, es autónoma, lo que significa que la red no colapsa si uno de sus nodos se mueve de lugar o se cae. Como se vió antes, una red con infraestructura puede usar ambas opciones de multiplexión, ya sea TDM o FDM, sin embargo en una red Ad-Hoc solo se puede usar TDM, ya que no hay un nodo central que haga el intercambio de frecuencias, por lo que la transmisión y recepción de datos deben ser por el mismo canal.

2.3. FUNDAMENTOS DE RADIOFRECUENCIA

2.3.1. Potencia de transmisión

"La relación de la pérdida de potencia entre dos sitios se conoce como *pérdida de propagación* o *pérdida en el espacio libre*. Esto se refiere a la energía que se pierde durante el tiempo en que se transmite entre dos puntos. Un factor importante es que la pérdida de propagación a lo largo de una ruta determinada normalmente es constante, sin importar la cantidad de potencia que se utiliza en el sitio transmisor; por tanto, las

variaciones debidas a la modulación se pueden reproducir en forma suficientemente fiel en el receptor.

La pérdida de propagación es importante para hacer cálculos, debido a que en un enlace RF (Radiofrecuencia) entre dos puntos se deben tomar en cuenta las distancias y las obstrucciones entre los transmisores y los receptores. Un diseño de enlace apropiado deberá proporcionar parámetros para las *pérdidas de propagación máximas permisibles*. Si la solución propuesta tiene una pérdida de propagación en el espacio libre que excede la pérdida de propagación máxima permisible, el sistema no contará con el ancho de banda necesario, ocasionará una falta de confiabilidad excesiva o simplemente no funcionará.

Para reponer las pérdidas de propagación excesivas, deberá incrementar la cantidad de potencia que se reciba en la antena receptora. Esto se puede efectuar de varias maneras. La más obvia es incrementar la potencia de transmisión hasta el límite que establezcan las autoridades reguladoras.

Entre otras técnicas, existe proporcionar más antenas direccionales, aumentar la ganancia (sensibilidad) de la antena receptora o incrementar las elevaciones de las antenas transmisora y receptora para librar las obstrucciones que causan la pérdida de propagación. También puede, en algunas instancias, usar repetidores. Puesto que las cantidades en la pérdida de propagación normalmente son órdenes de magnitud, en general se expresan en una escala de *decibeles* (determinados como dB), las cuales son logarítmicas.

El área que se encuentra más próxima a la antena transmisora tiene dos campos de energía que residen en el mismo espacio: un campo eléctrico y uno magnético. El campo que está más cerca de la antena se conoce como campo de *inducción*. Fuera de éste, la onda RF pierde cualquier identidad del campo eléctrico original. Por tanto, la onda RF existe independientemente de la corriente o voltaje original que la creó y continuará irradiándose a través de cualquier espacio en el que no existan conductores o puntos de absorción. Cuando la onda se acerca a un conductor, parte de la energía será

absorbida por este y creará copias miniatura de las corrientes y voltajes que originalmente se enviaron a través de la primera radiación.

Lo que no cambia a través del tiempo o durante la transferencia a través del espacio es la velocidad original (frecuencia) con la que se enviaron las ondas. Sin embargo, en términos de sistemas portátiles, mover un receptor hacia un transmisor o lejos de él induce un fenómeno conocido como el *Efecto Doppler*, que modifica la velocidad con la que se reciben las ondas. Si el receptor se dirige hacia el transmisor, las ondas se reciben con una velocidad que aumenta; por el contrario, si el receptor se aleja del transmisor, se reduce la velocidad en que se reciben las ondas. Cuando las ondas de radio pasan a través de un medio como la atmósfera o el agua, la longitud de las ondas disminuye pero la frecuencia continúa siendo la misma.

Parte de la longitud de onda se convierte en calor. El punto más importante de esto es que debido a que la frecuencia es el aspecto más confiable de la radiación RF, también es la medida más exacta de una señal en relación con otras, por ejemplo, la amplitud, la longitud de onda y la fase."[5]

2.3.2. Bandas de frecuencia

El uso de los diferentes grupos de frecuencias del espectro radioeléctrico generalmente se encuentra debidamente regulado en cada uno de los países del mundo. No hay alguna razón en específico de la forma en que estas divisiones son establecidas, sin embargo el hecho de que esas divisiones sean hechas con meticuloso cuidado es esencial para que los usuarios puedan obtener un uso eficiente y confiable del espectro.

En la tabla 2.1 se muestran las divisiones del espectro en diferentes bandas con cierto rango de frecuencias cada una, así como del tamaño de longitud de onda en el espectro de extremo inferior. Cabe señalar que la tabla muestra las frecuencias de acuerdo a la división que se hizo en el gobierno de los Estados Unidos.

[5] Ibídem. P. 41 – 43.

Banda	Rango de Frecuencia	Longitud de Onda
Frecuencia muy baja (VLF)	0 kHz – 30 kHz	100 km
Frecuencia baja (LF)	30 kHz – 300 kHz	10 km
Frecuencia intermedia (MF)	300 kHz – 3 MHz	1 km
Frecuencia alta (HF)	3 MHz – 30 MHz	100 metros
Frecuencia muy alta (VHF)	30 MHz – 300 MHz	10 metros
Frecuencia ultraalta (UHF)	300 MHz – 3 GHz	1 metro
Frecuencia superalta (SHF)	3 GHz – 30 GHz	100 metros
Frecuencia extremadamente alta (EHF)	30 GHz – 300 GHz	10 metros

Tabla 2.1. Distribución de Frecuencias.

2.3.3. Modulación

"Cualquier señal que se puede traducir en una forma eléctrica, como audio, video o datos, se puede modular y enviar a través del aire. Los datos son los más fáciles de modular de manera confiable, mientras que el video junto con la voz conllevan la dificultad más alta para retener la fidelidad de la fuente original. La *modulación* es la técnica de convertir los bits en algo que se transmite a través de la frecuencia portadora de la onda y a través del aire.

La frecuencia portadora de la onda no tiene inteligencia; los datos modulados contienen la inteligencia (datos) entre dos puntos. *La frecuencia portadora de la onda de un radio 802.11b y 802.11g es de 2.4 GHz a 2.485 GHz*. No existen menos de tres rangos de frecuencias portadoras de ondas para el estándar 802.11a, los cuales son 5.125 GHz a 5.225 GHz, 5.325 GHz a 5.425 GHz y 5.785 GHz a 5.825 GHz. La modulación es la diferencia entre una señal RF estable (una señal que no sufre cambios, que se conoce como un tono de onda continua [*Continuos Wave, CW*, por sus siglas en inglés]) y una que transporta información.

La selección de los esquemas de modulación se basa en la relación entre la maximización del ancho de banda a través de una alta eficiencia espectral y la pérdida de bits provocada por la complejidad del esquema. Además mientras sea mejor la eficiencia espectral será peor la eficiencia de la potencia, y viceversa. Los sistemas más simples como la modulación de fase por desplazamiento (*Phase – Shift Keying, PSK*, por sus

siglas en inglés), son muy sólidos y fáciles de implementar debido a que usan velocidades lentas de datos.

En la modulación PSK, la forma de la onda no se modifica en la amplitud o en la frecuencia, sino en la fase. Con las frecuencias más bajas, la selección de un esquema de modulación es muy importante debido a que, en forma inherente, existe menos ancho de banda en general con el que se puede trabajar que con el de frecuencias más altas. El término adecuado que se relaciona con este fundamento es la *eficiencia espectral*; es decir, la forma en que es posible obtener el máximo del ancho de banda disponible. El término más común que se usa para la eficiencia espectral son los bits por hertz.

La densidad espectral depende ampliamente del esquema de modulación seleccionado. El objetivo de un esquema de modulación es el de transformar unos y ceros en ondas que se puedan transmitir y recibir por la frecuencia portadora de la onda de un enlace de radio. La *frecuencia portadora de la onda* no transporta por sí misma la información, sino que ésta viaja a través de la frecuencia portadora. En la tabla 2.2 se muestra una lista parcial de los esquemas de modulación.

Los diferentes esquemas que se mencionan en la tabla 2.2, generalmente recaen o se relacionan con uno de los tres tipos más importantes de modulación:

- *Modulación de amplitud*.- La potencia de salida del transmisor es variable, mientras que la frecuencia y la fase de la onda senoidal permanecen constantes.
- *Modulación de frecuencia*.- La potencia de salida y fase permanecen constantes en tanto que la frecuencia varía de acuerdo con un rango pequeño.
- *Modulación de fase*.- La amplitud y la frecuencia permanecen constantes, pero la fase dentro de la frecuencia portadora de la onda cambia con respecto a un rango pequeño.

En general, los esquemas de modulación más comunes que se usan en la actualidad para los radios son BFSK, QPSK y QAM. BFSK enviará un *uno* a través de una frecuencia y un *cero* por medio de otra *frecuencia*. BFSK enviará dos estados, un *uno*

con una fase y un *cero* a través de otra *fase*. QPSK se vuelve más completo y tiene cuatro estados para representar ya sea un 00, 01, 11 o 10, cuatro estados de fase, y todos mantienen la onda portadora con la misma amplitud y frecuencia. En la figura 2.3 se puede observar la *constelación* de QPSK, que es un conjunto de combinaciones máximas que se permite entre la fase y la amplitud.

Símbolo	**Esquema de Modulación**
AM	Modulación en amplitud
FM	Modulación en frecuencia
SSB	Banda lateral única
PM	Modulación de fase
CCK	Modulación por codificación complementaria
CW	Onda continua (telegrafía)
PCM	Modulación por codificación de pulsos
VSB	Banda lateral residual
BMAC	Componentes analógicos de multiplexión de ondas tipo B
QAM	Modulación de amplitud del cuadrante
DSSS	Espectro extendido de secuencia directa
FHSS	Espectro extendido de salto de frecuencia
BFSK	Modulación de frecuencia por desplazamiento binario
PBCC	Codificación compleja de paquetes binarios
QPSK	Modulación de fase por desplazamiento en cuadrante
DQPSK	Modulación de fase por desplazamiento en cuadrante diferencial
DBPSK	Modulación de fase por desplazamiento binario diferencial
GFSK	Modulación de frecuencia por desplazamiento gaussiano

Tabla 2.2 Esquemas de Modulación.

Donde la modulación se vuelve compleja es con la *modulación de amplitud del cuadrante* (*Quadrature Amplitude Modulation, QAM*, por sus siglas en inglés) una técnica que modula la frecuencia portadora de la onda tanto en su fase como en la amplitud (cuando se usan los portadores senoidales y cosenoidales que tienen una diferencia de 90 grados). En QAM a medida que el número de bits se incrementa de manera lineal, el número de combinaciones de fase/amplitud crece exponencialmente, lo que proporciona una densidad espectral muy alta, incluso en 64 QAM. La relación de QAM respecto a los bits por cada una de las ondas senoidales transmitidas se muestra en la tabla 2.3.

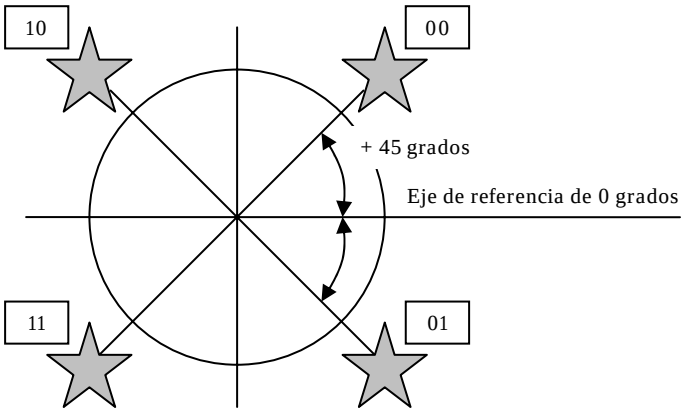

Figura 2.3 Constelación de QPSK.

Combinaciones de amplitud/fase	Bits por onda senoidal
16 QAM	4
32 QAM	5
64 QAM	6
128 QAM	7
256 QAM	8

Tabla 2.3 Relación QAM de bits por onda senoidal

Cuando dos valores de amplitud se transportan por medio de una sola frecuencia portadora de onda, el enlace puede llevar 2 bits a diferencia de 1 bit, por tanto tiene una *densidad espectral* más alta, lo que significa que se transporta más información en relación con una carga de energía determinada desde el transmisor. La frecuencia dentro de la portadora no cambia, pero la cantidad de los datos que se transmiten crece a medida que la complejidad de la modulación aumenta.

A medida que aumenta la complejidad de la modulación, también lo hace la probabilidad de que ocurra un error en la transmisión. Los errores en la transmisión significan la falla de malinterpretar un uno por un cero o viceversa, o no ser capaz de descifrar la energía como uno o cero en la parte receptora. La medida para la cantidad relativa de errores se conoce como el índice de errores de bit (*Bit Error Ratio, BER*, por sus siglas en inglés), que es la proporción de los bits que no se pueden usar respecto a los bits que pueden ser remodulados.

Mientras más sensible sea el tráfico a la latencia, como el de voz y de video, más pequeña tiene que ser la tasa de errores de bits. Los sistemas actuales contemplan la *negociación de velocidad automática*, que ocurre cuando los radios automáticamente cambian a una modulación menos compleja y técnicas de propagación con el fin de mantener niveles más altos de robustez. Las frecuencias más altas o distancias más extensas tienden a favorecer las modulaciones menos complejas. Las señales bajas en enlaces con ruido y esquemas de modulación más simples generalmente funcionan mejor y casi siempre tienen un BER más bajo.

La desventaja de esta simplicidad es que ofrece una capacidad de salida más baja. El concepto de ciclos por bit conduce al concepto de *símbolo*, el cual es una señal única identificable que contiene un número de bits específico, determinado por la complejidad de la modulación. Los símbolos individuales se distinguen por atributos, por ejemplo, duración, amplitud, frecuencia o fase.

El número de *bits por símbolo* es uno de los métodos más comunes para determinar la densidad espectral. Cuatro o más bits por símbolo por lo común se considerarían altamente eficientes en cuanto al espectro, mientras que uno o dos bits por símbolo serán menos eficientes en cuanto al espectro a pesar de proporcionar un buen servicio."[6]

En conclusión, "la modulación, que es una función de la capa física y es un proceso en el cual el radio transmisor prepara la señal digital dentro de la NIC para la transmisión a través del aire. La modulación es el proceso de agregar datos a la frecuencia portadora, mediante la alteración de la amplitud, la frecuencia o la fase de la portadora en una forma controlada.

La tabla 2.4 muestra los detalles de modulación y los tipos de códigos de esparcimiento usados con WLAN con FHSS (Frequency Hopping Spread Spectrum) y DSSS (Direct Sequence Spread Spectrum) en la banda ISM (Industrial, Scientific and

[6] Ibídem. P. 44 – 49.

Medical) de los 2.4 GHz. El *Barker Code* y el *Complementary Code Keying* (*CCK*, de sus siglas en inglés) son los tipos de códigos de esparcimiento usados en WLAN 802.11 y 802.11b. Bluetooth y HomeRF son también tecnologías FHSS que usan tecnología de modulación GFSK.

	Código de esparcimiento	Tecnología de modulación	Velocidad de datos
802.11 2.4 GHz FHSS	Barker Code	2GFSK	1 Mbps
	Barker Code	4GFSK	2 Mbps
802.11b 2.4 GHz DSSS	Barker Code	DBPSK	1 Mbps
	Barker Code	DQPSK	2 Mbps
	CCK	DQPSK	5.5 Mbps
	CCK	DQPSK	11 Mbps

Tabla 2.4 Modulación para 802.11 y 802.11b.

Conforme más altas tasas de transmisión son especificadas, las técnicas de modulación cambian a modo de proveer más rendimiento en los datos. Los equipos WLAN 802.11g y 802.11a especifican el uso de Multiplexión por División de Frecuencias Ortogonales (OFDM, por sus siglas en inglés), permitiendo velocidades de hasta 54 Mbps. En la tabla 2.5 se muestran las técnicas de modulación usadas por redes 802.11a.

Técnica de Codificación	Tecnología de Modulación	Velocidad de datos
OFDM	DBPSK	6 Mbps
OFDM	DBPSK	9 Mbps
OFDM	DQPSK	12 Mbps
OFDM	DQPSK	18 Mbps
OFDM	16QAM	24 Mbps
OFDM	16QAM	36 Mbps
OFDM	64QAM	48 Mbps
OFDM	64QAM	54 Mbps

Tabla 2.5 Modulación para 802.11a.

El estándar 802.11g provee compatibilidad con 802.11b mediante el uso de de códigos CCK y eventualmente también mediante el uso de codificación convolucional binaria de paquetes (*Packet binary convolution coding, PBCC,* por sus siglas en inglés) como una opción. La tabla 2.6 muestra los tipos de modulación usados en 802.11g."[7]

[7] Cfr. *Certified Wireless Network Administrator. Official Study Guide.* Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 312 – 315.

Método requerido para transmisión	Método de transmisión opcional	Velocidad de datos
Barker		1 Mbps
Barker		2 Mbps
CCK	PBCC	5.5 Mbps
OFDM	CCK – OFDM	6 Mbps
OFDM	OFDM, CCK – OFDM	9 Mbps
CCK	PBCC	11 Mbps
OFDM	CCK – OFDM	12 Mbps
OFDM	OFDM, CCK – OFDM	18 Mbps
OFDM	PBCC	22 Mbps
OFDM	CCK – OFDM	24 Mbps
OFDM	PBCC	33 Mbps
OFDM	OFDM, CCK – OFDM	36 Mbps
OFDM	OFDM, CCK – OFDM	48 Mbps
OFDM	OFDM, CCK – OFDM	54 Mbps

Tabla 2.6 Métodos de transmisión para 802.11g

2.3.4. Técnicas de propagación

"Existe una gran confusión entre técnicas de modulación y técnicas de propagación. *La diferencia entre una técnica de modulación y una de propagación es que una técnica de propagación distribuye la información a través de una variedad de canales, en tanto que una técnica de modulación modula la información a través de cada uno de los canales.*

El Espectro extendido de secuencia directa (DSSS), el Espectro extendido de saltos de frecuencia (FHSS), el *Acceso multiplexado de división de código* (CDMA) y la Multiplexión por división ortogonal de frecuencia (OFDM) son ejemplos de técnicas de propagación. La *multiplexión por división ortogonal de frecuencia codificada* (*Coded Orthogonal Frecuency Division Multiplexing*, *COFDM*, por sus siglas en inglés) es la técnica de propagación que se usa en 802.11a y 802.11g.

2.3.4.1. FHSS y DSSS

Normalmente DSSS tiene un desempeño mejor, en tanto que FHSS por lo general es más resistente a la interferencia. Aunque OFDM es una técnica para propagar la señal a

través de un ancho de banda determinado, no es una técnica de espectro extendido. Las duplicaciones de carga de datos son comunes en el espectro extendido de modo que cuando lleguen datos corruptos de manera excesiva, o no logran llegar al destino, las redundancias inherentes a esta arquitectura proporcionan un enlace de datos más sólido.

En la figura 2.4 se muestra un esquema del comportamiento de FHSS.

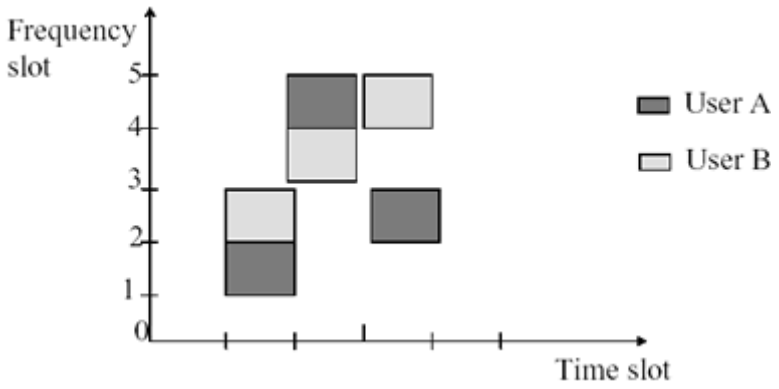

Figura 2.4 Frequency Hopping Spread Spectrum.

En los sistemas FHSS, ciertas frecuencias (canales) se evitan hasta que desaparece la interferencia. La interferencia tiende a cubrir más de un canal a la vez. Por tanto, los sistemas DSSS tienden a perder más datos debido a la interferencia, ya que la información se envía a través de canales secuenciales. Los sistemas FHSS *saltan* entre los canales con un orden no secuencial. El mejor de los sistemas FHSS ajusta la selección de los canales, de manera que los canales con interferencia alta se evitan cuando se mide en ellos tasas de bits excesivamente bajas.

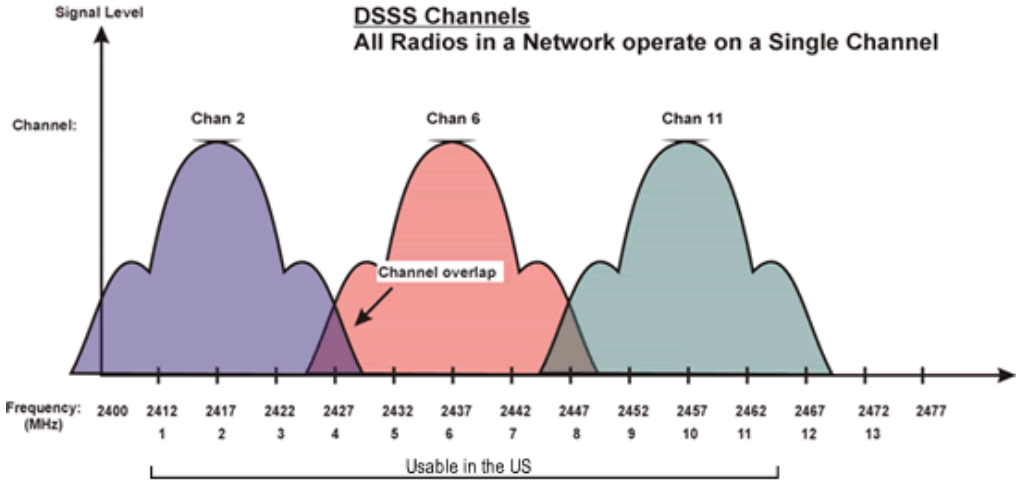

Figura 2.5 Direct Sequence Spread Spectrum.

En la figura 2.5 se muestra un esquema de DSSS para el estándar IEEE802.11 usado en la frecuencia de 2.4 GHz.

2.3.4.2. OFDM

Los sistemas vistos anteriormente usan espectro extendido, sin embargo, el sistema de OFDM usa *división de frecuencias*, lo cual significa que el ancho de banda disponible se divide en múltiples portadoras de datos. Luego, los datos que se van a transmitir se dividen entre estos subportadores. En la figura 2.6 se muestra la distribución de canales en OFDM.

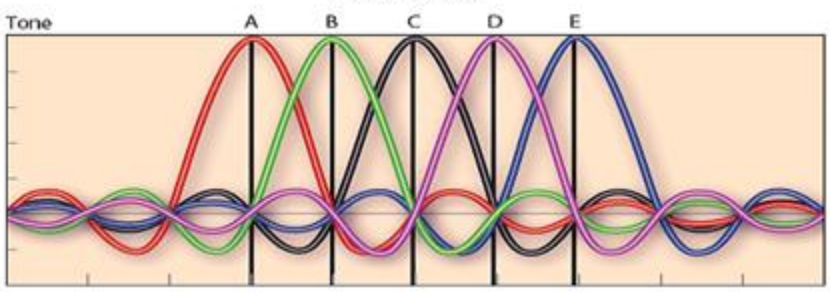

Figura 2.6 Orthogonal Frequency Division Multiplexing.

Debido a que cada portadora se considera independiente de las otras, la frecuencia de una banda de protección debe estar colocada en torno a ella, lo que es otra forma de decir que no se pueden transportar los datos sobre una frecuencia adyacente. Esta banda de protección disminuye la eficiencia del ancho de banda.

En OFDM se usan múltiples frecuencias portadoras de la onda (o tonos) para dividir los datos a lo largo del espectro disponible, de modo similar a un sistema FDM, sin embargo, en OFDM, se considera que cada código es ortogonal (independiente o sin relaciones) a los tonos adyacentes. Otra diferencia importante entre OFDM, FHSS y DSSS es que no obstante que cada uno de los canales envía energía en forma *secuencial* en FHSS y DSSS, dentro de OFDM, toda la energía se envía a lo largo de todos los canales *al mismo tiempo*.

En OFDM cada tono es un entero (un número completo) de frecuencia apartada de la frecuencia adyacente y, por tanto, no se requiere una banda de protección alrededor de cada tono. Debido a que OFDM sólo requiere de bandas de protección en torno a un conjunto de tonos, y de frecuencias piloto para sincronización y para medición de las condiciones del canal, tiene una eficiencia espectral más alta que FDM. Además, ya que OFDM está compuesto de muchos tonos de banda angosta, la interferencia de banda angosta sólo degradará una pequeña parte de la señal y no tiene ningún efecto, o sólo un poco, en el resto de os componentes de la frecuencia. La figura 2.7 muestra la diferencia entre FDM y OFDM

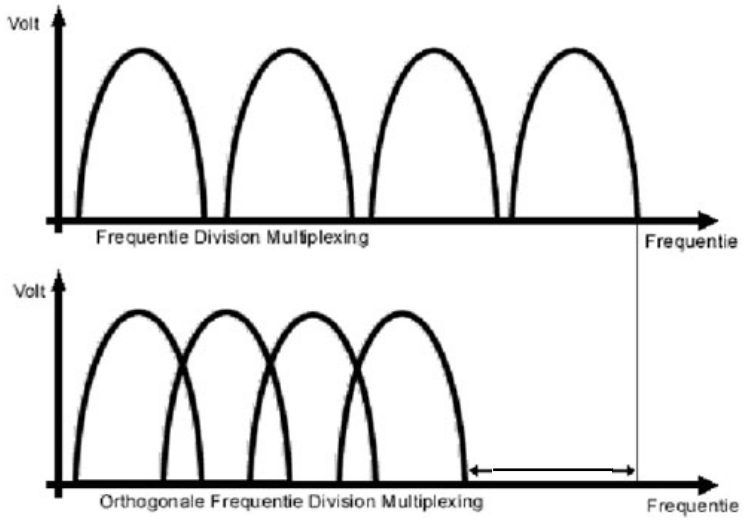

Figura 2.7 Diferencia entre FDM y OFDM.

OFDM es una parte obligatoria del estándar 802.11g, y la compatibilidad con productos anteriores para radios 802.11b también es obligatoria. El estándar permite tanto al manipulador de código complementario (*CCK*, por sus siglas en inglés) como OFDM, además de la *codificación compleja de paquetes binarios* (*Packet Binary Convolutional Coding*, *PBCC*, por sus siglas en inglés).

El estándar requiere que los fabricantes de equipo incluyan los formatos CCK/OFDM o PBCC/OFDM, pero no ambos. CCK es el formato de modulación básico para los radios 11b y 11g y es un esquema de *portador único*, es decir, solo opera sobre un rango muy angosto de frecuencia. PBCC también es un método de portador único, pero emplea el método 8 PSK para CCK, que es más complejo que BPSK y QPSK, y una

estructura de código complejo en lugar de una estructura de código de bloque más sencilla como la que usa CCK. Para las velocidades de datos de 11 Mbps o inferiores, normalmente CCK se considera un método aceptable para la transmisión de datos.

Para las velocidades de datos superiores a eso, OFDM es el formato que permite las velocidades de datos más altas de la de 54 Mbps que pueden alcanzar los usuarios de 11a y 11g. Generalmente el *preámbulo y el encabezado se envían mediante el uso de la modulación CCK, y la carga con las velocidades de datos por arriba de 20 Mbps se enviarán a través de OFDM*. El formato PBCC permite una velocidad de datos máxima de 33 Mbps, mientras que el formato CCK con OFDM proporciona una velocidad de datos a 54 Mbps.[8]

2.3.5. Técnicas de duplexión

Existen varias maneras fundamentales en las que un enlace logra establecer la comunicación de un extremo a otro. Los enlaces de radio más complejos efectúan comunicaciones dúplex completas (*Full duplex*), pero los radios de los productos 802.11 no cuentan con este tipo de duplexión, generalmente son Half dúplex. En la tabla 2.7 se muestran los distintos tipos de técnicas de duplexión.

Tipo	**Comunicación**
Simplex	La comunicación viaja sólo en un sentido
Half dúplex	Existe comunicación entre los dos extremos, pero solo uno puede usar el canal a la vez
Full dúplex	Ambas partes pueden transmitir y recibir al mismo tiempo

Tabla 2.7 Técnicas de duplexión.

[8] Cfr. Reid, Neil. Op. Cit. P. 50 – 56.

CAPÍTULO III
EL ESTÁNDAR IEEE 802.11

El estándar IEEE 802.11 forma parte del estándar IEEE 802, la forma en que estos se relacionan así como otros temas importantes de este mismo estándar se pueden encontrar por completo en la documentación de dichos estándares.

3.1. PILA DE PROTOCOLOS DE 802.11

"A modo de recordatorio, OSI (*Open Systems Interconnection*, por sus siglas en inglés) significa *Sistemas abiertos de interconexión* y su modelo de referencia es un modelo de arquitectura de red que se acepta en todo el mundo. Este modelo está formado por siete capas, cada una de las cuales sirve para funciones de red particulares, como direccionamiento, control de flujo, control de errores, cifrado y transferencia confiable de mensajes.

La capa más baja, Capa 1 es la que está más cerca de la tecnología de medios, que en este caso sería el radio. Las dos capas OSI inferiores (Capa 1 y Capa 2) están implementadas en el hardware y software, mientras que las cinco capas superiores sólo se implementan en el software. La capa más alta (la capa de aplicación) es la más cercana al usuario. El modelo de referencia OSI se usa universalmente como un método para enseñar y entender la funcionalidad de las redes. Las siete capas de la pila OSI son las capas física, de enlace de datos, red, transporte, sesión, presentación y aplicación."[9]

"Los protocolos usados por todas las variantes de los estándares 802, incluso Ethernet, tienen una estructura muy similar. La capa física de 802.11 corresponde

[9] Ibídem. P. 66.

bastante bien a la capa física del modelo OSI, pero la capa de enlace de datos en todos los protocolos 802 se dividen en dos o más subcapas. En el caso del estándar IEEE 802.11 la capa de enlace de datos está dividida en dos: subcapa de control de enlace lógico y subcapa de control de acceso al medio. Un bosquejo parcial de la pila del protocolo 802.11 se puede observar en la figura 3.1 donde se muestran las diferentes capas físicas que soporta el estándar IEEE 802.11."[10]

"Los radios que usan los equipos 802.11 integran tres elementos principales, sin importar si el dispositivo es un punto de acceso (AP), un dispositivo PCMCIA, un puente u otro dispositivo similar; esos tres elementos principales son:

- Radio.- Genera y recibe energía, la cual se envía y recibe desde una antena.
- Capa de Control de acceso a medios (MAC).- La capa que controla el flujo de paquetes entre dos o más puntos de una red.
- Antena.- Están disponibles dentro de una amplia variedad de configuraciones, tamaños y niveles de desempeño.

<table>
<tr><td colspan="6"></td><td>Capas Superiores</td></tr>
<tr><td colspan="6">Control de enlace Lógico</td><td>Capa de Enlace de Datos</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>Subcapa MAC</td></tr>
<tr><td>802.11
Infrarrojo</td><td>802.11
FHSS</td><td>802.11
DSSS</td><td>802.11a
OFDM</td><td>802.11b
HR-DSSS</td><td>802.11g
OFDM</td><td>Capa Física</td></tr>
</table>

Figura 3.1 Parte de la pila de protocolos del estándar IEEE 802.11.

El estándar 802.11 especifica la *capa física* del radio. Mientras que las antenas ayudan a los radios a adquirir suficientes electrones de modo que se muevan de manera relativamente unísona en la antena transmisora, para que tenga un efecto detectable en los electrones de la antena receptora. Las antenas de radio efectúan dos funciones esenciales:

[10] Méndez, Luis. Tesis: *Diseño, implementación y evaluación de un protocolo MAC con alto reuso espacial para redes inalámbricas con infraestructura y Ad – Hoc.* Fac. Ingeniería, UNAM. 2005. P. 21 – 22.

- Mejoran en gran medida el desempeño de un radio.
- Dan forma a la energía radiada para la comodidad del usuario. ”[11]

3.2. CAPA FÍSICA

“La capa física del estándar 802.11 es la interfase entre el MAC y el medio inalámbrico donde los paquetes son transmitidos y recibidos. La capa física proporciona tres funciones. Primero, proporciona una interfase para intercambiar paquetes con la capa superior MAC para transmisión y recepción de datos. Segundo, emplea modulación de espectro disperso y de la señal portadora para transmitir paquetes de datos sobre el medio inalámbrico. Tercero, proporciona indicación de detección de portadora hacia el MAC para verificar actividad en el medio.”[12]

“La capa física tiene dos subcapas, las cuales son el *Protocolo de convergencia de la capa física* (*Physical Layer Convergence Protocol*, *PLCP*, por sus siglas en inglés) y la subcapa *Dependiente del medio físico* (*Physical Medium Dependent*, *PMD*, por sus siglas en inglés). La diferencia entre las dos es que la capa PLCP se encarga de aspectos como la codificación Barker y CCK, además de las técnicas de modulación como QPSK y la técnica de propagación DSSS, mientras que la capa PMD crea la interfaz hacia la capa MAC para la sensibilidad de la portadora a través de su C*omprobación de canal libre* (*Clear Channel Assessment*, *CCA*, por sus siglas en inglés).

El PLCP consiste en un preámbulo de 144 bits que se usa para sincronizar los AP con los clientes, determinar la ganancia del radio y establecer la CCA. El encabezado PLCP cuenta con tres campos: señal, servicio y longitud, además de Revisión de errores en el encabezado (HEC), lo que asegura la integridad del encabezado y el preámbulo. El campo de señal indica la velocidad a la que será transmitida la carga, la cual para 802.11g es 1, 2, 5.5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48, 54. El campo de servicio está reservado para un uso futuro. El campo de longitud indica el tamaño de la carga, e

[11] Cfr. Neil, Reid. Op. Cit. P. 66, 68, 78.
[12] http://www.intelligraphics.com/articles/80211_article.html

incluye los 16 bits de HEC, que se efectúa mediante una verificación por redundancia cíclica (CRC). El PLCP siempre se transmite a 1 Mbps, debido a que la confiabilidad y solidez de la señal son muy importantes y tienen prioridades sobre la velocidad. Sin embargo, este encabezado no impacta la velocidad general de un enlace, debido a que 24 bits de cada paquete se envían a 1 Mbps. Debido a que la carga del encabezado de 192 bits se transmite a 1 Mbps, 802.11 tiene, cuando mucho, sólo un 85 por ciento de eficiencia en la capa física. La capa física realiza por lo menos tres funciones esenciales:

- Funciona como la interfaz entre la capa MAC en dos o más ubicaciones geográficas.
- Realiza la detección real de los sucesos CSMA/CD, mismos que ocurren dentro de la capa MAC.
- Efectúa la modulación y demodulación de la señal entre dos puntos geográficos en los que residen equipos 802.11."[13]

3.2.1. Capa física de 802.11g

"802.11g provee la misma máxima velocidad de 802.11a, acoplada a la compatibilidad con dispositivos 802.11b. Con esta compatibilidad se puede hacer una actualización a redes WLAN de forma simple y barata. El IEEE 802.11g especifica la operación en la banda ISM de 2.4 GHz. Para adquirir estas altas tasas de datos encontradas anteriormente solo en dispositivos 802.11a, los dispositivos que cumplen con 802.11g utilizan tecnología OFDM."[14]

"Debido a que 802.11g opera en la banda de 2.4 GHz, todos los aspectos de la capa física y principalmente, todas las regulaciones internacionales que se aplican a 802.11b también se aplican a 802.11g. Ya que 802.11g transmite en la banda de 2.4 GHz, puede aprovechar la forma de onda relativamente larga y la puede llevar más lejos que la forma de onda de 5 GHz que usa 802.11a, considerando que los demás aspectos siguen igual.

[13] Cfr. Neil, Reid. Op. Cit. P. 31, 71, 72.
[14] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 67 – 73.

A pesar de que varían alrededor del mundo, las regulaciones de 2.4 GHz normalmente permiten una potencia de transmisión más grande que la que se permite en las bandas de 5 GHz que usa 802.11a. Más aún, la misma ganancia de la antena relativamente alta que se permite para los dispositivos 802.11b también se permite para los dispositivos 802.11g. Sin embargo, existen algunas limitantes para 802.11g.

La banda de 2.4 GHz sólo permite el uso de tres canales, debido a que estos están en función del ancho de banda, a diferencia de los ocho canales de la banda de 5 GHz que está disponible en muchos países. Otro inconveniente es que la banda de 2.4 GHz está saturada con dispositivos 802.11b y teléfonos inalámbricos, además del uso de otros dispositivos de uso casero.

A pesar de que 802.11g usa los mismos medios de transmisión que 802.11a y proporciona las mismas velocidades de datos, es común que en la práctica no logre proporcionar una capacidad de salida tan alta como la de 802.11a. Los altos aspectos de la interferencia en la banda de 2.4 GHz producen una reducción en la capacidad de salida debido a los errores en la transmisión y los reenvíos asociados a esto.

Debido a que el estándar requiere que los radios 802.11g cuenten con interoperabilidad con otros radios 802.11g y 802.11b, los primeros deben asumir algunas definiciones heredadas de 802.11b cuando operan e un entorno 802.11b y 802.11g mixto en donde 802.11b interactúa con otros radios 802.11g."[15]

3.3. SUBCAPA DE CONTROL DE ACCESO AL MEDIO

La subcapa de control de acceso al medio (MAC, por sus siglas en inglés) es un subconjunto de la capa de enlace, que a su vez, es adyacente a la capa física en una red.

[15] Cfr. Neil, Reid. Op. Cit. P. 130 – 131.

"La capa MAC controla la conectividad de dos o más puntos a través de un esquema de direcciones. Cada computadora portátil o punto de acceso tiene una dirección MAC. Lo que hace que una WLAN sea diferente de una LAN Ethernet es, la capacidad de los usuarios a trasladarse de un punto de la red a otro y seguir conectados. La forma en la que opera MAC en 802.11 bajo este estándar es lo que permite que los niveles más altos de la pila OSI funcionen normalmente. En otras palabras, la capa MAC es la que controla los aspectos de movilidad de una red 802.11.

Es por esta razón que una capa MAC 802.11 está obligada a hacerse cargo de ciertas funcionalidades que normalmente son responsabilidad de capas más altas de la pila OSI, por ejemplo, la capa de sesión, que controla el inicio y la terminación de sesiones. En el estándar MAC 802.11, el flujo de información se realiza mediante un método del mejor esfuerzo, que también se conoce como *sin conexión*. Los enlaces sin conexión son en los que el extremo receptor del enlace no verifica la recepción de los datos con el enlace transmisor.

La técnica que usa la capa MAC se conoce como Acceso múltiple de sensor de portadora (*Carrier Sense Multiple Access*, *CSMA*, por sus siglas en inglés) que es una técnica que requiere que el transmisor *escuche* lo que ocurre en el entorno local, para asegurarse de que no existen otras transmisiones en la frecuencia asignada. La detección real se efectúa en la Capa 1, pero el control de tiempo para las transmisiones se controla en la capa MAC. En una arquitectura del mejor esfuerzo, es posible que no exista alguna garantía de que los datos que se envían podrán recibirse de manera exitosa. Algo que hace el sistema 802.11 para ayudar a asegurar la recepción exitosa de información es enviar la información de manera repetida, lo que se conoce como *repiqueteo*.

Otra función que proporciona la capa MAC 802.11 es la de seguridad, la que normalmente se controla en la capa de presentación (Capa 6). La medida de seguridad compatible con este estándar es la Privacidad equivalente al cableado (WEP, por sus siglas en inglés) que es un método para manejar claves y cifrar datos."[16]

[16] Ibídem. P. 32 – 33.

Las funciones esenciales de la capa MAC son: exploración, autenticación, asociación, seguridad, ahorro de energía y fragmentación. Estas funciones a su vez conllevan a otras funciones muy importantes.

"La subcapa MAC es responsable de los procedimientos de asignación de canal, direccionamiento de unidades de datos de protocolo (PDU), formato de tramas, chequeo de error, fragmentación y reagrupación. El medio de transmisión puede operar en el modo de contención exclusivamente, requiriendo que todas las estaciones contiendan por el acceso al canal por cada paquete transmitido. El medio también puede alternar entre el modo de contención, conocido como el *periodo de contención* (CP), y el *periodo libre de contención* (CFP). Durante el CFP, el uso del medio esta controlado por el punto de acceso, por consiguiente se elimina la necesidad de las estaciones de contender por el acceso al canal."[17]

3.3.1. Estructura de las tramas usadas en 802.11

"Una vez que un cliente inalámbrico se ha unido a una red, el cliente y el resto de la red se comunicarán mediante el intercambio de tramas a través de la red, en casi la misma forma en que se hace en otras redes del IEEE 802. Sin embargo, las WLAN no usan tramas Ethernet 802.3. Las tramas WLAN contienen más información de la que contiene una trama común Ethernet. Existen tres diferentes tipos de tramas inalámbricas en una WLAN: control, administración y datos. Cada tipo de trama está constituida de forma diferente a las otras y porta información relacionada a su nombre.

Una trama Ethernet 802.3 tiene un tamaño de trama máximo de 1518 bytes antes de la fragmentación que es requerida por este estándar, pero puede ser incrementada hasta 9000 bytes (llamadas *Tramas Jumbo*). Las tramas mayores a 1518 bytes normalmente son fragmentadas para cumplir con el estándar. Las tramas WLAN tienen un tamaño máximo de trama de 2346 bytes (de los cuales 2312 bytes están disponibles para la carga) antes de que el estándar 802.11 requiera fragmentación. Sin embargo, las

[17] Cfr. Crow, Brian. *IEEE 802.11 Wireless Local Area Networks*. Ed. IEEE Communications Magazine. Septiembre 1997.

tramas inalámbricas son generalmente fragmentadas a 1518 bytes por el punto de acceso debido a la conversión de datos entre Ethernet alámbrico (802.3) y el medio inalámbrico (802.11).

En una trama inalámbrica el preámbulo siempre es enviado a 1 Mbps para proveer una tasa de datos común que cualquier receptor pueda interpretar. Existen dos tamaños de preámbulo (también llamado preámbulo PLCP) – largo (128 bits) y corto (56 bits). Para preámbulos largos, tanto el preámbulo como la cabecera son enviados a 1 Mbps. Para preámbulos cortos, el preámbulo es enviado a 1 Mbps, y la cabecera es enviada a 2 Mbps.

La tasa de datos o campo *DR (Data Rate)* en la cabecera especifica la tasa en la cual serán transmitidos los datos. Después de enviar la cabecera, el transmisor puede entonces cambiar la tasa de datos a cualquiera que especifique la cabecera. Esta misma premisa se aplica a las señales denominadas beacons, las cuales también son enviadas a 1 Mbps por las mismas razones. Existen tres diferentes categorías de tramas generadas dentro de los confines de todos estos formatos de tramas. Las tres categorías y los tipos que cada una tienen son:

- Tramas de Administración
  - ✓ Trama de petición de asociación
  - ✓ Trama de respuesta de asociación
  - ✓ Trama de petición de reasociación
  - ✓ Trama de respuesta de reasociación
  - ✓ Trama de petición de sondeo
  - ✓ Trama de respuesta de sondeo
  - ✓ Trama Beacon
  - ✓ Trama ATIM
  - ✓ Trama de disociación
  - ✓ Trama de autentificación
  - ✓ Trama de deautentificación

- Tramas de Control
  - ✓ Petición de envío (RTS)
  - ✓ Listo para enviar (CTS)
  - ✓ Contestación de recibido (ACK)
  - ✓ Sondeo de ahorro de energía (PS Poll)
  - ✓ Terminación de libre contención (CF End)
  - ✓ CF End + CF Ack
- Tramas de Datos"[18]

"Las tramas de administración son usadas para la asociación o disociación de los nodos con los puntos de acceso, así como también para temporización, sincronización, autentificación, deautentificación y otras señalizaciones. Las tramas de control son usadas para verificar los periodos de contención, la señal inicial y las confirmaciones positivas durante estos periodos, y para terminar el periodo libre de contención.

Mientras que las tramas de datos son usadas para la transmisión de datos mientras dura el periodo de contención y el periodo libre de contención, en tanto se este transmitiendo en el periodo libre de contención las tramas de datos se pueden combinar con tramas de confirmación o petición. En la figura 3.2 se puede observar el formato que se maneja para las tramas de datos en el estándar 802.11, para después analizar su contenido.

Bytes	2	2	6	6	6	2	6	0 - 2312	4
Control de Trama	Duración	Dirección 1	Dirección 2	Dirección 3	# de Sec.	Dirección 4	Datos	Check - Sum	

Bits	2	2	4	1	1	1	1	1	1	1	1
Versión	Tipo	Subtipo	A DS	De DS	MF	Retry	Power Mngt.	More	W	O	

Figura 3.2 Formato de tramas de datos en el estándar 802.11.

[18] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 294 – 296.

- Campo de control de trama.- Está compuesto por 2 bytes distribuidos de la siguiente manera:
  - ✓ Versión.- En este campo se verifica el tipo de versión del protocolo.
  - ✓ Tipo.- En este campo se identifica la categoría de la trama, es decir, si se trata de una trama de administración, de control o de datos.
  - ✓ Subtipo.- Aquí se puede identificar el tipo de trama de acuerdo a su categoría, es decir, si es RTS, CTS, etc.
  - ✓ A DS.- Indica si la trama se dirige hacia el sistema de distribución (DS).
  - ✓ De DS.- Indica si la trama proviene del sistema de distribución.
  - ✓ MF.- Significa More Fragments y es usado para indicar que la trama será dividida en varios fragmentos y aún se realizarán más
  - ✓ Administración de energía.- Por medio de este campo se indica desde el punto de acceso al receptor si debe reactivarse o seguir dormido.
  - ✓ More.- Este campo indica que aún existen paquetes en el transmisor con dirección al mismo receptor.
  - ✓ W.- Aquí se puede saber si los datos fueron o no encriptados usando el algoritmo WEP.
  - ✓ O.- Por medio de este campo se indica al receptor que debe procesar en orden los paquetes que le transmitan en una secuencia de paquetes.

- Campo de duración.- Por medio de este campo las estaciones que no están transmitiendo ni recibiendo pueden saber cuanto tiempo ocupará el canal el paquete y su confirmación para así cada nodo poder actualizar su NAV (vector de asignación de red).

- Campos de dirección.- Se puede observar que existen cuatro campos de dirección, dos son usados para la dirección del transmisor y la del receptor deseado, mientras que los otras dos son usados para obtener las direcciones de punto de acceso fuente y destino cuando existe tráfico entre celdas.

- Campo de número de secuencia.- Cuando los paquetes son fragmentados permite que estos fragmentos sean numerados. De los 2 bytes disponibles, es

decir 16 bits, 12 bits identifican al paquete mientras que los otros 4 identifican al fragmento.

- Campo de datos.- En este campo viaja la información o carga útil y puede ser desde 0 hasta 2312 bytes para el estándar 802.11.

- Campo de CheckSum.- Este campo esta provisto de un algoritmo de chequeo de redundancia cíclica (CRC) de 32 bits para poder detectar si hubo o no errores en la transmisión.

El tipo de trama que se acaba de analizar corresponde mejor a una trama de datos, ya que las tramas de administración están restringidas a una sola celda, por lo que no necesitan llevar dirección de puntos de acceso, mientras que las tramas de control son aún más cortas debido a que generalmente tienen solo una o dos direcciones cuando mucho y no contienen campos de datos ni de secuencia, de tal modo que la información importante en este tipo de tramas se encuentra en el subtipo de trama."[19]

3.3.2. El problema de los nodos ocultos y expuestos

Generalmente en las diferentes topologías de una red inalámbrica no se puede considerar que todos los nodos se encuentren conectados entre sí, mientras que en una red cableada sí. Es por lo anterior que se da lugar al problema denominado nodo o terminal oculto y nodo o terminal expuesta. "Dado que no toda las estaciones están dentro del rango de uno a otro, las transmisiones en curso en una parte de la celda pueden no ser recibidos en alguna otra parte dentro de la misma celda.

El problema de la terminal oculta puede ser ejemplificado por medio de la figura 3.3 (a), donde la estación C está transmitiendo a la estación B. Si A sensa el canal, no escuchara a nadie y falsamente concluirá que puede empezar a transmitir hacia B, ocasionando una colisión. Por otro lado existe el problema inverso, de la terminal

[19] Méndez, Luis. Op. Cit. P. 29 – 31.

expuesta, que se ilustra en la figura 3.3 (b). En este caso B desea enviar hacia C, para poder hacer esto la estación B escucha el canal, cuando escucha una transmisión, falsamente concluye que no puede enviar hacia C aún cuando la estación A esta transmitiendo hacia la estación D. Además, la mayor parte de los radios son half-duplex, lo que significa que no pueden transmitir y escuchar al mismo tiempo en la misma frecuencia."[20]

Para resolver los problemas citados anteriormente, "el estándar 802.11 permite dos formas de acceso a medios (acceso a los canales de radio asignados). Conocidos como *Función de coordinación distribuida* (*Distributed Coordination Function, DCF,* por sus siglas en inglés) y *Función coordinada de punto* (*Point Coordination Function, PCF*, por sus siglas en inglés). DCF es un protocolo obligatorio dentro de las especificaciones de 802.11, en tanto que PCF es un protocolo opcional que se emplea para el tráfico sensible a la latencia, por ejemplo, el de voz y video."[21]

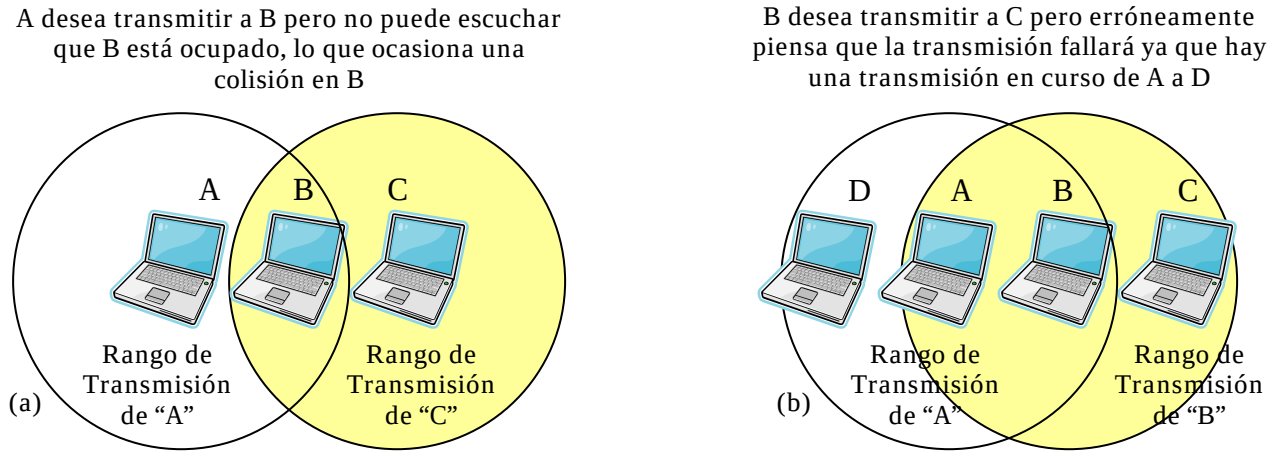

Figura 3.3 (a) Terminal oculta, (b) Terminal expuesta.

3.3.3. Funciones básicas de la subcapa MAC

- *Exploración*.- "Existen dos tipos de exploración dentro el protocolo 802.11, activa y pasiva. En este contexto, *exploración* se refiere a los clientes que buscan AP y

[20] Ibídem, P. 31.
[21] Neil, Reid. Op. Cit. P. 72.

puentes para grupos de trabajo, por mencionar algunos. La exploración pasiva es importante, debido a que muchas instalaciones 802.11 tienen canales traslapados para la cobertura de un área, con el fin de asegurar los niveles de desempeño más altos y una cobertura omnipresente.

Las señales denominadas *beacons* (radioeléctricas) se emiten periódicamente por los AP, y las tarjetas las reciben mientras realizan el proceso de exploración. Los beacons incluyen a los identificadores de establecimiento de servicio (*Service Set Identifiers*, *SSID*, por sus siglas en inglés) y otra información relevante. Posteriormente, el cliente se conecta con el AP a través de la señal más favorable. El propósito principal de la exploración es asegurar que el cliente se asocie con el AP más adecuado dentro del área.

La exploración activa es un protocolo opcional dentro de 802.11 y en esencia efectúa el mismo proceso que una exploración pasiva, la única diferencia es que el cliente envía una *trama de prueba* y todos los AP dentro del rango responden con una *respuesta de prueba*.

- *Autenticación*.- La autenticación es el proceso mediante el cual los clientes previamente aprobados pueden integrarse a un dominio de colisión. La autenticación ocurre antes de la asociación, debido a que es durante el proceso de asociación en el que las direcciones del protocolo Internet (IP) son reveladas por el AP y asignadas al cliente. La retención de esta información es muy importante para prevenir la *falsificación de direcciones*, un término de seguridad que se refiere a la emulación de un cliente o AP autorizado en la WLAN. Existen dos tipos de autenticación dentro del protocolo 802.11:

  a) *Autenticación de sistema abierto*.- Obligatoria dentro de la especificación 802.11. se realiza cuando el cliente envía una solicitud de autenticación con un SSID a un AP, el cual a su vez responde con la autorización o desaprobación de la autenticación.

b) *Autenticación de clave compartida*.- El fundamento del protocolo WEP, que se reconoce ampliamente como un protocolo de seguridad ineficaz para cualquier tipo de WLAN, pero en particular en aquellas que se usan en las redes de empresas pequeñas y medianas, en comparación con las WLAN que se usan en compañías y universidades grandes y campus universitarios.

- *Asociación*.- Después de que se ha realizado el proceso de autenticación, la tarjeta del cliente inicia una asociación cuando envía una trama de solicitud de asociación que contiene un SSID y las velocidades de datos soportadas. El AP responde mediante una trama de respuesta de asociación que contiene un ID de asociación junto con otra información relacionada con el AP específico.

- *Seguridad*.- Mediante WEP el cliente cifra el cuerpo, pero no el encabezado de la trama, antes de la transmisión usando una clave WEP. El AP descifra la trama cuando la recibe usando la misma clave. WEP se considera como inseguro, en esencia, debido a que los piratas informáticos ya encontraron una manera de adquirir suficiente información de la clave WEP para construir una clave completa."[22]
  Cabe señalar que WEP no es el único método que hay en las WLAN para obtener seguridad. En la actualidad se cuenta con mecanismos como claves dinámicas de cifrado y autenticación vía servidores RADIUS, entre otras.

- *Ahorro de energía*.- "La capa MAC proporciona la opción de reducir el uso de energía, lo que puede ser importante donde los usuarios tienen clientes, por ejemplo en computadoras portátiles o PDA. Cuando está activado el modo de ahorro de energía, el cliente envía un mensaje al AP indicando que se irá a dormir, lo que se realiza por medio del bit de estado localizado en el encabezado de cada trama que se envía desde el cliente. Al recibir la solicitud de ir a dormir, enseguida a AP coloca en el búfer los paquetes correspondientes al cliente.

[22] Ibídem. P. 74 – 76.

El modo de uso de energía predeterminado para los clientes es el Modo siempre activo (*Constant Awake Mode*, *CAM*, por sus siglas en inglés), lo cual es esencialmente lo que parece; el cliente permanece constantemente en un modo de estado activo. Pero si el usuario lo desea, puede utilizar un modo de energía más bajo, denominado *Modo de acceso de sondeo* (*Polled Access Mode*, *PAM*, por sus siglas en inglés). Sin embargo, incluso cuando está en el modo de dormir, el cliente debe *activarse* en forma periódica para recibir desde el AP un paquete llamado *Mapa de información de tráfico* (*Traffic Information Map*, *TIM*, por sus siglas en inglés), el cual es una notificación al cliente de que existe tráfico esperando en el AP.

Cuando el tráfico ha sido transferido desde el AP hacia el cliente, éste regresará a dormir. Debido a que el cliente no se activará después de un tiempo aleatorio, sino después de un tiempo muy específico, tendrá una probabilidad estadística alta de no perder el tráfico en espera. Dependiendo del volumen del tráfico, colocar un cliente en el modo PAM puede ahorrarle enormes cantidades de energía.

- *Fragmentación*.- La fragmentación en el contexto del protocolo 802.11 se refiere a la capacidad de un AP para dividir paquetes en tramas más pequeñas. Con frecuencia, esto se hace de modo que la interferencia RF sólo elimina a los paquetes más pequeños. La fragmentación de paquetes también permite el incremento de las cantidades e tiempo libre en el canal. Además de evitar las colisiones y pérdidas en la señal, la capa MAC es responsable de identificar las direcciones fuente y de destino del paquete que se envía, además del CRC. Cada nodo en una red 802.11 es identificado mediante su dirección MAC y usa un esquema de direccionamiento que es idéntico al de Ethernet, el cual es un valor de 6 bytes – 48 bits."[23]

- *Roaming 802.11*.- "El estándar 802.11 proporciona el recorrido (*roaming*) de clientes 802.11 entre múltiples AP, sin importar si el *nuevo* AP está transmitiendo en la misma frecuencia que el anterior que estaba asociado con el cliente. Esto se efectúa mediante las tramas señalizadoras desde los AP y los mismos principios de la

[23] Ibídem. P. 77 – 78.

exploración activa y pasiva de los clientes se pueden aplicar para los propósitos de roaming."[24]

3.3.4. Cambios de tasa de datos

"La selección adaptable (o automática) de tasa (ARS por sus siglas en inglés) y la tasa dinámica cambiante (DRS por sus siglas en inglés) son usadas para describir métodos de ajuste dinámico de velocidad en los clientes de una WLAN. Este ajuste de velocidad ocurre conforme la distancia incrementa o disminuye entre el cliente y el punto de acceso o conforme la interferencia se incrementa. Como se ha visto, los sistemas modernos de espectro disperso y de frecuencias ortogonales están diseñados para hacer saltos discretos solo en las tasas de datos especificadas.

Cuando la distancia se incrementa entre una estación y el punto de acceso, la fuerza de la señal disminuirá al punto en el que la tasa actual de datos no puede ser mantenida. Cuando esta disminución en la fuerza de la señal ocurre, la unidad de transmisión bajará su tasa de datos a la siguiente tasa de datos más baja especificada."[25] La figura 3.4 muestra las diferentes tasas de datos de acuerdo a las distancias entre los nodos, la cual se usará para realizar las pruebas de acuerdo al estándar 802.11g.

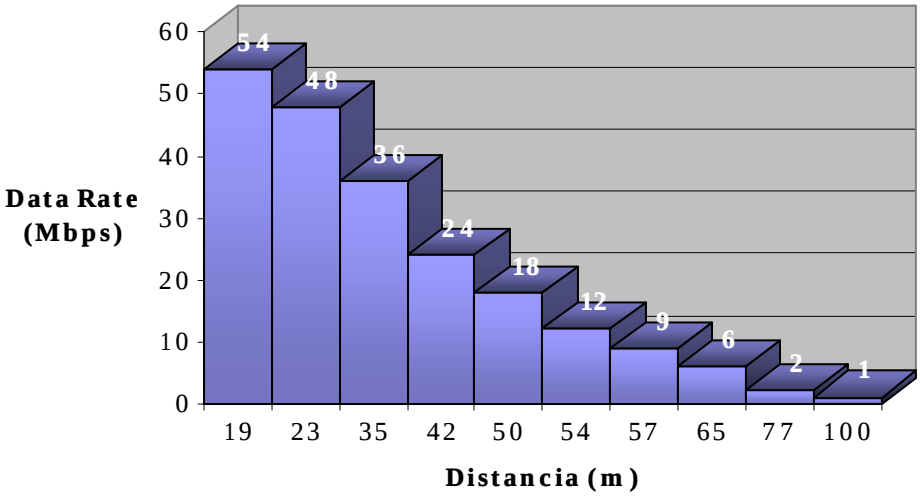

Figura 3.4 Data Rate VS Distancia.

[24] Neil, Reid. Op. Cit. P. 78.
[25] Cfr. *Certified Wireless Network Administrator. Official Study Guide.* Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 300.

3.3.5. Función de coordinación distribuida DCF

"El DCF es un método de acceso especificado en el estándar 802.11 que permite a todas las estaciones en una WLAN contender por el acceso al medio de transmisión (RF) compartido usando el protocolo CSMA/CA. En este caso, el medio de transmisión es una porción de la banda de radio frecuencias que la WLAN está usando para enviar datos. Tanto los conjuntos de servicios básicos (BSS), como los conjuntos de servicios extendidos (ESS), y los conjuntos de servicios básicos independientes, pueden usar el modo DCF. Los puntos de acceso en estos conjuntos de servicios actúan en la misma manera en que lo hacen los concentradores basados en el IEEE 802.3 para transmitir sus datos, y DCF es el modo en el cual los puntos de acceso envían los datos."[26]

"El DCF es el método de acceso fundamental usado para soportar transferencia de datos asíncronos sobre el principio básico de mejor esfuerzo. La DCF opera únicamente en la red Ad – Hoc y opera ya sea únicamente o coexiste con la función de coordinación puntual (PCF) en una red de infraestructura. La DCF esta directamente encima de la capa física y soporta servicios de contención."[27]

3.3.6. Sensor de Portadora

Debido a que en las WLAN se usa la radio frecuencia como medio de transmisión, y este medio es compartido, se tiene que trabajar con la posibilidad de colisiones tal y como se hace en una red cableada. La diferencia entre las redes cableadas y las redes inalámbricas radica en que mientras un nodo transmite en una red cableada, este puede en un momento dado determinar si se esta llevando a cabo una colisión, mientras que en las redes inalámbricas eso no es posible.

Lo anterior se debe a que los equipos de las redes cableadas son Full Duplex, lo que les permite transmitir y escuchar al mismo tiempo, de tal modo que mientras se

[26] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 301.
[27] Méndez, Luis. Op. Cit. P. 32.

encuentran transmitiendo escuchan el medio para ver si la transmisión es exitosa o se genera una colisión. En cambio en las WLAN los radios son Half Duplex, de modo que mientras estos equipos transmiten no pueden escuchar ni recibir información, así que no pueden darse cuenta mientras transmiten que esta ocurriendo una colisión.

"La *búsqueda de portadora* se refiere a la frecuencia real, o energía de radio, que es transmitida por un radio 802.11 y que se recibe y reconoce como nativa del dominio de colisión. La información reside dentro de la onda portadora. Por tanto, el estándar 802.11 usa un protocolo que se conoce como *Accesos múltiples de sensor de portadora con prevención de colisiones* (*Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA*, por sus siglas en inglés) para asegurar que la cantidad de colisiones dentro de un dominio se mantenga a un nivel mínimo. En CSMA/CA, una plataforma 802.11, por ejemplo un AP, efectúa las funciones siguientes dentro del orden indicado:

1. Detecta el canal de radio asignado.
2. Si el canal no está transportando tráfico, se considera inactivo; en cuyo punto el AP o cliente envía un paquete.
3. Si el canal asignado está ocupado, el transmisor que intenta enviar la transmisión espera hasta que termine la transmisión actual y luego espera un periodo aleatorio, conocido como el *periodo de contención*. Esto permite a todos los transmisores un acceso equitativo al canal de radio. El periodo de contención para los sistemas DSSS es de 20 microsegundos.
4. Si el canal asignado está libre de tráfico en el extremo de la transmisión de otra plataforma *además* del periodo de contención, el transmisor que ha esperado para llevar a cabo la transmisión comienza a enviarla"[28]

"La gran diferencia entre CSMA/CA y CSMA/CD es que CSMA/CA evita las colisiones y usa respuestas positivas (ACKs) en lugar del uso del arbitraje del medio cuando las colisiones ocurren. El uso de contestaciones, o ACKs, trabaja en una forma muy simple. Cuando una estación inalámbrica envía un paquete, la estación receptora

[28] Neil, Reid. Op. Cit. P. 72 – 73.

envía de regreso un ACK una vez que esta recibe el paquete. Si la estación transmisora no recibe un ACK, el transmisor asume que hubo una colisión y reenvía los datos.

CSMA/CA, sumado a la larga cantidad de datos de control usados en las WLAN, causa sobrecarga que usa aproximadamente el 50% del ancho de banda disponible en una WLAN. Esta sobrecarga más la sobrecarga adicional de protocolos como RTS/CTS que mejoran la prevención de colisiones, es la responsable del actual rendimiento (throughput) de aproximadamente 5.0 a 5.5 Mbps en una red WLAN 802.11b de 11 Mbps. CSMA/CD también genera sobrecarga, pero solo el 30% en una red con tráfico promedio. Cuando una red Ethernet se congestiona, CSMA/CD puede causar sobrecarga de arriba del 70%, mientras que en una red inalámbrica congestionada permanece constante en alrededor de un 50 a 55 % de rendimiento." [29]

3.3.7. Función coordinada de punto PCF

"La función coordinada de punto (PCF) es un modo de transmisión que permite transferencias de tramas libres de periodos de contención en una WLAN mediante el uso de un mecanismo de encuesta. PCF tiene la ventaja de garantizar una cantidad de latencia conocida de tal modo que las aplicaciones que requieren calidad de servicio (QoS), voz o video por ejemplo, pueden ser usadas. Cuando se usa PCF, el punto de acceso en una WLAN realiza las encuestas. Es por esta razón que PCF no puede ser utilizado en una red Ad-Hoc, ya que una red Ad-Hoc no tiene puntos de acceso para realizar las encuestas.

En PCF primero una estación inalámbrica debe decirle al punto de acceso que la estación es capaz de responder una encuesta. Entonces el punto de acceso pregunta, o encuesta, a cada estación inalámbrica para ver si esa estación necesita enviar una trama de datos a través de la red. PCF, a través de las encuestas, genera una cantidad significante de sobrecarga en una WLAN. Cuando se usa PCF, solo un punto de acceso debe estar en un canal no superposicionado para evitar que el rendimiento sea

[29] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 297.

degradado debido a la interferencia de canales. DCF puede ser usado sin PCF, pero PCF no puede ser usado sin DCF. DCF es escalable debido a su diseño basado en contención, mientras que PCF, por diseño, limita la escalabilidad de la red inalámbrica debido a la sobrecarga adicional de las tramas de encuesta."[30]

3.3.8. Intervalos de tiempo entre tramas

"Antes de que un nodo obtenga el acceso al medio, debe transmitir un valor denominado *Valor de asignación de red* (*Network Allocation Value, NAV*, por sus siglas en inglés), que es representativo de la longitud de un paquete que desean para enviar información. Todos los nodos presentan el NAV con el que quieren transmitir información y éste indica la cantidad de tiempo de transmisión que la trama *anterior* en la cola necesita para terminar.

El valor NAV debe ser cero antes de que el nodo (AP o cliente) pueda enviar la siguiente trama en la cola. Antes de que la siguiente trama en la cola sea transmitida, el nodo calcula la cantidad de tiempo de transmisión que requerirá la trama siguiente. Cuando todos los nodos dentro de un dominio de colisión reciben el NAV, lo usan como base para establecer sus tiempos de transmisión."[31]

De este modo, "todas las estaciones en una WLAN están sincronizadas en tiempo. El espacio entre tramas (IFS, por sus siglas en ingles) es el término que se usa para referirse a los espacios de tiempo estandarizados que son usados en las WLAN 802.11. Existen principalmente tres intervalos de espaciado: SIFS, DIFS y PIFS. Cada tipo de espacio entre tramas es usado por una WLAN para enviar ciertos tipos de mensajes a través de la red o para administrar los intervalos en los que las estaciones contienden por el medio de transmisión.

[30] Cfr. *Certified Wireless Network Administrator. Official Study Guide.* Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 302.
[31] Neil, Reid. Op. Cit. P. 73.

Hay un cuarto espacio entre tramas llamado espacio entre tramas extendido (EIFS, por sus siglas en ingles). EIFS es un intervalo de longitud variable usado como un periodo de espera cuando una transmisión de una trama resulta en una mala recepción de la trama debido a un valor incorrecto de FCS.

Los espacios entre tramas están medidos en microsegundos y son usados para aplazar los accesos de una estación al medio y para proveer varios niveles de prioridad. Mediante el uso de estos espacios, cada nodo conoce cuando y si está supuesto a realizar una cierta acción en la red.

- *Espacio entre tramas corto (SIFS)*.- SIFS es el espacio entre tramas más corto. Los SIFS son espacios de tiempo antes y después de los datos, en los que los siguientes tipos de mensajes son enviados.

  - ✓ RTS.- (Request to Send) Trama de petición de envío, usada para reservar el medio por las estaciones.
  - ✓ CTS.- (Clear to Send) Trama de listo para enviar, usada como una respuesta por los puntos de acceso a las tramas RTS generadas por una estación en orden de asegurarse de que todas las estaciones hayan dejado de transmitir.
  - ✓ ACK.- (Acknowledge) Trama de respuesta o acuse de recibo, usada para notificar a la estación que envía que los datos llegaron con un formato legible en la estación receptora.

  El SIFS provee el mayor nivel de prioridad en una WLAN. La razón para que el SIFS tenga la mayor prioridad es que las estaciones constantemente escuchan el medio (sensado de portadora) esperando por un medio libre. Una vez que el medio está libre, una estación debe esperar un tiempo determinado antes de proceder con la transmisión. La cantidad de tiempo que una estación debe esperar está determinada por la función a realizar por la estación.
  Cada función en una red inalámbrica cae en una categoría espaciado. Las tareas que tienen alta prioridad caen en la categoría SIFS. Si una estación solamente tiene que esperar un periodo corto de tiempo después de que el medio está libre para comenzar

su transmisión, ésta tendrá mayor prioridad sobre las estaciones que tienen que esperar periodos grandes de tiempo. El SIFS es usado para funciones que requieren un periodo de tiempo muy corto, aún necesitando alta prioridad en orden de alcanzar su finalidad.

- *Espacio entre tramas de función de coordinación de punto (PIFS)*.- Un espacio entre tramas PIFS no es ni el más corto ni el más largo espacio entre tramas fijo, así que tiene más prioridad que el DIFS pero menos que el SIFS. Los puntos de acceso usan los espacios PIFS *solamente* cuando la red está trabajando en modo de función de coordinación de punto, el cual es manualmente configurado por el administrador.

  El PIFS es más corto en duración que el DIFS, de tal modo que el punto de acceso siempre ganará control sobre el medio antes que otras estaciones contendientes en el modo DCF. PCF solo trabaja con DCF, una vez que el punto de acceso ha terminado de encuestar, otras estaciones pueden terminar su contención por el medio de transmisión usando el modo DCF.

- *Espacio entre tramas de función de coordinación distribuida (DIFS)*.- DIFS es el espacio entre tramas más largo y es usado por defecto en todas las estaciones que cumplen con 802.11 que están usando la función de coordinación distribuida. Cada estación en la red usando el modo DCF está requerida a esperar hasta que el DIFS haya expirado antes de que cualquier estación pueda contender en la red. Todas las estaciones operando de acuerdo a DCF usando DIFS para transmitir tramas de datos y tramas de administración. El espaciado hace que la transmisión de estas tramas sea de menor prioridad que las transmisiones basadas en PCF.

  En vez de que todas las estaciones asuman que el medio está libre y arbitrariamente comenzar transmisiones simultáneamente antes que el DIFS (lo cual causará colisiones), cada estación usa el algoritmo de retroceso aleatorio para determinar que tanto debe esperar para enviar sus datos. El periodo de tiempo que sigue directamente al DIFS es conocido como el periodo de contención (CP). Todas las estaciones en el modo DCF usan el algoritmo de retroceso aleatorio durante el

periodo de contención. Durante el proceso de retroceso aleatorio, una estación escoge un número aleatorio y lo multiplica por el tiempo de slot para obtener el tiempo total a esperar. Las estaciones hacen una cuenta regresiva de estos tiempos de slots uno a uno, realizando una evaluación de canal libre (CCA) después de cada tiempo de slot para ver si el medio está ocupado. Cuando el tiempo de retroceso aleatorio de la estación expira, esa estación hace un CCA y proveída de un medio que está libre y de que su NAV tiene un valor de cero, comienza la transmisión.

Una vez que la primera estación ha comenzado su transmisión, todas las otras estaciones sensan que el medio se encuentra ocupado, y recuerdan la cantidad restante de su tiempo de retroceso aleatorio del CP anterior. Esta cantidad restante de tiempo es usada en lugar de otro nuevo número aleatorio durante el siguiente CP. Este proceso asegura un acceso justo al medio para todas las estaciones. Una vez que el periodo de retroceso aleatorio se termina, la estación transmisora envía sus datos y recibe de regreso un ACK de la estación receptora. Este proceso entero entonces se repite.

- *Tiempos de slot*.- Un tiempo de slot, el cual está pre-programado en el radio, en la misma manera que lo están los tiempos SIFS, PIFS y DIFS, es un periodo de tiempo estándar en una red inalámbrica. Los tiempos de slot son usados dentro de los CP, en la misma forma en que la manecilla de los segundos de un reloj lo hace. Un nodo inalámbrico marca los tiempos de slot como el reloj marca los segundos. Estos tiempos de slot están determinados por la tecnología de la WLAN que se está utilizando."[32]

3.3.9. Sistema RTS / CTS

"Existen dos mecanismos de sensor de portadora usados en redes inalámbricas. El primero es el *sensor de portadora físico*. El sensor de portadora físico funciona mediante el chequeo de la fuerza de la señal, llamado Indicador de Fuerza de la Señal

[32] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 302 – 306.

Recibida (RSSI), en la señal portadora RF para ver si hay una estación actualmente transmitiendo. El segundo es el *sensor de portadora virtual.*

El sensor de portadora virtual trabaja mediante el uso del campo NAV, el cual actúa como un temporizador en la estación. De esta manera cualquier estación puede reservar el uso de la red para periodos de tiempo específicos. El sensor de portadora virtual está implementado con el protocolo RTS/CTS. El protocolo RTS/CTS es una extensión del protocolo CSMA/CA. Usando RTS/CTS permite a las estaciones transmitir su intención de enviar datos a través de la red."[33]

"Este protocolo es muy útil cuando existen *nodos ocultos*, dos o más clientes que no se detectan entre ellos debido a que están fuera de sus rangos respectivos. RTS/CTS elimina los problemas potenciales en el tiempo en las transmisiones entre los clientes que no pueden interactuar mediante RF. El protocolo RTS/CTS continúa funcionando mientras un cliente envíe paquetes más grandes del tamaño previamente establecido. Es importante observar que cada cliente puede tener tamaños de paquetes únicos, aunque el límite superior para el estándar es de 2312 bytes."[34]

"Como se podrá observar por la breve descripción, RTS/CTS causa sobrecarga significante en la red. Es por esta razón que generalmente el RTS/CTS se encuentra apagado por defecto en las WLAN. Si se esta experimentando una inusual cantidad de colisiones en una WLAN (evidenciada por la alta latencia y el bajo rendimiento) usar RTS/CTS puede incrementar el flujo de tráfico en la red mediante la disminución de colisiones. El uso del RTS/CTS no debe ser hecho al azar. El RTS/CTS debe ser configurado después de un estudio cuidadoso de las colisiones en la red, del rendimiento, de la latencia, etc.

La figura 3.5 muestra el proceso de 4 vías usado en RTS/CTS. En pocas palabras, la estación transmisora envía un RTS, seguido por la respuesta CTS de la estación receptora, ambas de las cuales pasan a través del punto de acceso. Posteriormente la

[33] Ibídem. P. 310.
[34] Neil, Reid. Op. Cit. P. 76 – 77.

estación transmisora envía su carga de datos a través del punto de acceso a la estación receptora, la cual inmediatamente responde con una trama ACK. Este proceso es usado para cada trama que es enviada a través de la red inalámbrica. Existen tres formas de configuración en la mayoría de los puntos de acceso y los nodos para RTS/CTS: Apagado, prendido y prendido con umbral.

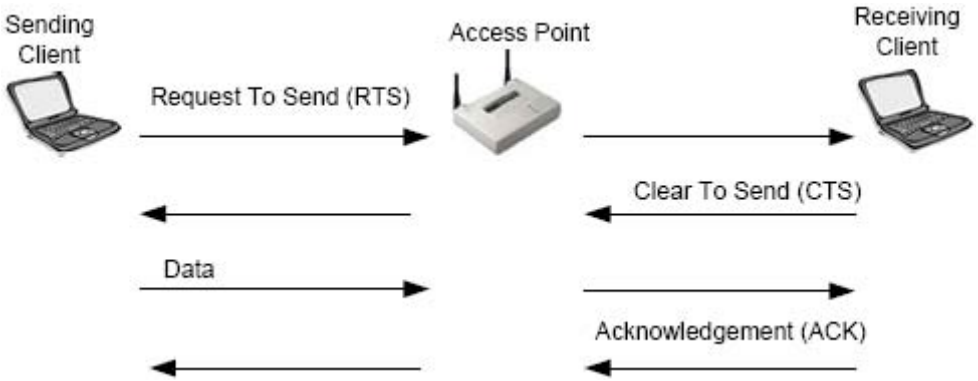

Figura 3.5 Proceso RTS/CTS.

La figura 3.6 muestra una red DCF usando el protocolo RTS/CTS para transmitir datos. Es de notar que las transmisiones RTS y CTS están espaciadas por un SIFS. El NAV es configurado mediante el RTS en todos los nodos y luego se resetea en todos los nodos mediante el siguiente inmediato CTS."[35]

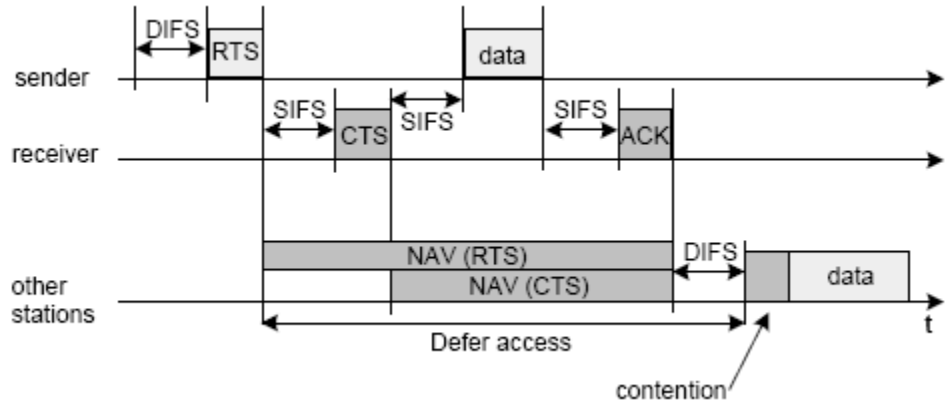

Figura 3.6 Transmisión de datos con RTS/CTS en modo DCF.

[35] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 311 – 312.

CAPÍTULO IV
PROTOCOLOS DE TRANSPORTE TCP Y UDP

En este capítulo se podrá observar a fondo como están constituidos TCP y UDP y de que manera funcionan. TCP y UDP son parte de la pila de protocolos de TCP/IP. Si se requiere conocer más sobre el modelo TCP/IP refiérase a los RFC (Request For Comments) 1122, *Requirements for Internet Hosts – Communication Layers*, RFC 1123, *Requirements for Internet Hosts – Application and Support*, RFC 1958, *Architectural Principles of the Internet*, así como a los RFC relacionados a estos y a los protocolos que en ellos se manejan.

4.1. INTRODUCCIÓN

"Los protocolos de la capa de transporte proveen lo siguiente:

- Una interfase para las aplicaciones de red – esto es, una manera para las aplicaciones de accesar a la red.

- Un mecanismo para multiplexaje/demultiplexado. *Multiplexaje*, en este caso significa aceptar datos de diferentes aplicaciones y computadoras y direccionar los datos a las diferentes aplicaciones destinadas en la computadora receptora. La capa de transporte debe ser capaz de simultáneamente soportar muchas aplicaciones de red y administrar el flujo de datos a la capa de Internet. En la parte receptora, la capa de transporte debe aceptar los datos de la capa de Internet y direccionarlos a múltiples aplicaciones. Otro aspecto del multiplexado/demultiplexaje es que una sola aplicación puede simultáneamente mantener conexiones con más de una computadora.

- Control de errores, control de flujo y verificación. El sistema de protocolos necesita un esquema que en su totalidad asegure la entrega de datos entre las máquinas transmisora y receptora.

Las cuestiones de garantía de calidad (calidad de servicio) siempre se equilibran en cuestiones de costo beneficio. Para proveer un nivel adecuado de garantía de calidad para una situación dada, existen dos alternativas de arquetipos de protocolos de red:

- Un *protocolo orientado a conexión* establece y mantiene una conexión entre las computadoras que se comunican y monitorea el estado de esa conexión durante el curso de la transmisión. Cada paquete de datos enviado a través de la red recibe un acuse de recibo, y la máquina transmisora guarda información del estado para asegurarse que cada paquete es recibido sin errores, retransmitiendo los datos si es necesario. Al final de la transmisión, las computadoras transmisora y receptora elegantemente cierran la conexión.

- Un *protocolo no orientado a conexión* envía un datagrama de una vía a su destino y no se preocupa de notificar a la máquina receptora que los datos son de una vía. La máquina receptora recibe los datos y no se preocupa de regresar información de estado a la máquina fuente.

La capa de transporte provee un método para direccionar los datos a aplicaciones en particular. En el sistema TCP/IP, las aplicaciones pueden direccionar los datos a través de cualquiera de los módulos de los protocolos TCP o UDP usando números de puertos. A modo de repaso un puerto es una dirección interna predefinida que sirve como un camino de las aplicaciones a la capa de transporte o de la capa de transporte a las aplicaciones. Un puerto bien conocido es un número de puerto que está asignado a una aplicación en específico por la ICANN (Internet Corporation for Assigned Names and Numbers). Combinado con la dirección IP, los puertos se convierten en la dirección del socket destino."[36]

[36] Casad, Joe. *Sams Teach Yourself TCP/IP in 24 Hours*. Ed. Sams Publishing. Edic. 3ª. U.S.A. 2003. P. 84 – 90.

Los sockets (zócalos, referido a los enchufes de conexión de cables) son mecanismos de comunicación entre programas a través de una red TCP/IP. De hecho, al establecer una conexión vía Internet se están utilizando sockets, los sockets realizan la interfase entre la aplicación y el protocolo TCP/IP.

Dichos mecanismos pueden tener lugar dentro de la misma máquina o a través de una red. Se usan en forma cliente – servidor: cuando un cliente y un servidor establecen una conexión, lo hacen a través de un socket. Los sockets tienen asociado un puerto

4.2. EL PROTOCOLO DE CONTROL DE TRANSMISÓN TCP

TCP es el protocolo más complejo en el conjunto de protocolos de Internet. Ofrece una serie de servicios para garantizar la transmisión de información satisfactoria. A continuación se analizará la organización global de este protocolo y se describirá las estructuras de datos que usa para administrar la información. Nota: Debido a que es uno de los protocolos más comunes en la mayoría de los equipos de cómputo se decidió realizar este trabajo basándose en este protocolo tan amplio y complejo; sin embargo en este apartado solo se verá a TCP como una herramienta y se analizará un poco más a fondo la parte de control de flujo que maneja TCP.

"TCP es un protocolo orientado a conexión que utiliza los servicios del nivel de Internet y al igual que cualquier protocolo orientado a conexión consta de tres fases:

- *Establecimiento de la conexión*.- Se inicia con el intercambio de tres mensajes, garantiza que los dos extremos de la transmisión estén preparados para la transferencia de datos y permite que ambos acuerden los números iniciales de secuencia (cada extremo elige un número de forma aleatoria).

- *Transferencia de los datos*.- La unidad de datos que utiliza es el segmento y su longitud se mide en *octetos*. La transmisión es fiable ya que permite la recuperación

ante datos perdidos, erróneos o duplicados, así como garantiza la secuencia de entrega, para lo que se añade a la cabecera del segmento de datos un número de secuencia y un código de control. La fiabilidad de la recepción se consigue mediante la confirmación de la recepción, los temporizadores de espera de confirmación y la retransmisión de segmentos.

- *Liberación de la conexión*.- Cuando una aplicación comunica que no tiene más datos que transmitir, TCP finaliza la conexión en una dirección. Desde ese momento, TCP no vuelve a enviar datos en ese sentido, permitiendo que los datos circulen en sentido contrario hasta que el emisor cierra también esa conexión.

TCP permite multiplexación, es decir, una conexión TCP puede ser utilizada simultáneamente por varios usuarios. Como normalmente existe más de un proceso de usuario o aplicación utilizando TCP de forma simultánea, es necesario identificar los datos asociados a cada proceso. Para ello, se utilizan los puertos."[37]

"TCP ofrece un servicio de flujo confiable, controlado y de extremo a extremo entre dos máquinas con velocidades de procesamiento variables, empleando para la comunicación el mecanismo IP.

Al igual que la mayoría de los protocolos de transporte más confiables, TCP usa el tiempo de espera con retransmisión para lograr la confiabilidad. Sin embargo, a diferencia de la mayor parte de los demás protocolos de transporte, TCP está construido en forma minuciosa para operar correctamente incluso cuando los datagramas se demoran, se duplican, se pierden o son entregados en desorden o con los datos dañados o incompletos. Además TCP permite que las máquinas en comunicación se reinicien y restablezcan conexiones en forma aleatoria, sin ocasionar confusión sobre qué conexiones están abiertas y cuáles son nuevas."[38]

"TCP tiene otras cualidades importantes que garantizan la transmisión:

[37] Raya, Jose Luis. *TCP/IP para Windows 2000 Server*. Ed. Alfaomega. Colombia 2001. P. 94.
[38] Comer, Douglas E. *Interconectividad de Redes con TCP/IP. Vol II*. Ed. Pearson Educación. Edic. 3ª. México 2000. P. 193.

- *Procesamiento orientado a flujos*.- TCP procesa los datos en un flujo. TCP puede aceptar datos un byte a la vez en lugar de un bloque preformado. TCP formatea los datos en segmentos de longitud variable, los cuales pasa a la capa de Internet.

- *Resecuenciamiento*.- Si los datos llegan a su destino fuera de orden, el modelo TCP es capaz de resecuenciar los datos para restaurar el orden original.

- *Control de flujo*.- La característica de control de flujo de TCP asegura que la transmisión de datos no sobrecargará o derrumbará la capacidad de la máquina destino de recibir los datos. Esto es especialmente crítico en un ambiente diverso en el cual hay variaciones considerables de velocidades de procesamiento y tamaños de búfer.

- *Precedencia y seguridad*.- Las especificaciones del departamento de defensa para TCP cuenta con niveles opcionales de seguridad y prioridad que pueden ser colocados para conexiones TCP. Muchas implementaciones de TCP, sin embargo, no proveen estas cualidades de seguridad y prioridad.

- *Cierre elegante*.- TCP es tan cuidadoso para cerrar una conexión tanto como lo es para abrirla. La característica de cierre elegante asegura que todos los segmentos han sido enviados y recibidos antes de que la conexión se cierre.

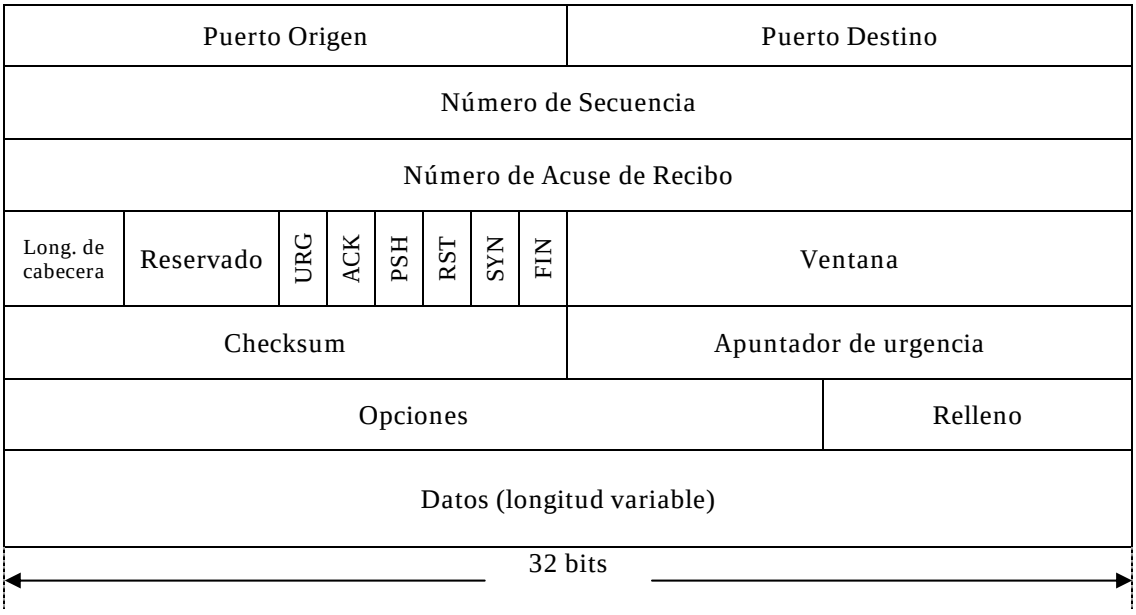

Figura 4.1 Cabecera de TCP.

4.2.1. Cabecera de TCP

El formato de la cabecera TCP se muestra en la figura 4.1. La complejidad de esta estructura revela la complejidad de TCP y las muchas facetas de sus funcionalidades.

- *Puerto origen* (16 bit).- Es el número de puerto asignado a la aplicación en la máquina fuente.

- *Puerto destino* (16 bit).- Es el número de puerto asignado a la aplicación en la máquina destino.

- *Número de secuencia* (32 bit).- Es el número de secuencia del primer byte en este segmento en particular, a menos que la bandera SYN (Synchronization) este puesta en 1. Si la bandera SYN está puesta en 1, el campo de número de secuencia provee el número de secuencia inicial (ISN, Initial Sequence Number), el cual es usado para sincronizar los números de secuencia. Si la bandera SYN esta puesta en 1, el número de secuencia del primer octeto es una vez mayor que el número que aparece en este campo (en otras palabras, ISN+1).

- *Número de acuse de recibo* (32 bit).- El número de acuse de recibo contesta de un segmento recibido. El valor es el siguiente número de secuencia que la computadora receptora está esperando recibir, en otras palabras, el número de secuencia del último byte recibido + 1.

- *Longitud de cabecera* (4 bits).- Es un campo que dice al software TCP receptor de que tamaño es la cabecera y, por lo tanto, donde empiezan los datos. La longitud de cabecera está expresada como un número entero de palabras de 32 bits.

- *Reservado* (6 bits).- Reservado para uso futuro. El campo de reservado provee alojamiento para acomodar futuros desarrollos de TCP y debe ser todo ceros.

- *Banderas de control* (1 bit cada una).- Las banderas de control comunican información especial sobre el segmento.
  - ✓ *URG (Urgent)*.- Un valor de 1 anuncia que el segmento es urgente y el campo de apuntador de urgente es importante.
  - ✓ *ACK (Acknowledgment)*.- Un valor de 1 anuncia que el campo de número de acuse de recibo es importante.
  - ✓ *PSH (Push)*.- Un valor de 1 dice al software TCP que debe avanzar todos los datos enviados a través de las líneas de transmisión a la aplicación receptora.
  - ✓ *RST (Reset)*.- Un valor de 1 resetea la conexión.
  - ✓ *SYN (Synchronization)*.- Un valor de 1 anuncia que los números de secuencia serán sincronizados, marcando el inicio de la conexión.
  - ✓ *FIN (Finish)*.- Un valor de 1 significa que la computadora transmisora no tiene datos por transmitir. Esta bandera es usada para cerrar la conexión.

- *Ventana* (16 bit).- Es un parámetro usado para control de flujo. La ventana define el rango de números de secuencia más allá del último número de secuencia contestado que la máquina transmisora está libre para transmitir sin necesidad de más acuses de recibo.

- *Checksum* (16 bit).- Es un campo usado para verificar la integridad del segmento. La computadora receptora realiza un cálculo checksum basado en el segmento y compara el valor con el valor guardado en este campo. TCP y UDP incluyen una pseudo cabecera con información de direccionamiento IP en el cálculo del checksum.

- *Apuntador de urgencia* (16 bit).- Es un apuntador de compensación que apunta al número de secuencia que marca el comienzo de cualquier información urgente.

- *Opciones*.- Especifica uno de un pequeño conjunto de ajustes opcionales.

- *Relleno*.- Son bits de cero extras (tantos como se necesiten) para asegurar que los datos comenzarán en un límite de 32 bits.

- *Datos*.- Son los datos que están siendo transmitidos con el segmento."[39]

4.2.2. Establecimiento de la conexión

"Todas las acciones en TCP ocurren en el contexto de una conexión. TCP envía y recibe datos a través de una conexión, la cual debe ser pedida, abierta y cerrada de acuerdo a las reglas de TCP.

En orden de proveer una conexión a través de los puertos, la interfase de TCP a la aplicación debe estar abierta. TCP soporta dos estados de abertura:

- Abierto pasivo.- Un proceso de aplicación dado notifica a TCP que está preparado para recibir conexiones entrantes a través de un puerto TCP. De este modo, el camino de TCP a la aplicación es abierto anticipadamente de una petición de conexión entrante.

- Abierto activo.- Una aplicación pide a TCP iniciar una conexión con otra computadora que está en el estado de abierto pasivo. (Actualmente, TCP también puede iniciar una conexión con una computadora que está en el modo abierto activo, en caso de que ambas computadoras estén tratando de abrir una conexión al mismo tiempo.

Para que el sistema de secuencia/acuse de recibo trabaje, las computadoras deben sincronizar sus números de secuencia. Esta sincronización de números de secuencia es llamada *saludo de tres pasos*. El saludo de tres pasos siempre ocurre al comienzo de una conexión TCP."[40]

"En la figura 4.2 se muestra el proceso de establecimiento de una conexión de TCP. El lado iniciador establece una conexión enviando un segmento con la bandera SYN en 1 y el número de secuencia inicial propuesto en el campo de número de

[39] Casad, Joe. Op. Cit. P. 92 – 95.
[40] Ibídem. P. 96 – 97.

secuencia (seq = X). En cuanto recibe este segmento, el lado que responde toma nota del valor del número de secuencia para el sentido entrante y luego devuelve un segmento con las banderas SYN y ACK en 1, su propio valor asignado para el sentido opuesto (seq = Y) en el campo de número de secuencia, y el valor X + 1 (ack = X+1) en el campo de confirmación para indicar que ya tomó nota del valor inicial para su sentido entrante. Al recibir esto, el lado iniciador toma nota de Y y devuelve un segmento con únicamente la bandera ACK puesta en 1 y el valor Y + 1 en el campo de confirmación.

Si sucediera que ambos lados enviaran un segmento SYN al mismo tiempo, parte (b) de la figura 4.2, cada lado se limitaría a devolver un segmento ACK confirmando el número de secuencia apropiado. Así quedan establecidos ambos lados de la conexión y pueden empezar a transmitir datos en forma independiente."[41]

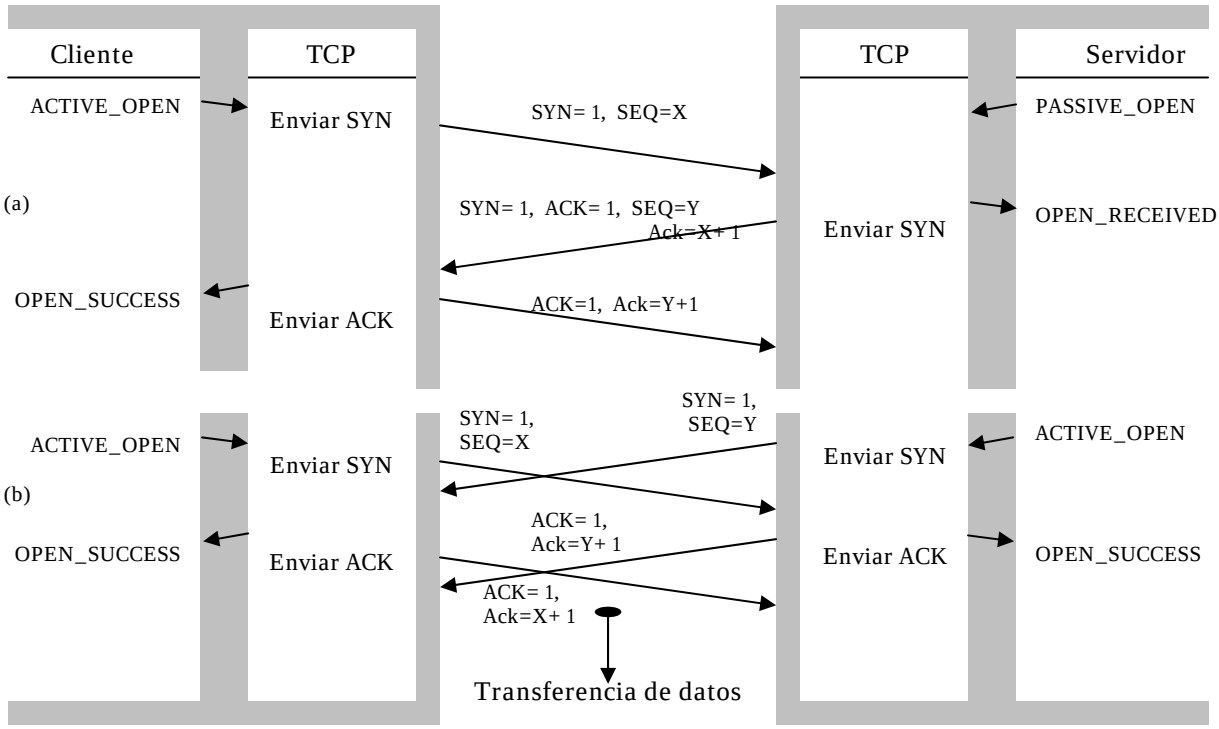

Figura 4.2 Establecimiento de una conexión TCP:
(a) Saludo de tres pasos; (b) Posibilidad de colisión

[41] Hallsal, Fred. Op. Cit. P. 684 – 685.

4.2.3. Transferencia de datos

"Transferir información es sencillo. Para cada bloque de datos recibido por TCP desde protocolos de aplicación de la máquina origen, TCP lo encapsula y lo envía a la máquina destino con un número de secuencia incrementado. Después de que la máquina destino recibe el mensaje, ésta responde con un segmento de acuse de recibo que incrementa el número de secuencia (y por lo tanto indica que ha recibido todo lo de ese número de secuencia).

El servicio de transporte de datos de TCP actualmente incorpora seis subservicios:

- *Full Duplex*.- Habilita a los ambos extremos de la conexión para transmitir en cualquier tiempo, aún simultáneamente.

- *Líneas de tiempo*.- El uso de temporizadores asegura que los datos sean transmitidos dentro de una cantidad de tiempo razonable.

- *Ordenamiento*.- Los datos enviados desde una aplicación son recibidos en el mismo orden en el otro extremo. Esto ocurre a pesar del hecho de que los datagramas pueden ser recibidos en desorden a través de IP, debido a que TCP reensambla el mensaje en el orden correcto antes de pasarlo a capas superiores.

- *Etiquetado*.- Todas las conexiones tienen una precedencia convenida y un valor de seguridad.

- *Control de flujo*.- TCP puede regular el flujo de la información a través del uso de búferes y límites de ventanas como lo se verá más adelante.

- *Corrección de errores*.- El checksum asegura que los datos estén libres de errores (dentro de los límites del algoritmo del checksum)."[42]

[42] Parker, Tim. *Teach Yourself TCP/IP in 14 days*. Ed. Sams Publishing. Edic. 2ª. Indianapolis U.S.A. P. 114 – 115.

"TCP coordina las actividades de transmisión, recepción y retransmisión de cada conexión TCP, a través de una estructura de datos que es compartida por todos los procesos. A esta estructura de datos se le conoce como *bloque de control de transmisión* o *TCB*. TCP mantiene un TCB para cada conexión activa."[43]

4.2.4. Retransmisión adaptiva

"TCP se adapta a los cambios en la demora del viaje de ida y vuelta de una conexión dada, haciéndola confiable aún cuando el sistema de conmutación de paquetes subyacente experimente congestionamiento o fallas temporales.

La retransmisión adaptiva reside en el corazón de TCP y es importante para su éxito. La retransmisión adaptiva emplea el comportamiento del pasado reciente para predecir el comportamiento futuro. Requiere que TCP mida la demora del viaje de ida y vuelta de cada transmisión y que utilice técnicas estadísticas para combinar las medidas individuales dentro de una estimación atenuada de la demora media del viaje. Además TCP actualiza en forma continua su estimación del viaje de ida y vuelta al adquirir nuevas medidas.

En principio, la estimación del viaje de ida y vuelta debería ser fácil. Sin embargo, los problemas en una determinada red imponen serias dificultades. Varios segmentos o acuses de recibo podrían perderse o tener demoras, lo que hace imprecisas las mediciones individuales del viaje de ida y vuelta. El tráfico explosivo de diversos orígenes puede ocasionar que las demoras fluctúen ampliamente. Además, la carga impuesta aun por una sola conexión, puede congestionar una red o una puerta de enlace. Por último, la retransmisión después de pérdidas de segmentos puede ocasionar congestionamiento o aumentarlo.

[43] Comer, Douglas E. Op. Cit. P. 197.

4.2.5. Inicio lento e impedimento del congestionamiento

Aunque los mecanismos de inicio lento e impedimento del congestionamiento forman parte del mecanismo de retransmisión adaptiva de TCP, serán tratados aparte, debido a que este trabajo está basado por completo en la adaptación de tales mecanismos a las redes inalámbricas.

"Cuando se congestiona una red que transporta segmentos TCP, las transmisiones adicionales pueden agravar la situación. Para ayudar a recuperarse de un congestionamiento, este estándar ahora requiere que TCP reduzca su velocidad de transmisión. En particular, TCP da por hecho que la pérdida de paquetes es resultado del congestionamiento y de inmediato usa una técnica conocida como *inicio lento* durante la recuperación. Para mejorar aún más el rendimiento y evitar que se agreguen nuevas conexiones al congestionamiento, TCP emplea el inicio lento siempre que comience a enviar nuevos datos en una conexión recién establecida.

El inicio lento es lo opuesto a la disminución multiplicativa; proporciona un incremento multiplicativo. De nuevo, la idea es simple: inicie la ventana de congestionamiento al tamaño de un solo segmento (el MSS) y envíela. Si la comunicación tiene éxito y llega un acuse de recibo antes de que expire el temporizador de retransmisión, sume un segmento al tamaño de la ventana de congestionamiento. Continúe sumando un segmento a la ventana de congestionamiento cada vez que llegue un acuse de recibo. Por lo tanto, si ambos segmentos llegan con éxito en la segunda ronda de transmisiones, la ventana de congestionamiento aumentará a 4 segmentos y seguirá aumentando en forma exponencial hasta que alcance el umbral establecido por la disminución multiplicativa.

Una vez que la ventana de congestionamiento alcanza el umbral, TCP se hace más lento. En vez de sumar un nuevo segmento a la ventana de congestionamiento cada vez que llega un acuse de recibo, TCP aumenta un segmento al tamaño de la ventana por cada tiempo de un viaje de ida y vuelta. Para calcular el tiempo de ida y vuelta, el código emplea el tiempo de envío y recepción de acuses de recibo para los datos de una ventana.

TCP no espera a que se envíe toda una ventana de datos con su acuse de recibo para incrementar el tamaño de la ventana de congestionamiento. En su lugar suma un pequeño incremento a la ventana de congestionamiento cada vez que llega un acuse de recibo. Este pequeño incremento se elige de manera que el incremento promedio sea aproximadamente de un segmento sobre toda la ventana."[44]

4.2.6. Cierre de la conexión

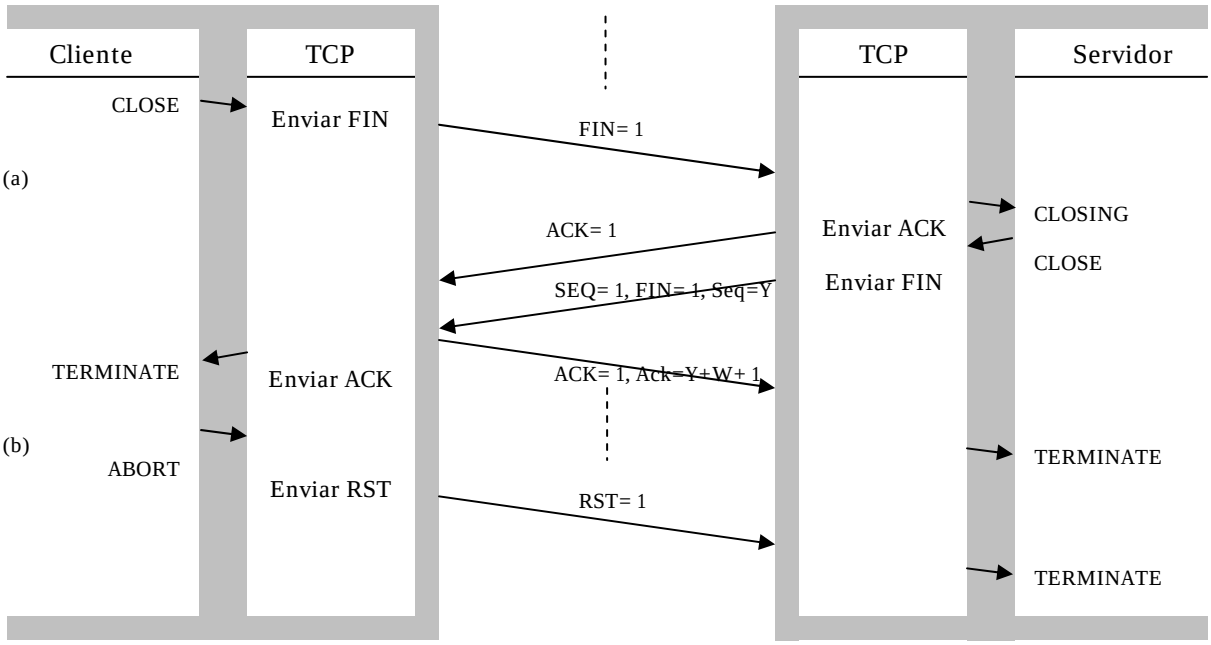

Figura 4.3 Terminación de una conexión TCP:
(a) Normal; (b) Abortar

"Cuando es tiempo de cerrar la conexión, la computadora que inicia el cierre, computadora A, pone un segmento en la cola con la bandera FIN igual a 1. Entonces la aplicación entra en lo que es llamado un *estado de espera de fin*. En este estado, el software de TCP de la computadora A continúa recibiendo segmentos, y notifica a la aplicación local que un FIN ha sido recibido. La computadora B envía un segmento FIN a la computadora A, al cual la computadora A responde de recibido y la conexión es cerrada."[45] Este proceso se encuentra ilustrado en la figura 4.3.

[44] Ibídem. P. 300 – 301.
[45] Casad, Joe. Op. Cit. P. 99.

4.3. EL PROTOCOLO DE DATAGRAMA DE USUARIO UDP

"UDP es por mucho más simple que TCP, y realiza muy pocas de las funciones hechas por TCP. UDP tiene la capacidad de realizar un chequeo de errores limitado. El datagrama UDP incluye un valor de suma de comprobación que la máquina receptora puede usar para probar la integridad del paquete. Sin embargo, esta característica es opcional y puede ser deshabilitada en la máquina receptora para acelerar el procesamiento de los datos entrantes.

El datagrama UDP incluye una pseudo-cabecera que abarca la dirección destino del datagrama, esto provee un significado de verificación para datagramas mal direccionados. También, si un módulo UDP recibe un datagrama direccionado a un puerto inactivo o indefinido, este regresa un mensaje ICMP (Internet Control Message Protocol) notificando a la máquina fuente que el puerto es inalcanzable.

UDP no ofrece el resecuenciamiento de datos que provee TCP. El resecuenciamiento es muy significante en una gran red, tal como lo es Internet, donde los segmentos de datos pueden tomar diferentes caminos y experimentar retrasos significantes en los buffers de ruteadores.

En redes locales, la carencia de una cualidad de resecuenciamiento en UDP típicamente no recae en una recepción no fiable. La inclinación de UDP hacia un diseño no orientado a conexión lo convierte en el protocolo de elección para situaciones de broadcast en la red.

El propósito primordial de UDP es entregar los datagramas a la capa de Aplicación. El protocolo UDP es muy pequeño y utiliza una estructura de cabecera muy simple. El RFC que describe este protocolo, RFC 768, es de sólo tres páginas. UDP no retransmite los paquetes perdidos o erroneos, no resecuencia los datagramas recibidos fuera de orden, no elimina los datagramas duplicados, no da acuse de recibo de los datagramas recibidos, ni establece o termina una conexión.

UDP es primordialmente un mecanismo para programas de aplicación para enviar y recibir datagramas sin el *overhead* de una conexión TCP. La aplicación puede proveer alguna o todas estas funciones, si son necesarias para la aplicación.

La cabecera UDP consiste en cuatro campos de 16 bits, representados en la figura 4.4.

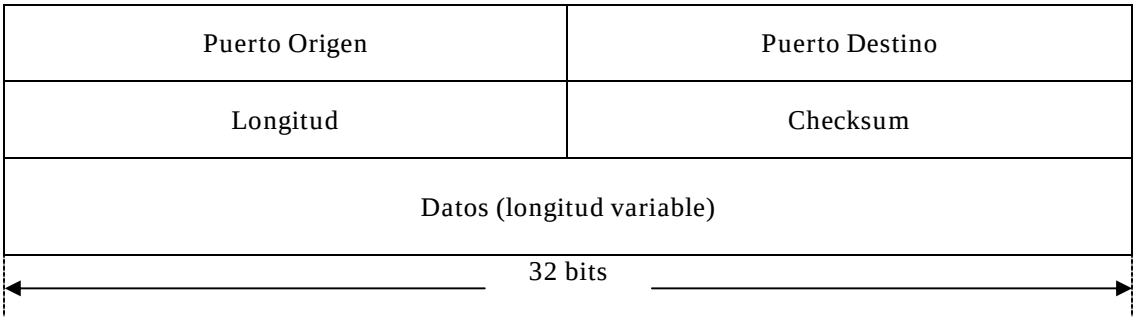

Figura 4.4 Cabecera de UDP.

- *Puerto Origen*. – Este campo ocupa los primeros 16 bits de la cabecera de UDP. Este campo típicamente contiene el número de puerto UDP de la aplicación que está enviando el datagrama. El valor contenido en este campo es usado por la aplicación receptora como la dirección de respuesta para cuando este lista para enviar una respuesta. Este campo es considerado opcional, y no es necesario que la aplicación emisora incluya su puerto origen.

  Si la aplicación origen no incluye su número de puerto, la aplicación debe poner 16 bits en cero en dicho campo. Obviamente si no hay una dirección de puerto válida, la aplicación receptora no será capaz de enviar una respuesta. Sin embargo, esto puede ser la funcionalidad deseada, como lo es en el caso de un mensaje snmr-trap, el cual es un mensaje unidireccional donde ninguna respuesta es esperada.

- *Puerto Destino*. – Este campo de 16 bits contiene la dirección del puerto destino en el cual el software UDP en la máquina receptora entregará su datagrama.

- *Longitud*. – Este campo de 16 bits identifica el tamaño en octetos del datagrama UDP. La longitud incluye la cabecera UDP como también la carga de datos UDP. Debido a que la cabecera UDP es de ocho octetos de longitud, el valor siempre será al menos 8.

- *Checksum*. – este campo de 16 bits es usado para determinar cuando un datagrama fue corrompido durante la transmisión. El checksum es el resultado de un cálculo especial realizado en una cadena de datos binarios. En el caso de UDP, el checksum es calculado basado en una pseudo-cabecera, la cabecera UDP, los datos UDP y posiblemente los octetos de cero de relleno para construir una entrada de checksum de londitug par de octetos. El checksum es generado en el origen y es verificado en el destino permitiendo a la aplicación destino el determinar si un datagrama ha sido corrompido.

Debido a que la cabecera actual de UDP no incluye la dirección IP fuente ni destino, es posible que el datagrama haya sido entregado a la computadora o servicio erróneo. Parte de los datos usados para el cálculo del checksum son una cadena de valores extraidos de la cabecera IP conocida como pseudo-cabecera.

La pseudo-cabecera provee la información de la dirección IP destino así que la computadora receptora puede determinar cuando un datagrama UDP se ha entregado a quien no correspondía."[46]

[46] Ibídem, P. 99 – 101.

CAPÍTULO V
EL SIMULADOR DE REDES NS -2

5.1. INTRODUCCIÓN

"El simulador de redes NS2 es un simulador de eventos discretos enfocado a la investigación de conexión de redes. El NS2 provee soporte substancial para simulación de protocolos TCP, de ruteo[47], y de multicast a través de redes cableadas e inalámbricas (locales y satelitales).

El NS comenzó como una variante del simulador de redes REAL en 1989 y ha sido envuelto considerablemente en los últimos años."[48]

"El simulador toma como entrada un *escenario,* el cual es una descripción de topologías de redes, protocolos, cantidad de trabajo y parámetros de control. Este produce tantas estadísticas de salida como el número de paquetes enviados por cada fuente de datos, el retraso de encolamiento de cada punto de encolamiento, y el número de paquetes tirados o descartados y paquetes retransmitidos."[49]

En 1995 el desarrollo del NS fue soportado por DARPA (Defense Advanced Research Projects Agency) a través del proyecto VINT (Virtual InterNetwork Testbed). "Actualmente el desarrollo del NS está soportado a través de DARPA con el proyecto de Simulación Aumentada para la Medición y Análisis de Redes (SAMAN) y a través del NSF (National Science Foundation) con la Simulación en Colaboración para la Educación y la Investigación (CONSER), ambos en colaboración con otros

[47] La palabra ruteo no existe propiamente en el idioma español, sin embargo en el orbe de las computadoras es un término muy común que proviene de la palabra router del idioma ingles, por lo que lo tomaremos como tal, así como todas sus derivaciones como son, rutear, ruteador, etc.
[48] http://www.isi.edu/nsnam/ns/
[49] http://www.cs.cornell.edu/skeshav/real/overview.html

investigadores incluyendo ACIRI [AT&T Center for Internet Research at ICSI (Internet architecture research institute)].”[50]

El NS es ampliamente usado por la comunidad de redes y es ampliamente considerado como un componente crítico de infraestructura de investigación. NS siempre ha incluido contribuciones substanciales de otros investigadores incluyendo código para redes inalámbricas de los proyectos UCB Daedelus y CMU Monarch.

El simulador NS ha sufrido muchos cambios desde su primera versión hasta llegar a la versión ns-2.1b1 ahora llamada ns-2.17 la cual ya incorporaba cualidades de trazado para el NAM (Network Animator) versión nam1.0a2 en noviembre de 1997. Durante febrero de 2005 se liberó la versión ns-2.28 la cual se uso para este proyecto al ser esta la última versión disponible cuando se comenzó la investigación. Actualmente la versión más nueva es la ns-2.33 que fue liberada en marzo de 2008.

Este simulador se puede obtener de forma gratuita en la página web http://www.isi.edu/nsnam/ns donde se pueden encontrar sus diferentes versiones, así como todo lo necesario para que funcione adecuadamente en un ambiente Unix, Linux o Cygwin.

5.2. GENERALIDADES DE LINUX Y CYGWIN

“Para instalar el NS es necesaria una computadora y un compilador de C++. El NS se ha desarrollado en diferentes tipos de Unix (FreeBSD, Linux, SunOS, Solaris), así que es menos conflictivo instalarlo ahí. El NS también se puede instalar y correr sobre Windows a través del Cygwin. Los escenarios simples corren bien en una máquina razonable, pero escenarios más grandes se ven beneficiados con cantidades más grandes de memoria.”[51]

[50] http://www.isi.edu/nsnam/ns/
[51] http://www.isi.edu/nsnam/ns/ns-build.html

Para el desarrollo de este trabajo se utilizó una PC tipo Laptop porque es necesario correr el NS ya sea en Linux o en Cygwin. A continuación se analizarán algunas cualidades básicas de ambos sistemas. El que escribe ha probado el NS tanto en Cygwin como en Linux y ha descubierto una serie de cosas que el lector podrá observar y de ahí deducir el porque se ha seleccionado a Linux para realizar la investigación.

El paquete de instalación *allinone* requiere alrededor de 350 MBytes de espacio en disco para construirse. Construir el NS en partes puede ahorrar algo de espacio en disco. Existen dos maneras de construir el NS: en partes o todo a la vez. Si se quiere probar el NS rápidamente se debe construir en todo a la vez. Si se requiere hacer desarrollo a nivel C, o ahorrar tiempo de descarga o espacio en disco, o se tienen problemas al instalarlo de todo a la vez, se debe construir en modo en partes. El NS depende de la disponibilidad de muchos componentes externos.

El paquete Ns-allinone es un paquete que contiene los componentes requeridos y algunos componentes opcionales usados en la ejecución del NS. El paquete contiene un script llamado *install* para configurar automáticamente, compilar e instalar estos componentes.

5.2.1. Linux

“Linux está basado en el sistema operativo UNIX, sin embargo Linux no es UNIX. Es un sistema operativo propio, con sus propios matices, sus propios rasgos y sus características especiales. Fue escrito desde sus cimientos por centenares de desarrolladores repartidos por todo el globo, desarrollándose en su mayor parte sobre Internet.

La idea original que está detrás de Linux surgió a principios de los años 90, en la Helsinki University Technology en Finlandia, de manos de un estudiante sueco llamado Linus Torvalds. Lo que empezó en 1991 como un proyecto para suministrar una alternativa al sistema operativo Minix.

Es importante reconocer que el *kernel* de Linux y las partes requeridas para obtener un sistema de trabajo son solamente una pequeña parte de una distribución. Durante algún tiempo, parecía que todo el mundo quería hacerse su propia distribución de Linux. La mayor parte de estas distribuciones se diferenciaban solamente en los conjuntos de software que incluían. A medida que pasaba el tiempo, las diferentes distribuciones diversificaron sus ofertas, añadiendo algunas veces software escrito específicamente para las propias distribuciones en un esfuerzo por diferenciarse del resto."[52]

5.2.2. Cygwin

"Cygwin es un ambiente parecido a Linux para Windows. Este consiste de dos partes:

- Un DLL (cygwin1.dll) el cual actúa como una capa de emulación API de Linux proveyendo substancial funcionalidad del API de Linux.

- Un conjunto de herramientas, las cuales proveen una apariencia de Linux y funcionalidades similares.

El DLL de Cygwin trabaja con todas las versiones de 32 bits de Windows para ix86, desde la versión Windows 95, con excepción de la versión Windows CE y las versiones Beta. Cygwin no es una manera de correr aplicaciones nativas de Linux en Windows. Usted tiene que reconstruir su aplicación *desde la fuente* si usted quiere correrla en Windows.

Cygwin no es una manera de hacer mágicamente aplicaciones nativas de Windows, estando conciente de las funcionalidades de UNIX, como las señales, etc. De nuevo, usted necesita construir sus aplicaciones *desde la fuente* si usted quiere tomar la ventaja de las funcionalidades de Cygwin."[53]

[52] Bandel, David. *Edición Especial Linux*. Ed. Prentice Hall. Edic. 6ª. Madrid, España 2001. P. 5 – 7.
[53] http://www.cygwin.com/

"Desde la versión 2.1b9, el NS ha sido probado, construido y validado en Windows 9x/2000/XP usando Cygwin. El ns-allinone se desempaqueta y se construye correctamente, sin embargo algunas pruebas de validación pueden fallar. La plataforma principal de desarrollo del NS son varias versiones de Unix, así que los problemas de construcción y validación sobre Windows son más frecuentes."[54]

Una vez instalado Cygwin es necesario tener el X11 en este sistema. Dependiendo de la versión de Cygwin que se este usando, puede ser Xfree86 o X.org. Adicionalmente se necesitarán también los siguientes paquetes instalados en Cygwin: gcc, gcc-g++, gawk, tar, gzip, make, patch, perl, and w32api. Cualquier paquete faltante será detectado por el programa de instalación del NS y puede ser agregado con elsetup.exe del Cygwin.

Después que se realizó todo lo anterior, se instaló la versión 2.28 del NS y se verificó que funcionara correctamente es necesario cargar la interfaz gráfica X11 mediante el comando *startx*, ya que el NAM no correrá como una aplicación independiente de Windows, sino en conjunto con el servidor X de Cygwin. También es necesario contar con algún editor de textos como VI o Emacs para poder editar los archivos de simulación y poder observar los archivos de salida, así como los archivos que se van a modificar del simulador.

5.3. INTERFASE AL INTÉRPRETE

"El NS es un simulador orientado a objetos, escrito en C++, con un interprete Otcl (Object Tool Command Language) como interfaz de usuario. El simulador soporta la jerarquía de clases en C++ (también llamada jerarquía compilada), y una jerarquía de clases similar dentro del intérprete OTcl (también llamada jerarquía interpretada). Las dos jerarquías están cercanamente relacionadas una a otra; desde la perspectiva del usuario, existe una correspondencia uno a uno entre una clase en la jerarquía interpretada y una clase en la jerarquía compilada.

[54] http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin

La raíz de esta jerarquía es la clase TclObject. Los usuarios crean nuevos objetos de simulación a través del intérprete; estos objetos están instanciados dentro del intérprete, y están estrechamente representados por un objeto correspondiente en la jerarquía compilada. La jerarquía de clase interpretada está automáticamente establecida a través de métodos definidos en la clase TclClass. Los objetos instanciados de usuario están representados a través de métodos definidos en la clase TclObject.

5.3.1. Conexión OTcl y C++

El NS usa dos lenguajes porque tiene dos diferentes tipos de cosas que necesita realizar. Por una parte, las simulaciones detalladas de protocolos requieren un lenguaje de programación de sistemas el cual pueda manipular eficientemente los bytes, las cabeceras de paquetes, e implementar algoritmos que corran sobre un gran conjunto de datos. Para estas tareas la velocidad del tiempo de ejecución es importante mientras que el tiempo de vuelta completa o *turn around* (correr la simulación, encontrar fallas, corregir las fallas, recompilar, volver a correr la simulación) es menos importante.

Por otro lado, una gran parte de las investigaciones de redes involucran ligeramente distintos parámetros o configuraciones, o la rápida exploración de un número de escenarios. En estos casos, el tiempo de iteración (cambiar el modelo y volver a correr) es más importante. Puesto que la configuración corre una vez (al principio de la simulación), el tiempo de ejecución de esta parte de la tarea es menos importante.

El NS hace frente a estas dos necesidades con dos lenguajes, C++ y OTcl. C++ es más rápido para correr pero más lento para cambiar, haciéndolo adecuado para la implementación de protocolos detallados. OTcl corre mucho más lento pero puede ser cambiado muy rápidamente (e interactivamente), haciéndolo ideal para la configuración de la simulación. El NS (a través de tclcl) provee un ligamiento para hacer que los objetos y las variables aparezcan en ambos lenguajes.

El tener dos lenguajes suscita la cuestión de cual lenguaje debe ser usado para que propósito. El consejo es usar OTcl para: configuración, instalación, y cosas de *una sola vez* y para manipulación de objetos C++ existentes. Y usar C++ para: cualquier cosa que requiera procesamiento de cada paquete de un flujo de datos, y para cambiar el comportamiento de una clase C++ existente en maneras que no estaban anticipadas."[55]

5.4. ARQUITECTURA GENERAL DEL NS – 2

"La figura 5.1 muestra la arquitectura general de NS. En esta figura un usuario general (no un desarrollador de NS) se puede pensar que se encuentra en la esquina inferior izquierda, diseñando y corriendo simulaciones en Tcl utilizando los objetos de simulador en la librería de OTcl. Los organizadores de eventos y la mayor parte de los componentes de red están implementados en C++ y disponibles hacia OTcl a través de una vinculación que está implementada usando tclcl. Todo esto junto hace NS, el cual es un intérprete extendido Tcl orientado a objetos con librerías de simulador de redes.

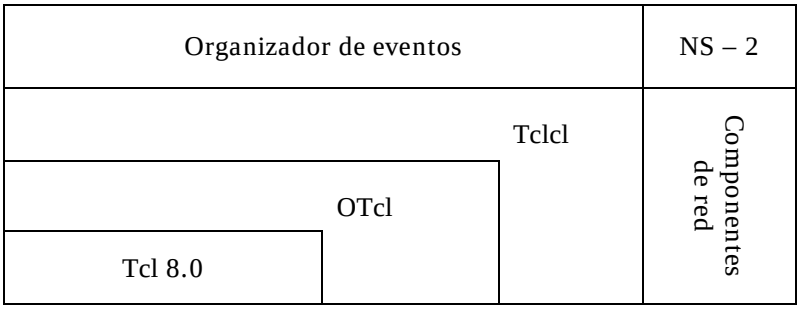

Figura 5.1 Vista de la arquitectura de NS.

En este punto, uno se debe de preguntar acerca de como obtener resultados de simulación en NS. Cuando termina una simulación, NS produce uno o más archivos de salida de texto que contienen datos detallados de simulación, si se especifica que haga esto en el script de OTcl. Los datos pueden ser para análisis de simulación o como una entrada a una herramienta de visualización de simulación gráfica, NAM.

[55] The Vint Proyect. *The NS Manual.* 2001. P. 17.

El NAM tiene una interfaz gráfica de usuario similar a un reproductor de CD, además tiene una pantalla de control de velocidad. Adicionalmente, puede presentar información gráfica tal como rendimiento y número de paquetes tirados en cada link, aunque la información gráfica no puede ser usada para análisis preciso de la simulación."[56]

"Desde la perspectiva del usuario, NS es un interpretador de scripts Orientados a objetos de TCL (OTcl) que tiene un organizador de eventos de simulación, objetos de componentes de red y librerías de módulos de organización de red (plumbing). En otras palabras, para usar NS, se programa en el lenguaje OTcl.

Otro componente importante además de los objetos de red, es el organizador de eventos. Un evento en NS es el ID de paquete que es único para cada paquete y tiene un tiempo de registro además de un apuntador a un objeto que maneja el evento.

En NS, un organizador de eventos mantiene información del tiempo de simulación y despacha todos los eventos en la cola de espera programados para el tiempo actual mediante el llamado de componentes de red apropiados, estos componentes normalmente son aquellos que expidieron los eventos, y se les permite realizar una acción apropiada, asociada con el paquete apuntado por el evento.

Los componentes de red se comunican entre si pasándose paquetes, sin embargo esta tarea no consume tiempo de simulación. Todos los componentes de red que necesitan gastar algún tiempo de simulación al manejar un paquete (esto es, un retardo) utilizan el organizador de eventos mediante la expedición de un evento para el paquete y esperan que este evento sea despachado por si mismo antes de realizar cualquier acción en el manejo del paquete.

Otro uso del organizador de eventos es el de temporizador; los temporizadores utilizan el organizador de eventos en una manera similar que lo hacen los retardos. La única diferencia es que el temporizador mide un valor de tiempo asociado con un

[56] Méndez, Luis. Op. Cit. P. 65.

paquete y realiza una acción apropiada relacionada al paquete después de que cierto tiempo ha pasado, y no simula un retardo."[57]

Existen actualmente cuatro organizadores disponibles en el simulador, cada uno de los cuales está implementado usando una estructura de datos diferente: una lista simplemente ligada, pila, cola calendarizada (por defecto), y un tipo especial llamado *tiempo real*. El organizador corre mediante la selección del siguiente evento que llegó primero, ejecutándolo hasta su finalización y regresando a ejecutar el siguiente evento. La unidad de tiempo usada por el organizador son los segundos."[58]

5.5. CARENCIAS DEL SIMULADOR NS – 2

El NS soporta una gran variedad de protocolos, incluyendo casi todas las variantes de TCP, diversas formas de multicast, de redes cableadas, varios protocolos de ruteo Ad-Hoc y varios modelos de propagación, difusión de datos, comunicaciones satelitales y otras cosas.

Sin embargo, "en el modelo de simulación para TCP de una vía no hay anuncios de ventana dinámica, los cómputos de número de segmento y de Ack están en unidades de paquetes y no hay un establecimiento/fin de la conexión del tipo SYN/FIN. Por otra parte el modelo de simulación para TCP de dos vías es muy similar a TCP 4.x BSD (Berkeley Software Distribution), excepto que no hay anuncios de ventan dinámica, no hay estados de persistencia o espera 2MSL, no hay segmentos de datos urgentes ni de Reset.

Recientemente, las funcionalidades SACK (Selective ACK), Newreno y Tahoe han sido añadidas al FullTCP. Finalmente, existen un número de protocolos contribuidos

[57] Ibídem. P. 63 – 64.
[58] The Vint Proyect. *The NS Manual*. 2001. P. 37 – 38.

descritos en sus propias páginas web. Estos protocolos son a menudo para distribuciones específicas del NS y pueden no trabajar en la distribución actual."[59]

En la parte de redes inalámbricas no se ha experimentado mucho con el NS, por lo cual este no puede simular redes celulares, redes 802.11a, b o g, ni otros tipos de comunicaciones inalámbricas que están teniendo gran éxito en el mundo y es necesario su análisis y comprensión. No se hablará más de todas las carencias del simulador, sino solo de las que a este trabajo interesan.

El simulador solo puede realizar pruebas de redes inalámbricas 802.11 con anchos de banda de 1 y 2 Mbps. En la actualidad existen diversos equipos de investigadores que han desarrollado algunas mejoras al NS, como lo es el grupo Monarca que ya tiene una versión con la que se puede trabajar limitadamente con redes 802.11b, al igual que con el proyecto *Enhanced NS*.

Sin embargo estas contribuciones al código del NS no sirven de mucho para esta tesis ya que no son capaces de realizar simulaciones del estándar 802.11g, así como de realizar la adaptación automática de tasa tan fundamental para este trabajo.

Es por todo lo anterior que es necesario realizar diversas modificaciones al código del NS para poder simular de modo satisfactorio las redes inalámbricas 802.11g que son las de interés para esta tesis.

5.6. LAS REDES INALÁMBRICAS Y EL NS – 2

A continuación se analizará el modo en que el simulador NS2 maneja los modelos de redes inalámbricas, también se podrá ver como modelar estas redes y como interpretar los resultados que entrega el simulador. Básicamente se describe el modelo inalámbrico que fue aportado al código original del simulador por el grupo Monarca CMU como una

[59] http://www.isi.edu/nsnam/ns/ns-tests.html

extensión al NS. Se podrá entender como se modelan el soporte para trazado CMU y la generación de movimiento de los nodos, así como los archivos de escenario y de tráfico. Las redes inalámbricas como se ha visto anteriormente, pueden ser con infraestructura o sin esta, de tal modo que solo se analizarán los modelos de las redes inalámbricas con infraestructura.

5.6.1. Modelo de redes inalámbricas en NS – 2

"El modelo inalámbrico esencialmente consiste en un nodo móvil en el núcleo, con cualidades de soporte adicional que permiten la simulación de redes ad-hoc multi-hop, WLAN, etc. El objeto MobileNode es un objeto dividido. La clase MobileNode C++ está derivada de la clase padre Nodo. De este modo, un nodo móvil es el objeto nodo básico con funcionalidades agregadas de un nodo inalámbrico y móvil como la habilidad de moverse dentro de una topología dada, la habilidad de recibir y transmitir señales a y desde un canal inalámbrico, etc. Las redes inalámbricas en el simulador están dadas por un nodo móvil, sus mecanismos de ruteo, los protocolos de ruteo DSDV (Destination Sequenced Distance Vector Routing), AODV (Ad-hoc On-demand Distance Vector), TORA (Temporally-Ordered Routing Algorithm) y DSR (Dynamic Source Routing), por la creación de pilas de red que permiten el acceso al canal en MobileNode, por el soporte de trazado y por la generación de escenarios de movimiento y tráfico."[60]

5.6.2. Simulación de redes inalámbricas en NS – 2

"Un nodo móvil consiste de componentes de red como capa de enlace (LL), Cola de espera (IFQ), capa MAC, etc. En el inicio de una simulación inalámbrica, se necesita definir el tipo de cada uno de estos componentes de red. Además, se necesita definir otros parámetros como el tipo de antena, el modelo de propagación, el tipo de protocolo de enrutamiento empleado por los nodos móviles y algunos otros que se pueden

[60] The Vint Proyect. *The NS Manual*. 2001. P. 143.

observar en la tabla 5.1. El Script en OTcl empieza con una lista de estos diferentes parámetros. Donde val( ) es un arreglo que se utiliza para definir estas variables."[61]

Una vez que se han definido todas las opciones a utilizar por el simulador es necesario crear e iniciar el organizador de eventos del que ya se ha hablado. "Un evento es la ejecución de un procedimiento Tcl programado para ocurrir en un tiempo determinado."[62] Para poder crear e iniciar el organizador de eventos se utilizan diferentes parámetros y comandos, los más simples y generalmente siempre usados son:

`set val(chan)`	`Channel/WirelessChannel`	`;# Tipo de canal`
`set val(prop)`	`Propagation/TwoRayGround`	`;# Modelo de propagación`
`set val(netif)`	`Phy/WirelessPhy`	`;# Tipo de Interfase de red`
`set val(mac)`	`Mac/802_11`	`;# Tipo de MAC`
`set val(ifq)`	`Queue/DropTail/PriQueue`	`;# Tipo de cola de espera`
`set val(ll)`	`LL`	`;# Tipo de capa de enlace`
`set val(ant)`	`Antenna/OmniAntenna`	`;# Tipo de antena`
`set val(x)`	`670`	`;# X dimensión de la topografía`
`set val(y)`	`670`	`;# Y dimensión de la topografía`
`set val(ifqlen)`	`50`	`;# Max no. de paquetes en ifq`
`set val(seed)`	`0.0`	`;# Semilla para generar número aleatorio`
`set val(adhocRouting)`	`DSR`	`;# Protocolo de enrutamiento`
`set val(nn)`	`3`	`;# Número de nodos`
`set val(cp)`	`"../mobility/scene/cbr-3-test"`	`;# Archivo de tráfico`
`set val(sc)`	`"../mobility/scene/scen-3-test"`	`;# Archivo de scenario`
`set val(stop)`	`400.0`	`;# tiempo de simulación`

Tabla 5.1 Definición de opciones para la simulación.

- `set ns_ [new Simulator]` ;# Sirve para crear el organizador de eventos
- `$ns at <tiempo> <evento>` ;# Sirve para programar diversos eventos en diferentes tiempos donde los eventos pueden ser cualquier comando permitido del tipo ns/tcl
- `$ns_ run` ;# Sirve para iniciar el organizador de eventos. Se coloca al final de nuestro script

[61] Méndez, Luis. Op. Cit. P. 67.
[62] Idem.

Posteriormente es necesario crear la topología con la que correrá nuestra simulación. Los comandos que se utilizan para crear la topología son:

- `set topo [new Topography]`
- `$topo load_flatgrid $val(x) $val(y)`

Una vez que se definió la topología, es necesario indicar al simulador que la salida de la simulación se dirija directamente a diversos archivos mediante la opción de trazado que contempla el NS. Para lo anterior es necesario agregar al script los comandos:

- `set tracefd [open misimulacion.tr w]`
- `$ns_ trace-all $tracefd`

Independientemente de este archivo se puede crear otro archivo con información necesaria para que a través del NAM, se pueda analizar la simulación. Para obtener este archivo es necesario agregar estas líneas:

- `set namtrace [open misimulacion.nam w]`
- `$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)`

En ambos archivos el nombre es necesario, aquí a modo de ejemplo tiene el nombre *misimulación*, pero puede ser el que el usuario quiera. Una vez realizado todo lo anterior se puede empezar la creación y configuración de los nodos, que para este caso se tratan de nodos móviles. Esto se logra a través de una interfase de programación de aplicación, los llamados APIs, la cual configura los nodos móviles con todos sus valores respectivos.

Las cualidades de los nodos que configura este API son: tipo de capa de enlace, tipo de capa MAC a usar, tipo de antena, tipo de capa física, tipo de canal, instancia de topología, agente de trazado, entre otras. La forma en que se debe operar es la siguiente:

```
$ns_ node-config –adhocRouting $val(adhocRouting) \
                 -llType   $val(ll) \
                 -macType  $val(mac) \
                 -ifqType  $val(ifq) \
                 -ifqLen   $val(ifqlen) \
                 -antType  $val(ant) \
                 -propType $val(prop) \
                 -phyType  $val(netif) \
                 -channelType  $val(chan) \
                 -topoInstance $topo \
                 -agentTrace  ON \
                 -routerTrace ON \
                 -macTrace    ON \
```

La parte de agentTrace es muy importante, ya que si se deshabilita esta opción no se tendrán trazados, de tal modo que no se podrán analizar los resultados en los archivos de salida antes mencionados. Una vez que se terminó de configurar el modo en que operarán los nodos es necesario crearlos, para este fin se utiliza un ciclo *for* que sirve para crear el número de nodos que se definió al principio del script mediante la variable $val (nn), de tal modo que el ciclo queda de la siguiente manera:

```
for {set i 0} {$i < $val(nn)} {incr i}
    {
     set node_($i) [$ns_ node]
     $node_($i) random-motion 0
    }
```

El valor colocado a random-motion sirve para habilitar o deshabilitar el movimiento aleatorio, en este caso es cero, lo que significa deshabilitado ya que se desea configurar los nodos para que sigan un patrón de movimiento diseñado por nosotros para poder realizar las simulaciones que se requieren.

“Existen dos mecanismos para inducir el movimiento en los nodos móviles. En el primer método, la posición inicial del nodo y sus destinos futuros deben ser asignados explícitamente. Estas directivas son normalmente incluidas en un archivo de escenario de movimiento separado.

La posición inicial y los destinos futuros de un nodo móvil deben ser asignados mediante el uso de los siguientes APIs:

- `$node set X_ <x1>`
- `$node set Y_ <y1>`
- `$node set Z_ <z1>`
- `$ns at $time $node setdest <x2> <y2> <speed>`

En el tiempo $time, el nodo empezará a moverse de su posición inicial de (x1, y1) hacia un destino (x2, y2) a la velocidad definida (speed). En este método las actualizaciones de movimiento de los nodos son desencadenadas cuando es requerido conocer la posición del nodo en un tiempo dado. Esto puede ser desencadenado por una pregunta de un nodo vecino, buscando conocer la distancia entre ellos, o la directiva setdest descrita anteriormente que cambia la dirección y velocidad del nodo. El segundo método emplea el movimiento aleatorio del nodo, visto anteriormente, la primitiva a ser usada es:

- `$mobilenode start`

La cual inicializa al nodo móvil con una posición aleatoria y tiene actualizaciones para cambiar la dirección y velocidad del nodo. Los valores de destino y velocidad son generados en una manera aleatoria."[63]

"Normalmente para grandes topologías, los patrones de movimiento y conexiones de tráfico están definidos en archivos separados por conveniencia. Estos archivos de movimiento y tráfico pueden ser generados usando los generadores CMUs de movimiento y conexión.

El generador para crear los archivos de movimiento se puede encontrar en el directorio *~ns/indep-utils/cmu-scen-gen/setdest*. Se debe compilar los archivos bajo setdest para crear un ejecutable. Se corre setdest con sus argumentos de la siguiente manera:

[63] The Vint Proyect. *The NS Manual.* 2001. P. 127.

- `./setdest -n <num_of_nodes> -p <pausetime> -t <simtime> -x <maxx> -y <maxy> > <outdir>/<miarchivomovimiento>`

Una vez que ya se definió el escenario junto con las combinaciones de movimientos que tendrán los nodos es necesario generar tráfico en la red, para poder observar como funciona la simulación y analizar variados aspectos que sólo si existe tráfico en la red se pueden observar.

El tráfico puede ser del tipo CBR (Constant Bit Rate) o TCP, se puede añadir el tráfico que se requiera a cada nodo, sin embargo para este propósito también existe utilidades que ayudan a generar estos tráficos. Estas utilidades trabajan en conjunto con el NS, ya que es necesario llamar al programa principal del NS para correr las utilidades de generación de tráfico, debido a que estas son scripts en tcl.

La ruta en donde se pueden localizar estos scripts para utilizarlos es *~ns/indep-utils/cmu-scen-gen* y tienen por nombre *cbrgen.tcl* y *tcpgen.tcl* y pueden ser usados para generar conexiones CBR y TCP respectivamente.

Al igual que la utilidad para escenario o movimiento de los nodos, es necesario dar ciertos parámetros al script para definir el número de nodos a usar, el tipo de tráfico a generar, el número máximo de conexiones, una semilla aleatoria y la tasa de datos que se requieren. También es necesario direccionar la salida de este script a un archivo para poderlo usar en la simulación. Este script tiene una sintaxis muy particular que es como sigue:

- `ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connection] [-rate rate] > <outdir>/<archivotrafico>`
- `ns tcpgen.tcl [-nn nodes] [-seed seed] > <outdir>/<archivotrafico>`

Una vez que se tiene el archivo de tráfico se puede modificar conforme las propias necesidades, en él se pueden variar los nodos y los tiempos en los que el tráfico fluye, los

tipos de tráfico, la tasa de datos, entre otros, ya que el script genera todo esto en base a una distribución aleatoria."[64]

En este momento es necesario colocar la ruta donde están localizados los archivos de escenario y tráfico en las opciones de simulación vistas anteriormente en las variables val(sc) y val(cp) respectivamente. Para que el script sea capaz de leer estos archivos se utilizan las siguientes directivas:

- `source $val(sc)`
- `source $val(cp)`

Una vez que se definieron todas las opciones del modelo de movilidad, así como todos los elementos necesarios para que este funcione, se debe definir la posición inicial de los nodos conjuntamente entre el simulador y la herramienta de interfaz gráfica.

Para ello es necesario definir el tamaño del que se requiere que los nodos tengan en esta interfaz mediante la opción <size> y también definir el número de nodos mediante $val(nn). Para todo lo anterior se utiliza un ciclo *for* para darle la misma característica a todos los nodos que se utilicen, de tal modo que las líneas en el script quedan como sigue:

```
for {set i 0} {$i < $val(nn)} {incr i}
    {
     $ns_ initial_node_pos $node_($i) <size>
    }
```

Por último es necesario decirle al simulador en que momento termina la simulación, para así este se detenga y reestablezca los componentes de los nodos móviles para que estos queden listos para futuras simulaciones. Para llegar a este fin se utilizan las siguientes directivas:

[64] Ibídem. P. 141 – 142.

```
for {set i 0} {$i < $val(nn)} {incr i}
    {
     $ns_ at <time0> “$node_($i) reset”;
    }
$ns_ at <time1> “stop”
$ns_ at <time2> “$ns_ halt”
proc stop {} {
     global ns_ tracefd
     close $tracefd
}
```

5.6.3. Análisis de resultados de simulación

“El soporte de trazado para simulaciones inalámbricas actualmente usa objetos cmu-trace. En el futuro esto será extendido para unirse con el soporte de trazado y monitoreo disponible en el NS, lo que también deberá incluir soporte de NAM para módulos inalámbricos. Los objetos cmu-trace son de tres tipos CMUTrace/Drop, CMUTrace/Recv y CMUTrace/Send. Estos son usados para rastrear los paquetes que son tirados, recibidos y enviados por agentes, routers, capas MAC o colas de interfase en NS.

Los métodos y procedimientos usados para implementar el soporte de trazado inalámbrico pueden ser encontrados en *~ns/trace.{cc,h}* y *~ns/tcl/lib/ns-cmutrace.tcl.*

El campo de tipo (descrito en la definición de clase Trace) es usado para diferenciar entre diferentes tipos de trazado. Para cmu-trace este puede ser *s* para enviando, *r* para recibiendo o *D* para un paquete descartado. Un cuarto tipo *f* es usado para denotar el reenvío de un paquete (cuando el nodo no es el creador del paquete).

Las funciones con que cuenta la programación del método CMUTrace llaman a diferentes formatos dependiendo del tipo de paquete que está siendo trazado. Una cuenta de la compensación para el buffer es guardada y es pasada a través de diferentes funciones de trazado. El formato más básico está definido por format_mac() y es usado para trazar todos los tipos pkt. Las otras funciones de formato imprimen información adicional como está definido por los tipos de paquetes. Un ejemplo de un trazado para un paquete tcp es como sigue:

- `r 160.093884945 _6_ RTR --- 5 tcp 1492 [a2 4 6 800] ----- -- [655 36:0 16777984:0 31 16777984] [1 0] 2 0`

Aquí se puede ver un paquete de datos TCP siendo recibido por un nodo con id de 6. El UID de este paquete es 5 con un tamaño de 1492 bytes. Los detalles de MAC muestran un paquete IP (ETHERTYPE_IP está definido como 0x0800), el mac-id de este nodo receptor es 6. Del cual el nodo transmisor tiene un id de 4 y el tiempo esperado para enviar este paquete de datos a través del canal inalámbrico es a2 (conversión hex/dec: 160+2 sec.). Adicionalmente, IP traza información sobre las direcciones IP fuente y destino.

La fuente traduce (usando un direccionamiento de nivel 3 de 8/8/8) a una cadena de dirección de 0.1.0 con un puerto de 0. La dirección destino es 1.0.3 con una dirección de puerto de 0. El valor TTL es 31 y el destino fue un salto fuera de la fuente. Adicionalmente el formato TCP imprime información sobre el número de secuencia de tcp de 1, y ack de 0. Existen otros formatos descritos en *~ns//cmu-trace.cc* para tipos de paquetes DSR, UDP, TCP/ACK y CBR."[65]

5.7. EL ARCHIVO DE MAC 802.11 DEL NS – 2

El estándar IEEE 802.11 está implementado en el simulador de una manera que solo funciona la primera versión del estándar. Como la mayor parte del simulador esta implementación se encuentra hecha en lenguaje C y para ella existen diversos archivos que maneja el simulador tanto para la capa física como para la capa MAC. Los archivos que más interesan a este trabajo son los de la capa MAC que se encuentran en *~ns/mac/mac-802_11.{cc,h}*. "Existen cuatro diferentes trayectorias que el código puede seguir:

- Transmitiendo un paquete
- Recibiendo un paquete destinado para sí mismo

[65] Ibídem. P. 135 – 138.

- Escuchando un paquete no destinado para sí mismo
- Paquetes colisionando

A continuación se analizará con detalle lo que sucede en las trayectorias de transmisión y recepción de paquetes que son las que más interesan en esta tesis.

5.7.1. Transmitiendo un paquete

Generalmente la transmisión toma la siguiente trayectoria (cuando no hay errores o congestión):

recv( ) ➔ *send( )* ➔ *sendDATA( )* y *sendRTS( )* ➔ iniciar *defer timer*
➔ *deferHandler( )* ➔ *check_pktRTS( )* ➔ *transmit( )*
➔ *recv( )* ➔ *receive timer* inicializado
➔ *recv_timer( )* ➔ *recvCTS( )* ➔ *tx_resume( )* ➔ iniciar *defer timer* ➔ *rx_resume( )*
➔ *deferHandler( )* ➔ *check_pktTx( )* ➔ *transmit( )*
➔ *recv( )* ➔ *receive timer* inicializado
➔ *recv_timer( )* ➔ *recvACK( )* ➔ *tx_resume( )* ➔ callback_ ➔ *rx_resume( )* ➔ Listo!

Cuando el primer RTS falla:

recv( ) ➔ *send( )* ➔ *sendDATA( )* y *sendRTS( )* ➔ iniciar *defer timer*
➔ *deferHandler( )* ➔ *check_pktRTS( )* ➔ *transmit( )* ➔ iniciar *send timer*
➔ *send_timer( )* ➔ *RetransmitRTS( )* ➔ *tx_resume( )* ➔ *backoff timer* inicializado
backoffHandler( ) ➔ *check_pktRTS( )* ➔ *transmit( )*
El resto es lo mismo que cuando no hay errores.

5.7.2. Recibiendo un paquete destinado a sí mismo

Generalmente la recepción de un paquete destinado a sí mismo toma la siguiente trayectoria (cuando no hay errores o congestión):

recv( ) ➔ *receive timer* inicializado

➔ *recv_timer( )* ➔ *recvRTS( )* ➔ *sendCTS( )* ➔ *tx_resume( )* ➔ iniciar *defer timer* ➔ *rx_resume( )*

➔ *deferHandler( )* ➔ *check_pkCTRL( )* ➔ *transmit( )*

➔ *recv( )* ➔ *receive timer* inicializado

➔ *recv_timer( )* ➔ *recvDATA( )* ➔ *sendACK( )* ➔ *tx_resume( )* ➔ iniciar *defer timer* ➔ uptarget_ ➔ recv( )

➔ *deferHandler( )* ➔ *check_pktCTRL( )* ➔ *transmit( )* ➔ iniciar *send timer*

➔ *send_timer( )* ➔ *tx_resume( )* ➔ Si nada ha pasado, Listo!

5.7.3. Funciones del MAC del NS – 2

A continuación se describirán brevemente las funciones más importantes utilizadas anteriormente por las trayectorias de recepción y envío de paquetes en la capa MAC.

- *recv( ) (DOWN).-* Como todos los conectores, de los cuales hereda el MAC, el paquete a ser enviado es recibido por la función recv(). Debido a que la función recv() es también llamada cuando un paquete viene del canal, recv() checa el campo de dirección en la cabecera del paquete. Si la dirección es DOWN (de bajada), significa que el paquete viene de una capa superior, y el paquete es pasado sobre la función send().

- *recv( ) (UP).-* La función recv() es llamada cuando un paquete es recibido de cualquiera de las capas superiores o inferiores. Si el paquete es recibido de una capa inferior, entonces el primer chequeo será saltado. En este punto la capa física ha recibido el primer bit del paquete entrante, pero el MAC no puede hacer nada con el paquete hasta que el paquete completo sea recibido. Si el paquete es recibido mientras el MAC está actualmente transmitiendo otro paquete, entonces el paquete recibido será ignorado. Si el MAC no está recibiendo ningún paquete, entonces el estado rx_state_ es cambiado a RECV y el CHECK BACKOFF TIMER es llamado.

Después, el paquete entrante es asignado a pktRx_ y el temporizador de recepción es inicializado para el txtime() del paquete. Si el MAC está recibiendo un paquete cuando este paquete ya llego, el MAC comparará el poder de recepción del nuevo paquete con el del paquete más antiguo. Si el poder del nuevo paquete es menor que el paquete viejo por lo menos por el umbral de captura, el nuevo paquete será ignorado y la función capture() es llamada. Si los niveles de poder de los dos paquetes están muy cercanos, habrá una colisión y el control se transfiere a collision(), el cual descartará el paquete entrante. El paquete original no será descartado hasta que su recepción esté completa. El control regresará al MAC cuando el temporizador de recepción expire, llamando recvHandler(), el cual en regreso va directamente a recv_timer().

- *send( )*.- La función send() primero checa el modelo de energía, descartando el paquete si el nodo está actualmente en modo dormido. Esta entonces ajusta callback_ al manipulador pasado con el paquete. Después, send() llama a sendDATA() y sendRTS el cual construye la cabecera MAC para el paquete de datos y para el paquete RTS para ir por lo largo con el paquete de datos el cual está guardado en pktTx_ y pktRTS_ respectivamente. La cabecera MAC para el paquete de datos es entonces asignada a un número de secuencia único (con respecto al nodo).

  Posteriormente, el MAC checa su temporizador de backoff o retroceso. Si el temporizador backoff no está actualmente en cuenta regresiva, entonces el nodo checa si el canal (medio) está libre, y si esto ocurre el nodo empezará a diferir. El nodo checa esto usando la función is_idle(). Si el medio es detectado ocupado, entonces el nodo inicializa su temporizador de backoff. En este punto, la función send() ha terminado y el control se resumirá cuando uno de los temporizadores expire, llamando entonces a deferHandler() o backoffHandler().

- *sendDATA( )*.- Esta función construye la cabecera MAC para el paquete de datos. Esto implica incrementar el tamaño del paquete, ajustando el tipo como datos, y el subtipo como datos. El paquete ahora debe tener una cabecera completa MAC adjunta a él. La función entonces guarda el txtime del paquete, el cual es computado

por la función txtime(). Mediante txtime, básicamente nos referimos al tamaño del paquete multiplicado por la tasa de datos. Usted verá que este cálculo es hecho dos veces – la primera vez es solo un desperdicio. Es calculado de nuevo porque un valor diferente de tasa de datos es usado si el paquete es un paquete de broadcast.

Además, si el paquete no es un paquete de broadcast, el campo de duración en la cabecera MAC es computado. Por duración, nos referimos a la cantidad de tiempo que esta comunicación aún necesita el canal después que el paquete ha sido transmitido. Para el caso de un paquete de datos, esto corresponde a la cantidad de tiempo para transmitir un ACK más un espaciado entre-trama corto (SIFS). Si el paquete se trata de un paquete de broadcast, este campo es ajustado a cero (no hay ACKs para paquetes de broadcast). Ahora, el MAC ha terminado de construir la cabecera MAC para el paquete y finalmente asigna la variable interna pktTx_ para apuntar al paquete con el que hemos estado trabajando. Esto esencialmente es una manera de guardar el paquete a ser transmitido en un buffer local en el MAC. Ahora el código regresa a la función send().

- *sendRTS( ).*- Esta función está a cargo de crear un paquete RTS con el destino especificado en conjunción con el paquete de datos que el MAC está tratando de enviar. La primer cosa que hace es inspeccionar el tamaño del paquete contra el RTSThreshold. Si el paquete es mas pequeño (o es broadcast) entonces ningún RTS es enviado antes de que los datos sean transmitidos (el mecanismo RTS/CTS no es usado). En este caso, la función simplemente regresa el control a la función send(). De otra forma, un nuevo paquete de marca es creado (actualmente hecho en la primer línea de la función) y sus campos son ajustados apropiadamente, es decir, el tipo es ajustado como un paquete MAC.

  Una estructura rts_frame es usada para llenar el resto de la cabecera del paquete y los valores apropiados son puestos en los campos rts. El campo de destino es llenado con los parámetros pasados a la función y el rf_ta (fuente?) es llenado con la dirección MAC. El campo de dirección es también calculado como el tiempo para transmitir un CTS, el paquete de datos (pktTx_) y un ACK (mas 3 SIFS). Después de

que el RTS ha sido construido, la variable interno de estado pktRTS_ es asignada a un apuntador al nuevo RTS. Después de esto, el control es regresado a la función send().

- *sendCTS( )*.- Esta función está a cargo de crear un paquete CTS y apuntar pktCTRL_ a este. Todo procede sencillamente, con campos dando valores obvios. El campo de duración es ajustado para ser el mismo que fue en el RTS, excepto menos el txtime de un CTS y un tiempo sifs_, desde que esa cantidad de tiempo ya ha transcurrido inmediatamente otra estación decodifica el paquete. Después de que la creación del paquete CTS está hecha, pktCTRL_ es apuntado al nuevo paquete y el control regresa a recvRTS().

- *sendACK( )*.- Esta función es responsable de crear un paquete ACK para ser enviado en respuesta a un paquete de datos. El paquete es creado y todos los campos son llenados con valores obvios. El campo de duración es ajustado a cero indicando a otros nodos que una vez que este ACK ha sido completado, ellos no necesitan diferir a otra comunicación. Una vez que el paquete ha sido construido satisfactoriamente, pktCTRL_ es apuntado al nuevo ACK y el control regresa a recvDATA().

- *transmit( )*.- Esta función tiene dos argumentos, un paquete y un valor de timeout. Esta ajusta una variable de bandera, tx_active_, a 1 para indicar que el MAC está actualmente transmitiendo un paquete. La función entonces realiza un chequeo ya que si este es un ACK siendo transmitido entonces es posible que el nodo esté recibiendo un paquete, en tal caso ese paquete se perderá. Este siguiente bloque checa si el MAC está actualmente recibiendo un paquete y si hay un ACK siendo transmitido, y si esto pasa, marca el paquete siendo recibido como sin errores. Después, el paquete es en realidad pasado a la interfase de red (clase WirelessPhy) la cual es apuntada por downtarget_.

  En realidad, solo una copia del paquete es enviada abajo en caso de que se necesite una retransmisión. Finalmente, dos temporizadores son inicializados – el temporizador enviar es inicializado con el valor timeout, el cual alerta al MAC que la

transmisión probablemente falló. También, el temporizador de interfase (mhIF_) es inicializado con el txtime() del paquete – cuando este temporizador expira, el MAC sabrá que la capa física ha completado la transmisión del paquete.

- *RetransmitRTS( ).*- Esta función es llamada en respuesta a un CTS que no ha sido recibido después que un RTS fue enviado. Primero, la función hace alguna señalización de coleccionismo, grabando este como un RTS fallido, y la cuenta de reintento corta (ssrc_) es incrementada. La cuenta de reintento corta es mantenida para que el MAC sepa cuando darse por vencido en este paquete y descartarlo, lo cual pasa cuando ssrc_ alcanza el valor de ShortRetryLimit en el MAC MIB. El descarto es manipulado llamando a la función discard() en el paquete RTS y reseteando el apuntador pktRTS_ a cero.

  Entonces el paquete de datos es también descartado mediante el llamado de la misma función discard(). El ssrc_ es reseteado a cero y la ventana de congestión es reseteada a su valor inicial. De otra manera, el mismo RTS apuntado a pktRTS_ es mantenido, pero un campo de reintento en el RTS es incrementado. Debido al mecanismo de prevención de congestión, la ventana de congestión es duplicada y luego el temporizador backoff es inicializado usando esta nueva ventana de congestión. Esto significa que el control eventualmente regresará a backoffHandler().

- *RetransmitDATA( ).*- Esta función es llamada cuando un ACK no es recibido en respuesta a un paquete de datos siendo enviado. Si el paquete de datos fue un paquete broadcast, un ACK no debe ser esperado así que el paquete es tratado como si hubiera sido transmitido satisfactoriamente y es liberado y la ventana de congestión es reseteada. El contador backoff es inicializado aunque, no estamos realmente seguros porque. Dos cuentas de retroceso separadas son mantenidas dependiendo en si o no un RTS está siendo usado para este paquete de datos.

  Si un RTS no está siendo usado, el límite de reintento corto es usado, de otra forma el límite de reintento largo es usado como un umbral. Si la cuenta de reintento ha excedido el umbral, entonces el paquete de datos es descartado usando la función

discard() y la cuenta de reintento y la ventana de congestión son reseteadas. Si la cuenta de reintento no ha sido excedida, el paquete de datos es preparado para retransmisión incrementando un campo de reintento en la cabecera MAC, duplicando la ventana de congestión, y luego inicializando el temporizador backoff. Esto significa que el control eventualmente regresará a backoffHandler().

- *recvRTS( )*.- Esta función es llamada por recv_timer después de que un paquete RTS completo ha sido recibido. Si el tx_state_ no esta desocupado, entonces el paquete no será escuchado, así que este es simplemente descartado. Además, si el MAC está actualmente respondiendo a otro nodo (pktCTRL_ no es cero) entonces el RTS será ignorado. De otra forma, el MAC está en un estado tal que puede recibir un paquete, así que se prepara para enviar un CTS de regreso llamando a sendCTS(). Después, el MAC detiene el tiempo de prórroga y llama a tx_resume() – el cual reiniciará el tiempo de prórroga por la cantidad apropiada de tiempo. El control entonces regresa a recv_timer().

- *recvCTS( )*.- Esta función es llamada por recv_timer después de que un paquete completo CTS ha sido recibido, significando que el MAC puede ahora enviar sus datos. Si el MAC no hace uso del paquete RTS este solo transmite, es liberado y el pktRTS_ es ajustado a cero. El temporizador de envío es detenido. El control entonces pasa a tx_resume(), el cual ajusta el tiempo de prórroga, y el control finalmente regresa a recv_timer().

- *recvACK( )*.- Esta función es llamada por el recv_timer después de que un paquete completo ACK ha sido recibido, indicando una transmisión de datos satisfactoria. Primero el MAC verifica si este realmente acaba de enviar un paquete de datos (tx_state = MAC_SEND) y descarta el ACK si no lo hizo. El MAC ahora conoce que acaba de transmitir su paquete de datos satisfactoriamente, así que libera el pktTx_ y lo ajusta a cero. El temporizador de envío es también detenido. El MAC entonces resetea la cuenta de reintento apropiada, corta si el RTS no fue usado, larga si lo fue. También, la ventana de congestión es reseteada y el MAC inicia su temporizador backoff así que este no enviará de nuevo inmediatamente. El control entonces va a

tx_resume() y luego de regreso a recv_timer(). En tx_resume(), una vez que no hay paquetes listos para enviar, la retirada será invocada, diciendo efectivamente a la cola de interfase que envíe otro paquete para transmisión.

- *recvDATA( ).*- Esta función es llamada por el recv_timer después de que un paquete de datos completo ha sido recibido, indicando que este nodo acaba de recibir un paquete de datos satisfactoriamente. Primero, el MAC quita la cabecera MAC del paquete, dejándolo listo para ser enviado a las capas superiores. Si el paquete de datos no fue un broadcast, los paquetes RTS están siendo usados, y el tx_state_ indica que el último paquete que el MAC envío fue un CTS, entonces ese CTS (pktCTRL_) es limpiado (liberado y el pktCTRL_ ajustado a cero). Y de nuevo, el temporizador de envío es detenido.

  Si el MAC no acaba de enviar un CTS cuando debería haberlo hecho, el paquete de datos es descartado porque los eventos no sucedieron en orden correcto y la función regresa. De otra manera, el paquete de datos fue recibido correctamente y el MAC se prepara para enviar un ACK llamando a sendACK() y luego a tx_resume() para iniciar el temporizador de prórroga apropiadamente. Si un CTS no fue enviado (porque no hubo un correspondiente RTS), entonces el MAC checa pktCTRL_. Si hay un paquete de control ahí, el MAC descartará el paquete de datos porque no hay lugar en el buffer para un paquete ACK (el ACK irá en pktCTRL_). De otra manera, sendACK() es llamado para crear un paquete ACK para enviarse. En este caso, si el temporizador de envío no está actualmente en cuenta regresiva, tx_resume() es llamado para iniciar el temporizador de prórroga.

  Después, el MAC actualiza su memoria cache de número de secuencia – si el paquete es solamente unicast. El paquete es verificado para asegurarse que el nodo origen cabrá en la cache – es posible para la cache que haya sido configurada con un tamaño incorrecto, es decir, menos que el número total de nodos en el sistema. Entonces el número de secuencia del paquete que se acaba de recibir es comparado con el número de secuencia mas recientemente recibido y si concuerdan, el paquete de datos es descartado ya que está duplicado (el mismo paquete se recibió dos veces). Si

el nodo origen no está en la cache (la cache es muy pequeña), algunas advertencias son desplegadas.

El paquete de datos es entonces pasado al uptarget_ - la capa sobre el MAC (usualmente capa de enlace). Esto significa que el paquete de datos ha sido recibido completo por el nodo y está en camino a la pila de protocolos."[66]

Las funciones descritas arriba son las más importantes del MAC, sin embargo existen más funciones, tantas como las que aparecen en las diferentes trayectorias para recibir y enviar paquetes. En lo respectivo a los temporizadores estos están definidos en los archivos *~ns/mac/mac-timers.{cc,h}* mientras que los manipuladores (funciones llamadas cuando los temporizadores expiran) están en *mac-802_11.cc*.

[66] Robinson, Joshua. *802.11 MAC code in NS – 2 (version 2.28)*. http://www.ece.rice.edu/~jpr/ns/docs/802_11.html

CAPÍTULO VI
IMPLEMENTACIÓN DEL ESTÁNDAR IEEE 802.11g EN NS – 2

Como se ha visto antes, el simulador no contempla el estándar 802.11G, de tal modo que fue necesario que se realizaran diversos cambios en el simulador para poder simular estas redes. A continuación se mostrarán las modificaciones a los archivos necesarios para poder llegar a este fin. Es importante mencionar que solo se mostrarán las modificaciones en cada archivo, mas no los archivos completos, ya que algunos son bastante extensos. Si usted desea observar los archivos completos, al principio de cada explicación de cada modificación se muestra el directorio donde se puede encontrar dicho archivo.

6.1. MODIFICACIONES AL ARCHIVO ns-mac.tcl

El archivo ns-mac.tcl tiene cierta importancia, ya que en él se definen diversos valores para la capa MAC tanto para redes cableadas como inalámbricas. En este trabajo solo se tocó la parte de redes inalámbricas, aunque cabe señalar que aún antes de que se modificara este archivo, el simulador ya funcionaba con los parámetros de 802.11g con otros cambios que realizamos, sin embargo, no está de más realizar estas modificaciones que son muy simples, al solo tener que cambiar los valores que trae el simulador por defecto por los que son del estándar 802.11g.

Este archivo se puede encontrar en la carpeta *~ns-allinone-2.28/ns-2.28/tcl/lan/ns-mac.tcl.* A continuación se muestra el código modificado de este archivo desde unas líneas antes de las modificaciones hasta unas líneas después, este archivo se encuentra escrito en lenguaje tcl:

```
#default bandwidth setting done during mac initialization (c++)

Mac set bandwidth_ 54Mb              ;# Ancho de Banda para 802.11g
Mac set delay_ 0us                   ;# Retardo inicial a nivel MAC 0 us

# IEEE 802.11G MAC settings
if [TclObject is-class Mac/802_11] {
     Mac/802_11 set delay_ 64us      ;# Retardo de 64 us para redes inalámbricas
     Mac/802_11 set ifs_ 16us        ;# Tiempo de 16 us para ifs
     Mac/802_11 set slotTime_ 16us   ;# Tiempo de Slot de 16 us
     Mac/802_11 set cwmin_ 15        ;# Ventana de contención mínima 15
     Mac/802_11 set cwmax_ 1023      ;# Ventana de contención máxima 1023
     Mac/802_11 set rtxLimit_ 16     ;# Limite de Retransmisión 16
     Mac/802_11 set bssId_ -1        ;# Identificador de bss -1
     Mac/802_11 set sifs_ 10us       ;# Tiempo de 10 us para sifs
     Mac/802_11 set pifs_ 12us       ;# Tiempo de 12 us para pifs
     Mac/802_11 set difs_ 16us       ;# Tiempo de 16 us para difs
     Mac/802_11 set rtxAckLimit_ 1   ;# Límite de retransmisión de Ack 1
     Mac/802_11 set rtxRtsLimit_ 3   ;# Límite de retransmisión de RTS 1
     Mac/802_11 set basicRate_ 6Mb   ;# Establece el BasicRate en 6 Mb
       ;#- se coloca en 0 para usar el BasicRate igual al DataRate
     Mac/802_11 set dataRate_ 54Mb   ;# Establece el DataRate en 54 Mb
       ;#- Se usa tanto para Control como para Datos.
}
```

6.2. MODIFICACIONES AL ARCHIVO ns-default.tcl

El archivo ns-default.tcl es uno de los más importantes en el simulador NS – 2, ya que en este archivo se guardan todos los valores que usará el simulador por defecto si es que no se le ajustan diferentes valores en la simulación. Para que no se tengan que estar definiendo en cada simulación los valores del estándar 802.11g, es necesario cambiar los valores que tomará por defecto el simulador por los que interesan.

Esto se hace en dos partes diferentes del código del archivo ns-default.tcl, primeramente en la parte de las variables para la capa MAC y posteriormente en las variables para la capa física. Este archivo se puede encontrar en el directorio *~ns-allinone-2.28/ns-2.28/tcl/lib/ns-default.tcl* y al igual que el archivo analizado anteriormente se encuentra escrito en tcl. A continuación se puede observar la primer parte de código que fue modificada para la capa MAC:

```
Mac/802_11 set CWMin_          15             ;# Ventana de Contención mínima
Mac/802_11 set CWMax_          1023           ;# Ventana de Contención máxima
Mac/802_11 set SlotTime_       0.000009       ;# Tiempo de Slot 9µs
Mac/802_11 set CCATime_        0.000004       ;# Tiempo Clear Channel Assessment
Mac/802_11 set RxTxTurnaroundTime 0.000002 ;# Tiempo ida y vuelta
Mac/802_11 set SIFS_           0.000010       ;# SIFS 10µs
Mac/802_11 set PreambleLength_   74           ;# Longitud del Preámbulo 144 bit
Mac/802_11 set PLCPHeaderLength_ 26           ;# 48 bits
Mac/802_11 set PLCPDataRate_   6.0e6          ;# 6Mbps
Mac/802_11 set RTSThreshold_   0              ;# bytes
Mac/802_11 set ShortRetryLimit_  7            ;# retransmisiones
Mac/802_11 set LongRetryLimit_   4            ;# retransmisiones

## Mac/802_11 set dataRate_ 54Mb
## Mac/802_11 set basicRate_ 6Mb
```

Arriba se puede observar que la parte de dataRate_ y basicRate_, los cuales son muy importantes, se encuentra comentada, esto es debido a que se realizó una modificación para que estos valores se ajusten automáticamente dependiendo de la distancia entre nodos como sucede en el estándar 802.11g, y de haber dejado estos valores estos se usarían durante toda la simulación sin importar la distancia entre nodos. En las siguientes líneas se puede observar la segunda parte del código del archivo ns-default.tcl que fue modificada para la capa física:

```
Phy/WirelessPhy set CPThresh_ 10.0          ;# Umbral de Colisiones
Phy/WirelessPhy set CSThresh_ 1.559e-11     ;# Umbral de Censado de Portadora
Phy/WirelessPhy set RXThresh_ 1e-8          ;# Umbral de Potencia de Recepción
Phy/WirelessPhy set bandwidth_ 54e6         ;# Ancho de banda para WLAN
Phy/WirelessPhy set Pt_ 0.030               ;# Potencia de Transmisión
Phy/WirelessPhy set freq_ 2.642e+6          ;# Frecuencia de Transmisión
Phy/WirelessPhy set L_ 1.0                  ;# Factor de pérdida del sistema
Phy/WirelessPhy set debug_ false
Phy/WiredPhy set bandwidth_ 10e6            ;# Ancho de banda para LAN
Phy/WiredPhy set debug_ false
Phy/Repeater set debug_ false
LanRouter set debug_ false
Phy/Sat set debug_ false
Mac/Sat set debug_ false
LL/Sat set debug_ false
```

Es de mencionar que no todos los valores fueron modificados, solo algunos de Phy/WirelessPhy.

6.3. MODIFICACIONES AL ARCHIVO packet.h

El archivo packet.h se encuentra escrito en C a diferencia de los archivos anteriores que están escritos en tcl. En este archivo se especifica todo lo relacionado a los paquetes que se manipulan en el simulador. Primeramente es necesario agregar algunas variables a la cabecera común de los paquetes con el fin de poder conocer su posición y potencia de recepción y transmisión en cualquier momento. Esto se realiza en la estructura hdr_cmn con que cuenta este archivo. Este archivo se puede encontrar en el directorio *~ns-allinone-2.28/ns-2.28/common/packet.h*. A continuación se puede observar el código aumentado a la estructura antes mencionada:

```
struct hdr_cmn {
     enum dir_t { DOWN= -1, NONE= 0, UP= 1 };
     packet_t ptype_;         // packet type (see above)
     int  size_;              // simulated packet size
     int  uid_;               // unique id
     int  error_;             // error flag
     int     errbitcnt_;      // # of corrupted bits jahn
     int     fecsize_;
     double ts_;              // timestamp: for q-delay measurement
     int  iface_;             // receiving interface (label)
     dir_t     direction_;    // direction: 0=none, 1=up, -1=down

     // Added by Jonathan Lopez
        double  miPt_;
        double  miPr_;
        double  rX_,rY_,rZ_;
        double  tX_,tY_,tZ_;
     // End
```

Una vez definidas estas variables es necesario llamarlas más adelante del código para que el archivo packet.cc las pueda usar. Eso se puede observar a continuación:

```
/* per-field member functions */
     inline packet_t& ptype() { return (ptype_); }
     inline int& size() { return (size_); }
     inline int& uid() { return (uid_); }
     inline int& error() { return error_; }
     inline int& errbitcnt() {return errbitcnt_; }
     inline int& fecsize() {return fecsize_; }
     inline double& timestamp() { return (ts_); }
```

```
    inline int& iface() { return (iface_); }
    inline dir_t& direction() { return (direction_); }
    // monarch_begin
    inline nsaddr_t& next_hop() { return (next_hop_); }
    inline int& addr_type() { return (addr_type_); }
    inline int& num_forwards() { return (num_forwards_); }
    inline int& opt_num_forwards() { return (opt_num_forwards_);}
    //monarch_end
    // Added by Jonathan Lopez
       inline double& miPt() { return (miPt_); }
       inline double& miPr() { return (miPr_); }
       inline double& rX() { return (rX_); }
       inline double& rY() { return (rY_); }
       inline double& rZ() { return (rZ_); }
       inline double& tX() { return (tX_); }
       inline double& tY() { return (tY_); }
       inline double& tZ() { return (tZ_); }
    // End
```

6.4. MODIFICACIONES AL ARCHIVO packet.cc

El archivo packet.cc se encuentra escrito en C al igual que su similar packet.h. En este archivo se carga todo lo relacionado a los paquetes que se manipulan en el simulador en conjunto con el archivo packet.h. Para poder obtener la ubicación de un nodo o su potencia de recepción o transmisión en determinado momento es necesario agregar diversas líneas a la clase CommonHeaderClass, lo cual se puede observar en las siguientes líneas de código. Este archivo se puede encontrar en el directorio *~ns-allinone-2.28/ns-2.28/common/packet.cc*:

```
class CommonHeaderClass : public PacketHeaderClass {
public:
     CommonHeaderClass() :
PacketHeaderClass("PacketHeader/Common",
                               sizeof(hdr_cmn)) {
          bind_offset(&hdr_cmn::offset_);
     }
     void export_offsets() {
          field_offset("ptype_", OFFSET(hdr_cmn, ptype_));
          field_offset("size_", OFFSET(hdr_cmn, size_));
          field_offset("uid_", OFFSET(hdr_cmn, uid_));
          field_offset("error_", OFFSET(hdr_cmn, error_));
```

```
        // Added by Jonathan Lopez

        field_offset("miPr_", OFFSET(hdr_cmn, miPr_));
        field_offset("miPt_", OFFSET(hdr_cmn, miPt_));
        field_offset("rX_", OFFSET(hdr_cmn, rX_));
        field_offset("rY_", OFFSET(hdr_cmn, rY_));
        field_offset("rZ_", OFFSET(hdr_cmn, rZ_));
        field_offset("tX_", OFFSET(hdr_cmn, tX_));
        field_offset("tY_", OFFSET(hdr_cmn, tY_));
        field_offset("tZ_", OFFSET(hdr_cmn, tZ_));

        // End
    };
} class_cmnhdr;
```

6.5. MODIFICACIONES AL ARCHIVO mac-802_11.h

Para el propósito de esta tesis los archivos más importantes del simulador son los de mac-802_11. En los dos archivos que hay, .h y .cc, se puede encontrar toda la información que requiere el simulador para operar con redes inalámbricas, en particular con redes 802.11, sin embargo, estos archivos originalmente fueron hechos para la primer versión del estándar, la cual solo contemplaba velocidades de transmisión de 1 y 2 Mbps, por lo que fue necesario realizar una serie de cambios en el archivo mac-802_11.h para poder soportar diversas tasas de transmisión correspondiendo con los demás parámetros del estándar 802.11g. Este archivo se encuentra en el directorio *~ns-allinone-2.28/ns-2.28/mac/mac-802_11.h* y las modificaciones hechas se muestran a continuación, primero se definen los parámetros que requerimos para posteriormente llamarlos.

```
// This is 802.11g by Jonathan Lopez
#define DSSS_CWMin 15
#define DSSS_CWMax 1023
#define DSSS_SlotTime 0.000009              // 9µs
#define DSSS_CCATime 0.000004               // 4µs
#define DSSS_RxTxTurnaroundTime 0.000002    // 2µs
#define DSSS_SIFSTime 0.000010              // 10µs
#define DSSS_PreambleLength 74 // 74 bits => Preamble of 120 bits
#define DSSS_PLCPHeaderLength 26            // 26 bits
```

```
#define DSSS_PLCPDataRate 6.0e6             // 6Mbps

// Added by Sushmita to support event tracing
#include "address.h"
#include "ip.h"
```

Una vez que se definieron los parámetros, se llaman en la clase PHY_MIB como se puede ver a continuación:

```
class PHY_MIB {
public:
     PHY_MIB(Mac802_11 *parent);

//Modification to obtain parameters of 802.11g by Jonathan Lopez
     inline u_int32_t getCWMin() { return(DSSS_CWMin); }
     inline u_int32_t getCWMax() { return(DSSS_CWMax); }
     inline double getSlotTime() { return(DSSS_SlotTime); }
     inline double getSIFS() { return(DSSS_SIFSTime); }
     inline double getPIFS() { return(DSSS_SIFSTime +
                                DSSS_SlotTime); }
     inline double getDIFS() { return(DSSS_SIFSTime + 2 *
                                DSSS_SlotTime); }
     inline double getEIFS() {
          // see (802.11-1999, 9.2.10)
          return(DSSS_SIFSTime + getDIFS()
                 + (8 *  getACKlen())/DSSS_PLCPDataRate);  }
     inline u_int32_t getPreambleLength() {
          return(DSSS_PreambleLength); }
     inline double getPLCPDataRate() { return(DSSS_PLCPDataRate); }
     inline u_int32_t getPLCPhdrLen() {
          return((DSSS_PreambleLength + DSSS_PLCPHeaderLength)
                   >> 3); }
//End of modification
     inline u_int32_t getHdrLen11() {
          return(getPLCPhdrLen() + sizeof(struct hdr_mac802_11)
                 + ETHER_FCS_LEN); }
```

6.6. MODIFICACIONES AL ARCHIVO mac-802_11.cc

El archivo mac-802_11.cc es donde se llevan a cabo todas las selecciones de rutas para los paquetes que maneja el simulador, ya sean entrantes o salientes en la capa MAC. Es por esto que siempre que se manipula un paquete se lee este archivo, por lo que es

necesario modificar este archivo para poder adaptar la tasa de transmisión de datos de acuerdo al estándar 802.11g para cada paquete que se manipule. La primera modificación es añadir una serie de librerías que ayudarán a obtener la posición de los nodos, esta modificación es:

```
// Added by Jonathan Lopez to support distance vector
#include "trace.h"
#include <dsr/hdr_sr.h>
#include "address.h"
#include "stdlib.h"
```

Los cambios más importantes se realizan en la clase Mac802_11 de la cual se muestra el principio y mis modificaciones.

```
Mac802_11::Mac802_11() :
     Mac(), phymib_(this), macmib_(this), mhIF_(this),
     mhNav_(this), mhRecv_(this), mhSend_(this), mhDefer_(this),
     mhBackoff_(this)
{
     nav_ = 0.0;
     tx_state_ = rx_state_ = MAC_IDLE;
     tx_active_ = 0;
     eotPacket_ = NULL;
     pktRTS_ = 0;
     pktCTRL_ = 0;
     cw_ = phymib_.getCWMin();
     ssrc_ = slrc_ = 0;
     // Added by Sushmita
        et_ = new EventTrace();
     sta_seqno_ = 1;
     cache_ = 0;
     cache_node_count_ = 0;
     // chk if basic/data rates are set
     // otherwise use bandwidth_ as default;


     // Added to obtain distance between two nodes, Jonathan Lopez

     Packet *p = Packet::alloc();
     struct hdr_cmn *ch = HDR_CMN(p);
     struct hdr_mac802_11 *dh = HDR_MAC802_11(p);
     struct hdr_arp *ah = HDR_ARP(p);
     double tX_,tY_,tZ_,rX_,rY_,rZ_;
```

```
     nsaddr_t txid=index_;
     nsaddr_t rxid;

     MobileNode *tx_node =  (MobileNode*)
                            (Node::get_node_by_address(txid));
     tx_node->getLoc(&tX_,&tY_,&tZ_);
     ch->tX()=tX_;   ch->tY()=tY_;  ch->tZ()=tZ_;
     if (strcmp(packet_info.name(ch->ptype()), "ARP") == 0)
          rxid=ah->arp_tpa;
     else
          rxid=ETHER_ADDR(dh->dh_ra);

     MobileNode *rx_node =   (MobileNode*)
                             (Node::get_node_by_address(rxid));
     rx_node->getLoc(&rX_,&rY_,&rZ_);

     dist = sqrt((rX_ - tX_) * (rX_ - tX_) + (rY_ - tY_) * (rY_ -
                 tY_)  + (rZ_ - tZ_) * (rZ_ - tZ_));

     // End of modification

     Tcl& tcl = Tcl::instance();
     tcl.evalf("Mac/802_11 set basicRate_");
          if (strcmp(tcl.result(), "0") != 0)
               bind_bw("basicRate_", &basicRate_);
          else
               basicRate_ = bandwidth_;

     tcl.evalf("Mac/802_11 set dataRate_");
          if (strcmp(tcl.result(), "0") != 0)
               bind_bw("dataRate_", &dataRate_);

// Added to obtain dataRate VS distance by Jonathan Lopez

          else if (dist<=100 & dist>77)
               dataRate_ = 1*1e6;
          else if (dist<=77 & dist>65)
               dataRate_ = 2*1e6;
          else if (dist<=65 & dist>57)
               dataRate_ = 6*1e6;
          else if (dist<=57 & dist>54)
               dataRate_ = 9*1e6;
          else if (dist<=54 & dist>50)
               dataRate_ = 12*1e6;
          else if (dist<=50 & dist>42)
               dataRate_ = 18*1e6;
          else if (dist<=42 & dist>35)
```

```
            dataRate_ = 24*1e6;
        else if (dist<=35 & dist>23)
            dataRate_ = 36*1e6;
        else if (dist<=23 & dist>19)
            dataRate_ = 48*1e6;
        else
            dataRate_ = bandwidth_;
    // End of modification
```

Una vez que esta modificación está lista, el simulador está completo para poder simular redes 802.11g. De aquí en adelante el trabajo abordará la parte de simulaciones con TCP, de tal modo que se harán una serie de simulaciones en las que se pueda observar el funcionamiento de este protocolo en diversas condiciones de tráfico, de congestión, etc.

CAPÍTULO VII
EXPERIMENTOS, RESULTADOS Y RECOMENDACIONES

Para poder analizar a fondo lo que sucede con TCP cuando se aplica a redes inalámbricas 802.11G se realizaron experimentos con tres escenarios distintos, usando tanto a TCP como a UDP como protocolos de transporte, haciendo simulaciones en el NS – 2.

7.1. ESCENARIO 1

El primer escenario consta de un nodo móvil (NM) que se mueve en línea recta hacia un Punto de Acceso (AP) fijo. El nodo móvil se mueve a una velocidad de 1 m/s y se encuentra a 100 m de distancia del Punto de Acceso. La comunicación solo se realiza entre estos dos equipos. Esta situación está representada en la figura 7.1 para UDP y en la figura 7.2 para TCP.

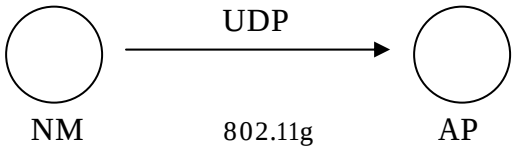

Figura 7.1. Escenario Inalámbrico UDP.

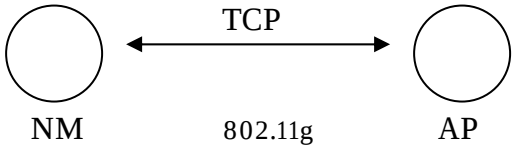

Figura 7.2. Escenario Inalámbrico TCP.

Las figuras anteriores representan solo el escenario de transmisión y recepción de la información, la simulación contempla a un Nodo Móvil que se mueve desde el instante cero a 1 m/s en línea recta hacia el Punto de Acceso; después de avanzar 99 metros se detiene a 1 metro del AP, se queda quieto durante 15 segundos y después regresa por donde llegó a la misma velocidad para detenerse a 99 metros de distancia del AP durante

otros 15 segundos y empezar de nuevo a acercarse hacia el AP a la misma velocidad anterior, para terminar la simulación a los 250 segundos. Los resultados que se obtuvieron de esta simulación, usando una liga inalámbrica 802.11G son los que se muestran en la figura 7.3 y 7.4 para UDP y TCP respectivamente. Lo que se pueden observar es el rendimiento real de la red, en las condiciones antes mencionadas.

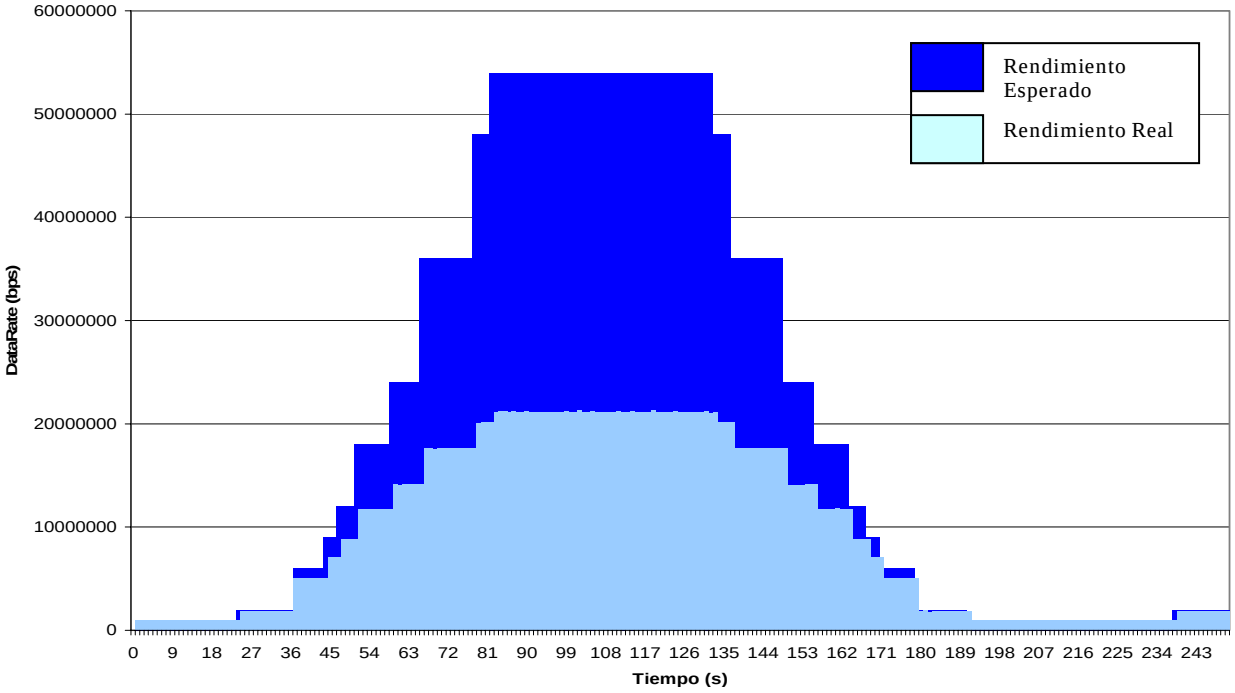

Figura 7.3. Rendimiento Real UDP WLAN.

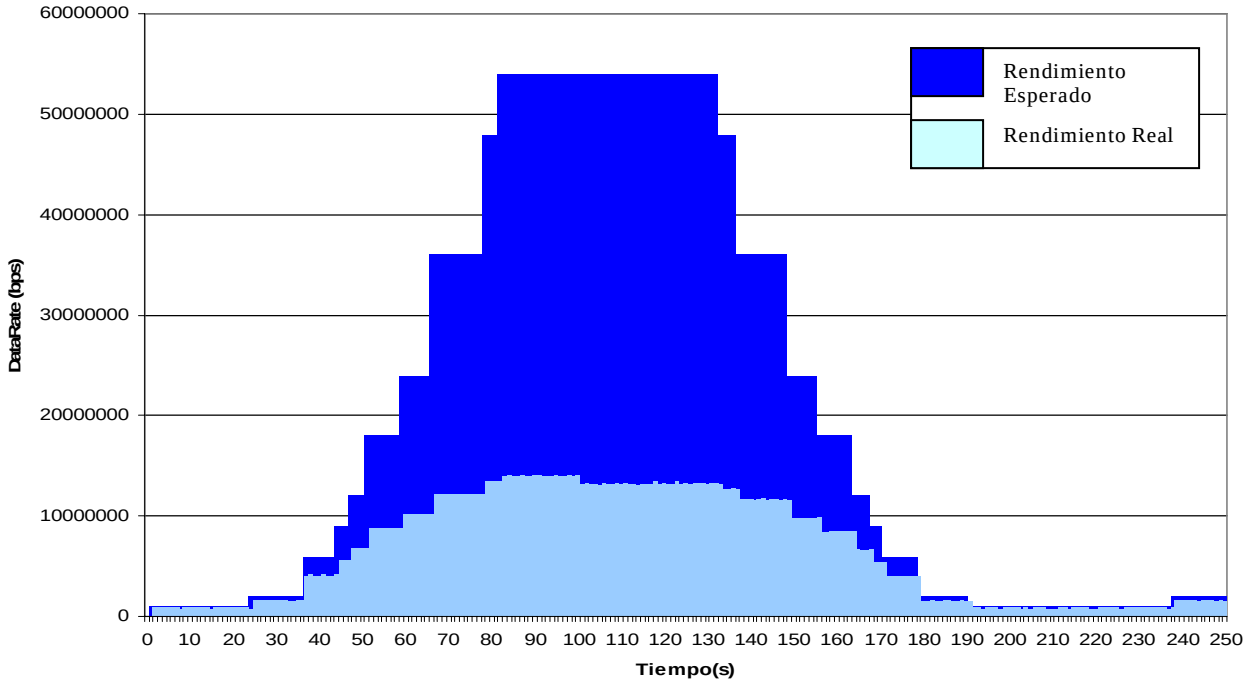

Figura 7.4. Rendimiento Real TCP WLAN.

De las gráficas anteriores se puede observar que mientras UDP maneja un rendimiento máximo promedio de 21 Mbps, TCP en cambio a velocidades de transmisión bajas se comporta bastante bien, sin embargo cuando la velocidad comienza a subir, la eficiencia de TCP se ve disminuida drásticamente.

Esto se debe a todas las pérdidas relacionadas con las redes inalámbricas, una de ellas es el hecho de que aún cuando se usa UDP, el mecanismo de 802.11G tiene que responder con ACKs en la capa MAC cuando un paquete fue transmitido satisfactoriamente, lo cual decrementa considerablemente el rendimiento de la red al agregar una gran cantidad de overhead. En cuanto a TCP, aunado a los ACKs del 802.11G, existen todos los mecanismos que TCP utiliza para garantizar la entrega satisfactoria de los datos, lo cual se traduce en una inestabilidad considerable en la red, debido al movimiento natural de los nodos móviles y el cambio de zonas de transmisión.

7.2. ESCENARIO 2

El segundo escenario consta de un nodo móvil que se mueve en línea recta hacia un AP fijo el cual tiene una conexión mediante Fast Ethernet 100 Mbps con un retardo de 10 ms hacia un nodo Fijo denominado N1. A su vez N1 está conectado con otro nodo fijo N0 mediante un enlace idéntico al de N1 con el AP. El nodo móvil se mueve a una velocidad de 1 m/s y se encuentra a 100 m de distancia del AP. La comunicación es realizada desde NM con destino a N0. Esta situación está representada en la figura 7.5 para UDP y en la figura 7.6 para TCP.

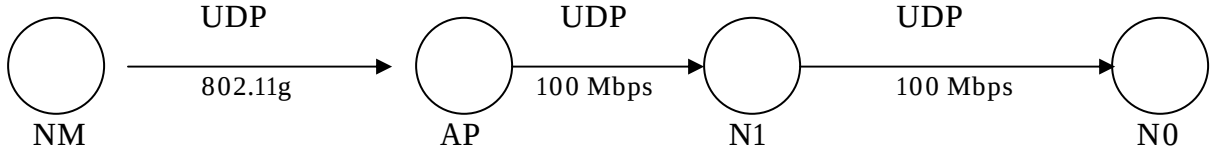

Figura 7.5. Escenario Inalámbrico/Alámbrico UDP.

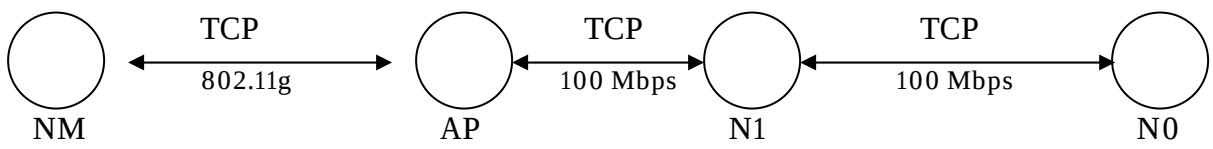

Figura 7.6. Escenario Inalámbrico/Alámbrico TCP.

En las figuras anteriores representan a un Nodo Móvil (NM) con el mismo patrón de movimiento que en el caso de las figura 7.1 y 7.2, para terminar la simulación a los 250 segundos. Los resultados que se obtuvieron de esta simulación, usando una liga inalámbrica 802.11G son los que se muestran en la figura 7.7 y 7.8 para UDP y TCP respectivamente. Lo que se puede observar es el rendimiento real de la red, en las condiciones previamente establecidas.

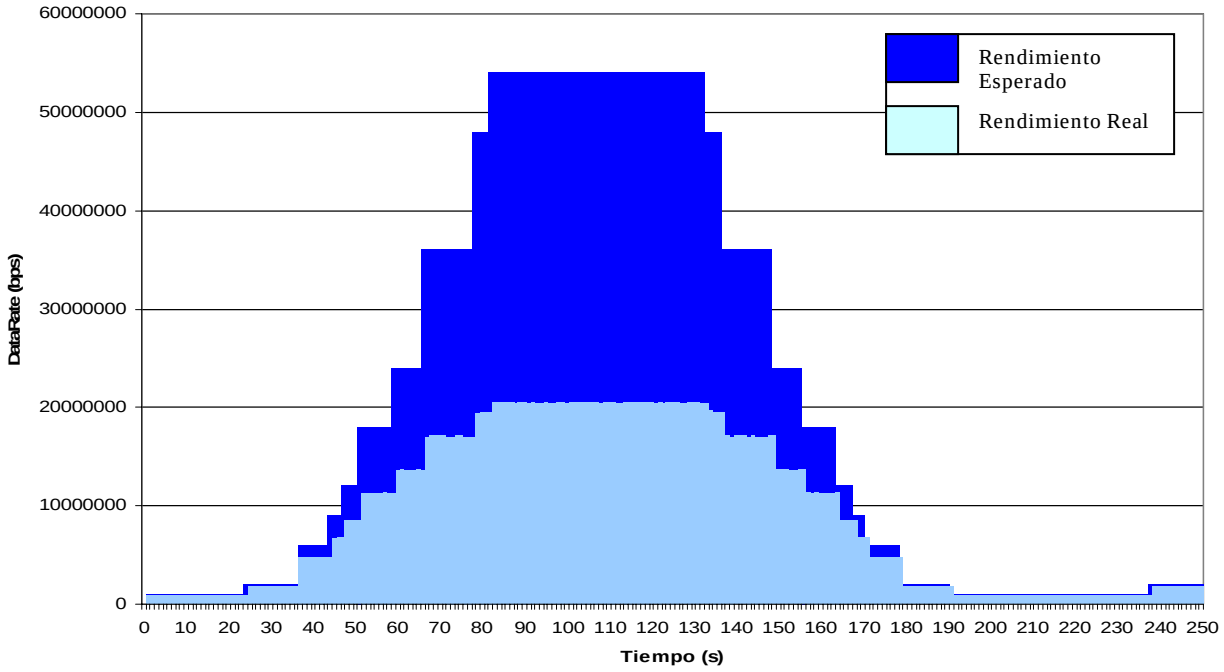

Figura 7.7. Rendimiento Real UDP WDCWL.

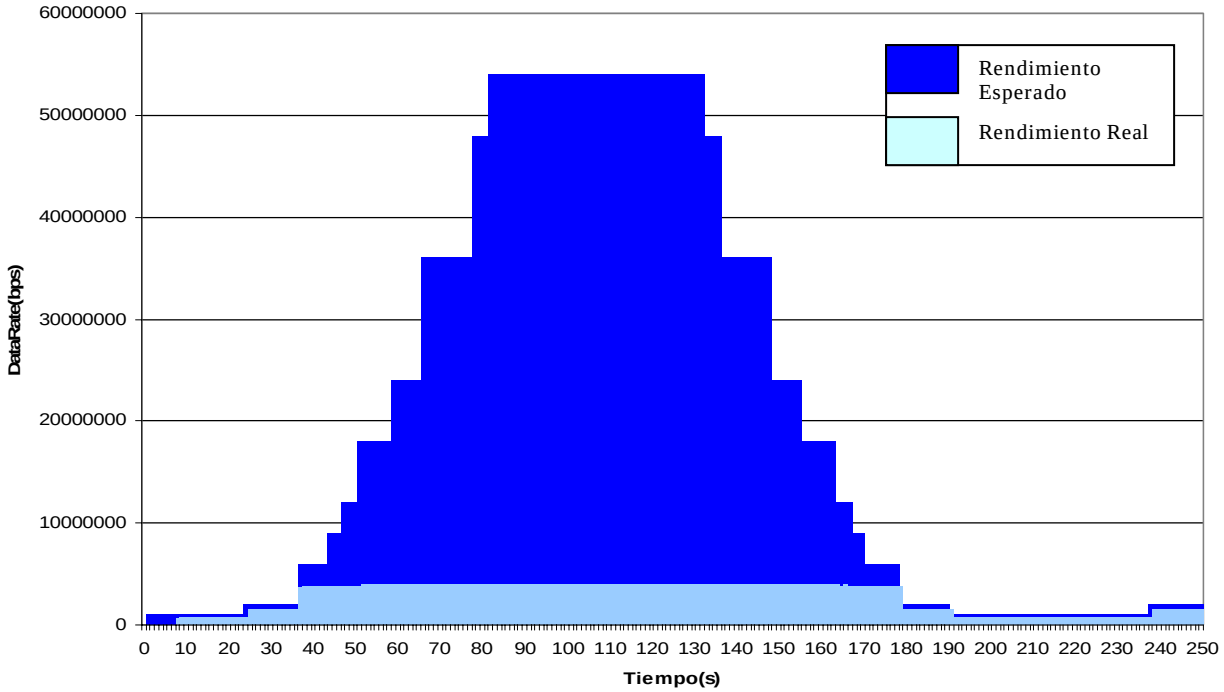

Figura 7.8. Rendimiento Real TCP WDCWL.

De las figuras 7.7 y 7.8 se puede deducir que la situación para UDP se mantiene un poco estable comparada con la situación del Escenario 1, de tal modo que en este nuevo escenario UDP alcanza un rendimiento promedio de 20.5 Mbps, mientras que TCP en el Escenario 1 había alcanzado un rendimiento promedio de 14 Mbps, cuando en el Escenario 2 tan sólo alcanza 3.9 Mbps.

Esta situación muestra que para UDP ser un protocolo no orientado a conexión le ayuda a mantener su rendimiento aún cuando el escenario cambia y lo único que le afecta es, como ya lo vimos, la propia red 802.11G, mientras que TCP se ve bastante afectado con este nuevo escenario debido al hecho de tener que hacer un control de flujo para una red híbrida, cuando fue hecho para manejarse bien en redes cableadas.

Al mismo tiempo, los agregados de los protocolos de ruteo hacen que el rendimiento sea mucho menor, además de que los tiempos de retraso también son mayores, lo que ocasiona que TCP tenga que jugar más con sus temporizadores y por si fuera poco también influyen los buffers de los equipos intermedios los cuales en determinado momento se pueden saturar o fallar y ocasionar una disminución del uso de la red con TCP.

7.3. ESCENARIO 3

El tercer escenario es idéntico al Escenario 2, solo cambia en el enlace entre N1 y N0 el cual es un enlace Fast Ethernet de 100 Mbps con un retraso de 2 s. La comunicación es realizada desde NM con destino a N0. Esta situación está representada en la figura 7.9 para UDP y en la figura 7.10 para TCP.

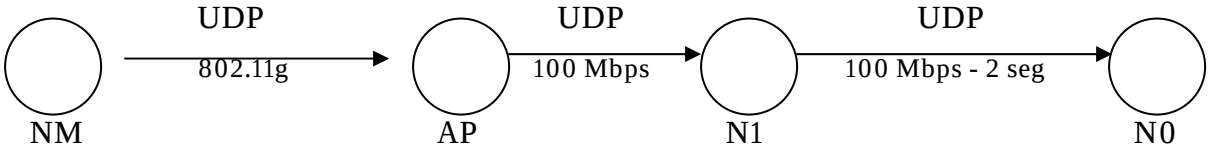

Figura 7.9. Escenario Inalámbrico/Alámbrico 2 UDP.

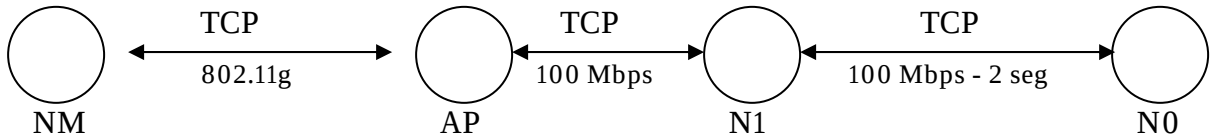

Fig. 7.10. Escenario Inalámbrico/Alámbrico 2 TCP

El patrón de movimiento de NM es idéntico a los casos anteriores, para terminar la simulación a los 250 segundos. Los resultados que se obtuvieron de esta simulación, usando una liga inalámbrica 802.11G son los que se muestran en la figura 7.11 y 7.12 para UDP y TCP respectivamente. Lo que podemos observar es el rendimiento real de la red, en las condiciones previamente establecidas.

Este par de análisis son muy importantes ya que ayudan a verificar que en el caso de UDP, aunque muchos paquetes se pierden, el rendimiento es muy similar a los escenarios anteriores, ya que UDP alcanza un rendimiento máximo de 20.5 Mbps, mientras que se puede observar que TCP se cae brutalmente a tal grado que no envía más que sólo 163 Kbps máximo en promedio cada 4 segundos.

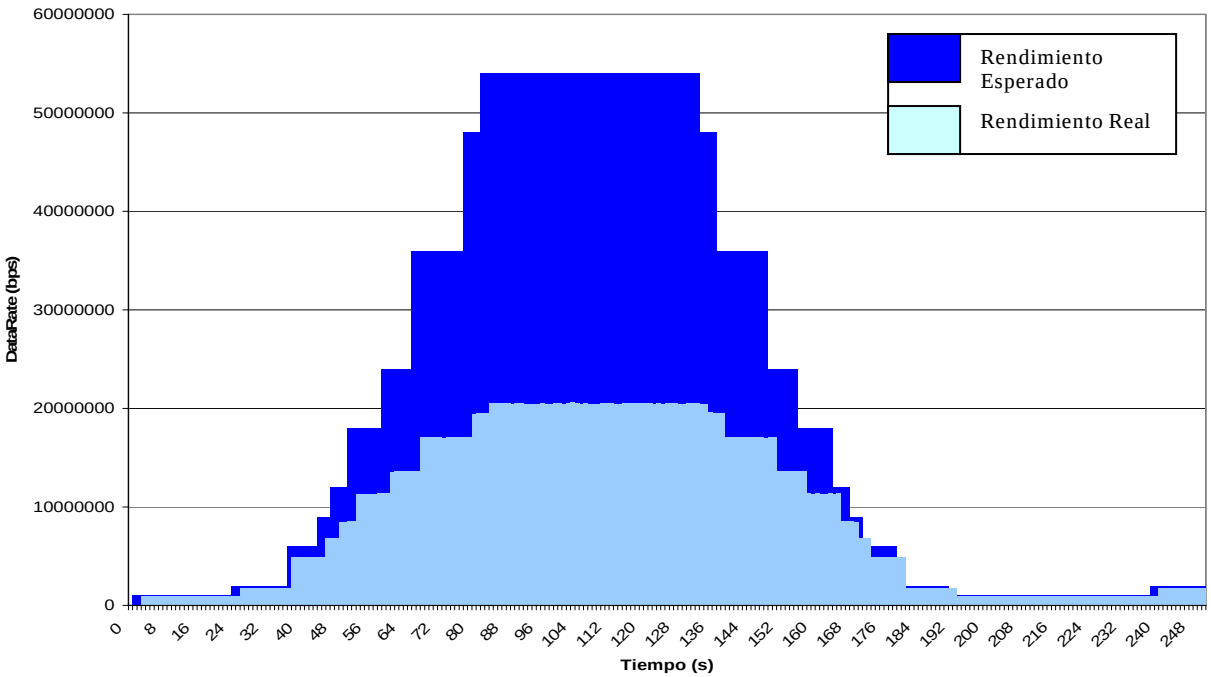

Figura 7.11. Rendimiento Real UDP WDCWL con retraso de 2 seg.

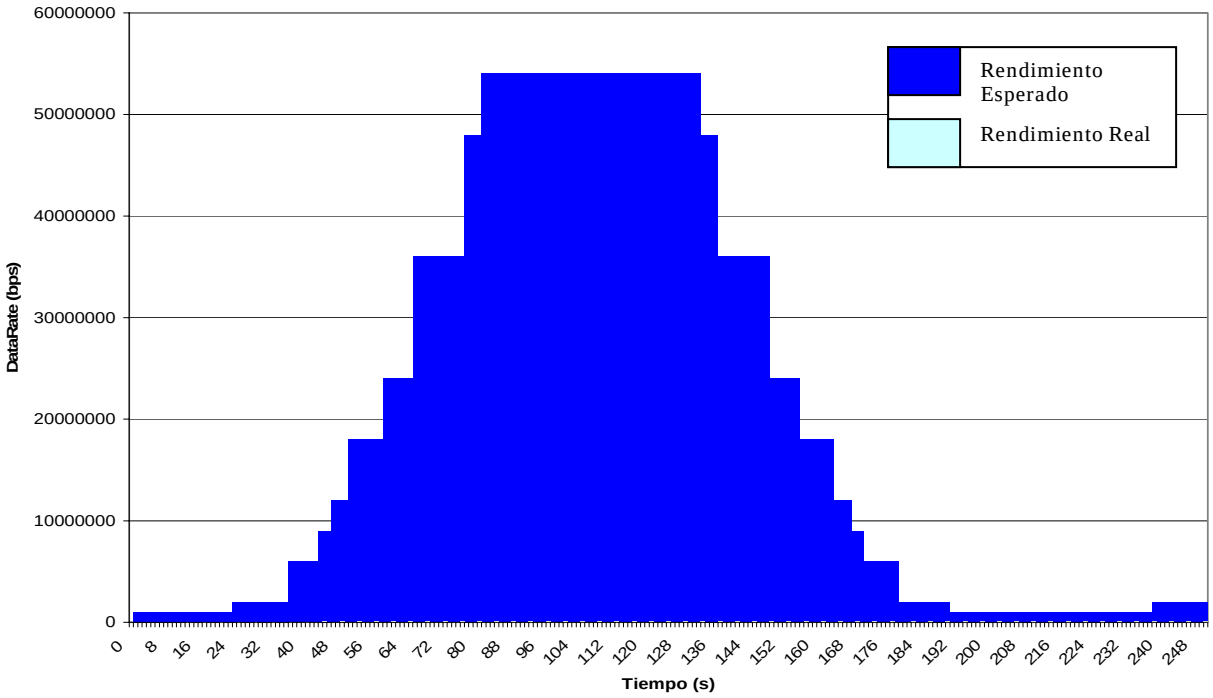

Figura 7.12. Rendimiento Real TCP WDCWL con retraso de 2 seg.

Aunque en la práctica es muy difícil encontrar retardos de más de 1 segundo en la transmisión común, se decidió poner un enlace de 2 segundos para poder observar lo que sucedía con TCP en condiciones extremas, teniendo como origen un nodo móvil. Lo que se pudo observar es que el rendimiento es casi nulo ya que el tiempo de transmisión de ida y vuelta es de más de 4 segundos, lo que ocasiona que todos los temporizadores que usa TCP para la estimación de la transmisión se vuelvan inestables. Aunado al manejo de las ventanas, TCP solo enviará la información posible cada poco más de 4 segundos para poder garantizar la entrega de la información, junto a esto están los casos anteriores que agudizan el bajo rendimiento como son los protocolos de ruteo, las propias técnicas de 802.11G, entre otros.

7.4. RECOMENDACIONES

Como se ha podido ver en el desarrollo de esta Tesis, existen muchos factores que contribuyen al bajo rendimiento de TCP en redes inalámbricas, esto es desde la misma forma en que fueron concebidas las redes inalámbricas hasta el modo en que TCP usa sus diferentes mecanismos para garantizar la entrega de la información.

7.4.1. Mejoras conjuntas de redes inalámbricas y TCP

Para poder mejorar el rendimiento en las redes inalámbricas 802.11G basadas en TCP, se pueden hacer recomendaciones de cambios en muchas áreas del modelo tanto OSI como TCP/IP de redes. Sin embargo por lo analizado se pueden reducir dichas recomendaciones a las más fundamentales que son:

- Ya que el hecho de dar un ACK por cada paquete de datos enviado en una red 802.11G en la capa MAC es el mayor causante del bajo rendimiento de estas redes, se puede crear un método alternativo similar al control basado en ventanas que usa TCP; lo que implicaría que la capa MAC del emisor mantenga un buffer con los paquetes enviados para, en caso de requerirse, estos sean reenviados. Por otra parte el receptor debe llevar otro mecanismo para poder en un solo ACK responder cuales paquetes le han llegado correctamente y al mismo tiempo el emisor pueda enviar determinado número de paquetes antes de requerir una confirmación.

- Por otro lado la movilidad de los nodos en redes inalámbricas conduce a que TCP tenga que estar trabajando con sus mecanismos de control arduamente, ya que en redes 802.11G existen diferentes zonas con diferentes anchos de banda disponibles para la transmisión de información y al momento en que un nodo cambia de una zona a otra hace que TCP tenga que adaptarse bruscamente a esta nueva zona, sin embargo en escenarios con retardos muy grandes puede tomarle mucho tiempo a TCP darse cuenta del cambio, de tal modo que esto se puede solucionar simplemente con pequeños paquetes que no causen mucha sobrecarga, los cuales lleven información de cuando exista un cambio de zona y los datos de dicha zona.

- Para poder manejar bien los retardos en redes híbridas se puede optar por una solución basada en algo similar a los Proxy, equipos que puedan servir de intermediarios entre los nodos móviles de la red inalámbrica y los nodos destinos en redes cableadas, para lo cual serían necesarios nuevos mecanismos para enviar la información hacia esos equipos en vez del destino real, para lo cual esos equipos

deberían poder responder como si estos fueran los destinos, mientras este proceso debe ser de forma transparente para el receptor y el transmisor.

7.4.2. Mejoras ya propuestas a TCP

"Para poder alcanzar la eficiencia y la solidez esperada, TCP debe mejorar en cinco áreas principales:

- *Temporizador y retraso de la retransmisión*.- TCP utiliza un esquema de *acuse de recibo acumulativo* en el que cada acuse lleva un número de secuencia. El número de secuencia especifica cuántos octetos contiguos del flujo de datos ha recibido correctamente del emisor. Puesto que los acuses de recibo no especifican segmentos individuales y puesto que pueden perderse, el emisor no puede distinguir si un acuse dado surgió a partir de una transmisión original o de la retransmisión de un segmento. Por lo tanto, el emisor no puede medir con precisión la demora del viaje de ida y vuelta de los segmentos retransmitidos. De este modo cuando hay enlaces con una alta demora, el uso de los distintos temporizadores es muy ineficaz, aunque hay alternativas que el mismo RFC de TCP especifica.

  El RFC especifica que TCP debe usar una técnica conocida como *algoritmo de Karn* para controlar el valor del temporizador de retransmisión. Durante la transferencia normal de datos, antes de que expire el temporizador de retransmisión llegan acuses de recibo para cada segmento. En estos casos, el algoritmo de Karn no interfiere con el proceso usual que mide la demora del viaje de ida y vuelta y que calcula el tiempo de espera de la retransmisión para el siguiente segmento a enviar. Sin embargo, debido a que TCP no puede asociar correctamente los acuses de recibo con las transmisiones individuales de un segmento, el algoritmo de Karn especifica que TCP debe ignorar las mediciones del viaje de ida y vuelta para todos los segmentos retransmitidos.

  Además, una vez que comienza la retransmisión, el algoritmo de Karn separa el tiempo de espera de retransmisión de la demora del viaje de ida y vuelta, y duplica el

tiempo de espera de cada retransmisión. Sin embargo aún con este mecanismo le cuesta mucho trabajo a TCP poder definir el tiempo que toma una transmisión para así adecuar sus temporizadores, ya que al haber enlaces con mucho retardo los paquetes constantemente se estarán retransmitiendo y al ir duplicando el tiempo de espera de retransmisión habrá un gran periodo de inactividad lo cual causa un bajo rendimiento en la red.

- *Control de flujo basado en ventanas*.- Cuando el TCP de la máquina receptora envía un acuse de recibo, incluye en el segmento una *notificación de ventana* para indicar al emisor cuánto espacio de búfer tiene disponible el receptor para datos adicionales. La notificación de ventana especifica siempre los datos que el receptor puede aceptar además de los datos que está recibiendo como acuse de recibo; y TCP establece que una vez que un receptor notifica una ventana dada, nunca podría notificar un subconjunto de esa ventana. Por supuesto, cuando llena la ventana notificada, el valor en el campo de acuse de recibo aumenta y el valor en el campo de la ventana podría reducirse hasta llegar a cero. Sin embargo, el receptor tal vez no podrá disminuir el punto en el espacio de secuencia a través del cual acordó aceptar datos. Por lo tanto, la notificación de ventana sólo puede disminuir si el emisor suministra datos o si el número de acuses de recibo se incrementa; no puede disminuir simplemente porque el receptor decida reducir el tamaño de su búfer.

  TCP emplea las notificaciones de ventana para controlar el flujo de datos a través de una conexión. Un receptor notifica tamaños de ventana pequeños para limitar los datos que puede generar un emisor. En el caso extremo, notificar un tamaño de ventana de cero detiene por completo la transmisión. Si un receptor notifica un espacio de búfer tan pronto como esté disponible, podría ocasionar un comportamiento conocido como el *síndrome de la ventana tonta*. El comportamiento de ventana tonta se caracteriza por una situación en la que la ventana del receptor oscila entre cero y un valor positivo pequeño, mientras que el emisor transmite pequeños segmentos para llenar la ventana tan pronto como ésta abre. Este comportamiento conduce a una baja utilización de la red, ya que cada segmento transmitido contiene pocos datos en comparación con la carga de los

encabezados TCP e IP. Para evitar que un punto TCP sea víctima del síndrome de la ventana tonta al momento de hacer la transmisión, TCP utiliza una técnica conocida como *impedimento de la ventana tonta del lado del receptor*. La regla de esta técnica establece que una vez que un receptor notifica una ventana de cero, debe demorar la notificación de una ventana diferente de cero hasta que tenga una cantidad no trivial de espacio en su búfer. Una cantidad no trivial de espacio en búfer se define como el espacio suficiente para un segmento de tamaño máximo, o como el espacio equivalente a una cuarta parte del búfer, lo que sea mayor.

Una vez que un receptor notifica una ventana de cero, el emisor entra en el estado de salida *PERSIST* y comienza a sondear al receptor. El receptor responde a cada sondeo enviando un acuse de recibo. En tanto la ventana permanezca cerrada, los sondeos continuarán y los acuses de recibo contendrán una notificación de ventana de cero. Tarde o temprano, cuando haya espacio suficiente disponible, los acuses de recibo llevarán una ventana diferente de cero y el emisor comenzará a transmitir nuevos datos.

Aunque el emisor tiene la responsabilidad final de sondear una ventana de cero, una pequeña optimización puede mejorar el rendimiento. La optimización consiste en hacer que el receptor genere *acuses de recibo gratuitos* que contengan el nuevo tamaño de ventana, sin esperar el siguiente sondeo. Cuando el emisor procesa el acuse de recibo, encuentra una notificación de ventana diferente de cero, regresa al estado *TRANSMIT* y continúa la transmisión de datos. Este mecanismo es bueno aunque podría haber una revisión exhaustiva y hacer que mejore mucho la eficiencia de TCP

- *Cálculo del tamaño máximo de segmento*.- Cuando TCP genera segmentos que llevan datos, limita su tamaño al *tamaño máximo de segmento* (MSS) permitido para esa conexión. Cuando intercambia solicitudes durante el acuerdo de conexión de tres vías, TCP negocia el MSS tanto para los segmentos entrantes como para los salientes. Una vez que establece un MSS en cada dirección, TCP nunca los cambia.

Para ayudar a evitar la fragmentación de IP, el documento de requerimientos del host especifica que TCP debe usar el tamaño máximo de segmento inicial de 536 octetos cuando la conexión pasa a través de una puerta de enlace. Para conexiones que residen en una red conectada de forma directa, TCP elige un valor inicial tal que los paquetes de red estarán tan llenos como sea posible (es decir, calcula un tamaño máximo de datos inicial restando el tamaño de los encabezados TCP e IP de la MTU de la red local empleada para alcanzar la máquina remota). Después de seleccionar un MSS inicial, TCP procesa la opción del tamaño máximo de segmento que se encuentra en los segmentos SYN entrantes. Un tamaño máximo de segmento sólo puede ser negociado durante el acuerdo de conexión de tres vías.

- *Impedimento y control de congestionamiento.-* Cuando ocurre un congestionamiento aumenta la demora, lo que ocasiona que TCP retransmita segmentos. En el peor de los casos, las retransmisiones incrementan el congestionamiento y producen un efecto conocido como *colapso de congestionamiento*. Para impedir que el congestionamiento aumente, el estándar especifica ahora que TCP debe emplear estrategias para reducir la transmisión cuando ocurre una demora o una pérdida de paquetes. A la primera estrategia se le conoce como *disminución multiplicativa*. La idea en que se basa la disminución multiplicativa es simple: el lado del emisor de TCP mantiene una variable interna conocida como *ventana de congestionamiento,* la cual utiliza para limitar la cantidad de datos que serán enviados. Al transmitir, TCP utiliza el mínimo de la ventana notificada del receptor y la ventana interna de congestionamiento para determinar cuantos datos debe enviar.

  Para calcular el tamaño de la ventana de congestionamiento, suponga que el número de retransmisiones proporciona una medida del congestionamiento de la intrarred. Mientras no ocurra un congestionamiento o una pérdida, asigne al tamaño de la ventana de congestionamiento el tamaño de la ventana notificada del receptor. Es decir, use la ventana notificada del receptor para determinar cuantos datos enviar. Cuando comience el congestionamiento (es decir, cuando ocurra una retransmisión), reduzca el tamaño de la ventana de congestionamiento en una constante

multiplicativa. En particular, reduzca la ventana de congestionamiento a la mitad cada vez que ocurra una retransmisión, aunque nunca la reduzca a menos del tamaño requerido para un segmento.

Aunque la técnica se denomina *multiplicativa*, el umbral de la ventana de congestionamiento disminuirá en forma exponencial al ser medida en segmentos perdidos. La primera pérdida reducirá la ventana a la mitad de su tamaño original, la segunda a una cuarta parte, la tercera a un octavo y así sucesivamente. Aquí podemos observar que para conexiones con altos retardos, donde habrá muchas retransmisiones al principio, en lo que TCP adapta sus temporizadores la transmisión habrá decaído drásticamente y quizá nunca se recupere.

- *Estimación del viaje de ida y vuelta y del tiempo de espera*.- Desde un principio, se reconoció que el rendimiento de TCP dependía de su capacidad para estimar la medida del tiempo de viaje de ida y vuelta en una conexión. TCP emplea el historial de medidas para estimar la demora del viaje de ida y vuelta, y elije un tiempo de espera para la retransmisión a partir de esta estimación. Debido a que la demora del viaje de ida y vuelta varia con el tiempo, TCP da mayor peso a las medidas recientes que a las anteriores. Sin embargo, puesto que las medidas individuales de la demora del viaje de ida y vuelta difieren ampliamente del estándar cuando ocurre un congestionamiento, TCP no puede ignorar por completo el historial de mediciones.

  Estudios sobre el rendimiento han mostrado que TCP puede mostrar una velocidad real de transporte mayor si calcula la varianza, ya que existen buenos algoritmos de incremento. Por lo tanto, TCP mantiene un promedio corriente que se actualiza cada vez que obtiene una nueva medición. ”[67] Así mismo, en nuestros resultados podemos observar que TCP al medir el tiempo de ida y vuelta de un paquete este tiempo puede haber variado al momento de que un ACK se recibe en el transmisor por la misma naturaleza de movilidad de los ambientes estudiados.

[67] Cfr. Comer, Douglas E. Op. Cit. P. 287 – 303.

CONCLUSIONES

Para poder realizar este trabajo haciendo uso del simulador Ns – 2 se requirió instalar la distribución de Linux Mandrake versión 10.2, debido a que es una distribución de fácil comprensión e instalación; posteriormente la instalación del simulador NS2 se hizo a través del paquete allinone pero durante esta se encontraron muchos errores y era un tanto complicado corregirlos.

Originalmente Mandrake era un reempaquetador de Red Hat Linux cuyo objetivo eran las optimizaciones de KDE (K Desktop Environment). De este modo se decidió mudar la distribución de Linux a una ya un poco más revisada tipo Red Hat, para este trabajo la distribución más adecuada y actual era Fedora Core 3 que estaba disponible en esos momentos.

Una vez que se descarga el paquete todo en uno del NS de la página principal del propio simulador, el cual viene comprimido, primeramente se descomprime y luego se desempaqueta. Posteriormente se corre el script de instalación y teóricamente todo debería quedar listo, sin embargo en la distribución de Linux (Fedora Core 3) hubo un pequeño problema.

El problema que se encontró al instalar el NS en Linux fue que el NAM se compilaba, configuraba e instalaba correctamente, pero no aparecía el archivo ejecutable de NAM; una vez revisado el problema se pudo observar que había una discrepancia en el compilador de C++ que la distribución de Linux usada tenía, por lo que se tuvo que revisar el código de uno de los archivos del NAM para ver que estaba mal; se encontró que el compilador no permitía la asignación *NULL* a variables, la cual aparecía una vez en dicho código, de este modo se cambió esa asignación por un cero (0), y así se volvió a correr la compilación, configuración e instalación del NAM y todo resultó satisfactorio.

Una vez realizado lo anterior es necesario importar algunas rutas a la LD Library Path, y a la TCL Library, estos pasos son indicados al final de la instalación del NS si se realizó con el script de instalación. Por último es necesario agregar la ruta /.../ns-allinone-2.28/bin en el archivo .bash_profile con el fin de poder ejecutar el NS y el NAM desde cualquier ruta del sistema.

Conforme se realizaba esta investigación, fue liberada la distribución de Fedora Core 4 y como se tenían algunos conflictos en la computadora de prueba se creyó oportuno migrar el sistema a dicha distribución, de modo que se instaló y se procedió a construir de nuevo el simulador en esta distribución. Lo que se encontró fue que al contrario de Fedora Core 3, en esta nueva distribución había muchos más problemas para poder instalar satisfactoriamente el simulador, sin embargo en la página web del NS se proveía de una solución que era modificar una serie de archivos los cuales creaban algunos conflictos al momento de compilarse con la versión del compilador gcc con que cuenta Fedora Core 4. Buscando en Internet se pudo encontrar un parche que modificaba automáticamente todos los archivos que no reconocía correctamente el gcc de Fedora 4, se instaló y posteriormente se construyó el simulador y todo resultó satisfactorio.

Una vez que se tuvo el simulador corriendo eficientemente en Fedora Core 4, se realizaron diversas simulaciones para poder verificar que todo funcionara correctamente. Posteriormente se dio un enfoque a realizar los cambios que necesitaba el NS. Sin embargo en ocasiones era necesario realizar otras actividades independientes a esta investigación, pero Linux no estaba provisto de las herramientas necesarias, o por lo menos no las tenía instaladas, para poder realizar dichas actividades, de modo que en la computadora de prueba, la cual es una computadora portátil Compaq Presario Modelo 2130LA, se dispuso a instalar tanto Windows como Linux en la misma PC.

Una vez que los dos sistemas operativos corrían adecuadamente, se instaló el Cygwin para poder realizar las actividades necesarias de Windows y a la vez realizar las investigaciones con NS2.

Después de que se empezaron a realizar las modificaciones que se vieron en este trabajo, se observó que muchas de estas variantes no funcionaban correctamente en Linux, a pesar que la recompilación no marcaba errores, el simulador no hacia lo que se requería, sin embargo en Cygwin funcionaban de manera satisfactoria. Es por esto que se decidió seguir trabajando con Cygwin y dejar por un lado a Linux por ese momento, ya que la investigación es sobre TCP y redes inalámbricas y no sobre el simulador en si. Posteriormente al aplicar un par de modificaciones más, estas funcionaron satisfactoriamente en Linux, pero no en Cygwin así que al final se terminó trabajando al 100% con Linux.

Desafortunadamente, el simulador NS – 2 está hecho por diferentes personas, de tal modo que muchas partes del código son un tanto confusas, lo que contribuyó a que este trabajo se estancara por momentos para poder adaptarlo a los propósitos establecidos. Si el simulador pudiera manipular correctamente las WLAN sin necesidad de modificarlo, entonces tuviéramos una mejor idea de cómo afecta el movimiento a TCP al cambiar de zonas de transmisión.

El proyecto de adaptación de TCP a WLAN 802.11g es algo ambicioso, ya que TCP es un protocolo muy bien establecido en todo el mundo, la mayoría de las redes lo usan como protocolo de transporte, y consta de muchas subáreas que lo complementan a fondo, de tal modo que modificarlo o implantar un nuevo protocolo de transporte no es un trabajo sencillo. Sin embargo, se espera poder proveer a la comunidad de redes de computadoras de un análisis exhaustivo para que cualquiera que desee instalar, administrar, etc. una red inalámbrica sepa cuales son las cualidades de TCP que se pueden explotar para que el rendimiento de esa red sea el máximo y del mismo modo se espera que esta comunidad conozca los retos que conlleva la instalación, el mantenimiento o la administración de redes inalámbricas basadas en TCP como protocolo de transporte.

En conclusión a este trabajo se puede decir que hasta el momento se analizó a fondo el simulador NS – 2 y se vio que es muy útil para simular redes cableadas. Sin embargo, en redes inalámbricas es muy ineficaz, aún cuando existen diversos códigos de

mejora para este fin. A esto hay que añadir la complicación de unir las redes inalámbricas con las cableadas, lo cual es muy común en la vida real. Es importante decir que no todo es malo en este simulador, que si bien o mal, es gratuito, existe una gran comunidad que lo respalda, de tal modo que tiene una gran cantidad de módulos para simular diversos ambientes y diversas redes con todo el código que esta comunidad ha contribuido.

Finalmente se observó que en un medio inalámbrico la movilidad afecta el rendimiento de TCP, por lo que es necesario diseñar un algoritmo que defina la velocidad de movimiento del nodo móvil y pueda predecir hacia donde se lleva a cabo dicho movimiento, para de este modo poder adaptar la velocidad de transmisión según la zona a la que el nodo esté arribando, siempre y cuando TCP también lleve un control del buffer del nodo destino y del BER, ya que si nuestro algoritmo detecta que el nodo se está moviendo con dirección a una zona de mayor velocidad a la actual, pero el nodo destino tiene saturado su buffer, no podrá recibir la cantidad de paquetes que le enviáramos. Esta y otras adaptaciones de las que se hablaron en el último capítulo servirían de mucho para alcanzar el objetivo de mejoramiento del rendimiento de TCP en redes inalámbricas.

GLOSARIO

A

Ad-Hoc.- En redes de comunicación, dicha expresión hace referencia a una red (especialmente inalámbrica) en la que no hay un nodo central, sino que todos los ordenadores están en igualdad de condiciones. También se suele conocer como IBSS (Independent Basic Service Set).

B

Banda de Frecuencia.- Es un grupo de frecuencias del espectro radioeléctrico que generalmente se encuentra debidamente regulados en cada uno de los países del mundo.

BER (Bit Error Rate).- Porcentaje de bits recibidos con errores.

Bluetooth.- Tecnología de radio desarrollada por Ericsson y otras compañías. Construida alrededor de un chip que hace posible transmitir señales en distancias cortas, sin el uso de cables, entre computadoras y otros dispositivos.

BSS (Basic Service Set).- Consiste en dos o más nodos inalámbricos a veces conocidos como estaciones. Un BSS tiene dispositivos que se reconocen y trabajan en conjunto unos con otros para minimizar la cantidad de colisiones que existen dentro del dominio del BSS. Cada BSS es identificado por un SSID.

C

Cabecera.- Información de control colocada antes de los datos cuando se encapsulan dichos datos para la transmisión por la red.

Campo de Inducción.- El campo que está más cerca de la antena de RF.

CRC (Cyclic Redundance Check).- Técnica de verificación de errores en la que el receptor de la trama calcula un resto dividiendo el contenido de una trama por un divisor binario primo y compara el resto calculado con un valor almacenado en la trama por el nodo emisor.

CSMA (Carrier Sense Multiple Access).- Mecanismo de acceso al medio dentro del cual los dispositivos que están listos para transmitir datos, primero verifican el canal en busca de una portadora. El dispositivo puede transmitir si no detecta ninguna portadora durante un periodo de tiempo determinado.

Cygwin.- Es una colección de herramientas desarrollada por Cygnus Solutions para proporcionar un comportamiento similar a los sistemas Unix en Windows. Su objetivo es portar software que ejecuta en sistemas POSIX a Windows con una recompilación a partir de sus fuentes. Se distribuye habitualmente bajo los términos de la GPL con la excepción de que permite ser enlazada con cualquier tipo de software libre cuya licencia esté de acuerdo con la definición de software libre.

D

DB (Decibel).- Diez veces la relación logarítmica entre dos magnitudes.

DIFS (DFC Interframe Space).- DIFS es el espacio entre tramas más largo y es usado por defecto en todas las estaciones que cumplen con 802.11 que están usando la función de coordinación distribuida. Cada estación en la red usando el modo DCF está requerida a esperar hasta que el DIFS haya expirado antes de que cualquier estación pueda contender en la red.

DSSS (Direct Sequence Spread Spectrum).- es uno de los métodos de modulación en espectro extendido para transmisión de señales digitales sobre ondas radiofónicas que más se utilizan. DSSS es una técnica de modulación que utiliza un código de pseudorruido para modular directamente una portadora, de tal forma que aumente el ancho de banda de la transmisión y reduzca la densidad de potencia espectral (es decir, el nivel de potencia en cualquier frecuencia dada). La señal resultante tiene un espectro muy parecido al del ruido, de tal forma que a todos los radiorreceptores les parecerá ruido menos al que va dirigida la señal.

E

Efecto Doppler.- Consiste en la variación de la longitud de onda de cualquier tipo de onda emitida o recibida por un objeto en movimiento.

Eficiencia espectral.- Es una medida de lo bien aprovechada que está una determinada banda de frecuencia usada para transmitir datos (bits). Cuando mayor es este valor, mejor aprovechada está dicha banda. La eficiencia espectral es uno de los muchos parámetros con los que se mide la calidad de una modulación digital. Otros factores a tener en cuenta son la velocidad de transmisión, el BER y la energía por bit (Eb / N).

Espectro Radioeléctrico.- Es el espacio que permite la propagación sin guía artificial de ondas electromagnéticas cuyas bandas de frecuencias se fijan convencionalmente por debajo de los 3000 gigahertz. El espectro radioeléctrico es un recurso natural, de carácter limitado, que constituye un bien de dominio público, sobre el cual el Estado ejerce su soberanía.

ESS (Extended Service Set).- Es la referencia apropiada para los clientes y AP en una red de infraestructura, debido a que este término incluye dispositivos que provienen de más de un BSS y normalmente está conectado mediante Ethernet a través de un sistema de distribución, como una LAN, a lo largo de toda una empresa.

F

FDM (Frequency Division Multiplexing).- Técnica por la cual a la información procedente de varios canales se le puede asignar ancho de banda en un único medio de transmisión, basándose en la frecuencia.

FHSS (Frequency Hopping Spread Spectrum).- Es una técnica de modulación en espectro extendido en la que la señal se emite sobre una serie de radiofrecuencias aparentemente aleatorias, saltando de frecuencia en frecuencia síncronamente con el transmisor. Los receptores no autorizados escucharán una señal ininteligible. Si se intentara interceptar la señal, sólo se conseguiría para unos pocos bits.

H

HomeFR.- La idea de este estándar se basa en el Teléfono inalámbrico digital mejorado (Digital Enhaced Cordless Telephone, DECT) que es un equivalente al estándar de los teléfonos celulares GSM. Transporta voz y datos por separado, al contrario que protocolos como el WiFi que transporta la voz como una forma de datos. Los creadores de este estándar pretendían diseñar un aparato central en cada casa que conectara los teléfonos y además proporcionar un ancho de banda de datos entre las computadoras.

I

IEEE (Institute of Electrical and Electronics Engineers).- Es una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en informática e ingenieros en telecomunicaciones.

IEEE 802.11.- Es un estándar de protocolos de comunicaciones del IEEE que define el uso de los dos niveles inferiores de la arquitectura OSI (capas física y de enlace de datos), especificando sus normas de funcionamiento en una WLAN. En general, los protocolos de la rama 802.x definen la tecnología de redes de área local.

ISM (Industrial, Scientific and Medical).- Son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones WLANo WPAN. El uso de estas bandas de frecuencia está abierto a todo el mundo sin necesidad de licencia, respetando las regulaciones que limitan los niveles de potencia transmitida. Este hecho fuerza a que este tipo de comunicaciones tengan cierta tolerancia frente a errores y que utilicen mecanismos de protección contra interferencias, como técnicas de espectro extendido.

L

LAN (Local Area Network).- Una red de área local, o red local, es la interconexión de varias computadoras y periféricos. Su extensión esta limitada físicamente a un edificio o a un entorno de pocos kilómetros. Su aplicación más extendida es la interconexión de computadoras personales y estaciones de trabajo en oficinas, fábricas, etc., para compartir recursos e intercambiar datos y aplicaciones.

Lenguaje de programación C.- C es un lenguaje de programación creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Linux.- Es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye como software libre. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds. Conocido mayormente por el uso en servidores y super-computadores, cuenta con el soporte de corporaciones como Dell, Hewlett-Packard, IBM, Novell, Oracle, Red Hat y Sun Microsystems.

LMDS (Local Multipoint Distribution Service).- Es una tecnología de conexión vía radio inalámbrica que permite, gracias a su ancho de banda, el despliegue de servicios fijos de voz, acceso a internet, comunicaciones de datos en redes privadas, y video bajo demanda.

Longitud de Onda.- La longitud de una onda describe cuán larga es la onda. La distancia existente entre dos crestas o valles consecutivos es también a lo que llamamos longitud de onda.

M

MMDS (Multichannel Multipoint Distribution Service).- Identifica a una tecnología inalámbrica de telecomunicaciones, usada para el establecimiento de una red de banda ancha de uso general o, más comúnmente, como método alternativo de recepción de programación de televisión por cable.

Mobilenode.- Es el objeto básico Nodo en el NS con funcionalidades agregadas como movimiento, habilidad de transmitir y recibir en un canal que permite ser usado para crear ambientes de simulación inalámbricos móviles.

Modelo de Referencia OSI.- Modelo de arquitectura de red desarrollado por la ISO y la ITU-T. El modelo de referencia OSI está compuesto por siete capas, cada una de las cuales especifica funciones de red individuales, tales como el direccionamiento, el control de flujo, el control de errores, la encapsulación y la transferencia fiable de mensajes.

Modulación.- Es el conjunto de técnicas para transportar información sobre una onda portadora, típicamente una onda sinusoidal. Estas técnicas permiten un mejor aprovechamiento del canal de comunicación lo que posibilita transmitir más información en forma simultánea, protegiéndola de posibles interferencias y ruidos. Es un proceso por el cual se transforman las características de las señales eléctricas para representar información.

MSS (Maximum Segment Size).- Es el tamaño más grande de datos, especificado en bytes, que un dispositivo de comunicaciones puede manejar en un único trozo, sin fragmentar. Para una comunicación óptima la suma del número de bytes del segmento de datos y la cabecera debe ser menor que el número de bytes del MTU de la red.

MTU (Maximum Transfer Unit).- La unidad máxima de transferencia (Maximum Transfer Unit - MTU) es un término de redes de computadoras que expresa el tamaño en bytes del datagrama más grande que puede pasar por una capa de un protocolo de comunicaciones.

N

NAM (Network Animator).- Es una herramienta de animación basada en Tcl/TK para observar las trazas de una simulación de red. Soporta diseño de topologías, animación a nivel de paquetes y varias herramientas de inspección de datos.

NAV (Network Allocation Value).- Representa la longitud de un paquete en el que un nodo desea enviar información. Todos los nodos presentan el NAV con el que quieren transmitir información y éste indica la cantidad de tiempo de transmisión que la trama anterior en la cola necesita para terminar. Cuando todos los nodos dentro de un dominio de colisión reciben el NAV, lo usan como base para establecer sus tiempos de transmisión.

Network Simulator – 2.- Es un simulador de redes de eventos discretos. Utilizado principalmente en ambientes académicos debido a que está escrito en código abierto y a la abundancia de documentación en línea. Se pueden simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes moviles ad-hoc. Puede simular una amplia gama de protocolos tanto para redes cableadas o redes wireless, así como mixtas.

NIC (Network Interface Card).- Es una placa de circuito impreso que proporciona las capacidades de comunicación de red hacia y desde una computadora personal. También se denomina adaptador de LAN; se enchufa en la tarjeta madre y proporciona un puerto de conexión a la red. Esta tarjeta se puede diseñar como una tarjeta Ethernet, Token Ring, FDDI, etc.

Nodo.- La palabra nodo se utiliza de forma genérica para hacer referencia a cualquier entidad que tenga acceso a una red y frecuentemente se utilizan indistintamente nodo y dispositivo.

O

OFDM (Orthogonal Frequency Division Multiplexing).- Es una modulación que consiste en enviar la información modulando en QAM o en PSK un conjunto de portadoras de diferentes frecuencias. Normalmente se realiza la modulación OFDM tras pasar la señal por un codificador de canal con el objetivo de corregir los errores producidos en la transmisión, entonces esta modulación se denomina COFDM, del inglés Coded OFDM.

P

PAU (Portable Access Unit).- Mediante este dispositivo se obtiene acceso a una computadora servidor conectado a una LAN por cable. Por lo regular, el campo de cobertura de la PAU es de 50 a 100 m, y en una instalación grande hay muchas de estas unidades distribuidas dentro de un sitio.

PDU (Protocol Data Unit).- Término OSI que equivale a paquete. Es una agrupación lógica de información que incluye una cabecera que contiene la información de control y los datos del usuario. Los paquetes se usan a menudo para referirse a las unidades de datos de capa de red. Los términos datagrama, trama, mensaje y segmento también se usan para describir agrupamientos lógicos de información en las diversas capas del modelo de referencia OSI.

Pérdida de propagación.- O pérdida en el espacio libre es la relación de la pérdida de potencia entre dos sitios.

PIFS (PCF Interframe Space).- Un espacio entre tramas PIFS no es ni el más corto ni el más largo espacio entre tramas fijo, así que tiene más prioridad que el DIFS pero menos que el SIFS. Los puntos de acceso usan los espacios PIFS solamente cuando la red está trabajando en modo de función de coordinación de punto, el cual es manualmente configurado por el administrador.

PLCP (Physical Layer Control Protocol).- Subcapa de la capa física que se encarga de aspectos como la codificación Barker y CCK, además de las técnicas de modulación como QPSK y la técnica de propagación DSSS

PMD (Physical Medium Dependence).- Subcapa de la capa física que realiza la interfaz directa con el medio físico y lleva a cabo las funciones más básicas de transmisión de bits de la red. También crea la interfaz hacia la capa MAC para la sensibilidad de la portadora a través de su C*omprobación de canal libre*

Potencia de Transmisión Inalámbrica.- Es el proceso que se da en un sistema donde la energía eléctrica es transmitida de una fuente de poder a una carga eléctrica, sin interconectar cables.

Protocolo.- Descripción formal de los formatos que deberán presentar los mensajes para poder ser intercambiados entre diferentes equipos de cómputo, también define las reglas y convenciones que deben seguir dichos equipos para poder lograr una comunicación.

Puerto.- Es una dirección interna predefinida que sirve como un camino de las aplicaciones a la capa de transporte o de la capa de transporte a las aplicaciones.

R

RF (Radiofrecuencia).- Término genérico para referirse a las frecuencias que corresponden a las transmisiones de radio.

Roaming.- Es un concepto utilizado en comunicaciones inalámbricas que está relacionado con la capacidad de un dispositivo para moverse de una zona de cobertura a otra. Roaming es una palabra del idioma inglés que significa vagar o rondar. El término más adecuado en castellano es *itinerancia*. Cuando es utilizado en las redes Wi-Fi, significa que el dispositivo Wi-Fi cliente puede desplazarse e ir registrándose en diferentes bases o puntos de acceso.

RTS/CTS (Request to Send/Clear to Send).- Es un mecanismo usado en el estándar 802.11 para reducir las colisiones de tramas introducidas por el problema de la terminal oculta.

S

SIFS (Short Interframe Space).- Es el espacio entre tramas más corto. Los SIFS son espacios de tiempo antes y después de los datos.

Socket.- Designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada. Un socket queda definido por una dirección IP, un protocolo y un número de puerto.

Subcapa de Control de Enlace Lógico.- La más alta de las dos subcapas de la capa de enlace de datos definidas por el IEEE. La subcapa LLC maneja el control de errores, control de flujo, entramado y direccionamiento de la subcapa MAC. El protocolo LLC más generalizado es IEEE 802.2, que incluye variantes no orientadas a conexión y orientadas a conexión.

Subcapa de Control de Acceso al Medio.- La inferior de las dos subcapas de la capa de enlace de datos definidas por el IEE. La subcapa MAC administra el acceso a medios compartidos, como, por ejemplo, si se utilizará transmisión de testigos o contención.

T

TCL (Tool Command Languaje).- es un lenguaje de script creado por John Ousterhout, que ha sido concebido para su fácil aprendizaje, pero que resulta muy potente en las manos adecuadas. Se usa principalmente para el desarrollo rápido de prototipos, aplicaciones "script", interfaces gráficas y pruebas.

TCP (Transmission Control Protocol).- Protocolo de la capa de transporte orientado a conexión que proporciona una transmisión dúplex fiable de datos. TCP es parte de la pila de protocolos TCP/IP y está descrito en el RFC 793.

TCP/IP (Transmission Control Protocol / Internet Protocol).- Nombre común para la suite de protocolos desarrollados por el DoD de Estados Unidos en los años 70 para facilitar la construcción de redes a nivel mundial. Debe su nombre a los protocolos TCP e IP los cuales son los más conocidos y usados de la suite.

TDM (Time Division Multiplexing).- Técnica mediante la cual se puede asignar ancho de banda a la información procedente de múltiples canales en un solo medio de transmisión, en base a espacios de tiempo asignados previamente. El ancho de banda se asigna a cada canal sin tomar en cuenta si la estación tiene datos para transmitir.

Técnica de Propagación.- Es un mecanismo que distribuye la información a través de una variedad de canales

U

UDP (User Datagram Protocol).- Protocolo sin conexión de capa de transporte de la pila de protocolos TCP/IP. UDP es un protocolo simple que intercambia datagramas sin confirmación o garantía de entrega y que requiere que el procesamiento de errores y las retransmisiones sean manejadas por otros protocolos. UDP se define en el RFC 768.

UNIX.- Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

V

Ventana de Transmisión.- Cantidad de octetos que el remitente desea aceptar. Se refiere al número de mensajes que pueden ser transmitidos mientras se espera una confirmación.

W

WEP (Wired Equivalent Privacy).- Es el sistema de cifrado incluido en el estándar IEEE 802.11 como protocolo para redes inalámbricas que permite cifrar la información que se transmite. Proporciona un cifrado a nivel capa 2. Está basado en el algoritmo de cifrado RC4 y utiliza claves de 64 bits (40 bits más 24 bits del vector de iniciación IV) o de 128 bits (104 bits más 24 bits del IV).

WLAN (Wireless Local Area Network).- Es un sistema de comunicación de datos inalámbrico flexible, muy utilizado como alternativa a las redes LAN cableadas o como extensión de éstas. Utiliza tecnología de radiofrecuencia que permite mayor movilidad a los usuarios al minimizar las conexiones cableadas.

WLL (Wireless Local Loop).- Es el uso de un enlace de comunicaciones inalámbricas como la conexión de "última milla" para ofrecer servicios de telefonía (POTS) e Internet de banda ancha a los usuarios.

BIBLIOGRAFÍA Y REFERENCIAS

BIBLIOGRAFÍA

- Bandel, David. *Edición Especial Linux*. Ed. Prentice Hall. Edic. 6ª. 983 P. Madrid, España, 2001.
- Casad, Joe. *Sams Teach Yourself TCP/IP in 24 Hours*. Ed. Sams Publishing. Edic. 3ª. 480 P. U.S.A., 2003.
- *Certified Wireless Network Administrator*. Official Study Guide. Ed. Mc-GrawHill. Edic. 2ª. 578 P. California, U.S.A., 2003.
- Comer, Douglas E. *Interconectividad de Redes con TCP/IP. Vol II*. Ed. Pearson Educación. Edic. 3ª. 660 P. México, 2000.
- Coulouris, G. *Sistemas Distribuidos. Conceptos y diseño*. Ed. Pearson Educación. Edic. 3ª. 744 P. Madrid, España, 2001.
- Crow, Brian. *IEEE 802.11 Wireless Local Area Networks*. Ed. IEEE Communications Magazine. 126 P. Septiembre 1997.
- Deitel, Harvey. *Cómo programar en C++*. Ed. Pearson Educación. Edic. 4ª. 1320 P. México, 2003.
- *Diccionario de Términos de Computación*. Ed. Grupo Editorial Tomo, S. A. de C. V. 216 P. México D.F., 2000.
- ElAarag, Hala. *Performance Evaluation of TCP implementation in Wireless Networks*. School of Computer Science, University of Central Florida. 78 P. 1999.
- Halsall, Fred. *Comunicación de Datos, Redes de Computadores y Sistemas Abiertos*. Ed. Pearson Educación. Edic. 4ª. 955 P. México, 1998.
- *IEEE Std. 802.11G TM* – 2003. Ed. IEEE Computer Society. 67 P.
- Leiden, Candance. *TCP/IP para Dummies*. Ed. ST Editorial, Inc. Edic. 4ª. 291 P. Panamá, 2001.
- Méndez, Luis. Tesis: *Diseño, implementación y evaluación de un protocolo MAC con alto reuso espacial para redes inalámbricas con infraestructura y Ad – Hoc*. Fac. Ingeniería, UNAM, 2005. 150 P.
- Navarro, Anna. *Diccionario de Términos de Comunicaciones y Redes*. Ed. Pearson Educación, S. A. 616 P. Madrid, España, 2003.
- Parker, Tim. *Teach Yourself TCP/IP in 14 days*. Ed. Sams Publishing. Edic. 2ª. 512 P. U.S.A.
- Raya, Jose Luis. *TCP/IP para Windows 2000 Server*. Ed. Alfaomega. 768 P. Colombia, 2001.

- Reid, Neil. *802.11 (Wi-Fi) Manual de Redes Inalámbricas*. Ed. Mc Graw Hill. 364 P. México, 2004.
- The Vint Proyect. *The NS Manual*. 420 P. 2001.

SITIOS DE INTERNET

- Chen, Xuan. *CONSER Project Overview*. http://www.isi.edu/conser/overview.html. 2001.
- *Cygwin Information and Installation*. http://www.cygwin.com/ 2008.
- Heidemann, John. *SAMAN, Simulation Augmented by Measurement and Analysis for Networks*. http://www.isi.edu/saman/index.html. 2001.
- *Historia de las redes de computadoras*. http://galeon.hispavista.com/redeslanabedulmo/historia.html.
- *IEEE 802.11*. http://es.wikipedia.org/wiki/IEEE_802.11, 2008.
- Intelligraphics, Inc. *Introduction to IEEE 802.11*. http://www.intelligraphics.com/articles/80211_article.html. 2007
- Jones, Evan. *Basic 802.11 Statistics*. http://evanjones.ca/basic-80211-stats.html, 2004.
- Kumar, Satish. *VINT Project Overview*. http://www.isi.edu/nsnam/vint/project_overview.html. 1997.
- Kumar, Satish. *VINT, Virtual InterNetwork Testbed*. http://www.isi.edu/nsnam/vint/index.html. 1997.
- Martínez, Evelio. *Estándares WLAN*. http://www.eveliux.com/articulos/estandareswlan.html. 2002.
- Naranjo, Alice. *Redes de computadoras*. http://www.monografias.com/trabajos5/redes/redes.shtml. 2008.
- *NS-2 Running NS and NAM Under Windows 9x/2000/XP Using Cygwin*. http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin. 2008.
- *NS-2 User Information*. http://nsnam.isi.edu/nsnam/index.php/User_Information. 2008.
- *NS Change History*. http://www.isi.edu/nsnam/ns/CHANGES.html.
- *Protocolos TCP/IP*. Soto, Miguel Alejandro. http://usuarios.lycos.es/janjo/janjo1.html, 1998.
- Raman, Bhaskaran. *The Enhanced Network Simulator (Release Version 1.2)*. http://www.cse.iitk.ac.in/users/braman/tens/.
- *Red de Computadoras*. http://es.wikipedia.org/wiki/Red_de_computadoras, 2008.
- Robinson, Joshua. 802.11 MAC code in NS – 2 (version 2.28). http://www.ece.rice.edu/~jpr/ns/docs/802_11.html

- Robinson, Joshua. *Making NS-2 simulate an 802.11b link*. http://www.ece.rice.edu/%7Ejpr/ns-802_11b.html.
- S. Keshav. *REAL 5.0 Overview*. http://www.cs.cornell.edu/skeshav/real/overview.html 1998.
- *The ICSI Networking Group*. http://www.icir.org/. 2008.
- *The Network Simulator - ns–2*. http://www.isi.edu/nsnam/ns/.
- *The Network Simulator: Building NS*. http://www.isi.edu/nsnam/ns/ns-build.html.
- *The Network Simulator NS-2: Validation Tests*. http://www.isi.edu/nsnam/ns/ns-tests.html.