UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

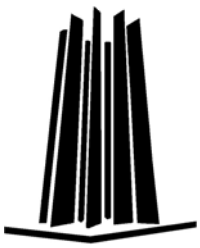
INGENIERÍA EN COMPUTACIÓN

"ADAPTACIÓN DE TCP A REDES INALÁMBRICAS 802.11G"

T R A B A J O E S C R I T O
E N L A M O D A L I D A D D E
C R É D I T O S D E M A E S T R Í A
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A

JONATHAN EMMANUEL LÓPEZ FIGUEROA



ASESOR:
ING. JOSÉ MANUEL QUINTERO CERVANTES

MÉXICO 2006

DEDICATORIAS

A mis padres:
Ana Elizabeth Figueroa Morales y Gustavo López Bustos por todo el amor que me dan, por todos los valores que me han enseñado los que me han hecho ser una buena persona, por todo el apoyo incondicional, ya que si he llegado hasta aquí es por ustedes y mucho de lo que soy lo debo a ustedes…

A mis abuelos:
Ana María Morales Carvajal y Luis Figueroa Cruz ✝ por estar conmigo en los momentos en que mis padres no podían hacerlo, por darme su cariño y consejos que siempre me ayudaron a seguir adelante…

AGRADECIMIENTOS

Al resto de mi familia y amigos:
Tíos, primos, abuelos paternos, etc. con quienes he podido compartir alegrías y tristezas y por todo el apoyo que me han dado…

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 Ing. José Manuel Quintero Cervantes:
Por la confianza puesta en esta tesis para así dirigirla y apoyarme en todo lo necesario para poder terminarla…

A mi asesor de maestría, el Dr. Javier Gómez Castellanos: Por la asesoría que me ha brindado desde que acudí a él, y así apoyarme en todo el proceso del postrado que estudio…

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:
Ing. Antonia Navarro González
Ing. Jessica Eugenia Alcalá Jara
Mat. Luis Ramírez Flores
M. en C. Marcelo Pérez Medel
Por todas las sugerencias y comentarios que me ayudaron a mejorar esta tesis…

A mi novia, Araceli Reyes Velázquez:
Por todos los momentos que ha compartido conmigo, tanto buenos como malos, por darme su amor incondicional y siempre alentarme, por comprenderme durante todo el proceso de desarrollo de este trabajo… ¡Te Amo!

ÍNDICE

INTRODUCCIÓN 1

CAPÍTULO I
ANTECEDENTES DE TCP Y REDES 4
1.1. Definición de red inalámbrica 4
1.2. Clasificación de las redes inalámbricas 6
1.3. Fundamentos de radiofrecuencia 10
1.3.1. Potencia de transmisión 10
1.3.2. Bandas de frecuencia 12
1.3.3. Modulación 13
1.3.4. Técnicas de propagación 20
1.3.4.1. FHSS y DSSS 20
1.3.4.2. OFDM 21
1.3.5. Técnicas de duplexión 22
1.4. El estándar IEEE 802.11 23
1.4.1. Pila de protocolos de 802.11 23
1.4.2. Capa física 26
1.4.2.1. 802.11g 28
1.4.3. Subcapa de Control de Acceso al Medio 30
1.4.3.1. Estructura de las tramas usadas en 802.11 32
1.4.3.2. El problema de los nodos ocultos y expuestos 36
1.4.3.3. Funciones básicas de la subcapa MAC 37
1.4.3.4. Cambios de tasa de datos 43
1.4.3.5. Función de coordinación distribuida DCF 44
1.4.3.6. Sensor de portadora 44
1.4.3.7. Tiempo de retroceso aleatorio 47
1.4.3.8. Función coordinada de punto PCF 48
1.4.3.9. Intervalos de tiempo entre tramas 49
1.4.3.10. Sistema RTS / CTS 53
1.5. El protocolo de control de transporte TCP 56
1.5.1. Introducción 56
1.5.2. Cabecera de TCP 61
1.5.3. Establecimiento de la conexión 64
1.5.4. Transferencia de datos 66
1.5.5. Control de flujo y retransmisión adaptiva 67
1.5.6. Inicio lento e impedimento del congestionamiento 73
1.5.7. Cierre de la conexión 74

CAPÍTULO II
ESTADO ACTUAL DEL PROYECTO 76
2.1. El proyecto de adaptación de TCP a redes inalámbricas 802.11G 76
2.1.1. Objetivos del proyecto 77
2.1.2. Contribución y relevancia 77
2.1.3. Metas del proyecto 78
2.1.4. Metodología 78
2.2. El simulador de redes NS – 2 79
2.2.1. Introducción 79
2.2.2. Generalidades de Linux y Cygwin 83
2.2.2.1. Linux 85
2.2.2.2. Cygwin 88
2.2.3. Interfase al intérprete 90
2.2.3.1. Conexión OTcl y C++ 91
2.2.4. Arquitectura general del NS – 2 93
2.2.5. Carencias del simulador NS – 2 96
2.2.6. Las redes inalámbricas y el NS – 2 100
2.2.6.1. Modelo de redes inalámbricas en NS – 2 100
2.2.6.2. Simulación de redes inalámbricas en NS – 2 101
2.2.6.3. Análisis de resultados de simulación 108
2.2.7. El archivo de MAC 802.11 del NS – 2 110
2.2.7.1. Transmitiendo un paquete 111
2.2.7.2. Recibiendo un paquete destinado a sí mismo 111
2.2.7.3. Funciones del MAC del NS – 2 112
2.3. Implementación del estándar IEEE 802.11G en NS – 2 119
2.3.1. Modificaciones al archivo ns-mac.tcl 119
2.3.2. Modificaciones al archivo ns-default.tcl 120
2.3.3. Modificaciones al archivo packet.h 122
2.3.4. Modificaciones al archivo packet.cc 123
2.3.5. Modificaciones al archivo mac-802_11.h 124
2.3.6. Modificaciones al archivo mac-802_11.cc 126
2.4. Últimos avances del proyecto 126

CONCLUSIONES 130

APÉNDICE 1. CONCEPTOS GENERALES 132

APÉNDICE 2. TCP/IP 175

BIBLIOGRAFÍA Y REFERENCIAS 200

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 he decidido realizar una tesis que involucrara algún aspecto relacionado a las comunicaciones. En la maestría he enfocado 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 hablara sobre comunicaciones inalámbricas.

La tesis realizada lleva por título *Adaptación de TCP a Redes Inalámbricas 802.11g*; he optado 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 sería la vida sin las redes inalámbricas, ya que muchas de las tecnologías se están enfocando hacia este tipo de comunicaciones. En el gran mundo de las redes inalámbricas, el estándar IEEE 802.11 es el más usado en todo el mundo, y en particular en nuestro país, por lo que decidí tomarlo como base para mi 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 de que también definen las reglas que deben seguir dichos equipos para poder lograr esa comunicación.

El protocolo TCP forma parte de la pila de protocolos de TCP/IP, la cual provee una plataforma de comunicación ínter operable entre todo tipo de hardware y software, así que en la actualidad la mayoría de las aplicaciones de redes de computadoras, requieren transmitir datos usando TCP como protocolo de transporte. Por todo lo mencionado anteriormente he optado por realizar una tesis donde se involucraran por una parte las redes inalámbricas y por otra TCP. Básicamente esta tesis hablará sobre el uso de TCP en redes inalámbricas y de que forma la movilidad de un equipo de cómputo, limitado por las capacidades del estándar IEEE 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 nos apoyaremos en el simulador de redes *Network Simulator NS-2*.

Esta tesis consta básicamente de dos capítulos, sin embargo no porque sean pocos significa que el contenido será muy breve, por el contrario los contenidos de cada capítulo son algo extensos, de tal modo que realicé un par de apéndices para poder cubrir muchas cosas que en este par de capítulos quedaran al aire. En el primer capítulo se podrá leer una gran diversidad de temas, incluyendo temas como *fundamentos de radiofrecuencia*, hasta temas un tanto avanzados como un análisis del *Estándar IEEE 802.11g* en el primer capítulo y temas como análisis del *Simulador de Redes NS2*, de sus siglas en ingles Network Simulator Versión 2, y simulación de ambientes inalámbricos IEEE 802.11g en el segundo capítulo.

En el primer capítulo veremos algunos conceptos básicos con los que me apoyaré para proseguir el desarrollo de la tesis, así como los conceptos primordiales que se usan en las redes inalámbricas. Veremos primeramente aspectos básicos del estándar *IEEE 802.11*.

También en el mismo primer capítulo el lector podrá observar a fondo el Protocolo de Control de Transporte, que de aquí en adelante llamaremos *TCP* por sus siglas en inglés.

En el segundo capítulo hablaré sobre diversas cosas relacionadas directamente con el proyecto que estoy realizando en la maestría, daré una amplia explicación de lo que consiste en sí el proyecto. Analizaré el simulador de redes NS-2 incluyendo su historia, evolución, aplicaciones y la forma en que he trabajado con él, manipulándolo para poder simular los escenarios necesarios para mi proyecto, ya que fue necesario hacer diversas modificaciones al código del simulador así como al código de las simulaciones a realizar. También hablaré sobre el ambiente de desarrollo *LINUX* y *Cygwin*, en los cuales me encuentro trabajando.

Por último en este mismo capítulo daré una explicación de todo el avance que he tenido en este proyecto y en que estado se encuentra actualmente, para poder dar una visión más amplia de las capacidades y limitaciones del proyecto.

Después se podrán observar las conclusiones donde se mostrarán todas nuestras memorias y resumiremos el trabajo realizado hasta la fecha, analizando si es viable la ambición del proyecto, la cual es básicamente analizar el protocolo TCP en redes inalámbricas para luego, 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, crear un nuevo protocolo de transporte inmune a la movilidad.

Por último, al final de este trabajo se hallan un par de apéndices, donde el lector podrá encontrar información más detallada en lo que a las redes y a TCP se refiere, para así aclarar cualquier duda que a este le surja.

CAPÍTULO I
ANTECEDENTES DE TCP Y REDES

Empezaremos este capítulo con una definición de los que son las redes inalámbricas, para poder apoyarnos en esto y seguir adelante. Si el lector requiere más información, como un poco de historia de estas redes y algunos otros conceptos, acuda al Apéndice 1 donde podrá encontrar todo esto.

1.1. 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, Internet, desde cualquier lugar. Por una parte es más fácil para el proveedor de servicios de telecomunicaciones e Internet brindar a sus usuarios acceso sin alambres que cablear a cada uno de ellos, además de que es más fácil la incorporación de un nuevo usuario a una red inalámbrica.

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.

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. Nosotros manejaremos 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 o algún otro equipo electrónico, nosotros llamaremos Nodo a cualquiera de los anteriores que se encuentre conectado a la red inalámbrica. Aunque en realidad "un nodo o estación es una plataforma individual, como un punto de acceso o una tarjeta de interfaz de cliente (por ejemplo, una tarjeta PCMCIA o mini-PCMCIA)."[1]

La señal transmitida por un nodo solo puede ser percibida dentro de cierta área de transmisión, la cual llamaremos rango del nodo.

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

1.2. CLASIFICACIÓN DE LAS REDES INALÁMBRICAS

Las redes inalámbricas se pueden clasificar en tres categorías: WAN/MAN, LAN y PAN.

En la primer categoría WAN/MAN (redes de área amplia/redes de área metropolitana) pondremos a las redes que cubren desde decenas de metros hasta miles de kilómetros. En la segunda categoría LAN (red de área local), pondremos las redes que comprenden varios metros hasta decenas de metros. Y en la última y más nueva categoría PAN (redes de área personal) pondremos a las redes que comprenden desde centímetros hasta 30 metros.

En la categoría MAN/WAN tenemos primeramente el acceso por telefonía celular. Aunque originalmente la telefonía celular fue utilizada para la transferencia de voz, muy pronto se desarrollaron protocolos para poder transferir datos a través de esta tecnología inalámbrica. Otras tecnologías WAN/MAN inalámbricas que permiten el acceso a redes de datos o Internet son MMDS, LMDS, WLL, enlaces de microondas terrestres, enlaces vía láser infrarrojo y comunicaciones vía satélite.

En la segunda categoría las redes locales inalámbricas se han vuelto muy populares hoy en día, éstas pueden proveer de acceso a redes cableadas e Internet. En esta categoría tenemos diferentes estándares que existen para poder transmitir de diferentes maneras y a diferentes velocidades, referirse al Apéndice 1 para mayor información

Por otra parte, las redes tipo PAN son una tecnología que cubre distancias cortas y cerradas. Algunas de estas tecnologías son Bluetooth, 802.15 y HomeRF.

Dentro de las categorías mencionadas anteriormente nosotros nos enfocaremos 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."[2]

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."[3] Generalmente las redes inalámbricas centralizadas son una extensión de una red cableada. Este tipo de redes están ilustradas en la fig. 1.1.

La PAU que se mencionó antes, 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 un círculo 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

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

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. Estas redes pueden operar tanto por duplexaje por división de tiempo (TDD) como por duplexaje por división de frecuencia (FDD), los cuales se explicarán más adelante.

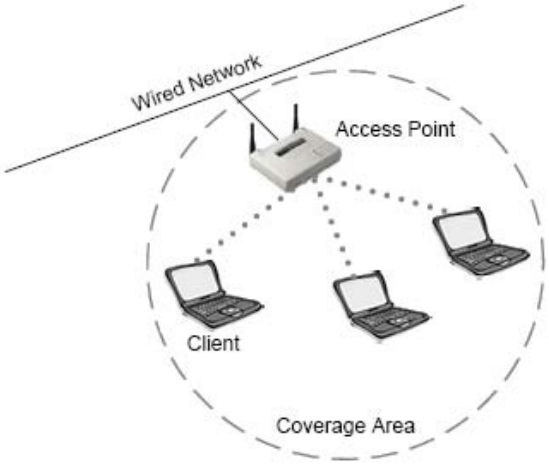


Fig. 1.1 Red inalámbrica con infraestructura

"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."[4]

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),

[4] Reid, Neil. Op. Cit. P. 69.

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."[5]

La fig. 1.2 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 la llamaremos de aquí en adelante.

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 vimos antes, una red con infraestructura puede usar ambas opciones de duplexaje, ya sea TDD o FDD, sin embargo en una red Ad-Hoc solo se puede usar TDD, 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.

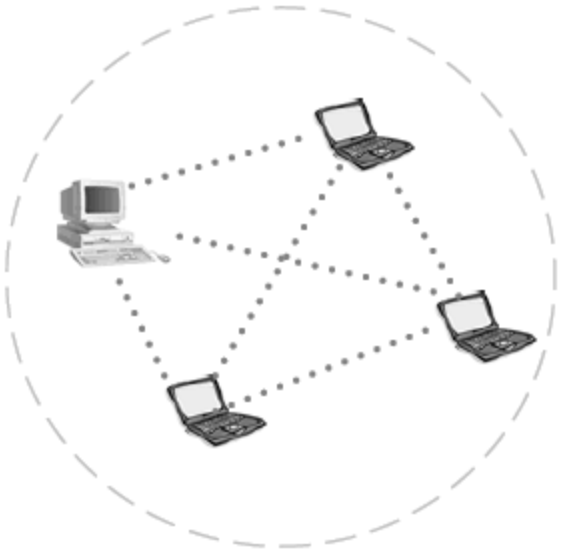


Fig. 1.2 Red inalámbrica Ad-Hoc

[5] Ibídem. P. 68.

1.3. FUNDAMENTOS DE RADIOFRECUENCIA

En este apartado veremos los aspectos fundamentales de la radiofrecuencia que más interesan para poder realizar mi trabajo. Si se desea más información, como son las propiedades del medio inalámbrico refiérase al Apéndice 1 donde podrá encontrar estos temas de radiofrecuencia.

1.3.1. Potencia de transmisión

"La transmisión RF es un proceso de transferencia de energía, sin embargo éste es muy ineficiente; por fortuna, muchos receptores sensibles compensan esta ineficiencia tan baja. La recepción de una millonésima de la señal que envía el transmisor se considera en realidad como una señal buena en términos de fuerza. Una señal RF típica que se envía entre dos sitios es, con frecuencia, 10 000 veces más débil que eso; no obstante, sorprendentemente es muy aprovechable y en realidad es algo que ocurre en forma regular.

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. Este factor no se debe confundir con el concepto de que es constante la pérdida de propagación a través de una ruta predeterminada; en lugar de esto, la pérdida total a través de una ruta es relativamente constante para los distintos niveles de potencia o tipos de modulación. La pérdida de propagación a través del espacio libre ocurrirá a un ritmo diferente del que sucede cuando una ruta queda bloqueada de manera parcial.

La pérdida de propagación es importante para hacer cálculos, debido a que un enlace RF entre dos puntos debe 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, están 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. Una vez más, estas *copias* son tan pequeñas que debemos usar la escala dB para expresar de manera más sencilla

este cambio en la fuerza. 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.

Cada átomo a lo largo de la ruta de transmisión es afectado por los electrones, los cuales deben transmitir la energía mediante un método que usa un átomo a la vez. Entre más átomos existan, la energía se transferirá más veces. Debido a que existe una pérdida minúscula de energía cada vez que ésta se transfiere de un átomo al siguiente, la onda no sólo disminuirá en términos de energía sino que también en cuanto a la velocidad.

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."[6]

1.3.2. 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.

[6] Ibídem. P. 41 – 43.

En la tab. 1.1 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.

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

Tab. 1.1. Distribución de Frecuencias

1.3.3. 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. La fase se puede considerar como un cambio en el tiempo. 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.

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

Tab. 1.2 Esquemas de Modulación

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 tab. 1.2 se muestra una lista parcial de los esquemas de modulación.

Los diferentes esquemas que se acaban de mencionar en la tab. 1.2, 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 fig. 1.3 se puede observar la *constelación* de QPSK, que es un conjunto de combinaciones máximas que se permite entre la fase y la amplitud.

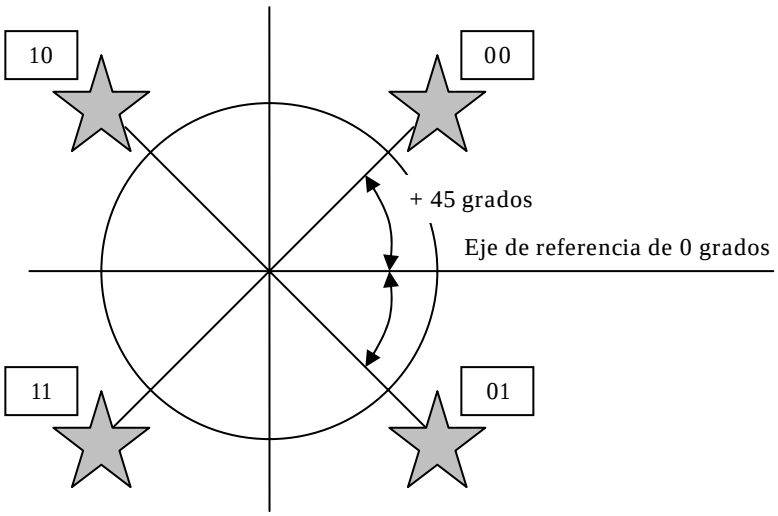


Fig. 1.3 Constelación de QPSK

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 tab. 1.3.

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

Tab. 1.3 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 radio 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."[7]

En conclusión, "la modulación, que es una función de la capa física, 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 tab. 1.4 muestra los detalles de modulación y los tipos de códigos de esparcimiento usados con WLAN con FHSS y DSSS en la banda ISM 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

Tab. 1.4 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 tab. 1.5 se muestran las técnicas de modulación usadas por redes 802.11a.

[7] Ibídem. P. 44 – 49.

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

Tab. 1.5 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 tab. 1.6 muestra los tipos de modulación usados en 802.11g."[8]

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

Tab. 1.6 Métodos de transmisión para 802.11g

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

1.3.4. 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.

1.3.4.1. 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 corrompidos de manera excesiva, o no logren llegar al destino, las redundancias inherentes a esta arquitectura proporcionen un enlace de datos más sólido.

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. Cualquiera de los enfoques es apropiado, y la selección depende de los requerimientos del cliente. El criterio de selección se involucra

principalmente con la existencia de trayectorias múltiples o la interferencia en un entorno RF.

1.3.4.2. OFDM

Multiplexión por división ortogonal de frecuencia. 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. 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, 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.

Además de la aplicación de los principios estándar de OFDM, el uso de la diversidad espacial puede incrementar la tolerancia al ruido, interferencias y trayectorias múltiples del sistema.

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.[9]

1.3.5. 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

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

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 tab. 1.7 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

Tab. 1.7 Técnicas de duplexión

1.4. 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 en el Apéndice 1.

1.4.1. Pila de protocolos de 802.11

"Es muy importante notar que todo el equipo de interconectividad de redes inalámbricas debe ser considerado como elementos de red y no simplemente radios, debido a que en este equipo se manejan por lo menos la capa física y la capa MAC de la pila de protocolos de referencia OSI.

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 (también conocida como L1) 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."[10]

"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 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 enlace al medio.

Un bosquejo parcial de la pila del protocolo 802.11 se puede observar en la fig. 1.4 donde se muestran las diferentes capas físicas que soporta el estándar IEEE 802.11."[11]

"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:

[10] Ibídem. P. 66.
[11] 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.

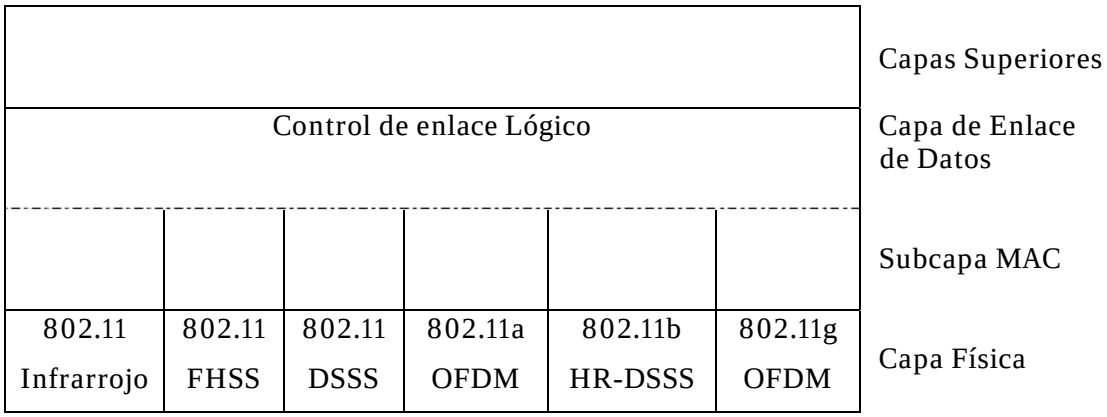


Fig. 1.4 Parte de la pila de protocolos del estándar IEEE 802.11

- 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.

El estándar 802.11 especifica los elementos L1 del radio; en otras palabras, determina 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:

- Mejoran en gran medida el desempeño de un radio.

- Dan forma a la energía radiada para la comodidad del usuario. ”[12]

[12] Cfr. Neil, Reid. Op. Cit. P. 66, 68, 78.

1.4.2. 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."[13]

"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 Dependence*, *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. Este preámbulo está formado por 128 bits para la sincronización, seguido de un campo de 16 bits que consiste del patrón 1111001110100000. Esta secuencia se usa para marcar el inicio de una trama y se conoce como el Delimitador de inicio de trama (*Start Frame Delimiter*, *SFD*, por sus siglas en inglés). Los siguientes 48 bits se conocen en conjunto como el encabezado PLCP, el cual 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.

[13] http://www.intelligraphics.com/articles/80211_article.html

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 incluye los 16 bits de HEC, que se efectúa mediante una revisión de 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."[14]

El estándar IEEE 802.11 define la capa física para cada una de sus versiones, a continuación veremos la capa física para la versión 802.11g, si se desea observar como están constituidas las capas físicas de las demás versiones del estándar 802.11, vaya al Apéndice 1.

[14] Cfr. Neil, Reid. Op. Cit. P. 31, 71, 72.

1.4.2.1. 802.11g

Una vez visto que con nuevos esquemas de transmisión como OFDM se pueden alcanzar tasas de datos más elevadas que los dos primeros estándares de 802.11 que usan espectro disperso, el IEEE se dedicó a hacer un nuevo estándar que soportara tasas similares a las de 802.11a pero que fuera compatible con las versiones anteriores del estándar.

"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, la cual ya se ha explicado anteriormente. Estos dispositivos pueden automáticamente cambiar a modulación DQPSK para poder comunicarse con dispositivos más lentos de 802.11b y 802.11."[15]

"Debido a que 802.11g opera en la banda de 2.4 GHz, todos los aspectos de la 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. Sin embargo no todas las demás cosas permanecen sin cambios. 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.

[15] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 67 – 73.

En 802.11g, a diferencia de 802.11a, no existen inconvenientes ni restricciones limitantes relacionadas con la antena y la integración de dispositivos. Al igual que 802.11b, es posible seleccionar entre varios tipos de antenas y conectarlas directamente a un punto de acceso o ubicarlas en un lugar remoto, conectándolas con el punto de acceso a través de un cable.

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. Como ejemplo, el estándar 802.11g solicita intervalos de reenvío que están basados en los intervalos de 802.11b, los cuales son más largos que los intervalos que se requieren cuando se transmite a través de OFDM. Obviamente, el estándar 802.11a, que proporciona soporte para OFDM, sólo usa el intervalo de reenvío más corto y por lo tanto proporciona una capacidad de salida más alta."[16]

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

1.4.3. Subcapa de control de acceso al medio

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

"La capa MAC es una subcapa de la Capa 2 de la pila OSI y 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. El estándar 802.11 define la forma en que funciona esta asignación de direcciones.

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.

CSMA es un protocolo que tiene como propósito resolver los conflictos de transmisión. El transmisor determinará si existe una transmisión en la frecuencia

asignada de un punto de acceso o adaptador cliente. Cuando una transmisión está en proceso, el punto de acceso o puente esperará un periodo específico, después del cual determinará si el canal de radio está desocupado o no. Los radios están programados de manera que es aleatorio el tiempo entre los intentos para determinar si un canal de radio en particular está disponible. La cantidad de tiempo entre la repetición de intentos con frecuencia se conoce como *tiempo de retroceso*.

Sin embargo, en la mayor parte de los sistemas 802.11 el tiempo de retroceso disminuye de manera uniforme hasta que el transmisor determina que existe un canal abierto. Al hacer que el tiempo disminuya uniformemente con periodos distintos, una WLAN obtiene eficiencia.

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."[17]

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

[17] Ibídem. P. 32 – 33.

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."[18]

A continuación analizaremos a fondo algunas de las funciones que realiza la capa MAC, para poder entender de mejor forma a las WLAN.

1.4.3.1. 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 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). Las tramas alámbricas tienen un tamaño máximo de 1518 bytes (de los cuales 1500 bytes están disponibles para

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

la carga), es por esta razón que generalmente las tramas inalámbricas son fragmentadas a 1518 bytes.

En una trama inalámbrica el preámbulo (una serie de 1's y 0's usada para la sincronización de bit al principio de cada trama) siempre es enviada 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). Es importante que los nodos a cada fin de liga inalámbrica usen el mismo tipo de preámbulo. Después de que el preámbulo es enviado, la cabecera (también llamada cabecera PLCP) es enviada. 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* 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"[19]

"Las tramas de administración, como podemos ver en sus nombres, 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 fig. 1.5 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	

Fig. 1.5 Formato de tramas de datos en el estándar 802.11

[19] 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 acabamos 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."[20]

1.4.3.2. El problema de los nodos ocultos y expuestos

Anteriormente hemos visto algunas diferencias entre las redes inalámbricas y las redes cableadas, sin embargo existe otra diferencia fundamental; 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 fig. 1.6 (a), donde la estación C está transmitiendo a la

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

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 expuesta, que se ilustra en la fig. 1.6 (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."[21]

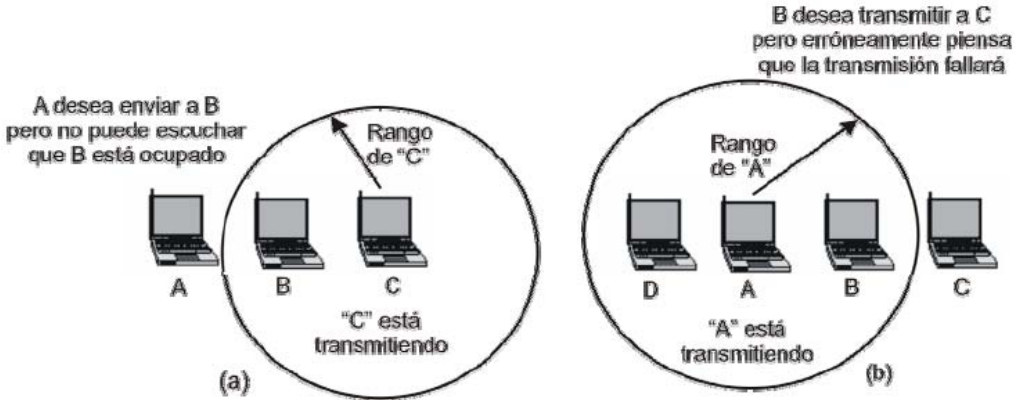


Fig. 1.6 (a) Terminal oculta, (b) Terminal expuesta

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."[22] Más adelante analizaremos un poco más a fondo estas dos técnicas de acceso.

1.4.3.3. 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

[21] Ibídem, P. 31.
[22] 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. Las 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*. La diferencia operativa entre la exploración pasiva y activa es que cuando un cliente explora de manera activa, no espera las señales radioeléctricas programadas regularmente que envían los AP; en otras palabras, los AP responden de acuerdo con la recepción de la exploración activa. A pesar de que la exploración activa puede ofrecer una pequeña ventaja en términos del tiempo necesario para identificar el AP óptimo con el que se puede conectar, también requiere de una carga de trabajo adicional debido a las tramas de transmisiones de prueba recurrentes y las respuestas correspondientes.

- *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.
   Una revisión resumida de la manera en que funciona WEP es que un cliente envía una solicitud de autenticación a un AP, pero en lugar de que el AP responda con una aprobación o desaprobación como en el caso de la autenticación abierta, responde mediante el envío de un *texto de interrogación* dentro del cuerpo de la trama que usa para responder. El texto de interrogación en realidad no es nada más que un texto que está cifrado con el propósito de determinar si el cliente tiene o no la clave apropiada para descifrar el texto. Al recibirlo, el cliente usa su clave WEP correspondiente para descifrar el texto y luego vuelve a enviarlo hacia el AP. Al recibir este texto de interrogación, el AP lo descifra y compara con el texto que se envío originalmente al cliente en respuesta a la solicitud inicial de autenticación del cliente. Cuando el texto de interrogación recibido por el AP coincide correctamente, le envía al cliente una trama de autenticación seguida por la información de direcciones IP necesarias del AP y la dirección IP asignada al cliente para esa sesión en particular.

- *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. Cuando el cliente y el AP se han asociado, comienza el proceso de autenticación.

- *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. Cuando el pirata informático tiene la capacidad de proporcionar la suma de verificación *correcta* del texto de interrogación, la red WLAN generalmente estará incapacitada para diferenciar a un residente WLAN legítimo de un pirata informático."[23]

  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.

  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

[23] Ibídem. P. 74 – 75.

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.

Los AP pueden configurarse de manera que indiquen el tráfico mediante un *Mapa de información de entrega del tráfico* (*Delivery Traffic Information Map, DTIM*, por sus siglas en inglés). El temporizador DTIM siempre está configurado de acuerdo con un múltiplo del temporizador TIM y el administrador de red puede ajustarlo en el AP. Al configurar este valor lo suficientemente alto, los clientes permanecerán inactivos por un periodo más largo, sin embargo, la desventaja de esta estrategia es que reducirá el tiempo de respuesta del cliente, debido a que podría estar inactivo cuando un paquete se envía a él desde el AP.

- *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. El estándar 802.11 permite al usuario establecer el umbral del tamaño de la trama máximo antes de que la plataforma fragmente el paquete. Cuando el usuario ha establecido sus umbrales de fragmentación, ninguna trama será más grande el tamaño máximo permitido que estableció el usuario.

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."[24]

"La fragmentación de paquetes en fragmentos más pequeños agrega sobrecarga y reduce la eficiencia del protocolo (disminuyendo la eficiencia de la red) cuando no se observan errores, pero reduce el tiempo gastado en retransmisiones si los errores ocurren. Los paquetes más grandes tienen mayor posibilidad de colisionar en la red; por lo tanto, un método de variación del tamaño del fragmento de paquete es necesario."[25]

- *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 señalizadotas desde los AP y los mismos principios de la exploración activa y pasiva de los clientes se pueden aplicar para los propósitos de roaming.

  Cuando un cliente entra en un nuevo dominio de colisión, lo primero que debe ocurrir es que él debe estar hecho para operar en el mismo canal que el AP preferido. Aunque el AP estará preconfigurado para operar en uno de los canales disponibles de la especificación 802.11 b, a o g, el cliente adquirirá el canal usado por el AP al explorar un canal en un *canal de barrido*. Un canal de barrido es donde un cliente *barre* a través de los canales y permanece en cada canal por un período específico para capturar la señal radiada desde el AP. En un *modo de barrido completo*, el conjunto completo de 11 canales será examinado, o el cliente puede estar configurado para efectuar un *barrido corto*, lo que ocurre cuando el cliente sólo barrerá un conjunto seleccionado de canales. Las lecciones de barrido corto con frecuencia se basan en los canales que se usaron previamente con el cliente. Cuando el canal ha

[24] Ibídem. P. 77 – 78.
[25] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 297 – 298.

sido adquirido, el dispositivo se autenticará y asociará de la forma normal prescrita en el estándar 802.11."[26]

1.4.3.4. 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 hemos 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."[27]

La fig. 1.7 muestra las diferentes tasas de datos de acuerdo a las distancias entre los nodos, la cual usaremos para realizar nuestras pruebas de acuerdo al estándar 802.11g.

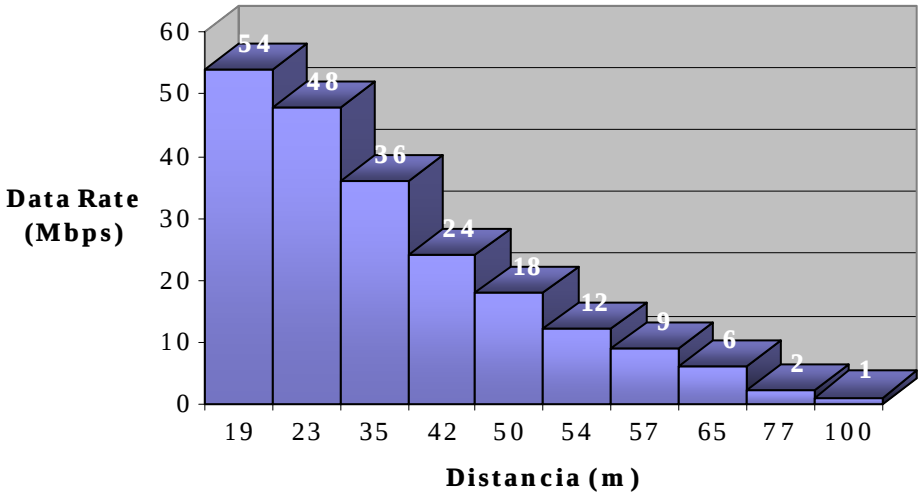


Fig. 1.7 Data Rate VS Distancia

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

1.4.3.5. 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."[28]

"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. Los servicios de contención implican que cada estación con una unidad de datos de servicio MAC (MSDU) en la cola de espera para transmisión debe contender por el acceso al canal y, una vez que el MSDU es transmitido, debe volver a contender para tener acceso al canal para todos los paquetes subsecuentes. Los servicios de contención promueven acceso justo al canal para todas las estaciones."[29]

1.4.3.6. Sensor de Portadora

Como se acaba de mencionar, el método DCF usa el protocolo CSMA/CA para la búsqueda de la frecuencia 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

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

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 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. Por lo anterior es necesario en las WLAN utilizar técnicas un tanto diferentes a las de las redes cableadas para poder transmitir exitosamente.

"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 (normalmente está previamente establecido por la administración de la red o los técnicos de instalación).
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*, que precede la transmisión de cualquier transmisor. Esto permite a todos los transmisores un acceso equitativo al canal de radio que se asigna a una LAN en particular. El

periodo de contención para los sistemas DSSS es de 20 microsegundos. Esta cantidad de tiempo permite al transmisor y al receptor reconocer la recepción exitosa de información.

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"[30]

Las redes cableadas usan como protocolo CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) es decir el protocolo CSMA con detección de colisiones por lo mencionado anteriormente.

"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 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 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. El protocolo CSMA/CA prevé la probabilidad de colisiones a través de que las estaciones compartan el medio mediante el uso de un *tiempo de retroceso aleatorio* si el mecanismo de sensado de la estación física o lógica indica que el medio está ocupado.

[30] Neil, Reid. Op. Cit. P. 72 – 73.

1.4.3.7. Tiempo de retroceso aleatorio

El periodo de tiempo inmediato a cuando se indica que el medio está ocupado es cuando existe la mayor probabilidad de colisiones, especialmente en altas utilizaciones. En este punto en tiempo, muchas estaciones pueden estar esperando a que el medio se encuentre desocupado e intentarán transmitir al mismo tiempo. Una vez que el medio se encuentra desocupado, un tiempo de retroceso aleatorio aplaza a una estación de la transmisión de una trama, minimizando la oportunidad de que las estaciones colisionen." [31]

"Los temporizadores de retroceso aleatorio se involucran cuando un nodo determina que ya existe tráfico en el canal que desea usar. Si el nodo transmisor tiene que esperar que se libere un canal, *espera un periodo aleatorio antes e intentar acceder al medio otra vez*. Esto asegura que los múltiples nodos que esperan transmitir no intenten enviar su información al mismo tiempo, debido a que los temporizadores de retroceso aleatorio aseguran que los nodos que hacen cola esperarán cantidades de tiempo distintas antes de intentar transmitir. El temporizador de tiempo es un método efectivo, simple y poco costoso para reducir las colisiones dentro de una sola LAN (dominio de colisión)."[32]

"El tiempo de retroceso aleatorio o *backoff* es un valor entero que corresponde a un número de ranuras de tiempo. Inicialmente, la estación calcula el tiempo backoff en el rango de 0-7. Después de que el medio se torna libre después de un periodo DIFS, del cual hablaremos más adelante, las estaciones decrementan su temporizador de backoff hasta que el medio se torna ocupado otra vez o el temporizador llega a cero. Si el temporizador no ha llegado a cero y el medio se torna ocupado, la estación congela su temporizador. Cuando el temporizador finalmente se ha decrementado hasta cero, la estación transmite su paquete. Si dos o mas estaciones decrementan a cero al mismo tiempo, ocurrirá una colisión, y cada estación tendrá que generar un nuevo tiempo de

[31] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 297.
[32] Neil, Reid. Op. Cit. P. 73.

backoff en el rango de 0-15. Por cada intento de retransmisión, el tiempo de backoff crece $[2^{2+i} \bullet ranf(\ )] \bullet Slot\_Time$, donde i es el número de veces consecutivas que una estación intenta transmitir una unidad de datos de protocolo de comunicación (MPDU – Message Protocol Data Unit), $ranf(\ )$ es una variable aleatoria uniforme entre (0,1), y $[x]$ representa el entero mas grande menor que o igual a x. El periodo inactivo después de un periodo DIFS se le conoce como *ventana de contención* (CW).

La ventaja de este método de acceso al canal es que promueve igualdad entre las estaciones, pero su debilidad es que probablemente no podría soportar servicios distribuidos de tiempo limitado (DTBS de sus siglas en inglés). La igualdad se mantiene porque cada estación debe volver a competir por el canal después de cada transmisión de un MSDU. Todas las estaciones tienen igual probabilidad de ganar el acceso al canal después de cada intervalo DIFS. Los servicios limitados en tiempo típicamente soportan aplicaciones tales como video o voz en paquetes que deben ser mantenidos con un retardo mínimo especificado. Con DCF, no hay un mecanismo que garantice un retardo mínimo a las estaciones que soportan servicios limitados en tiempo."[33]

1.4.3.8. 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

[33] Méndez, Luis. Op. Cit. P. 38.

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 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."[34]

PCF solo es un protocolo opcional en el estándar 802.11 que se emplea para el tráfico sensible a la latencia.

"Cuando la WLAN está intentando transportar tráfico sensible a la baja latencia, como el de video o voz, la especificación 802.11 permite la función PCF. PCF otorga el acceso a un nodo después de que la raíz AP escoge uno entre todos los clientes de un dominio de colisión. El tráfico PCF se permite dentro de los periodos de contención DCF."[35]

1.4.3.9. 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

[34] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 302.
[35] Neil, Reid. Op. Cit. P. 74.

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."[36]

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 (espacios entre tramas): 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.

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. En una red inalámbrica, todo está sincronizado y todas las estaciones y puntos de acceso usan cantidades de tiempo (espacios) estándar para realizar varias tareas. Cada nodo conoce estos espacios y los usa apropiadamente. Un conjunto de espacios estándar es especificado para Infrarrojo, FHSS, DSSS y OFDM. 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.

[36] Ibídem. P. 73.

- ✓ RTS.- Trama de petición de envío, usada para reservar el medio por las estaciones.
- ✓ CTS.- 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.- 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."[37]

1.4.3.10. 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 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. Si una estación desea transmitir su intención de usar la red, la estación envía una trama a la estación destino, la cual pondrá el campo NAV en todas las estaciones que escuchan la trama al tiempo necesario para que la estación complete su transmisión, más la trama ACK de regreso. 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."[38]

"Lo que no se conoce muy bien acerca de RTS/CTS es que los usuarios cliente pueden establecer el tamaño máximo de la longitud de la trama que se usará en este protocolo dentro del dominio de colisión. Por ejemplo, cuando el usuario establece la

[37] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 302 – 306.
[38] Ibídem. P. 310.

longitud máxima de la trama en 1000 bytes, el cliente usará RTS/CTS en todas las tramas que tengan 1000 bytes o más.

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."[39]

"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 fig. 1.8 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 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.

[39] Neil, Reid. Op. Cit. P. 76 – 77.

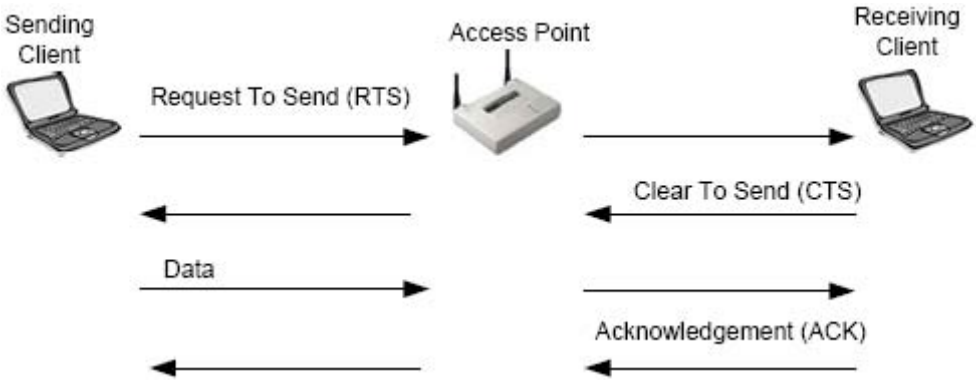


Fig. 1.8 Proceso RTS/CTS

Cuando RTS/CTS está prendido, cada paquete que viaja a través de la red inalámbrica es anunciado y confirmado de listo para transmitir entre los nodos transmisor y receptor antes de la transmisión, creando una cantidad significante de sobrecarga y bajando significativamente el rendimiento. Generalmente, RTS/CTS debe ser usado en redes en el diagnostico de problemas de red y solo cuando paquetes muy grandes están viajando a través de una red inalámbrica congestionada, lo cual es muy raro.

Sin embargo, la configuración prendido con umbral permite controlar cuales paquetes (sobre cierto tamaño – llamado umbral) son anunciados y confirmados de listos para enviar por las estaciones. Ya que las colisiones afectan más a los paquetes grandes que a los pequeños, se puede configurar el umbral del RTS/CTS para trabajar solo cuando un nodo desea transmitir paquetes sobre cierto tamaño dado. Esta configuración permite personalizar la configuración de RTS/CTS para nuestro tráfico de datos en nuestra red y optimizar el rendimiento en la red inalámbrica.

La fig. 1.9 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."[40]

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

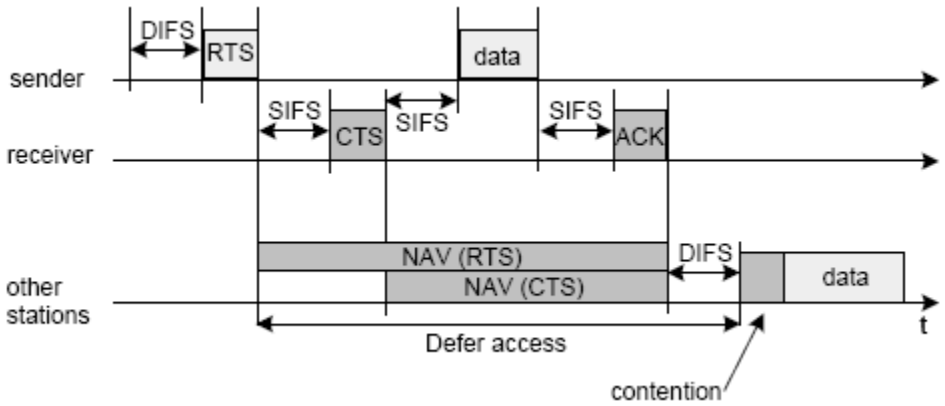


Fig. 1.9 Transmisión de datos con RTS/CTS en modo DCF

1.5. EL PROTOCOLO DE CONTROL DE TRANSPORTE TCP

En este apartado podremos observar a fondo como está constituido TCP y de que manera funciona. TCP es parte de la pila de protocolos de TCP/IP, si el lector desea tener más referencia sobre TCP/IP y todo lo que este protocolo conlleva, se puede encontrar información en el Apéndice 2.

1.5.1. Introducción

"La capa de Internet de TCP/IP está llena de protocolos útiles que son efectivos para proveer la información de direccionamiento necesaria, de tal modo que los datos pueden hacer su viaje a través de la red. El direccionamiento y encaminamiento, sin embargo, son solo parte de la transmisión. Los desarrolladores de TCP/IP sabían que necesitaban otra capa sobre la de Internet que cooperara con IP proveída de características necesarias adicionales. Específicamente, ellos querían que los protocolos de la capa de transporte proveyeran lo siguiente:

- Una interfase para las aplicaciones de red – esto es, una manera para las aplicaciones de accesar a la red. Los diseñadores querían ser capaces de direccionar los datos no sólo a la computadora destino, sino a una aplicación en particular corriendo en la computadora destino.

- 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. Esta característica se conoce como demultiplexaje, permite a una computadora soportar simultáneamente múltiples aplicaciones de red. 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.

Servicio	Número de Puerto TCP	Breve descripción
Tcpmux	1	Puerto de servicio multiplexor TCP
commpressnet	2	Utilidad de administración
commpressnet	3	Utilidad de compresión
echo	7	Echo
discard	9	Descartar o invalidar
systat	11	Usuarios
daytime	13	Tiempo
netstat	15	Estado de la red
qotd	17	Presupuesto del día
chargen	19	Generador de caracteres
ftp - data	20	Protocolo de transferencia de archivos de datos
ftp	21	Protocolo de transferencia de archivos de control
telnet	23	Conexión de red de terminal
smtp	25	Protocolo de transporte de correo sencillo
nsw - fe	27	Sistema de usuarios NSW
time	37	Servidor de tiempo
name	42	Servidor de nombres de anfitrión
domain	53	Servidor de nombres de dominio (DNS)
nameserver	53	Servidor de nombres de dominio (DNS)
DHCP	67	Protocolo de configuración de anfitrión dinámico
gopher	70	Servicio gopher
rje	77	Entrada de trabajo remoto
finger	79	Teclear
http	80	Servicio WWW
link	87	Liga TTY
supdup	95	Protocolo SUPDUP
hostnames	101	Servidor de nombres de anfitrión sri – nic
iso - tsap	102	ISO – TSAP
x400	103	Servicio de correos X.400
x400 - snd	104	Envío de correos X.400
pop	109	Protocolo de oficina postal
pop2	109	Protocolo de oficina postal 2
pop3	110	Protocolo de oficina postal 3
portmap	111	
sunrpc	111	Servicio RPC de SUN
auth	113	Servicio de autentificación
sftp	115	FTP seguro
path	117	Servicio de camino UUCP
uucp - path	117	Servicio de camino UUCP
nntp	119	Protocolo de transferencia de noticias de red Usenet
nbsession	139	Servicio de sesión NetBIOS
NeWS	144	Noticias
tcprepo	158	Repositorio TCP

Tab. 1.8 Puertos bien conocidos TCP

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. La tab. 1.8 muestra los *puertos bien conocidos* para TCP, cabe señalar que solo se presentan los más importantes. Un puerto bien conocido es un número de puerto que está asignado a una aplicación en específico por la ICANN. Combinado con la dirección IP, los puertos se convierten en la dirección del socket destino."[41]

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 analizaremos la organización global de este protocolo y describiremos las estructuras de datos que usa para administrar la información. Debido a que es uno de los protocolos más comunes en la mayoría de los equipos de cómputo he decido realizar mi trabajo basándome en este protocolo tan amplio y complejo, sin embargo, en este apartado solo veremos a TCP como una herramienta y analizaremos un poco más a fondo la parte de control de flujo que maneja TCP, ya que esa parte es la más importante para mi trabajo.

"TCP es un protocolo orientado a conexión que utiliza los servicios del nivel de Internet. 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

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

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."[42]

"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."[43]

"TCP tiene otras pocas cualidades importantes que garantizan la transmisión:

[42] Raya, Jose Luis. *TCP/IP para Windows 2000 Server*. Ed. Alfaomega. Colombia 2001. P. 94.
[43] 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. En otras palabras, 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.

1.5.2. Cabecera de TCP

El formato de la cabecera TCP se muestra en la fig. 1.10. La complejidad de esta estructura revela la complejidad de TCP y las muchas facetas de sus funcionalidades.

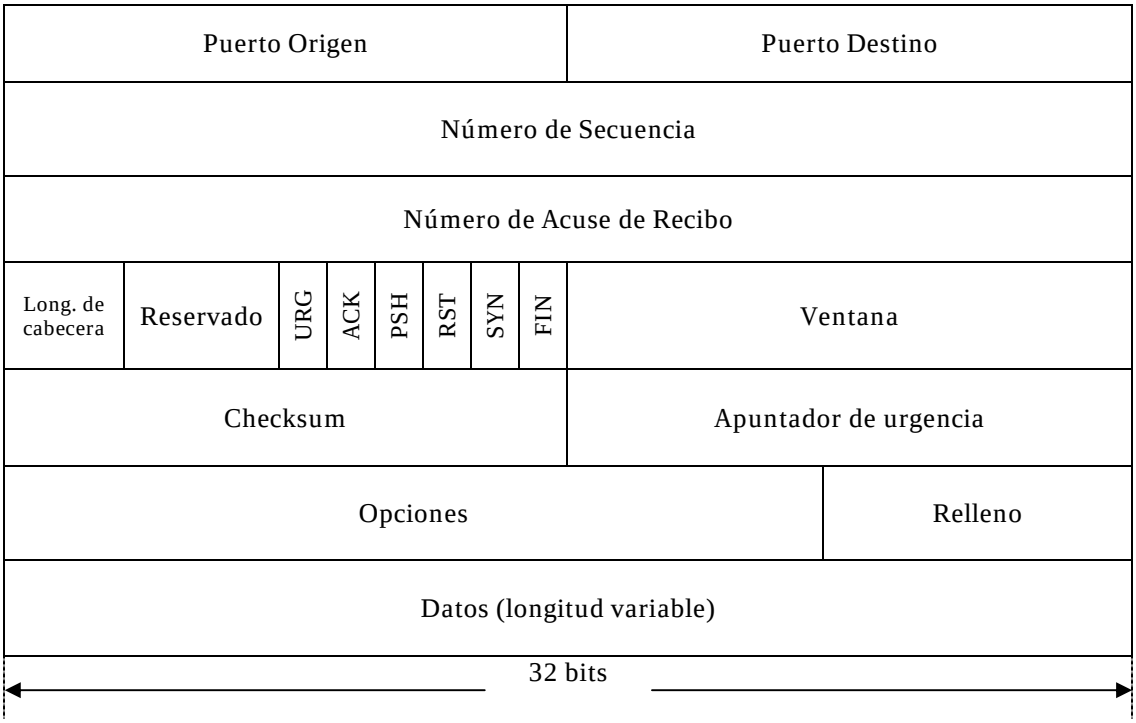


Fig. 1.10 Cabecera de TCP

- *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 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), 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*.- Un valor de 1 anuncia que el segmento es urgente y el campo de apuntador de urgente es importante.
  - ✓ *ACK*.- Un valor de 1 anuncia que el campo de número de acuse de recibo es importante.
  - ✓ *PSH*.- 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*.- Un valor de 1 resetea la conexión.
  - ✓ *SYN*.- Un valor de 1 anuncia que los números de secuencia serán sincronizados, marcando el inicio de la conexión.
  - ✓ *FIN*.- 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, es de longitud variable."[44]

1.5.3. 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.

Una de las finalidades de TCP es proveer una interfase para que las aplicaciones puedan tener acceso a la red. Esta interfase está proveída a través de los puertos TCP, en orden de proveer una conexión a través de esos 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

[44] Casad, Joe. Op. Cit. P. 92 – 95.

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."[45]

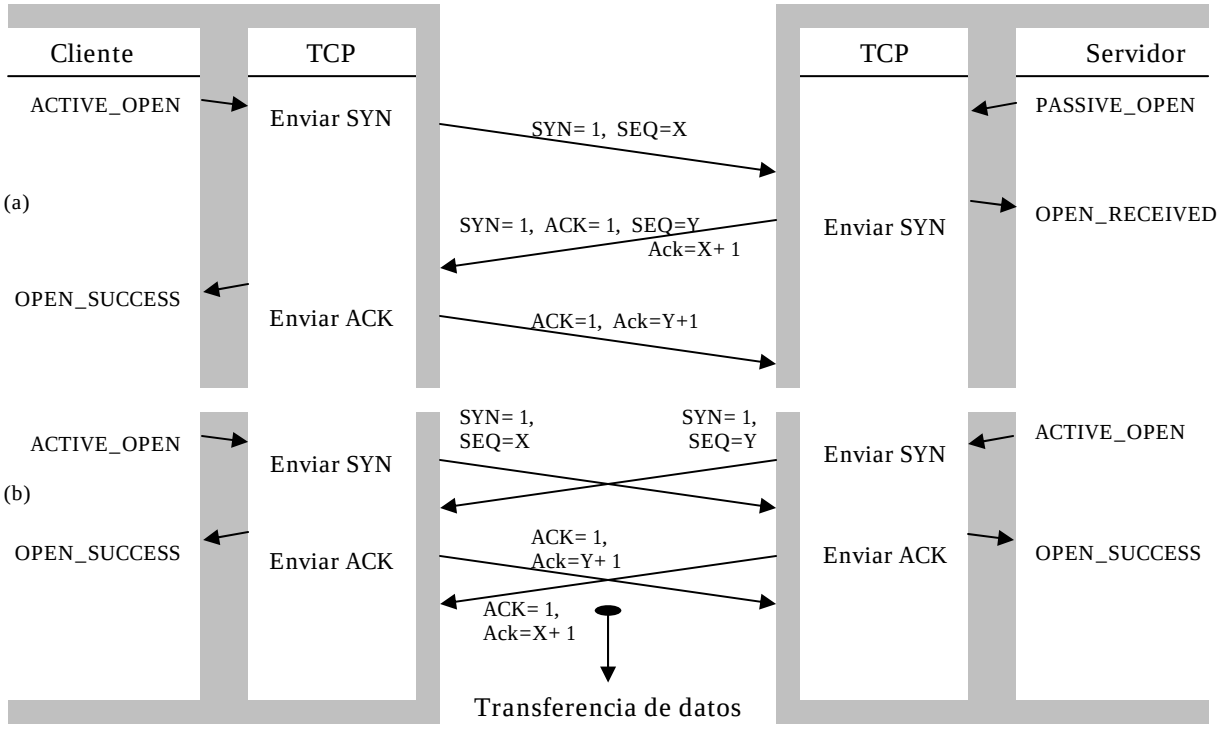


Fig. 1.11 Establecimiento de una conexión TCP:
(a) Saludo de tres pasos; (b) Posibilidad de colisión

"En la fig. 1.11 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 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

[45] Ibídem. P. 96 – 97.

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 fig. 1.11, 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."[46]

1.5.4. 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

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

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 veremos más adelante.

- *Corrección de errores*.- El checksum asegura que los datos estén libres de errores (dentro de los límites del algoritmo del checksum)."[47]

"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. El TCB contiene toda la información acerca de la conexión TCP, incluyendo las direcciones y números de puerto de los puntos terminales de la conexión, la estimación actual del tiempo de viaje de ida y vuelta, los datos enviados o recibidos, si es necesario un acuse de recibo o una retransmisión, así como todas las estadísticas que reúne TCP sobre el uso de la conexión.

Aunque el estándar del protocolo define la noción del TCB y sugiere parte de su contenido no proporciona todos los detalles. Por lo tanto, un diseñador debe elegir el contenido exacto."[48]

1.5.5. Control de flujo y 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.

[47] Parker, Tim. *Teach Yourself TCP/IP in 14 days*. Ed. Sams Publishing. Edic. 2ª. Indianapolis U.S.A. P. 114 – 115.
[48] Comer, Douglas E. Op. Cit. P. 197.

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 interred 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.

Para alcanzar la eficiencia y la solidez, la retransmisión adaptiva de 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 sitio. 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.

  El estándar 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.

- *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 cera 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.

- *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 pero 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.

- *Estimación del viaje de ida y vuelta y del tiempo de espera*.- Desde un principio, los investigadores reconocieron 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."[49]

[49] Ibídem. P. 287 – 303.

1.5.6. 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, he decidido tratarlo aparte, debido a que nuestro trabajo está basado por completo en la adaptación de estos mecanismos a las redes inalámbricas.

"Dijimos que cuando se congestiona una intrarred 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."[50]

1.5.7. Cierre de la conexión

"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 puesta en 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."[51]

A continuación veremos con más detalle este proceso a partir de "la fig. 1.12 en donde se representan las acciones de cada entidad de TCP en respuesta a los métodos de terminación de conexión alternativos. La parte (a) corresponde a una terminación elegante y la parte (b) a la secuencia de abortar.

En la parte (a) suponemos que el protocolo cliente ya terminó de enviar todos sus datos y simplemente desea terminar la conexión. Entonces, al recibir la primitiva CLOSE, el cliente envía un segmento con la bandera FIN en 1. Al recibir este segmento, el servidor emite una primitiva CLOSING al protocolo servidor y devuelve un segmento ACK al cliente para confirmar la recepción del segmento FIN.

Suponemos que el protocolo servidor también terminó de presentar datos y por tanto emite una primitiva CLOSE en respuesta a la primitiva CLOSING. Sin embargo, en

[50] Ibídem. P. 300 – 301.
[51] Casad, Joe. Op. Cit. P. 99.

el ejemplo suponemos que el TCP servidor todavía tiene datos pendientes por transmitir, que envía en un segmento con las banderas SEQ y FIN en 1. Al recibir este segmento, el TCP cliente emite una primitiva TERMINATE y devuelve un ACK por los datos que acaba de recibir. Al recibir este ACK, el TCP servidor emite su propia TERMINATE al protocolo servidor. Como alternativa, si el TCP servidor no recibe un ACK dentro de un lapso igual al doble del periodo de tiempo de vida, supone que el ACK fue alterado y emite una TERMINATE al servidor.

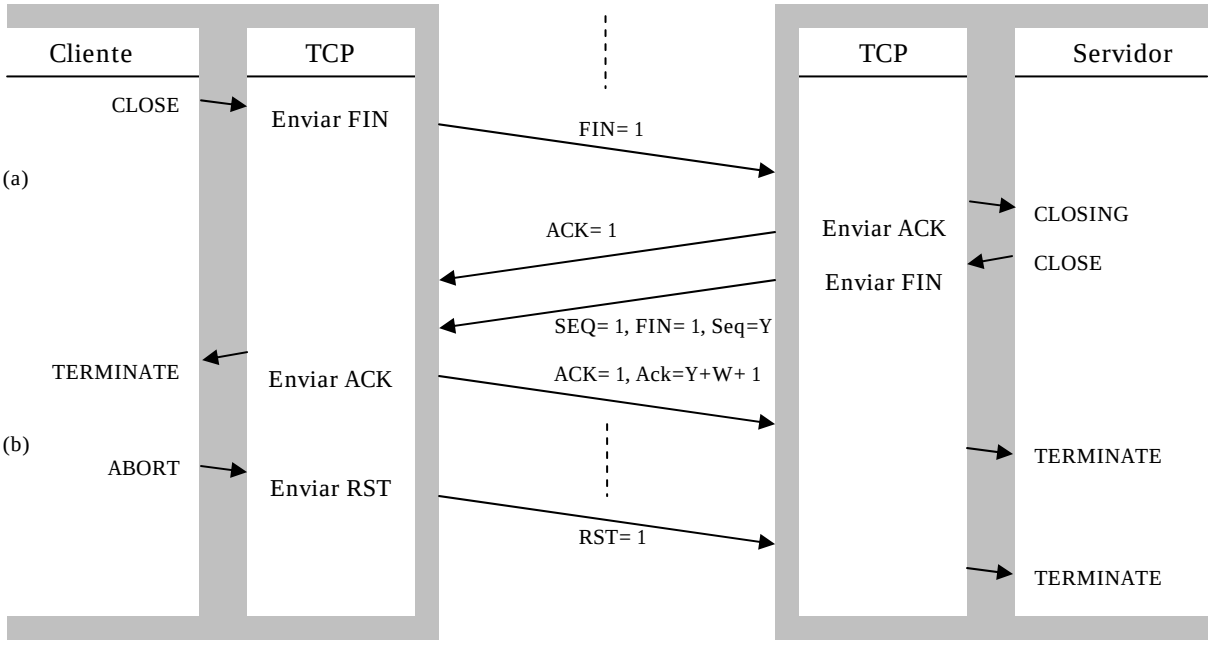


Fig. 1.12 Terminación de una conexión TCP:
(a) Normal; (b) Abortar

En la secuencia de aborto, el lado del cliente termina de inmediato ambos lados de la conexión y envía un segmento con la bandera RST en 1. Al recibir este segmento, el servidor termina ambos lados de la conexión abruptamente y emite una primitiva TERMINATE con el código de razón correspondiente a conexión abortada."[52]

[52] Hallsal, Fred. Op. Cit. P. 686 – 687.

CAPÍTULO II
ESTADO ACTUAL DEL PROYECTO

2.1. EL PROYECTO DE ADAPTACIÓN DE TCP A REDES INALÁMBRICAS 802.11G

Es aquí donde mi trabajo empieza a tomar forma; 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.

Como hemos visto, el mecanismo de inicio lento de TCP permite enviar datos siempre dentro del ancho de banda disponible para transmisión y con relación a otras variables como la ventana de congestionamiento y la MSS, 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 mi investigación se enfoca, mi trabajo de investigación en la maestría 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. Es por esto que el título de mi tesis es Adaptación de TCP a redes inalámbricas 802.11G.

2.1.1. Objetivos del proyecto

Los objetivos del proyecto son hacer las modificaciones necesarias para adaptar el simulador NS2 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 así analizar como afecta la movilidad de un equipo inalámbrico al comportamiento de los mecanismos de control del congestionamiento de la red de TCP, basándome en los resultados de las simulaciones hechas con el simulador.

Además de realizar los estudios pertinentes y presentar mis resultados, si el tiempo apremia queda como objetivo secundario 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.

2.1.2. Contribución y relevancia

Un mejor control del congestionamiento en la transmisión de datos se traduce en un mejor aprovechamiento del sistema, es decir, más bits por segundo para el usuario y sus aplicaciones. Los resultados que obtenga 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.

2.1.3. Metas del proyecto

Son varias las metas que me he propuesto 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 NS2.
- Realizar un estudio del control de congestión de TCP en el simulador NS2.
- Proponer mejoras a TCP en redes WLAN.
- 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.

2.1.4. Metodología

Para realizar la adaptación del simulador NS2 con 802.11G es necesario tener conocimientos del lenguaje de programación C así como de TCL 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. En estos momentos también estoy analizando los archivos wireless-phy.cc y wireless-phy.h para modificarlos y se encuentran escritos en C.

Se crearán simulaciones con ambientes inalámbricos en lenguaje TCL para poder ser simulados en el NS2, usando primero UDP como protocolo de transporte para después mudar 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.

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. 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.

2.2. EL SIMULADOR DE REDES NS – 2

2.2.1. 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[53], 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."[54]

"REAL es un simulador de redes originalmente hecho para estudiar el comportamiento dinámico de esquemas de control de flujo y de congestión en redes de datos de conmutación de paquetes. Este provee a los usuarios de una manera de especificar tales redes y simular su comportamiento. Provee alrededor de 30 módulos (escritos en C) que emulan exactamente las acciones de diversos protocolos de control de flujo bien conocidos (como lo es TCP) y cinco disciplinas de planificación de investigaciones. El diseño modular del sistema permite a nuevos módulos ser agregados al sistema con poco esfuerzo. El código fuente es proveído de tal forma que los usuarios interesados pueden modificar el simulador para sus propios propósitos. La documentación en línea y el código fuente son parte de la distribución.

[53] 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.
[54] http://www.isi.edu/nsnam/ns/

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. La versión 5.0 del simulador incluye una interfase de usuario gráfica (GUI) escrita en Java por Hani T. Jamjoom en la Universidad de Cornell. La GUI permite a los usuarios construir rápidamente escenarios con una interfase de apuntar y hacer clic.

REAL está escrito en C, y corre sobre sistemas Digital Unix/ SunOS/ Solaris/ IRIX/ BSD4.3/ Ultrix/ UMIPS en hardware VAX, SUN, SPARC, MIPS, Alpha, SGI, o DECstation, entre muchos más. Para una lista más completa refiérase al manual de instalación. La versión disponible actualmente es la 5.0 de agosto de 1997."[55]

En 1995 el desarrollo del NS fue soportado por DARPA a través del proyecto VINT. "VINT es un proyecto de investigación de DARPA cuyo propósito es construir un simulador de redes que permita estudiar la escala y la interacción de protocolos en el contexto de protocolos de redes actuales y futuras. VINT es un proyecto colaborativo que involucra a USC/ISI, Xerox PARC, LBNL y la UC Berkeley."[56]

"El sistema propuesto Virtual InterNetwork Testbed (VINT) espera transformar el diseño de protocolos de red y las prácticas de ingeniería en la misma manera que la simulación y los métodos basados en VHDL transformaron el nivel de diseño de chips y tarjetas. Las siguientes innovaciones permitirán a VINT llegar a esta finalidad:

- *Estructura de simulación desarrollable.*
- *Herramientas y técnicas de abstracción.*
- *Técnicas de visualización.*

[55] http://www.cs.cornell.edu/skeshav/real/overview.html
[56] http://www.isi.edu/nsnam/vint/index.html

- *Interfase de emulación*

- *Librerías de topologías de red y generadores de tráfico.*"[57]

"Actualmente el desarrollo del NS está soportado a través de DARPA con SAMAN y a través del NSF con CONSER, ambos en colaboración con otros investigadores incluyendo ACIRI."[58]

"El proyecto de Simulación Aumentada para la Medición y Análisis de Redes (SAMAN) está enfocado al problema de cómo hacer protocolos de red y operaciones más robustas a fallas. Su finalidad es entender, detectar, predecir y evitar condiciones de falla. SAMAN se encuentra dirigiendo este problema mediante la extensión de herramientas de simulación de redes actuales como es el NS2 con:

- Preprocesamiento analítico de escenarios de simulación para rápidamente descartar casos sin interés.

- Modelos de tráfico de nivel de aplicación.

- Herramientas para usar rápidamente medidas para parametrizar modelos de tráfico."[59]

Por otra parte, "la Simulación en Colaboración para la Educación y la Investigación (CONSER), propone desarrollar infraestructura direccionando a dos necesidades apasionantes en las redes hoy día:

- *Investigación* en el desarrollo y evaluación.

[57] http://www.isi.edu/nsnam/vint/proyect_overview.html
[58] http://www.isi.edu/nsnam/ns/
[59] http://www.isi.edu/saman/index.html

- *Enseñanza* de conceptos de conexión de redes actuales y protocolos de nuevos protocolos de conexión de redes.

El NS es ampliamente usado por la comunidad de redes y es ampliamente considerado como un componente crítico de infraestructura de investigación. El éxito continuo del NS como una herramienta de investigación requiere un soporte e integración de protocolos en desarrollo, y una explotación completa del NS para la educación de protocolos requiere de nuevos desarrollos. Se propone llegar a estas finalidades mediante:

- Integración de módulos de simulación para soportar el entendimiento en nuevos dominios de protocolos.

- Continuación del NS como una plataforma para diseminación de recientes resultados de investigación.

- Soporte del NS como una infraestructura de simulación para la comunidad a través de la publicación regular de versiones, reparaciones de errores, y tutoriales.

- Mejora del NAM como una herramienta de visualización y creación de escenarios para usos en salones de clase y laboratorios.

- Desarrollo de material para ilustrar conceptos de redes usando la simulación y animación."[60]

NS siempre ha incluido contribuciones substanciales de otros investigadores incluyendo código para redes inalámbricas de los proyectos UCB Daedelus y CMU Monarch.

[60] http://www.isi.edu/conser/overview.html

El simulador NS ha sufrido muchos cambios desde su primera versión, la cual fue una adaptación del simulador REAL en 1989. En julio de 1995 se publicó la versión v1.0a1, la cual sufrió diversas modificaciones liberándose nuevas versiones hasta llegar en diciembre del mismo año a la versión v1.0a17. En marzo de 1996 se liberó la versión v1.0b1, la cual llegó hasta la versión v1.0b5 de julio de 1996. Durante el mes de noviembre de 1996 se realizó la preliberación de la versión ns-2.0a1 que posteriormente se llamaría solamente ns-2.0. De aquí en adelante vendrían una serie de cambios radicales al simulador llevando toda una serie de versiones hasta llegar a la versión ns-2.1b1 ahora llamada ns-2.17 la cual ya incorporaba cualidades de trazado para el NAM versión nam1.0a2 en noviembre de 1997.

Pasaron los años y el simulador NS sufrió muchos cambios, pero no fue sino hasta febrero del 2003 cuando se liberó la versión ns-2.1b10 ahora llamada ns-2.26, esta nueva nomenclatura sería usada de aquí en adelante, dejando a un lado la nomenclatura anterior que incluía caracteres en su título. Durante febrero de 2005 se liberó la versión ns-2.28 la cual usamos para nuestro proyecto al ser esta la última versión disponible cuando comenzamos nuestra investigación. Actualmente la versión ns-2.29 fue liberada en octubre de 2005 y la versión 2.30 se encuentra en proceso de desarrollo y se encuentra pendiente su liberación.

2.2.2. Generalidades de Linux y Cygwin

“Para construir 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, pero debería correr en una computadora tipo Posix, posiblemente con algunos pequeños retoques. El NS también se puede construir 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.”[61]

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

Debido a que nosotros estamos usando computadoras del tipo PC es necesario correr el NS ya sea en Linux o en Cygwin. A continuación veremos algunas cualidades básicas de ambos sistemas. He probado el NS tanto en Cygwin como en Linux y descubrí una serie de cosas que el lector podrá observar y entender el porque he seleccionado al Cygwin para nuestra investigación.

El NS es bastante grande. El paquete de instalación *allinone* requiere de alrededor 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. A continuación se lista un resumen de los componentes necesarios para construir el NS. Debido a que los componentes dependen unos de otros, se deben construir en el orden listado:

- TCL/TK
- OTcl
- TclCL (el paquete es formalmente conocido como libTcl)
- NS-2
- NAM-1 (opcional)
- Xgraph (opcional, pero necesario para conjuntos de pruebas)
- Perl (opcional, pero necesario para conjuntos de pruebas)
- Tcl-debug (opcional, disponible para la ayuda de depuración de Tcl)
- Dmalloc (opcional, disponible para la depuración de memoria)
- Programa de conversión sgb2ns (opcional, necesario para convertir salidas GT-ITM a formato ns2)
- Programa de conversión tiers2ns (opcional, necesario para convertir salidas tires a formato ns2)
- Código fuente de Cweb y sgb (opcional, requeridos para crear librerías sgb que son usadas por los programas gt-itm y sgb2ns)

Por otra parte, 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.

2.2.2.1. 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.

En pocos años, el equipo de desarrollo de Linux, que se había expandido para incluir, no solamente desarrolladores de controladores de *kernel*, sino también desarrolladores de software y entusiastas que trabajaron febrilmente para portar software de código fuente abierto de UNIX a Linux, ha atraído una atención significativa por parte de la industria de las computadoras.

Lo que se consideraba un sistema Linux típico creció rápidamente hasta el punto de ser pesado y difícil de mantener. Con el tiempo, distintos grupos de personas empezaron a aunar sus esfuerzos para crear lo que se conoce como distribuciones de Linux, o conjuntos predefinidos de software empaquetado con el sistema operativo Linux. Éstas se distribuían generalmente en disquetes y se acompañaban con algún tipo de utilidad de instalación. Las distribuciones iniciales, tales como SLS o Slackware, se hicieron populares rápidamente entre los entusiastas de Linux por su relativa facilidad de instalación y frecuentes actualizaciones. Simultáneamente, cayeron los precios de las

unidades de CD-ROM en el momento en que estas distribuciones iniciales fueron ganando popularidad, y los CD-ROM se convirtieron rápidamente en el medio preferido para las distribuciones de Linux. 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.

Algunas ofrecían suscripciones, en donde se podían conseguir actualizaciones trimestrales mediante el pago de una tarifa anual fija; otras añadieron soporte técnico y se movieron en una dirección más comercial. Con el tiempo, se incluyeron hasta los paquetes comerciales de software que se habían portado a Linux, añadiendo viabilidad comercial a Linux.

Todavía existen muchas de las distribuciones originales, pero la mayoría de las veces éstas son utilizadas por aficionados o en escenarios académicos. Actualmente, sólo hay un puñado de distribuciones comerciales realmente producidas y mantenidas.

Los usuarios de Linux son miembros de la comunidad Linux y como miembros saben que lo que hace grande a Linux es la variedad, y que cualquiera que apoye a una distribución en definitiva estará apoyando a las otras. Las diferencias dan a los usuarios la capacidad de elección y la capacidad de elección es lo que distingue a Linux del resto de los sistemas operativos populares hoy en día."[62]

[62] Bandel, David. *Edición Especial Linux*. Ed. Prentice Hall. Edic. 6ª. Madrid, España 2001. P. 5 – 7.

Para mi proyecto comencé por instalar la distribución de Linux Mandrake versión 10.2, debido a que es una distribución de fácil comprensión e instalación; posteriormente realicé la instalación del simulador NS2 a través del paquete allinone pero encontré que existían muchos errores en la instalación y era un tanto complicado corregir todos estos errores.

Originalmente Mandrake era un reempaquetador de Red Hat Linux cuyo objetivo eran las optimizaciones de KDE. De este modo decidí mudar nuestra distribución de Linux a una ya un poco más revisada tipo Red Hat, para mi trabajo la distribución más adecuada y actual era Fedora Core 3 que estaba disponible en esos momentos. Una vez que descargue el paquete todo en uno del NS, el cual viene comprimido, primeramente se descomprime y luego se desempaqueta. Posteriormente corremos el script de instalación y teóricamente todo debería quedar listo, sin embargo en mi distribución de Linux (Fedora Core 3) hubo un pequeño problema.

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

Una vez realizado lo anterior es necesario importar algunas rutas a nuestra 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 de nuestro sistema.

Conforme realizaba mi investigación, fue liberada la distribución de Fedora Core 4 y como tenía algunos conflictos en mi computadora creí oportuno migrar mi sistema a

dicha distribución, de modo que la instalé y procedí a construir de nuevo el simulador en esta distribución. Lo que encontré 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 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 pude encontrar un parche que modificaba automáticamente todos los archivos que no reconocía correctamente el gcc de Fedora 4, lo instalé y posteriormente construí el simulador y todo resultó satisfactorio.

Una vez que tuve el simulador corriendo eficientemente en Fedora Core 4, realice diversas simulaciones para poder verificar que todo funcionara correctamente. Posteriormente me enfoque a realizar los cambios que necesitaba al NS, los cuales mencionaré más adelante. Sin embargo en ocasiones me era necesario realizar otras actividades independientes a mi 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 mi computadora, la cual es una computadora portátil Compaq Presario Modelo 2130LA, me dispuse a instalar tanto Windows como Linux en la misma PC.

Una vez que los dos sistemas operativos corrían adecuadamente, instalé el Cygwin para poder realizar mis actividades en Windows y a la vez realizar mis investigaciones con NS2.

2.2.2.2. 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."[63]

"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 pocas pruebas de validación pueden fallar. Sven Ehlert está actualmente manteniendo el NS2 para Windows/Cygwin.

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."[64]

Una vez instalado el Cygwin es necesario tener el X11 en este sistema. Dependiendo de la versión de Cygwin que se este usando, puede ser Xfree86 (paquetes XFree86-base, XFree86-bin, XFree86-prog, XFree86-lib, and XFree86-etc) o X.org (paquetes xorg-x11-bin, xorg-x11-bin-dlls, xorg-x11-devel, xorg-x11-libs-data, and xorg-x11-etc).

[63] http://www.cygwin.com/
[64] http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin

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 realice todo lo anterior, instalé la versión 2.28 del NS y verifiqué que funcionara correctamente. Para esto es necesario cargar la interfaz gráfica X11 mediante el comando *startx*, ya que sin esta 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 vamos a modificar del simulador.

Una vez que empecé a realizar las modificaciones que veremos más adelante, me di cuenta que muchas de estas variantes no funcionaban correctamente en Linux, a pesar que la recompilación no marcaba errores, el simulador no hacia lo que nosotros requeríamos, sin embargo en Cygwin funcionaban de manera satisfactoria. Es por esto que decidí seguir trabajando con Cygwin y dejar por un lado a Linux por el momento, ya que mi investigación es sobre TCP y redes inalámbricas y no sobre el simulador en si.

2.2.3. Interfase al intérprete

“El NS es un simulador orientado a objetos, escrito en C++, con un interprete Otcl 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. 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. Existen otras jerarquías en el código de C++ y en los scripts de OTcl; estas otras jerarquías no son representadas en la manera de un TclObject.

2.2.3.1. 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."[65]

"Existen un gran número de clases definidas en tclcl. Pero las clases que se usan en NS son:

- Clase Tcl.- Contiene los métodos que el código C++ usará para accesar al intérprete.
- Clase TclObject.- Es la clase base para todos los objetos del simulador que también son representados en la jerarquía compilada.
- Clase TclClass.- define la jerarquía de clase interpretada, y los métodos para permitir que el usuario instancíe TclObjects.
- Clase TclCommand.- Es usada para definir comandos simples de intérprete globales.
- Clase EmbeddedTcl.- Contiene los métodos para cargar comandos incorporados de alto nivel para hacer la configuración de las simulaciones más fácil.
- Clase InstVar.- Contiene métodos para accesar variables miembros de C++ como variables de instancia OTcl."[66]

[65] The Vint Proyect. *The NS Manual.* 2001. P. 17.
[66] The Vint Proyect. *The NS Manual.* 2001. P. 18.

2.2.4. Arquitectura general del NS – 2

"La fig. 2.1 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.

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 llamada Animador de redes (NAM- Network Animator). NAM tiene una interfase 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 (throughput) y número de paquetes tirados en cada link, aunque la información gráfica no puede ser usada para análisis preciso de la simulación."[67]

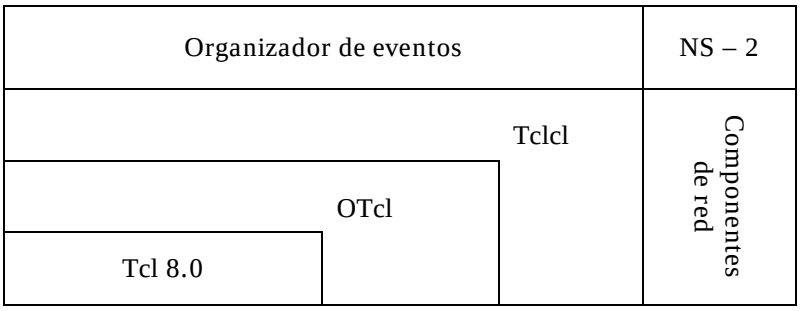


Fig. 2.1 Vista de la arquitectura de NS

[67] Méndez, Luis. Op. Cit. P. 65.

"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.

Para configurar y correr una red simulada, un usuario debe escribir un script en OTcl que inicia un organizador de eventos, configura la topología de red usando los objetos de red y las funciones de *plumbing* en la librería, y le dice a los generadores de tráfico cuando deben iniciar y cuando terminar de transmitir paquetes a través del organizador de eventos. El término *plumbing* se utiliza para la configuración de red, ya que configurar una red es conectar posibles trayectorias de datos entre objetos de red mediante la puesta del apuntador *neighbor* de un objeto a la dirección de otro objeto apropiado. Cuando un usuario desea hacer un nuevo objeto de red, puede hacerlo fácilmente ya sea escribiendo un nuevo objeto o haciendo un objeto compuesto de la librería de objetos, y conectar la trayectoria de datos a través del objeto. Esto puede sonar como un trabajo complicado, pero los módulos de conexión OTcl hacen el trabajo muy fácil. El poder de NS viene del Plumbing.

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 paquete y realiza una acción apropiada relacionada al paquete después de que cierto tiempo ha pasado, y no simula un retardo."[68]

"El simulador por completo está descrito por un simulador de clases Tcl. Este provee un conjunto de interfases para configurar una simulación y para escoger el tipo de organizador de eventos usado para manejar la simulación. Un script de simulación generalmente comienza mediante la creación de una instancia de esta clase y llamando varios métodos para crear nodos, topologías y configurar otros aspectos de la simulación. Una subclase del simulador llamada OldSim es usada para soportar compatibilidad con la versión 1 del NS.

Cuando un nuevo objeto de simulación es creado en tcl el procedimiento de inicialización realiza las siguientes operaciones:

- Inicializar el formato de los paquetes (llama a create_packetformat)
- Crear un organizador de eventos
- Crear un *agente nulo* (un destructor descartado usado en varios lugares)

La inicialización del formato del paquete coloca compensaciones de campo dentro de los paquetes usados por toda la simulación. El organizador corre la simulación en una manera de manejo de eventos y puede ser reemplazado por organizadores alternativos que proveen algunas diferentes semánticas. El agente nulo es creado con la siguiente llamada:

- `set nullAgent_ [new Agent/Null]`

[68] Ibídem. P. 63 – 64.

Este agente es generalmente útil como un destructor para los paquetes descartados o como un destino para los paquetes que no son contados o grabados.

El simulador es un simulador que maneja eventos. 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. Actualmente, el simulador está simplemente hilado, y solo un evento en ejecución en cualquier tiempo dado. Si más de un evento son calendarizados para ejecutarse al mismo tiempo, la ejecución es realizada sobre la forma primero en llegar – primero despachado. Los eventos simultáneos no son reordenados por los organizadores y todos los organizadores deben ceder el mismo orden de despacho dada la misma entrada."[69]

2.2.5. 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, excepto que no hay anuncios de ventan dinámica, no hay estados de persistencia o

[69] The Vint Proyect. *The NS Manual*. 2001. P. 37 – 38.

espera 2MSL, no hay segmentos de datos urgentes ni de Reset. Recientemente, las funcionalidades SACK, Newreno y Tahoe han sido añadidas al FullTCP.

Una vez que instalamos el simulador es necesario realizar diversas validaciones que corren todas las pruebas estándares actuales para verificar que todo funcione correctamente. Estas validaciones son opcionales, sin embargo pueden ayudar en determinado momento para saber si alguno de los protocolos implementados en el simulador no funciona correctamente en nuestra instalación. Las validaciones no consideran a todos los protocolos con los que cuenta el simulador. Los protocolos y módulos cubiertos al menos en parte por la validación del simulador incluyen los siguientes:

- *Nivel de Aplicación*:
  - ✓ HTTP, cacheo web e invalidación, TcpApp (test-suite- webcache.tcl)
  - ✓ Fuentes telnet y ftp (test-suite-simple.tcl)
  - ✓ Fuentes de bit rate constante (CBR) (test-suite-cbq.tcl)
  - ✓ Fuentes prendido/apagado (test-suite-intsserv.tcl)

- *Protocolos de transporte (UDP, TCP, RTP, SRM)*:
  - ✓ Funcionamiento básico de TCP (test-suite-simple.tcl, test-suite-v1{,a}.tcl)
  - ✓ TCP Tahoe, Reno, New-Reno, y SACK sobre diferentes pérdidas (test-suite-tcpVariants.tcl)
  - ✓ TCP FACK (validación limitada en test-suite-tcpVariants.tcl)
  - ✓ Vegas TCP (test-suite-vegas-v1.tcl)
  - ✓ TCP New-Reno (test-suite-newreno.tcl)
  - ✓ TCP SACK (test-suite-sack{,-v1,v1a})
  - ✓ Validación parcial de full TCP (test-suite-full.tcl)
  - ✓ TCP rate-based pacing (test-suite-rbp.tcp)
  - ✓ Funcionamiento de TCP RFC-2001 (Reno) (test-suite-rfc2001.tcl)
  - ✓ RTP (test-suite-friendly.tcl)
  - ✓ SRM (test-suite-srm.tcl)

- *Ruteo*:
  - ✓ Ruteo algorítmico (test-suite-algo-routing)
  - ✓ Ruteo jerárquico (test-suite-hier-routing.tcl)
  - ✓ Ruteo LAN y broadcast (test-suite-lan.tcl)
  - ✓ Ruteo manual (test-suite-manual-routing.tcl)
  - ✓ Multicast centralizado, multicast DM, no detailedDM, no multicast sobre LAN (test-suite-mcast.tcl)
  - ✓ Ruteo dinámico (test-suite-routed.tcl)
  - ✓ Simulación detallada usando clasificador virtual (test-suite-vc.tcl)
  - ✓ Simulación de nivel de sesión de modo mezclado (test-suite-mixmode.tcl)
  - ✓ Simulación de nivel de sesión (test-suite-session.tcl)

- *Mecanismos de ruteo (organización, administración de colas, control de admisión, etc.)*:
  - ✓ Diversos algoritmos de organización: FQ (Fair Queueing), SFQ (Stochastic Fair Queueing), DRR (Déficit Round Robin), FIFO (con drop-tail y administración de cola RED) (test-suite-schedule.tcl)
  - ✓ CBQ (tanto en modo v1 y v2) (test-suite-cbq{,-v1,-v1a})
  - ✓ Administración de cola RED (test-suite-red{,-v1,-v1a})
  - ✓ Funcionamiento ECN (e iteraciones TCP) (test-suite-ecn.tcl)
  - ✓ Algoritmos de control de admisión: MS, HB, ACTP, ACTO (test-suite-intserv.tcl)

- *Mecanismos de la capa de enlace*:
  - ✓ LANs con protocolos MAC CSMA/CD (test-suite-lan.tcl)
  - ✓ Snoop

- *Otros*:
  - ✓ Módulos de error (test-suite-ecn.tcl, test-suite-tcp-init-win.tcl, test-suite-session.tcl, y test-suite-srm.tcl)

Además, existe un número de protocolos en la distribución estándar del NS que no están cubiertos por las validaciones. Los protocolos y módulos en el núcleo que las validaciones no incluyen son:

- TCP Fack y Asym
- RTCP
- RTP
- LANs con protocolos MAC CSMA/CA (tcl/ex/mac-test.tcl), con MultihopMac (mac-multihop.cc), con 802.11 (mac-802_11.cc)
- RLM (Receiver Layered Multicast) (tcl/ex/test-rlm.tcl)
- Filtros de balde de muestras (tcl/ex/test-tfb.tcl)
- Fuentes de traza generada (tcl/ex*tg.tcl)
- Receptores de retrazo adaptativo (tcl/ex/test-rcvr.tcl)
- Módulos de retrazo (delaymodel.cc)
- IVS (ivs.cc)
- Modo de emulación
- Muchos otros protocolos no listados

Finalmente, existen un número de protocolos contribuidos descritos en sus propias páginas web. Estos protocolos son a menudo para distribuciones específicas del NS y pueden no trabajar en la distribución actual."[70]

Como acabamos de ver 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 hablaré más de todas las carencias del simulador, sino solo de las que a nosotros nos 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

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

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 mi trabajo 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 de mi interés.

2.2.6. Las redes inalámbricas y el NS – 2

A continuación veremos el modo en que el simulador NS2 maneja los modelos de redes inalámbricas, también veremos como modelar estas redes y como interpretar los resultados que nos entrega el simulador. Básicamente se describe el modelo inalámbrico que fue aportado por el grupo Monarca CMU como una extensión al NS. Podremos 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 hemos visto anteriormente, pueden ser con infraestructura o sin esta, de tal modo que solo veremos los modelos de las redes inalámbricas con infraestructura. Aunque actualmente existen modelos que pueden simular la combinación de redes cableadas e inalámbricas, así como MobileIP como una extensión a los modelos, estos no serán cubiertos por este trabajo al quedar un poco distantes de la esencia de la tesis en cuestión.

2.2.6.1. 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. Sin embargo, la mayor diferencia entre los nodos es que un nodo móvil no está conectado por medio de ligas a otros nodos o nodos móviles. Las redes inalámbricas en el simulador están dadas por un nodo móvil, sus mecanismos de ruteo, los protocolos de ruteo dsdv, aodv, tora y dsr, 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."[71]

2.2.6.2. 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

`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`

Tab. 2.1 Definición de opciones para la simulación

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

observar en la tab. 2.1. El Script en OTcl empieza con una lista de estos diferentes parámetros que acabamos de mencionar.

Donde val( ) es un arreglo que se utiliza para definir estas variables."[72]

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 hemos hablado. Hemos mencionado mucho los eventos, pero aún no definimos a que se refiere dicha palabra. "Un evento es la ejecución de un procedimiento Tcl programado para ocurrir en un tiempo determinado."[73] Para poder crear e iniciar el organizador de eventos se utilizan diferentes parámetros y comandos, los más simples y generalmente siempre usados son:

- `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

Posteriormente es necesario crear la topología con la que correrá nuestra simulación, ya que sin esta topología no podemos crear los nodos ya que se encontrarían en el vacío. Se pueden crear topologías de tres dimensiones, sin embargo generalmente se definen topologías de solo dos dimensiones usando las variables val(x) y val(y) que se encuentran en la definición de opciones de simulación como se puede observar en la tab. 2.1. Los comandos que se utilizan para crear la topología son:

- `set topo [new Topography]`

[72] Méndez, Luis. Op. Cit. P. 67.
[73] Idem.

- `$topo load_flatgrid $val(x) $val(y)`

Una vez que definimos 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. Con estos archivos podremos ver con detalle todo lo que aconteció en la simulación como veremos más adelante. Primeramente y con más detalle se puede generar un archivo que se pueda leer con diversos editores de texto. Para lo anterior es necesario agregar a nuestro 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, del que ya hemos hablado, podamos analizar nuestra 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 misimulación, pero puede ser el que nosotros queramos.

Una vez realizado todo lo anterior podemos empezar la creación y configuración de los nodos, que para nuestro 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 deshabilitamos esta opción no tendremos trazados, de tal modo que no podremos analizar nuestros resultados en los archivos de salida antes mencionados. Una vez que terminamos de configurar el modo en que operaran los nodos es necesario crearlos, para este fin se utiliza un ciclo *for* que nos sirve para crear el número de nodos que nosotros definimos al principio de nuestro 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 nos sirve para habilitar o deshabilitar el movimiento aleatorio, en nuestro caso es cero que significa deshabilitado ya que nosotros deseamos configurar los nodos para que sigan un patrón de movimiento diseñado por nosotros para poder realizar las simulaciones que requerimos. Existen diversos modos de hacer que los nodos se muevan.

“El objeto nodo móvil (mobilenode) está diseñado para moverse en una topología de tres dimensiones. Sin embargo, generalmente la tercera dimensión (Z) no es usada. Esto significa que el nodo móvil está asumido a siempre moverse en un terreno plano con Z siempre igual a cero. De este modo el nodo móvil tiene coordenadas X, Y, Z(=0)

que son continuamente ajustadas conforme el nodo se mueve. 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."[74]

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

"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:

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

Una vez que ya tenemos definido nuestro escenario junto con las combinaciones de movimientos que tendrán los nodos es necesario generar tráfico en nuestra 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 o TCP, podemos añadir el tráfico que requiramos a cada nodo, sin embargo para este propósito también existe una utilidad que nos ayuda a generar estos tráficos. Esta utilidad trabajo en conjunción con el NS, ya que es necesario llamar al programa principal del NS para correr la utilidad de generación de tráfico, debido a que esta es un script en tcl. La ruta en donde se puede localizar este script para utilizarlo es *~ns/indep-utils/cmu-scen-gen* y tiene por nombre *cbrgen.tcl* y *tcpgen.tcl* y puede ser usado 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 requerimos. También es necesario direccionar la salida de este script a un archivo para poderlo usar en nuestra 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 tenemos nuestro archivo de tráfico lo podemos modificar conforme nuestras necesidades, en él podemos 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."[75]

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

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

Una vez que definimos todas las opciones del modelo de movilidad, así como todos los elementos necesarios para que este funcione, debemos definir la posición inicial de los nodos conjuntamente entre el simulador y la herramienta de interfaz gráfica. Para ello es necesario definir el tamaño que queremos 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 utilicemos, de tal modo que las líneas en nuestro 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

[75] Ibídem. P. 141 – 142.

para que estos queden listos para futuras simulaciones. Para llegar a este fin se utilizan las siguientes directivas:

```
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
}
```

2.2.6.3. 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*.

Un objeto cmu-trace puede ser creado por el siguiente API:

```
set sndT [cmu-trace Send “RTR” $self]
```

el cual crea un objeto de trazado, sndT, del tipo CMUTrace/Send para rastrear todos los paquetes que son enviados fuera en un router. Los objetos de trazado pueden ser usados para rastrear paquetes en MAC, agentes (ruteo y otros), routers o cualquier otro objeto NS. El objeto cmu-trace CMUTrace está derivado de la clase base Trace.

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). Similar al método Trace::format(), el CMUTrace::format() define y dicta el formato del archivo de trazado.

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. Si el LOG_POSITION en el formato de MAC está definido, las coordenadas *x* y *y* del nodo móvil, también son impresas. Para todos los paquetes IP campos de cabecera IP adicionales son agregados al formato de trazado.

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 podemos 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, ETHERTYPE_IP es 0x0806), 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.

Otros formatos de trazado también son usados por agentes de ruteo (TORA y DSR) para apuntar ciertos eventos de ruteo especiales como *creación* (agregando una cabecera SR al paquete) o *corrida al final de una ruta fuente* indicando algún tipo de problema de ruteo con la ruta fuente, etc. Estos trazados de eventos especiales empiezan con S para DSR y con T para TORA y pueden ser encontrados en *~ns/tora/tora.cc* para TORA y en *~ns/dsr/dsrgent.cc* para el agente de ruteo DSR.

En un esfuerzo de fusionar el trazado inalámbrico, usando objetos cmu-trace, con el trazado NS, un nuevo formato de trazado ha sido introducido. Este soporte de formato revisado es compatible con el formato de trazo viejo."[76] Sin embargo nosotros no lo usaremos.

2.2.7. 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 nos interesan 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
- Escuchando un paquete no destinado para sí mismo
- Paquetes colisionando

[76] Ibídem. P. 135 – 138.

A continuación analizaremos con detalle lo que sucede en las trayectorias de transmisión y recepción de paquetes que son las que más nos interesan.

2.2.7.1. 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.

2.2.7.2. 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!

2.2.7.3. Funciones del MAC del NS – 2

A continuación describiré 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."[77]

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

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

los archivos *~ns/mac/mac-timers.{cc,h}* mientras que los manipuladores (funciones llamadas cuando los temporizadores expiran) están en *mac-802_11.cc*.

2.3. IMPLEMENTACIÓN DEL ESTÁNDAR IEEE 802.11G EN NS-2

Como hemos visto antes, el simulador no contempla el estándar 802.11G, de tal modo que fue necesario que realizara diversos cambios en el simulador para poder simular estas redes. A continuación mostraré las modificaciones a los archivos necesarios para poder llegar a este fin. Es importante mencionar que solo mostraré las modificaciones en cada archivo, mas no mostraré 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 conde se puede encontrar dicho archivo.

2.3.1. 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. Yo solo toque la parte de redes inalámbricas, aunque cabe señalar que aún antes de que modificara este archivo nuestro simulador ya funcionaba con los parámetros de 802.11g, 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             ;# Modified by Jonathan Lopez
Mac set delay_ 0us

# IEEE 802.11G MAC settings
if [TclObject is-class Mac/802_11] {
     Mac/802_11 set delay_ 64us
     Mac/802_11 set ifs_ 16us
     Mac/802_11 set slotTime_ 16us
     Mac/802_11 set cwmin_ 15       ;# Modified by Jonathan Lopez
     Mac/802_11 set cwmax_ 1023     ;# Modified by Jonathan Lopez
     Mac/802_11 set rtxLimit_ 16
     Mac/802_11 set bssId_ -1
     Mac/802_11 set sifs_ 10us      ;# Modified by Jonathan Lopez
     Mac/802_11 set pifs_ 12us
     Mac/802_11 set difs_ 16us
     Mac/802_11 set rtxAckLimit_ 1
     Mac/802_11 set rtxRtsLimit_ 3
     Mac/802_11 set basicRate_ 6Mb  ;# Modified by Jonathan Lopez
                   ;#- set this to 0 if want to use bandwidth_ for
     Mac/802_11 set dataRate_ 54Mb  ;# Modified by Jonathan Lopez
                   ;#- both control and data pkts
}
```

2.3.2. 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 tengamos 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 nos 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:

```
# Change for 802.11g parameters made by Jonathan Lopez

Mac/802_11 set CWMin_          15
Mac/802_11 set CWMax_          1023
Mac/802_11 set SlotTime_       0.000009       ;# 9µs
Mac/802_11 set CCATime_        0.000004
Mac/802_11 set RxTxTurnaroundTime 0.000002
Mac/802_11 set SIFS_           0.000010       ;# 10µs
Mac/802_11 set PreambleLength_   74           ;# 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            ;# retransmittions
Mac/802_11 set LongRetryLimit_   4            ;# retransmissions

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

# Mac/802_11 set MAC_ShortRetryLimit      7  ;# ReTX
# Mac/802_11 set MAC_LongRetryLimit       4  ;# ReTX
```

Arriba se puede observar que la parte de dataRate_ y basicRate_, los cuales son muy importantes, se encuentra comentada, esto es debido a que estamos trabajando en la forma de que estos valores se ajusten automáticamente dependiendo de diversos factores como sucede en el estándar 802.11g.

En las siguientes líneas se puede observar la segunda parte del código del archivo ns-default.tcl que fue modificada para la capa física:

```
# Other Changes for 802.11g parameters made by Jonathan Lopez

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 1e-8
Phy/WirelessPhy set bandwidth_ 54e6
Phy/WirelessPhy set Pt_ 0.030
Phy/WirelessPhy set freq_ 2.642e+6
Phy/WirelessPhy set L_ 1.0
Phy/WirelessPhy set debug_ false

Phy/WiredPhy set bandwidth_ 10e6
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.

2.3.3. 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
```

2.3.4. 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 conjunción 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 podremos 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;
```

2.3.5. Modificaciones al archivo mac-802_11.h

Para mi propósito 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 definimos 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); }
```

2.3.6. 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. Sin embargo en esto aún estoy trabajando como lo veremos en el subcapítulo siguiente llamado estado actual del proyecto.

2.4. ESTADO ACTUAL DEL PROYECTO

En estos momentos casi he terminado de realizar los cambios necesarios para que el simulador pueda funcionar con redes 802.11g, sin embargo hay un detalle que aún no he podido resolver, la adaptación automática de la tasa de transferencia. Con los cambios que he hecho podemos transmitir a diferentes tasas de datos, sin embargo esto lo hacemos manualmente y se configura para toda la simulación, de tal modo que no importa a que distancia se encuentren los nodos transmisor y receptor, ni la relación señal a ruido, siempre la tasa de transmisión se mantiene constante al valor que ajustamos antes de realizar la simulación.

He podido observar que el archivo mac-802_11.cc siempre se carga antes de enviar cualquier paquete y después de recibir estos mismos. De este modo estamos experimentando en la modificación de este archivo para poder realizar la adaptación automática de la tasa de datos en base a la distancia que haya entre los nodos transmisor y receptor como se mostró en la fig. 1.7.

La primera modificación es añadir una serie de librerías que nos ayudarán a obtener la posición de los nodos, esta modificación es:

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

Los cambios más importantes con los cuales estoy experimentando 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 esté lista, el simulador estará completo para poder simular redes 802.11g. De aquí en adelante el trabajo abordará la parte de simulaciones con TCP, de tal modo que haré una serie de simulaciones tan variadas que se pueda observar el funcionamiento de este protocolo en diversas condiciones de tráfico, de congestión, etc.

Una vez que obtenga los resultados de toda la serie de simulaciones, podré analizar a fondo el protocolo para proveer a la comunidad de un trabajo que ayude a saber qué esperar y qué no en las redes inalámbricas con TCP. Si el tiempo apremia y los resultados mostrados en las simulaciones muestran que TCP es un tanto ineficiente en las WLAN, haremos un protocolo alternativo que mejore el rendimiento actual de TCP en dichas redes.

CONCLUSIONES

La movilidad afecta mucho a TCP, por lo que es necesario diseñar un algoritmo que defina la velocidad de movimiento del nodo móvil y pueda predecir hacia donde se está moviendo, 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.

Desafortunadamente el simulador NS – 2 está hecho por diferentes personas, de tal modo que muchas partes del código son un tanto confusas, por lo que mi trabajo se ha estancado por momentos, para poder adaptarlo a mis propósitos. Si el simulador pudiera manipular correctamente las WLAN, en este momento tuviéramos una mejor idea de cómo afecta el movimiento a TCP al cambiar de zonas de transmisión, de modo que mi trabajo ya hubiese avanzado mucho más.

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, de tal modo que modificarlo o implantar un nuevo protocolo de transporte no es un trabajo sencillo. Sin embargo espero 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, del mismo modo espero 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 nuestro trabajo puedo decir que hasta el momento he analizado a fondo el simulador NS – 2 y he visto 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 de todo el código que esta comunidad ha contribuido.

Una vez que todo mi trabajo funcione correctamente, se piensa colocar una página Web para que todo aquel interesado en redes inalámbricas tenga acceso al trabajo que realizo en conjunción con mi asesor de maestría, el Dr. Javier Gómez Castellanos, y pueda hacer sus propias investigaciones. Así mismo, este trabajo de investigación se publicará en diversas revistas científicas (Journals) para una amplia difusión.

APÉNDICE 1
CONCEPTOS GENERALES

En este apéndice el lector podrá encontrar información adicional a la mencionada en el cuerpo principal de esta tesis. Los temas que se tratan a continuación son: historia de redes, conceptos generales de redes, historia de las redes inalámbricas, los diferentes estándares inalámbricos, aplicaciones de las WLAN, fundamentos de radiofrecuencia, y el estándar IEEE 802.11. A lo largo de la tesis hemos visto ya parte de estos temas, aquí solo se podrá encontrar la información que no haya quedado clara o no se haya revisado.

HISTORIA DE LAS REDES

En la década de los sesenta había una gran inquietud en todos los Estados Unidos de América, debido a la inminente guerra que se planteaba con la Unión Soviética. En el Departamento de Defensa de los Estados Unidos se dieron cuenta de que tenían una gran debilidad en su sistema de comunicaciones, porque si se requería un ataque nuclear la orden debería de ser directa del presidente, sin embargo ésta orden requería que el mismo presidente se comunicara mediante una línea conmutada con el Departamento de Defensa, de tal modo que si el enemigo bloqueaba de alguna forma esa línea, entonces la orden de ataque no podría ser enviada. De esta manera, a finales de los sesenta, varios científicos de la empresa *RAND* propusieron la tecnología de conmutación de paquetes, la cual sería un gran avance en las comunicaciones.

En estos momentos la red de conmutación de paquetes se introdujo en el sistema de comunicaciones de ese departamento, pero fue hasta 1969 cuando la Agencia de Proyectos de Investigación del mismo Departamento de Defensa construyo la primera red de datos del mundo, llamada *ARPANET* (de las siglas en ingles de *Advanced Research Proyects Agency* ARPA y NET de *network*). La ARPANET estaba compuesta

por cuatro nodos situados en la Universidad de California en Los Ángeles (*UCLA*), en el *Stanford Research Institute* (*SRI*), en la Universidad de California de Santa Bárbara (*UCBS*), y la universidad de *UTA*.

La primera comunicación entre computadoras se produjo entre la UCLA y Stanford el 20 de Octubre de 1969, esta red era de conmutación de paquetes y estaba basada en un conjunto de pequeñas computadoras interconectadas llamadas procesadores de mensajes con interfaz (*IMPs*). Estos IMPs son los precursores de los modernos dispositivos de encaminamiento o enrutamiento que además ofrecían también almacenamiento. En ese mismo año la Universidad de Michigan crearía una red también basada en la conmutación de paquetes, con un protocolo llamado X.25. También en 1969 se empiezan a crear los primeros RFC (*Request For Comments*, Petición de Comentarios) que originalmente eran documentos que normalizaban el funcionamiento de las redes de computadoras basadas en TCP/IP y sus protocolos asociados.

En 1970 la ARPANET comienza a utilizar para sus comunicaciones un protocolo Punto a Punto llamado NCP, el cual es el predecesor del actual TCP/IP que se utiliza en toda la Internet. En ese mismo año, Norman Abramson desarrolla la *ALOHANET* que era la primera red de comunicación de paquetes inalámbrica.

"Ya en 1971 la ARPANET estaba compuesta por 15 nodos y 21 máquinas que se unían mediante conmutación de paquetes. En 1972 se elige el popular @ como tecla de puntuación para la separación del nombre del usuario y de la máquina donde estaba dicho usuario. Se realiza la primera demostración pública de la ARPANET con 40 computadoras.

En 1973 se produce la primera conexión internacional de la ARPANET. Dicha conexión se realiza con el colegio universitario de Londres (Inglaterra) En ese mismo año Bob Metcalfe expone sus primeras ideas para la implementación del protocolo Ethernet que es uno de los protocolos más importantes que se utiliza en las redes locales. A mediados de ese año se edita el *RFC454* con especificaciones para la transferencia de archivos, a la vez que la universidad de Stanford comienza a emitir noticias a través de la

ARPANET de manera permanente. En ese momento la ARPANET contaba ya con 2000 usuarios y el 75% de su tráfico lo generaba el intercambio de correo electrónico.

En 1974 Cerf y Kahn publican su artículo, un protocolo para interconexión de redes de paquetes, que especificaba con detalle el diseño del protocolo de control de transmisión (*TCP*). En 1975, Se prueban los primeros enlaces vía satélite cruzando dos océanos (desde Hawai a Inglaterra) con las primeras pruebas de TCP de la mano de Stanford, UCLA y UCL. En ese mismo año se distribuyen las primera versiones del programa UUCP (Unix-to-Unix CoPy) del sistema operativo UNIX por parte de AT&T.

La parada generalizada de la ARPNET el 27 de octubre de 1980 da los primeros avisos sobre los peligros de la misma. Ese mismo año se crean redes particulares como la CSNET que proporciona servicios de red a científicos sin acceso a la ARPANET. En 1982 es el año en que la DCA y la ARPA nombran a TCP e IP como el conjunto de protocolos TCP/IP de comunicación a través de la ARPANET. El 1 de enero de 1983 se abandona la etapa de transición de NCP a TCP/IP pasando este último a ser el único protocolo de la ARPANET. Se comienza a unir redes y países ese mismo año como la CSNET, la MINET europea y se crearon nuevas redes como la EARN.

En 1985 se establecen responsabilidades para el control de los nombres de dominio y así el ISI (*Información Sciences Institute*) asume la responsabilidad de ser la raíz para la resolución de los nombres de dominio. El 15 de marzo se produce el primer registro de nombre de dominio (symbolics.com) a los que seguirían cmu.edu, purdue.edu, rice.edu, ucla.edu y .uk."[78]

En 1986 la Fundación de Ciencia Nacional de Estados Unidos comenzó el desarrollo de una espina dorsal de alta velocidad llamada NSFnet, para conectar los centros de supercomputación de la nación. A medida que fue creciendo la demanda del ancho de banda de esta espina dorsal, la NSF dio origen a Merit en una muy productiva unión con MCI e IBM, para desarrollar una espina dorsal de 1.5 Mbps. NSFnet comenzó

[78] http://galeon.hispavista.com/redeslanabedulmo/historia.html

a servir como el mayor soporte entre redes, dando origen a la red Internet, que por supuesto reemplazó a la ARPANET.

Fuera de la Merit, surgió una nueva corporación sin ánimo de lucro para redes avanzadas y servicios: la ANS. Esta corporación, fue inicialmente responsable del desarrollo de la espina dorsal de redes y servicios de la NSF a 45 Mbps, la red TCP/IP más veloz del mundo.

La red Internet, nació en los años 1990, creciendo desde más o menos 500,000 equipos hasta más de 10 millones que había en 1996. Actualmente el número de computadoras es mucho mayor y sigue creciendo día a día. El World Wide Web, desarrollado por el CERN (Laboratorio Europeo de Física Nuclear), ha sido la mayor fuerza cercana al exponencial crecimiento de la red Internet.

CONCEPTOS GENERALES DE REDES

Es muy importante que en el momento de instalar nuevos equipos de cómputo, se tomen en cuenta la clase de técnicas y recursos de transmisión y recepción de datos que les permitan comunicarse con otros equipos de cómputo. Existen numerosas y variadas redes de comunicación, de tal modo que es necesario conocer y comprender los diferentes requerimientos de cada interfaz para cada tipo de red.

Definición de Red

He mencionado mucho las redes de computadoras, sin embargo no he definido aún lo que ese término significa en realidad. A continuación daremos algunas definiciones para poder entender mejor el concepto.

- "Interconexión entre varias terminales o sistemas de computadoras por medio de líneas de comunicación de datos. Puede estar compuesta de dos o más computadoras que se comunican entre sí."[79]

- "Agrupación de computadoras, impresoras, routers, switches y otros dispositivos que se pueden comunicar entre sí a través de un medio de transmisión."[80]

- "Una red de computadoras (también llamada red de ordenadores, red informática o red a secas) es un conjunto de dos o más computadores o dispositivos conectados entre sí y que comparten información (archivos), recursos (CD-ROM, impresoras, etc.) y servicios (e-mail, chat, juegos), etc."[81]

- "Red es un sistema de comunicaciones, ya que permite comunicarse con otros usuarios y compartir archivos y periféricos. Es decir es un sistema de comunicaciones que conecta a varias unidades y que les permite intercambiar información."[82]

Nosotros manejaremos la definición como: un sistema de comunicaciones que permite que 2 o más computadoras o dispositivos electrónicos intercambien información, recursos y/o servicios.

Clasificación de las Redes

Existen diferentes tipos de redes y también diferentes modos de clasificarlas, entre estos modos de clasificación se encuentran por sistema operativo: Unix, Windows, Linux, Novell, etc. También se pueden clasificar según su topología: Anillo, Bus, Estrella, Malla, etc. Otro modo de clasificarlas es por su arquitectura en Centralizadas y Distribuidas. Del mismo modo se pueden clasificar por su extensión: Redes de Área Local (*LAN*), Redes de Área Metropolitana (*MAN*) y Redes de Área Amplia (*WAN*), también algunos

[79] *Diccionario de Términos de Computación*. Ed. Grupo Editorial Tomo, S. A. de C. V. México D.F. 2000. P. 168.
[80] Navarro, Anna. *Diccionario de Términos de Comunicaciones y Redes*. Ed. Pearson Educación, S. A. Madrid, España. 2003. P. 242.
[81] http://es.wikipedia.org/wiki/Red_de_computadoras
[82] http://www.monografias.com/trabajos5/redes/redes.shtml

autores marcan una cuarta extensión como Redes de Área Global (*GAN*) pero ésta clasificación se encuentra en desuso. También se pueden clasificar de acuerdo a la tecnología que emplean: Ethernet, TokenRing, FDDI, etc. Para nuestros fines la mejor clasificación es por extensión y solo manejaremos LAN, MAN y WAN.

La diferencia entre estos tres tipos de redes radica en que las LAN se extienden hasta 5 Km de longitud, las MAN se extienden hasta una ciudad y las WAN se extienden más allá de una ciudad. Sin embargo en la actualidad esta clasificación empieza a desaparecer con la llegada de nuevos medios y técnicas de transmisión y de nuevos dispositivos de interconexión. En nuestros días es cada vez es más común ver redes que se les hace llamar LAN, pero en cuanto a distancia van más allá de una ciudad, de tal modo que en el medio de las redes el término MAN empieza a desaparecer.

HISTORIA DE LAS REDES INALÁMBRICAS

Comúnmente a las redes que no usan cables como medio de transmisión de datos se les conoce como redes inalámbricas, sin embargo en realidad se trata de tecnología de radio. La tecnología de las redes inalámbricas comenzó su historia a mediados de los años ochenta, pero la tecnología de transmisión de radio ya existía desde mucho antes.

Una vez que Heinrich Hertz demostró en la década de 1880 que en la naturaleza existen realmente las ondas electromagnéticas que James Clerk Maxwell había anticipado, se inició una serie de estudios teóricos y experimentales para encontrar sus diversas propiedades. En los Estados Unidos, Nikola Tesla logró hacer varias demostraciones usando descargas de alto voltaje y de alta frecuencia, para lo cual inventó una bobina, llamada bobina de Tesla, que posteriormente fue de utilidad para las comunicaciones inalámbricas.

En 1892 William Crookes publicó un trabajo en el que proponía las bases para utilizar ondas electromagnéticas como medio para transmitir señales telegráficas a través del aire. En 1894 el físico inglés Oliver Lodge, basándose en el trabajo de Crookes,

desarrolló el primer sistema de comunicación inalámbrica que permitía recibir una señal transmitida inalámbricamente a 100 metros de distancia.

Durante 1895 Guglielmo Marconi construyo un aparato que anteriormente Hertz ya había descrito pero les añadió varias partes más, como una antena. Con estos aparatos Marconi logró enviar señales hasta distancias de un par de kilómetros. Para 1896 obtuvo su primera patente en Inglaterra y hacia 1898 el aparato de Marconi ya se usaba ampliamente en barcos y tierra firme. En 1901 Marconi logró una transmisión a través del Océano Atlántico: de Polhu en Cornualles, Inglaterra, hasta San Juan de Terranova, Canadá.

Posteriormente Thomas Edison fue quien impulsó los primeros sistemas inalámbricos que se empezaron a vender en América. El trabajo de Edison se basó en los trabajos realizados por Marconi, así como del trabajo de Tesla quien colaboró hombro a hombro con Edison.

Es importante mencionar que en 1923, el gobierno de los Estados Unidos de América comenzó el proceso de dividir el espectro de frecuencias de radio en asignaciones para usos y usuarios específicos.

"Lamarr y Antheil crearon un sistema para emitir comunicaciones de radio de banda angosta a través de una banda ancha en el espectro de frecuencia, como un medio para guiar torpedos hacia sus blancos de una manera que fuera menos susceptible a las técnicas de obstrucción de frecuencias o al espionaje. Esto se realizó a fin de lograr que la frecuencia que utilizaban el controlador y el torpedo para comunicarse se cambiara o *saltara* de un canal al siguiente, mediante un modelo predeterminado y coordinado. La patente que apareció como resultado, premiada el 11 de agosto de 1942, fue el primer sistema de espectro extendido."[83]

[83] Reid, Neil. Op. Cit. P. 6.

En 1934, la Federal Communications Commission (FCC por sus siglas en inglés) se establece y en 1985 ésta organización asigna las porciones del espectro radioeléctrico que las entidades Industriales, Científicas y Médicas (*ISM* por sus siglas en inglés) podrán utilizar sin necesidad de una licencia.

El verdadero precursor de las redes inalámbricas actuales fue la red "ALOHANET, un sistema inalámbrico que conectaba a las Islas Hawaianas. ALOHANET fue el primer sistema creado para enviar paquetes de datos a través de radios con una velocidad de operación de 9,600 bits por segundo (bps) y no solo es un precursor de las LAN inalámbricas, sino que también representa la base de la tecnología de área local *cableada* predominante: Ethernet.

Telxon Corporation de Akron, Ohio, desarrolló uno de los primeros sistemas inalámbricos mediante la integración de un módem de 1,200 bps a una Terminal de grupos de datos de proceso."[84]

"A pesar de que en 1981 la FCC había descalificado las patentes del espectro extendido y que, a petición de la FCC, el Instituto de ingenieros eléctricos y electrónicos (*Institute of Electrical and Electronics Engineers, IEEE*, por sus siglas en inglés) comenzara a estudiar las aplicaciones comerciales de las comunicaciones del espectro extendido, los radios que se asignaron para el servicio de los sistemas de adquisición de datos inalámbricos eran dispositivos de banda angosta y no estaban basados en la tecnología de espectro extendido. No fue sino hasta 1985 cuando se establecieron las regulaciones para permitir el uso público controlado de la tecnología de espectro disperso."[85]

"En 1985 gracias a las regulaciones propuestas se empezó a comercializar la tecnología inalámbrica para redes. En esta comercialización la compañía Telesystems SLW tuvo un papel muy importante en el desarrollo de estas redes. El sistema diseñado por esta compañía era en realidad de espectro extendido y usaba una variación de este

[84] Ibídem P. 7.
[85] Ibídem P. 8.

tema, que es distinta del sistema de cambio de frecuencia. En lugar de hacer que la señal de banda angosta saltara de una frecuencia a otra a través de un ancho de banda establecido, Telesystems empleó un sistema que se conoce como *secuencia directa*, donde una señal de banda angosta se extiende a través del ancho de banda determinado al multiplicar el ancho de la señal a través de un conjunto de frecuencias más grande.

En 1988 fue introducido al mercado el primer sistema comercial basado en la tecnología Secuencia directa en el espectro extendido (*Direct Sequence Spread Spectrum, DSSS*, por sus siglas en inglés). Estos sistemas usaban la banda libre alrededor de los 902 y 928 Mhz. Más tarde para llegar a los mercados ubicados fuera de Estados Unidos, Canadá o Australia, los fabricantes de sistemas inalámbricos comenzaron a producir radios que operaban en la banda de los 2.4 Ghz del espectro de frecuencia que estaba disponible para la operación libre de licencia a lo largo de la mayor parte de Europa y Japón.

Ya en 1993, los fundamentos para un estándar estaban establecidos, pero no fue sino hasta junio de 1997, que el estándar 802.11 de la IEEE, que tenía más de seis años en proceso de desarrollo, fue ratificado."[86]

LOS DIFERENTES ESTÁNDARES INALÁMBRICOS EXISTENTES

"Los estándares existen principalmente a fin de asegurar una base y acuerdo común para la forma y función de los dispositivos y servicios que nos rodean. Además, en los tiempos modernos, los estándares se emplean para ayudar en la protección de las personas que usan productos y servicios.

Los estándares también reducen el número de problemas mediante la capacidad de interoperabilidad e intercambio de información. Mientras más ampliamente esté adoptado un estándar, el mercado será más extenso para un grupo de proveedores de

[86] Cfr. Ibídem P. 9 – 12.

tecnología, donde el estándar de mayor preferencia será el que sea ratificado en forma global. Y también ocurre que mientras más ampliamente este adoptado un estándar, será más grande la cantidad de sinergia entre los proveedores de tecnología y el mercado que satisfacen. Sin embargo, los estándares internacionales son mucho más difíciles en la práctica que en la teoría.

Así mismo, los estándares definen por ley lo que ciertos valores representan, por ejemplo, el pie, el metro, el tiempo, un galón o litro de gasolina por mencionar algunos. Los estándares definen lo que un dispositivo o servicio específico es o no es, y permiten a los proveedores de tecnologías y servicios establecer, usar y adherirse en forma legítima a los estándares definidos de manera adecuada."[87]

"La definición de un estándar no es de ninguna manera banal. El aspecto aún más elemental de *cuando* se debe adoptar un estándar es tan importante como el estándar mismo, debido a que vivimos en un mundo donde los conceptos, desarrollo y despliegue de tecnología se realizan en periodos que disminuyen con rapidez.

En el mundo de las WLAN, tenemos la suerte de contar con ciertos estándares fundamentales que han sido ratificados a nivel global, por ejemplo, la frecuencia, la energía y el tiempo."[88]

"El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE de sus siglas en ingles) es el fabricador clave de estándares para la mayoría de las cosas relacionadas a las tecnologías de la información en los Estados Unidos. El IEEE crea sus estándares dentro del marco legal creado por el FCC. El IEEE crea muchos estándares de tecnología como la Llave de Criptografía pública (IEEE 1363), Firewire (IEEE 1394), Ethernet (IEEE 802.3), y WLAN (IEEE 802.11)."[89]

[87] Reid, Neil. Op. Cit. P. 14.
[88] Ibídem P.15.
[89] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 218.

"En ocasiones los estándares no emergen de entidades como la IEEE, sino que provienen de un consorcio de proveedores de tecnología, como el Foro OFDM o la Alianza de compatibilidad de Ethernet inalámbrico (*Wireless Ethernet Compatibility Alliance, WECA*, por sus siglas en inglés). A pesar de que estos foros no son entidades de estándares, varios de ellos se consideran seriamente en el mercado y por los proveedores de tecnología debido a que mantienen una relación cercana con la IEEE. Además, los foros como BWIF normalmente están compuestos por alguna de las compañías de tecnología más importantes, por ejemplo, Cisco. Los foros también ofrecen la ventaja del acceso a las arquitecturas abiertas."[90]

"El objetivo de WECA es el de certificar la interoperabilidad de los productos de alta velocidad IEEE 802.11 y de promover Wi-Fi como el estándar LAN inalámbrico global a través de todos los segmentos del mercado. WECA es uno de los grupos principales que lograron que *802.11b* se conociera como *Wi-Fi* o *Fidelidad inalámbrica*. WECA también intenta asegurar que si usted compra un producto con la insignia de Wi-Fi, de cualquier vendedor, sea compatible con otras tarjetas similares y, más importante aún, con los puntos de acceso con los que se conectan a la LAN o incluso Internet, que también cuenten con el logotipo de Wi-Fi.

Entre las compañías que actualmente apoyan a estos grupos de estándares están 3Com, Acere, Nokia, Apple Computer, Atmel y Cisco Systems; hoy día suman un total de aproximadamente 150 compañías. Una de las características más importantes respecto a WECA es que está adoptando y promoviendo un estándar (802.11) en lugar de implantar un estándar propio. En comparación está HomeRF, un consorcio de compañías cuyo objetivo principal es el mercado residencial de los dispositivos inalámbricos, no obstante que es interesante observar que WECA también trabaja con los Grupos de interés especial (*Special Interest Groups, SIG*, por sus siglas en inglés) de BlueTooth, debido a que la frecuencia de 802.11b y las frecuencias de BlueTooth son similares y ambas usan el espectro extendido. La alianza entre WECA y BlueTooth tiene como objetivo promover la capacidad de que ambos protocolos operen en un entorno físico común. En tanto que

[90] Reid, Neil. Op. Cit. P. 20 – 21.

los dispositivos WECA principalmente conectarán PC e impresoras, el propósito de BlueTooth es esencialmente enfocarse a los productos electrónicos más pequeños, como las cámaras digitales, PDA y teléfonos celulares, aunque ya existen dispositivos BlueTooth en PC.

WECA no solo está enfocado en el estándar 802.11b y la interoperabilidad entre los productos de distintos fabricantes, sino que también en el estándar 802.11a, que opera en una frecuencia de 5 GHz. También existe un tercer grupo de trabajo, el grupo 802.11g, que es similar a 802.11b, pero que utiliza OFDM como la técnica de propagación y la integración de QAM como una de las técnicas de modulación permitidas."[91]

"Por otra parte, existe también el Instituto de Estándares Europeos (ETSI, de sus siglas en inglés) que está constituido para producir estándares de comunicaciones para Europa en la misma forma que la IEEE lo es para los Estados Unidos. Los estándares ETSI se han establecido para competir directamente con los estándares creados por la IEEE. Han existido grandes discusiones en como la IEEE y el ETSI se unificarán en ciertas tecnologías inalámbricas.

El estándar original HiPerLAN del ETSI para redes inalámbricas, también llamado HiperLAN/1, soporta rangos de hasta 24 Mbps usando tecnología DSSS con un rango de aproximadamente 45.7 metros. HiperLAN/1 usa las bandas superiores y medias de la UNII, como lo hace HiperLAN/2, 802.11a y el nuevo estándar 802.11h. El nuevo estándar HiperLAN/2 soporta tasas de hasta 54 Mbps y usa todas las bandas de UNII. Por último mencionaré a la Asociación de redes de área local inalámbricas (WLANA, de sus siglas en inglés), la cual tiene la misión de educar y aumentar el conocimiento respecto al uso y disponibilidad de las WLAN y de promover estas redes en la industria en general. La WLANA es un recurso educacional para aquellos que buscan aprender más acerca de las redes inalámbricas. La WLANA también puede ayudar si se esta buscando un producto o servicio en específico para redes inalámbricas."[92]

[91] Ibídem P. 27 – 28.
[92] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 225 – 226.

Es importante mencionar que también existe la Asociación de Datos Infrarrojos (IrDA) que creo un estándar para comunicación de datos vía luz infrarroja, la cual es usada en calculadoras, impresoras, algunas conexiones edificio a edificio y otros dispositivos. Del mismo modo también existe otro estándar llamado OpenAir creado por el Foro de Interoperabilidad de WLAN (WLIF, de sus siglas en inglés), pero esta dejando de tener soporte de parte de la comunidad y los fabricantes por lo cual está por desaparecer.

HomeRF

"El estándar HomeRF tiene sus raíces en el Teléfono inalámbrico digital mejorado (*Digital Enhanced Cordless Telephone, DECT*, por sus siglas en inglés). Esto explica porque el estándar HomeRF puede transportar el tráfico de voz con la calidad de llamadas telefónicas normales, y de hecho está tomando un camino opuesto al de los estándares 802.11 y BlueTooth, lo que significa ir de voz a datos.

El estándar HomeRF utiliza una combinación de CSMA/CD para los datos en paquetes y TDMA para el tráfico de voz y video, con el fin de optimizar el flujo del tráfico sobre una base de prioridad.

La capa física utiliza la Manipulación por frecuencia (*Frequency Shift Keying, FSK*, por sus siglas en inglés) para proporcionar velocidades en bits variables de entre 800 Kbps y 1.6 Mbps en una banda de 2.4 GHz. Se hizo un intento de impulsar un segundo estándar HomeRF denominado HomeRF2, que permite velocidades de datos de 5 y 10 Mbps pero, además de Proxim, ninguna otra compañía ha invertido en esta tecnología. En HomeRF, la disminución del ancho de banda se efectúa a través del uso de 75 canales de 1 MHz para voz y canales de datos a 1.6 Mbps. El estándar HomeRF2 usa 15 canales de 5 MHz para canales de datos a 5 y 10 Mbps. La capa física también incluye el salto en frecuencia inteligente para evitar la residencia de transmisiones en canales que están muy congestionados debido a la interferencia. Para la seguridad, HomeRF usa un cifrado de 128 bits aumentado por medio de la mejora en la seguridad original de la modulación FHSS en la capa física.

BlueTooth

BlueTooth no compite directamente con 802.11 y compite sólo de una manera superficial con HomeRF. BlueTooth tiene como propósito ser un estándar con un rango nominal de aproximadamente 1 a 3 metros. Su intención es conectar computadoras portátiles con teléfonos celulares, PDA con computadoras y otros dispositivos similares. Está relativamente limitado en la velocidad con aproximadamente 1.5 Mbps. El estándar BlueTooth tiene dos puntos fuertes:

a) *Tamaño*.- El factor de la forma que ofrece le permite conectarse en relojes de mano, PDA y otros dispositivos electrónicos pequeños en los que el tamaño es un criterio de diseño importante.
b) *Ahorro de energía*.- BlueTooth usa 30 microamperes,

En términos de seguridad, BlueTooth cuenta con un método de cifrado; además usa un esquema de saltos FHSS de 1600 saltos por segundo y con su rango de 1 a metros, ocasionará que sea muy difícil interferir la comunicación a distancia."[93]

BlueTooth usa la banda de los 2.4 GHz y usa tecnología de salto en frecuencia. El nuevo borrador IEEE 802.15 para redes de área personal inalámbricas incluye especificaciones para BlueTooth. Los dispositivos BlueTooth operan en tres clases de poder de transmisión: 1 mW, 2.5 mW y 100 mW. Con el uso de la segunda clase, se pueden alcanzar distancias de operación de hasta 10 metros.

IEEE 802.11

Anteriormente ya he mencionado ampliamente el estándar IEEE 802.11 y esto es debido a que dicho estándar es la base para mi trabajo. Lo he decidido así por que este estándar es quizá el más importante en su área ya que la mayoría de los proveedores principales de tecnología han enfocado sus esfuerzos para desarrollarlo y cuente con la mayor parte

[93] Reid, Neil. Op. Cit. P. 34 – 36.

del mercado de las redes inalámbricas. Este estándar cuenta con una serie de variantes por lo que debe ser analizado con adicional detalle. El estándar 802.11 fue el primer estándar en describir la operación de las redes inalámbricas. Este estándar contiene todas las tecnologías de transmisión disponibles incluyendo el Espectro Disperso de Secuencia Directa (DSSS), Espectro Disperso en Salto en Frecuencia (FHSS) e Infrarrojo. El IEEE adoptó el estándar 802.11 IEEE en 1997 y se convirtió en el primer estándar WLAN, principalmente controla las Capas 1 y 2 de la pila de protocolos OSI.

Versión	Frecuencia de Portadora	Técnicas de Propagación	Máxima Velocidad de datos	Resumen
802.11	2.4 – 2.485 GHz	FHSS, DSSS	2 Mbps	Fue el primer estándar WLAN. La mayoría de los dispositivos solo usaban FHSS
802.11a	5.1 – 5.2 GHz 5.2 – 5.3 GHz 5.7 – 5.8 GHz	OFDM	54 Mbps	La potencia máxima de transmisión es 40 mW en la banda 5.1, 250 mW en la banda 5.2 y 800 mW en la banda 5.7 para los Estados Unidos, para otros países llega a cambiar.
802.11b	2.4 – 2.485 GHz	DSSS	11 Mbps	Es el estándar que se llegaba a vender más hace algunos años. Soporta velocidades de 1, 2, 5.5 y 11 Mbps
802.11d	N/D	N/D	N/D	Múltiples dominios reguladores.
802.11e	N/D	N/D	N/D	Calidad de servicio.
802.11f	N/D	N/D	N/D	Protocolo de conexión entre puntos de acceso (*Inter Access Point Protocol, IAPP*)
802.11g	2.4 – 2.485 GHz	OFDM	54 Mbps	Provee la misma velocidad que su similar 802.11a, sin embargo ofrece compatibilidad con dispositivos 802.11b
802.11h	N/D	N/D	N/D	Selección dinámica de frecuencia (*Dynamic Frequency Selection, DFS*)
802.11i	N/D	N/D	N/D	Seguridad

Tab. A1.1. Versiones de estándares IEEE 802.11

Las diferentes versiones del estándar y algunas breves características de cada una, se encuentran enlistadas en la Tabla A1.1.

APLICACIONES DE LAS WLAN

El mercado de las redes inalámbricas se encuentra en constante evolución, las recientes tecnologías hacen que cada vez más y más usuarios empiecen a tomar más interés en este tipo de comunicaciones. La gran diferencia entre el mercado de las redes y el de las redes inalámbricas es el crecimiento, en donde las segundas han tenido un rotundo éxito. Todo esto debido a la gran capacidad de flexibilidad que proveen las redes inalámbricas. A pesar de que estas redes tienen una gran variedad de aplicaciones solo mencionaremos algunas a modo de ejemplo para poder ver sus cualidades y beneficios.

Rol de Acceso

"Las redes inalámbricas LAN son más usadas en el papel de capa de acceso, es decir son usadas como un punto de entrada a una red cableada. Las WLAN son redes en la capa de liga de datos. Debido a la variación de velocidad, el ruido, y otros factores, las redes inalámbricas no son implementadas típicamente como parte de distribución o núcleo en otras redes. Por supuesto, en pequeñas oficinas no hay diferencia entre las capas de distribución, núcleo o acceso de la red. El núcleo de una red debe ser muy rápido y muy estable, capaz de manejar tremendas cantidades de tráfico con poca dificultad y sin pérdidas de tiempo. La capa de distribución de la red debe ser rápida, flexible y confiable. Las redes WLAN generalmente no cumplen con estas características para una solución en empresas.

Las WLAN ofrecen una solución específica a un problema complicado: la movilidad. Sin duda, las redes inalámbricas resuelven una multitud de problemas a corporaciones y también a usuarios en hogares, pero todos esos problemas apuntan a la necesidad de libertad de los cables de datos. Las soluciones celulares han estado disponibles desde hace un tiempo, ofreciendo a los usuarios la capacidad de moverse

mientras permanecen conectados, a bajas velocidades y precios muy elevados. Las WLAN ofrecen la misma flexibilidad sin esas desventajas. Las redes inalámbricas son rápidas, baratas y pueden ser localizadas donde sea.

Extensión de red

Las redes inalámbricas pueden servir como una extensión de una red cableada. Puede haber casos donde extender la red requiera instalar cableado adicional que es prohibitivo por los costos. Se descubrirá que contratar instaladores de cables y electricistas para construir una nueva sección de espacio de oficina para la red costará decenas o miles de dólares. O en el caso de grandes almacenes, las distancias pueden ser muy grandes para usar cables, de tal modo que habría que optar por nuevas opciones, como instalar equipos adicionales como switches o repetidores. Las redes inalámbricas pueden ser fácilmente implementadas para proveer conectividad a áreas remotas dentro de edificios. Ya que hay poco cableado requerido para instalar una red inalámbrica, los costos del cableado y de contratar instaladores del mismo pueden ser eliminados por completo. En la fig. A1.1 se muestra un ejemplo de este tipo de aplicaciones.

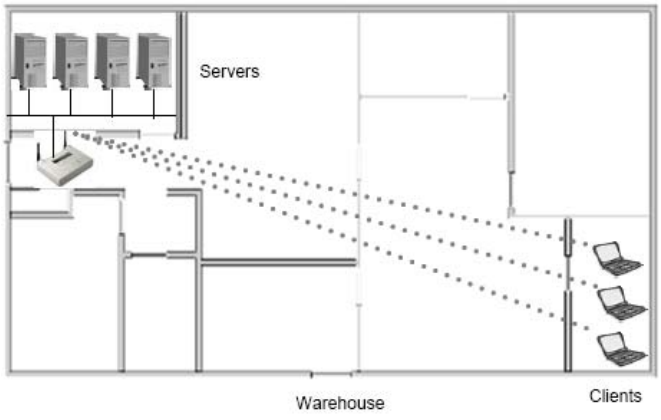


Fig. A1.1 Extensión de red con WLAN

Conectividad Edificio a Edificio

En una universidad o un pequeño ambiente como lo son dos edificios adyacentes, quizá hay necesidad de que los usuarios de la red de diferentes edificios tengan acceso directo

a la misma red de computadoras. En el pasado, este tipo de acceso y conectividad era resuelto tendiendo cables debajo del suelo de un edificio a otro, o rentando líneas muy caras de una compañía de teléfonos local. Usando tecnología inalámbrica, el equipo puede ser instalado fácil y rápidamente para permitir a dos o más edificios ser parte de la misma red. Con las antenas apropiadas, cualquier número de edificios pueden ser ligados en la misma red. Sin duda existen limitaciones al usar la tecnología WLAN, como lo hay en cualquier solución de conectividad de datos, pero la flexibilidad, velocidad y el costo que las WLAN introducen al administrador de la red la hacen indispensable.

Existen dos tipos diferentes de conectividad de edificio a edificio. La primera es llamada punto a punto (PTP), y la segunda es llamada punto a multipunto (PTMP). Los enlaces punto a punto son conexiones inalámbricas entre solamente dos edificios, como se ilustra en la fig. A1.2. Las conexiones PTP casi siempre usan antenas semidireccionales o antenas altamente direccionales al final del enlace.

Los enlaces punto a multipunto son conexiones inalámbricas entre tres o más edificios, típicamente implementados en un radio cúbico o en un modo topológico de estrella, donde un edificio es el punto de foco central de la red. Este edificio central debe tener el núcleo de la red, conectividad a Internet y el conjunto de servidores. Los enlaces punto a multipunto usualmente usan antenas omnidireccionales en el edificio concentrador central y antenas semidireccionales en los edificios que se encuentran en el radio de transmisión.

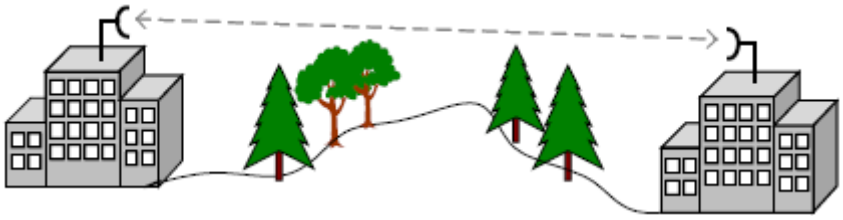


Fig. A1.2 Conectividad edificio a edificio

Entrega de datos de última milla

Los proveedores de servicios de Internet inalámbrico (WISP) están tomando ventaja en los recientes avances en la tecnología inalámbrica para ofrecer servicios de entrega de datos de última milla a sus clientes. *Última milla* se refiere a la infraestructura de comunicación – alámbrica o inalámbrica – que existe entre la oficina central de la compañía de telecomunicaciones o la compañía de cable y el usuario final. Actualmente las compañías tienen su propia infraestructura de última milla, pero con el interés de todos en la tecnología inalámbrica, los WISP están creando su propio servicio de entrega de datos de última milla.

Considere el caso donde tanto las compañías de cable o de telecomunicaciones están encontrando dificultades de expansión de sus redes para ofrecer conexiones de banda ancha en casas y empresas. Si usted vive en un área rural, existe la posibilidad de que no pueda tener acceso a conexiones de banda ancha. Es mucho más barato para los WISP ofrecer acceso inalámbrico a esas localidades remotas. Los WISP tienen también un conjunto de retos a superar. Existen problemas con los techos de las casas, con los árboles, montañas, rayos, torres y muchos otros obstáculos para la conectividad. Este tipo de servicio se ilustra en la fig. A1.3.

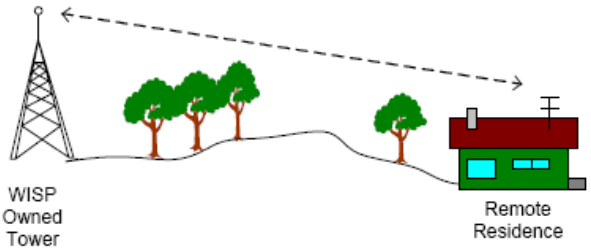


Fig. A1.3 Servicio de última milla

Movilidad

Como una solución en la capa de acceso, las WLAN no pueden reemplazar a las redes cableadas en términos de tasa de datos. Un ambiente inalámbrico usa conexiones intermitentes y tiene altas tasas de error, lo que conlleva a un ancho de banda limitado.

Como resultado, los protocolos de aplicaciones y de mensajes diseñados para el mundo alámbrico a veces operan pobremente en un ambiente inalámbrico. Las expectativas inalámbricas de los usuarios finales y de los gerentes de Tecnologías de la Información están puestas por el rendimiento y comportamiento de sus redes alámbricas. Lo que realmente ofrecen las redes inalámbricas es un incremento en la movilidad a cambio de velocidad y calidad de servicio.

En los almacenes, las redes inalámbricas son usadas para rastrear la localización de almacenaje y disposición de los productos. Estos datos entonces son sincronizados en la computadora central para los departamentos de ventas y de compras. Los escáneres inalámbricos de mano están siendo comunes en organizaciones con empleados que se mueven alrededor de los almacenes, procesando órdenes e inventarios. Esto se muestra claramente en la fig. A1.4. Algunas de las más nuevas tecnologías inalámbricas permiten a los usuarios moverse de un área de cobertura a otra sin perder conectividad.

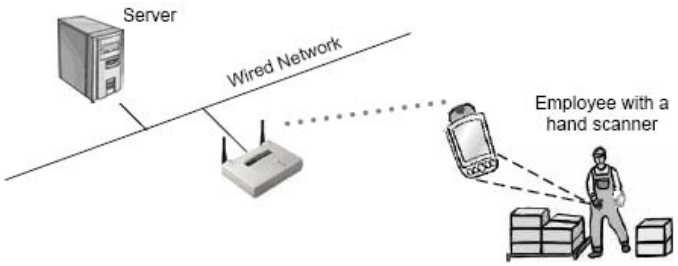


Fig. A1.4 Movilidad

Pequeñas oficinas – oficinas en casa

Muchas veces podemos tener más de una computadora en casa. De ser así, estas computadoras la mayoría de las veces están conectadas en red para poder compartir archivos, una impresora o una conexión de banda ancha. Este tipo de configuración es también usado por negocios que cuentan con pocos empleados. Estos negocios tienen la necesidad de compartir información entre los usuarios y una sola conexión a Internet para una mayor eficiencia y productividad.

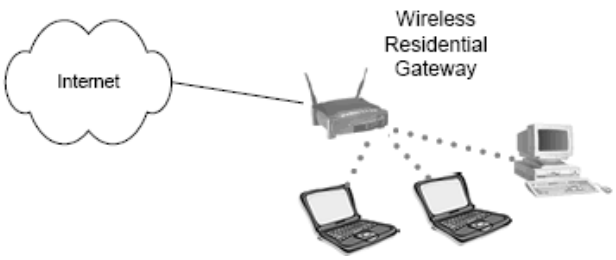


Fig. A1.5 Red inalámbrica SOHO

Para esas aplicaciones – pequeña oficina, oficina en casa (SOHO de sus siglas en ingles) – una WLAN es una solución muy simple y efectiva. La fig. A1.5 ilustra una solución típica de WLAN SOHO.

Oficinas móviles

Las oficinas o salones móviles permiten a los usuarios guardar rápidamente su equipo de cómputo y moverse a otro lugar. Debido a la sobrepoblación en los salones de clases, muchas escuelas ahora usan salones móviles. Usualmente estos salones consisten en camiones movibles grandes, que son usados mientras más estructuras permanentes son construidas. En orden de extender la red de computadoras a estos edificios temporales, cableado aéreo o debajo de tierra debe ser instalado a un alto costo. Las conexiones WLAN del edificio principal de la escuela a los salones móviles permiten configuraciones flexibles por una fracción del costo de la alternativa de cableado. En la fig. A1.6 se muestra el ejemplo de la universidad. Hay muchos grupos que pueden usar las redes movibles eficientemente. Algunos de estos incluyen a las olimpiadas, los circos, los carnavales, las ferias, festivales, compañías de construcción y otros."[94]

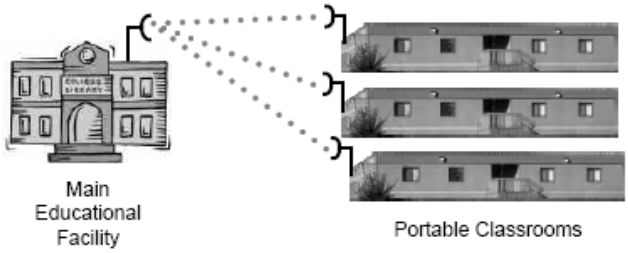


Fig. A1.6 Ejemplo de oficinas móviles

[94] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 5 – 13.

FUNDAMENTOS DE RADIOFRECUENCIA

"Para entender los conceptos fundamentales de la Frecuencia de radio (*Radio Frecuency, RF*, por sus siglas en inglés), el primer aspecto que se debe considerar es la energía electromagnética, misma que está presente en todo el planeta y es parte intrínseca de cualquier sistema eléctrico. Todos los sistemas o dispositivos eléctricos que transportan electricidad tienen un campo magnético.

El elemento fundamental del electromagnetismo es la actividad de los electrones, partículas que orbitan en un núcleo de protones y neutrones. La función esencial de RF es lograr que se muevan suficientes electrones de manera relativamente unísona dentro de la antena transmisora, y local, de modo que tengan un efecto detectable en los electrones de la antena receptora.

Los campos magnético y eléctrico se crean cuando se transfiere energía eléctrica de un punto a otro. Es este cambio en la energía lo que permite el fenómeno del magnetismo, y por ende, la propagación de frecuencias de radiodifusión. Por tanto, para transmitir información entre dos puntos, la energía no debe tener un estado constante; es decir, debe cambiar ya sea en amplitud o en frecuencia. La amplitud es la magnitud de la fuerza de una onda, en tanto que la frecuencia es el ritmo en que una onda completa traspasa un punto determinado en el espacio.

El cambio de energía se conoce como *modulación*, que esta disponible en muchos tipos, pero todos ellos conllevan cambios en la amplitud o frecuencia. El concepto prevaleciente es que RF y la conducta de los electrones son inseparables, debido a que siempre que se tiene una corriente eléctrica también aparece un campo electromagnético y viceversa. Los campos electromagnéticos afectan a los conductores. Las antenas son, en términos simplificados, conductores eléctricos de un tamaño y forma específica. Para irradiar una señal, la potencia de un transmisor debe impulsar electrones en forma alternativa hacia una antena y luego atraerlos. Un *ciclo* es el impulso y la atracción completa de electrones.

Para cada transmisión y recepción RF en particular, existirá una cantidad específica de ciclos que se realizan cada segundo. Este fenómeno se conoce como *frecuencia* y ocurre a velocidades muy altas.

Para referencia general el espectro completo de radiodifusión está dividido dentro de tres segmentos grandes de bandas de frecuencia. Las bandas bajas, normalmente inferiores a 1 GHz, se conocen como el espectro RF. La banda que se encuentra aproximadamente entre 1 GHz y 10GHz se conoce como el espectro de microondas, y la banda que se ubica entre 10 GHz y 100 GHz se denomina espectro de onda milimétrica."[95] De este modo podemos definir a las frecuencias de radio como señales de alta frecuencia de corriente alterna (AC) que son pasadas a través de un conductor de cobre y después irradiadas al aire a través de una antena. Una antena convierte o transforma la señal alámbrica a una señal inalámbrica y viceversa. Cuando la señal de alta frecuencia de AC es irradiada en el aire, se forman las ondas de radio. Estas ondas de radio salen de la antena transmisora en línea recta en todas direcciones a la vez.

Propiedades del medio inalámbrico

A continuación veremos algunos conceptos básicos que influyen sobre las señales transmitidas.

- *"Ganancia*.- La ganancia es el término que se usa para describir un incremento en la amplitud de la señal RF. La ganancia es usualmente un proceso activo, lo que significa que una fuente externa, como un amplificador RF, es usado para amplificar la señal o una antena con alta ganancia es usada para concentrar el ancho de emisión de una señal y así poder incrementar la amplitud de dicha señal. Sin embargo, también existen procesos pasivos que pueden producir ganancia. La fig. A1.7 muestra el efecto de la ganancia. Incrementar la fuerza de la señal RF puede traer resultados positivos o negativos.

[95] Reid, Neil. Op. Cit. P. 38 – 39.

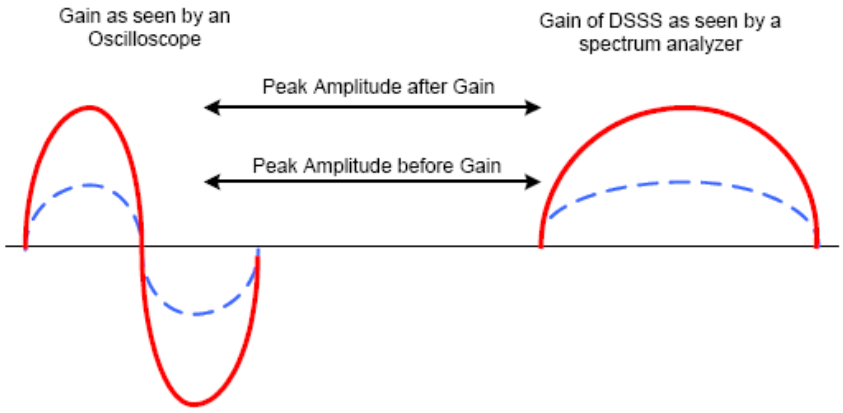


Fig. A1.7 Efecto de la ganancia

- *Pérdida*.- La pérdida describe un decremento en la fuerza de la señal. Muchos factores pueden influir en la pérdida de señal RF. La resistencia de cables y conectores causa pérdidas debido a la conversión de la señal AC a calor. Los objetos que se encuentran dentro del área de transmisión de la onda propagada pueden absorber, reflejar o destruir las señales RF. Existen muchos más factores que pueden causar pérdidas en la señal, sin embargo a veces es necesario hacer que haya pérdidas en un circuito mediante el uso de un atenuador RF. Los atenuadores RF son resistencias muy precisas que convierten la alta frecuencia AC en calor para reducir la amplitud de la señal en cierto punto del circuito. La fig. A1.8 muestra el efecto de la pérdida de señal. El ser capaz de medir y compensar la pérdida en una conexión RF o un circuito es muy importante debido a que los radios tienen un umbral de sensibilidad de recepción. El umbral de sensibilidad está definido como el punto donde un radio puede distinguir claramente una señal del ruido de fondo.

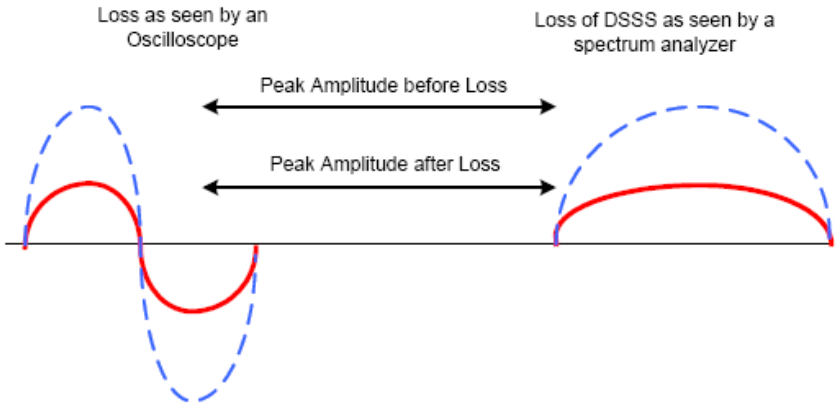


Fig. A1.8 Efecto de la pérdida de señal

- *Reflexión*.- La reflexión ocurre cuando las ondas electromagnéticas propagadas inciden en objetos que tienen dimensiones más grandes que la longitud de onda de las ondas propagadas. La reflexión puede ser causada por la tierra, edificios, paredes y muchos otros obstáculos. Si la superficie es lisa, la señal puede permanecer intacta, aunque existen un poco de pérdidas debido a la absorción y a la dispersión de la señal. La reflexión de la señal RF puede causar serios problemas a las WLAN. La reflexión de la señal principal debido a muchos objetos en el área de transmisión es conocida como *multitrayectorias* (*multipath*). Las multitrayectorias pueden traer severos efectos adversos en una WLAN, como es la degradación o cancelación de la señal principal y causar huecos o vacíos en el área de cobertura RF. Superficies como lagos, techos y puertas metálicas y otras pueden causar una reflexión severa y por lo tanto multitrayectorias. Una reflexión de esta magnitud no es recomendable y típicamente requiere funcionalidad especial (*diversidad de antenas*) dentro del hardware de las WLAN para compensar las pérdidas. La fig. A1.9 muestra una señal reflejada.

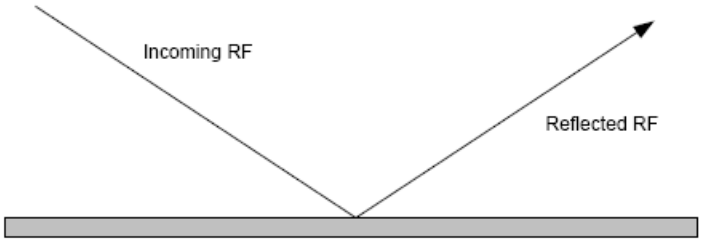


Fig. A1.9 Efecto de reflexión

- *Refracción*.- La refracción describe el encurvamiento de una onda de radio conforme ésta pasa a través de un medio de diferente densidad. Cuando una onda RF pasa por un medio muy denso (como una estanque de aire frío sobre un valle), la onda será doblada o curveada tal que su dirección cambia. Cuando la onda pasa a través de tal medio, parte de la onda será reflejada de la trayectoria deseada y otra parte será curveada a través del medio hacia otra dirección como se muestra en la fig. A1.10. La refracción puede convertirse en un serio problema en enlaces RF de larga distancia. Conforme las condiciones atmosféricas cambien, las ondas RF pueden cambiar su dirección, dirigiendo la señal fuera del blanco deseado.

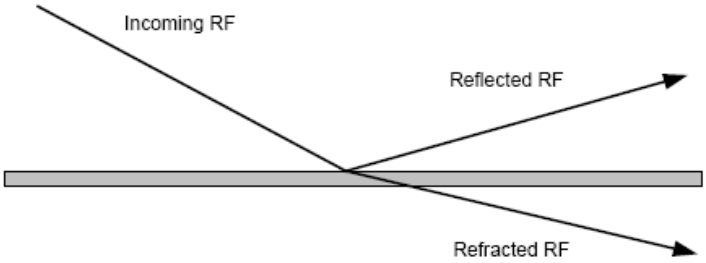


Fig. A1.10 Efecto de refracción

- *Difracción*.- La difracción ocurre cuando la trayectoria de radio entre transmisor y receptor está obstruida por una superficie que tiene irregularidades afiladas o de otro modo una superficie áspera. A altas frecuencias, la difracción como la reflexión, depende de la geometría del objeto que está obstruyendo y de la amplitud, fase y polarización de la onda incidente en el punto de difracción. La difracción describe una onda curveando alrededor de un obstáculo mientras que la refracción describe una onda curveando a través de un medio. Si el objeto es muy grande o demasiado dentado, la onda quizá no haga curva, sin embargo puede ser bloqueada. La difracción es el alentamiento del frente de la onda en el punto donde este frente golpea un obstáculo, mientras que el resto del frente de la onda mantiene la misma velocidad de propagación. Este fenómeno esta ilustrado en la fig. A1.11.

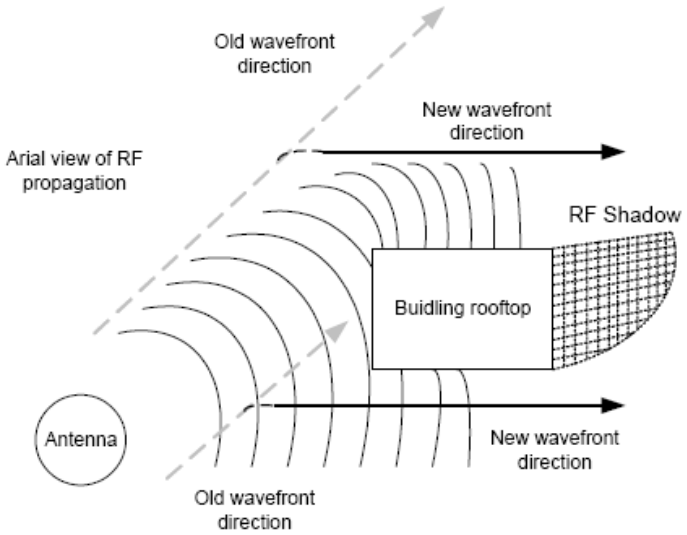


Fig. A1.11 Efecto de difracción

- *Dispersión*.- La dispersión ocurre cuando el medio en el que viajan las ondas consiste de objetos con dimensiones más pequeñas comparadas a la longitud de onda de la señal y el número de obstáculos por unidad de volumen es grande. La dispersión puede tener lugar de dos maneras principalmente. Primero, la dispersión puede ocurrir cuando la onda golpea una superficie irregular y es reflejada en muchas direcciones simultáneamente. Este tipo de dispersión produce muchas reflexiones de poca amplitud y destruye la señal RF principal. Cuando ocurre la dispersión en esta manera, la degradación en la señal RF puede ser significante al punto de interrumpir la comunicación intermitentemente o causar la pérdida total de la señal. Este efecto esta claramente ilustrado en la fig. A1.12. Segundo, la dispersión puede ocurrir cuando una onda de señal viaja a través de partículas en el medio como lo es el alto contenido de polvo en el aire. En este caso, las ondas RF son reflejadas individualmente en una escala muy pequeña de partículas diminutas.

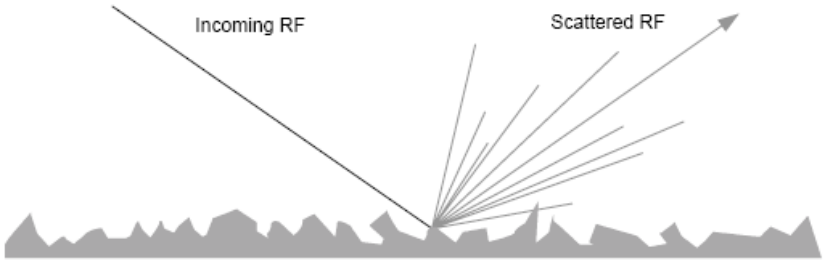


Fig. A1.12 Efecto de dispersión

- *Absorción*.- La absorción ocurre cuando la señal RF golpea un objeto y es absorbida dentro del material del objeto, de tal manera que la señal no pasa, no se refleja o no viaja alrededor de dicho objeto."[96] Este hecho se muestra en la fig. A1.13.

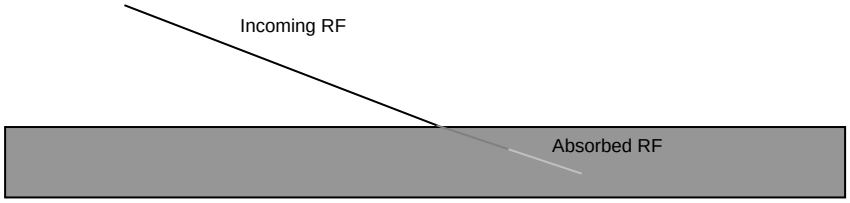


Fig. A1.13 Efecto de Absorción

[96] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 27 – 33.

EL ESTANDAR IEEE 802.11

Introducción

Los estándares de redes generalmente especifican las características para las capas físicas y de control de acceso al medio (MAC), o al menos así sucede para 802.11. El estándar IEEE 802.11 tiene diferentes variaciones de acuerdo a lo que se necesite, pero en general todas estas variaciones trabajan para redes inalámbricas de área local y de área metropolitana. La capa MAC en este estándar está diseñada para soportar diferentes unidades de la capa física, de acuerdo a la disponibilidad del espectro disponible que sea adoptado. La primer versión del estándar, solo contemplaba tres unidades para la capa física: dos unidades de radio, ambas operando en la banda de los 2.4 GHz y una unidad de banda base infrarroja. Al principio solo se contemplaba para las unidades de radio la tecnología FHSS, pero posteriormente se contempló DSSS. En general el estándar 802.11 y todas sus variantes son parte de la familia de estándares para redes de área local y metropolitana que son los estándares 802. La relación existente entre 802.11 y los demás estándares se muestra en la fig. A1.14.

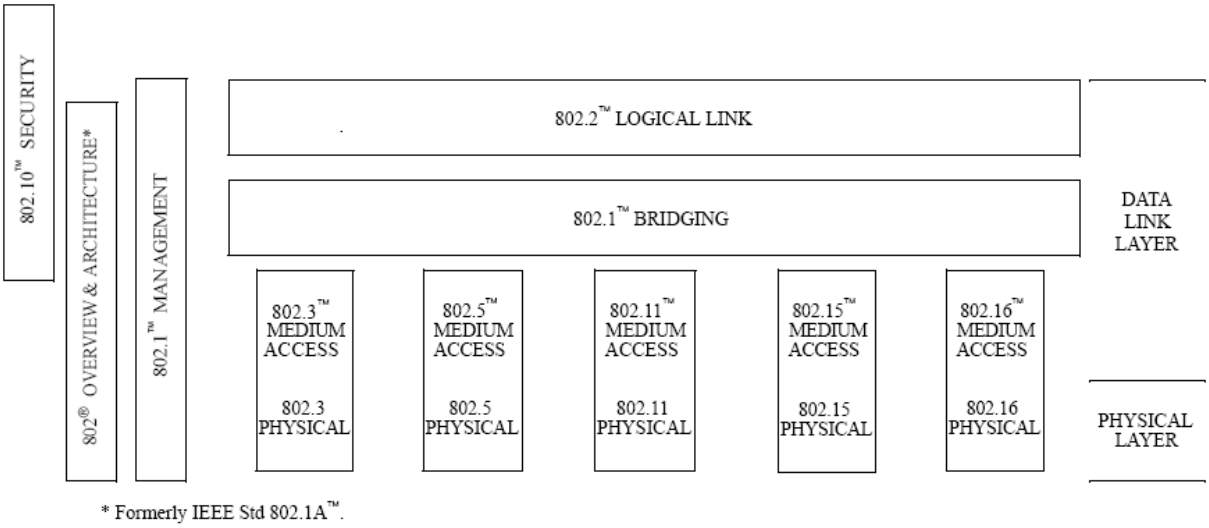


Fig. A1.14 Familia de estándares IEEE 802

Los estándares mostrados en la fig. A1.14, se describen brevemente a continuación:

- "IEEE Std 802.- *Perspectiva y arquitectura*. Este estándar provee una perspectiva de la familia de estándares IEEE 802.

- IEEE Std 802.1B y 802.1k [ISO/IEC 15802-2].- *Administración LAN/MAN*. Define una arquitectura OSI compatible de administración, servicios y elementos de protocolos para usarse en un ambiente LAN/MAN para realizar administración remota.

- IEEE Std 802.1D.- *Puenteo de Control de Acceso al Medio (MAC)*. Especifica la arquitectura y el protocolo para la interconexión de LANs IEEE 802 debajo del límite de servicios de MAC.

- IEEE Std 802.1E [ISO/IEC 15802-4].- *Protocolo de carga del sistema*. Especifica un conjunto de servicios y protocolos para esos aspectos que conciernen a la administración con la carga de sistemas en LANs IEEE 802.

- IEEE Std 802.1F.- *Definiciones comunes y procedimientos para información de administración de IEEE 802*.

- IEEE Std 802.1G [ISO/IEC 15802-5].- *Puenteo del Control de Acceso al Medio (MAC) remoto*. Especifica extensiones para la interconexión, usando tecnologías de comunicación de sistemas que no son LAN, de LANs IEEE 802 geográficamente separadas debajo del nivel del protocolo de control de enlace lógico.

- IEEE Std. 802.1H [ISO/IEC TR 11802-5].- *Práctica recomendada para puenteo de Control de Acceso al Medio (MAC) de Ethernet V2.0 en redes de área local IEEE 802.*

- IEEE Std. 802.1Q.- *Redes de Área Local Puenteadas Virtualmente*. Define una arquitectura para LANs puenteadas virtualmente, los servicios proveídos en este tipo de redes, y los protocolos y algoritmos involucrados en la provisión de esos servicios.

- IEEE Std. 802.2 [ISO/IEC 8802-2].- *Control de Enlace Lógico*.

- IEEE Std 802.3.- *Método de acceso CSMA/CD y especificaciones de la capa física*.

- IEEE Std. 802.5 [ISO/IEC 8802-5].- *Método de acceso Token Ring y especificaciones de la capa física*.

- IEEE Std. 802.10.- *Estándar para la seguridad de LAN interoperables (SILS)*. Actualmente aprobado: Intercambio de datos seguro (SDE).

- IEEE Std. 802.11 [ISO/IEC 8802-11].- *Subcapa de Control de Acceso al Medio (MAC) de LAN inalámbricas y especificaciones de la capa física*.
- IEEE Std. 802.15.- *Control de Acceso al Medio (MAC) inalámbrico y especificaciones para la Capa Física (PHY) para: Redes Inalámbricas de Área Personal*.

- IEEE Std. 802.16.- *Interfase aérea para sistemas fijos de acceso inalámbrico de banda ancha*."[97]

"Hacia 1993, los fundamentos para un estándar de redes inalámbricas estaban establecidos, y en junio de 1997, el estándar 802.11 del IEEE, que tenía más de seis años en el proceso de creación, fue ratificado. Este primer estándar 802.11 proporcionaba velocidades de datos de 1 y 2 Megabits por segundo (Mbps), una forma rudimentaria de cifrado de datos que tiene un nombre confuso: Privacidad equivalente al cableado (*Wired Equivalent Privacy*, *WEP*, por sus siglas en inglés), así como la transmisión a través de las tecnologías de secuencia directa y de salto de frecuencia sobre una banda de 2.4 GHz, además de rayos infrarrojos. El primer estándar 802.11 marcó el comienzo de una nueva era y estableció los fundamentos para el siguiente estándar, 802.11b que fue ratificado en 1999 y ofrece una velocidad de datos de hasta 11 Mbps."[98]

[97] IEEE Std. 802.11G™ – 2003.
[98] Cfr. Reid, Neil. Op. Cit. P. 12.

Por otra parte, a pesar de que el estándar 802.11a ha estado ratificado desde 1999, sólo fue hasta finales del 2001 que comenzaron a aparecer en el mercado los primeros productos compatibles con 802.11a. Por último durante el 2001 el estándar 802.11g fue aprobado, pero las especificaciones finales salieron durante el 2003.

El estándar 802.11 fue hecho debido a la necesidad de desarrollar las especificaciones necesarias de la capa física (PHY) y de la capa de Control de Acceso al Medio (MAC) para la conectividad de estaciones fijas, portátiles y móviles a redes de área local. De este modo se pueden definir básicamente dos propósitos primordiales de este estándar:

- Proporcionar conectividad inalámbrica a maquinaria automática, equipo o estaciones que requieran un rápido desarrollo, los cuales pueden ser portátiles, manuales o que puedan estar montados en vehículos en movimiento dentro de un área local.

- Ofrecer un estándar para usarse por cuerpos regulatorios con el fin de estandarizar el acceso a una o más bandas de frecuencia para el propósito de comunicaciones de área local.

"Otra de las especificaciones del estándar es el soporte obligatorio para transferencia de datos asíncrono así como el soporte adicional para servicios distribuidos de tiempo limitado (DTBS). La transferencia de datos asíncronos se refiere al tráfico que es relativamente insensible a retardos de tiempo como el correo electrónico y la transferencia de archivos. En cambio, el tráfico limitado en tiempo es el tráfico que esta limitado por los retardos de tiempo especificados para lograr una cantidad de servicio aceptable (QoS), ejemplo de este tipo de tráfico es la voz y el video en paquetes."[99]

"De particular interés en la especificación es el soporte de dos esquemas fundamentalmente diferentes de MAC para transportar servicios asíncronos y limitados

[99] Cfr. Crow, Brian. Op. Cit.

en tiempo. El esquema, Función de Coordinación Distribuida (DCF), es similar a las redes de paquetes tradicionales que soportan entrega de datos con mejor esfuerzo. El DCF está diseñado para transporte de datos asíncronos, donde todos los usuarios que tienen datos para transmitir tienen la misma oportunidad de acceder a la red. La Función de Coordinación Puntual (PCF) es el segundo esquema de MAC. El PCF está basado en encuestas (*polling*) que están controladas por un AP y está diseñado principalmente para la transmisión de tráfico sensible a retardos."[100]

Capa física

El estándar 802.11 especifica diferentes técnicas de propagación en la capa física, de acuerdo a la versión del estándar que se esté manejando. Anteriormente hemos analizado las diferentes técnicas existentes como son FHSS, DSSS y OFDM. A continuación analizaremos brevemente la capa física para las diferentes versiones el estándar 802.11, a excepción del 802.11g que ya se ha analizado en la tesis.

802.11 Infrarrojo (IR)

"La especificación de la técnica IR emplea transmisión difusa en el rango de longitudes de onda de 850 a 950 nm. Utiliza Modulación por posición de pulsos (PPM – Pulse Position Modulation) para transmitir datos usando radiación IR. PPM varía la posición de un pulso con el propósito de transmitir diferentes símbolos binarios. De esta manera la técnica de IR puede ser utilizada para transmitir información ya sea a 1 o 2 Mbps. La banda IR esta diseñada solo para uso dentro de edificios y opera con transmisiones no dirigidas.

Para transmitir a 1Mbps, se utilizan 16 símbolos para transmitir 4 bits de información (16-PPM), mientras que en el caso de 2 Mbps, se utilizan 4 símbolos para transmitir 2 bits de información (4-PPM). Los símbolos de datos siguen el código Gray.

[100] Méndez, Luis. Op. Cit. P. 19.

Este código tiene la propiedad que un pequeño error en la sincronización del tiempo produce un solo bit en error en la salida.

Las transmisiones IR tienen varias desventajas, estos sistemas comparten parte del espectro que utiliza el sol, lo cual lo hace práctico solo para ambientes dentro de edificios. Las lámparas fluorescentes también emiten radiaciones en el espectro IR causando degradación de la relación señal a interferencia (SIR) en los receptores, además tienen anchos de banda bajos y alcanzan rangos que raramente exceden 20m. Estas son algunas razones que hacen a los sistemas IR una opción no popular."[101]

802.11

El primer estándar ratificado para redes inalámbricas por la IEEE fue el 802.11 sin ningún otro índice. Este estándar usa técnicas de espectro extendido, en el rango de frecuencias de 2.4 GHz a 2.4835 GHz. Originalmente el primer estándar solo usaba técnicas FHSS pero posteriormente se le agregó DSSS para obtener un mejor desempeño, aunado a la alta resistencia de FHSS a las interferencias. Como ya se ha visto la interferencia tiende a cubrir más de un canal a la vez, de modo que los sistemas DSSS tienden a perder más datos debido a esta interferencia, ya que los datos se envían a través de canales secuenciales, mientras que en los sistemas FHSS se salta entre los canales con un orden no secuencial.

Debido a que el estándar 802.11 empezó manejando FHSS, en este inciso veremos ampliamente este tipo de sistemas, mientras los sistemas DSSS los analizaremos un poco más adelante.

"El FHSS es una técnica de espectro disperso que usa la agilidad de la frecuencia para esparcir los datos a través de más de 83 MHz. La agilidad de la frecuencia se refiere a la habilidad del radio para cambiar la frecuencia de transmisión abruptamente en base a la banda de frecuencias RF usable. En el caso de las redes inalámbricas de salto en

[101] Méndez, Luis. Op. Cit. P. 23.

frecuencia, la porción usable de la banda ISM de 2.4 GHz es de 83.5 MHz dado por el estándar 802.11. Un ejemplo de un sistema de salto de frecuencia simple puede verse en la fig. A1.15.

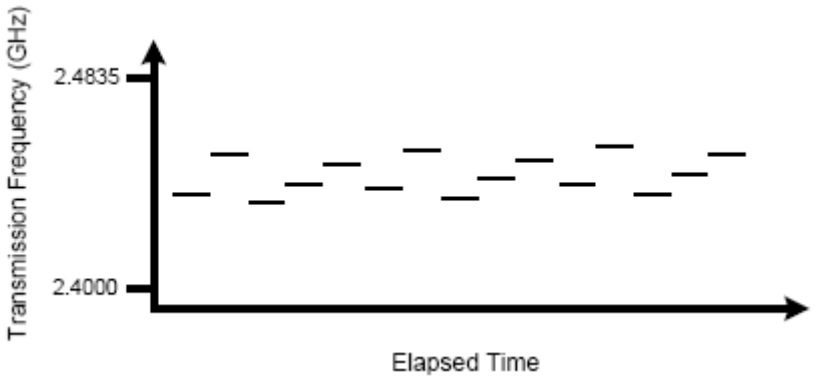


Fig. 1.15 Sistema de salto en frecuencia simple

Un sistema de salto en frecuencia operará usando un patrón de salto especificado llamado c*anal*. Los sistemas de salto en frecuencia típicamente usan el estándar 26 de la FCC de patrones de salto o un subconjunto de estos. Algunos sistemas de salto en frecuencia permiten patrones de salto personalizados para ser creados y otros incluso permiten la sincronización entre sistemas para eliminar completamente las colisiones en un ambiente co – alojado. Sin embargo es posible tener hasta 79 puntos de acceso sincronizados y co – alojados, con todos estos sistemas cada radio de salto en frecuencia requerirá una sincronización precisa con todos los otros, en orden de no interferir con otro radio de salto en frecuencia en el área. El costo de tal conjunto de sistemas es prohibitivo y generalmente no es considerado como una opción. Si se usan radios sincronizados, el costo tiende a dictar como máximo 12 sistemas co – alojados. En la fig. 1.16 se muestra un ejemplo de sistemas de salto en frecuencia co – alojados.

Si se usan radios no sincronizados, entonces pueden usarse hasta 26 sistemas co – alojados en una WLAN; este número es considerado como máximo en una WLAN con tráfico medio. Si el tráfico se incrementa significativamente o se envían paquetes muy grandes en la red, el límite práctico de sistemas co – alojados se reduce a 15. Por otra parte es importante hablar del *tiempo de vida*; cuando un sistema de salto en frecuencia transmite en cierta frecuencia, este lo debe hacer por cierta cantidad de tiempo. Este

tiempo es llamado tiempo de vida. Una vez que el tiempo de vida ha expirado, el sistema deberá cambiar a una frecuencia diferente y volver a iniciar la transmisión.

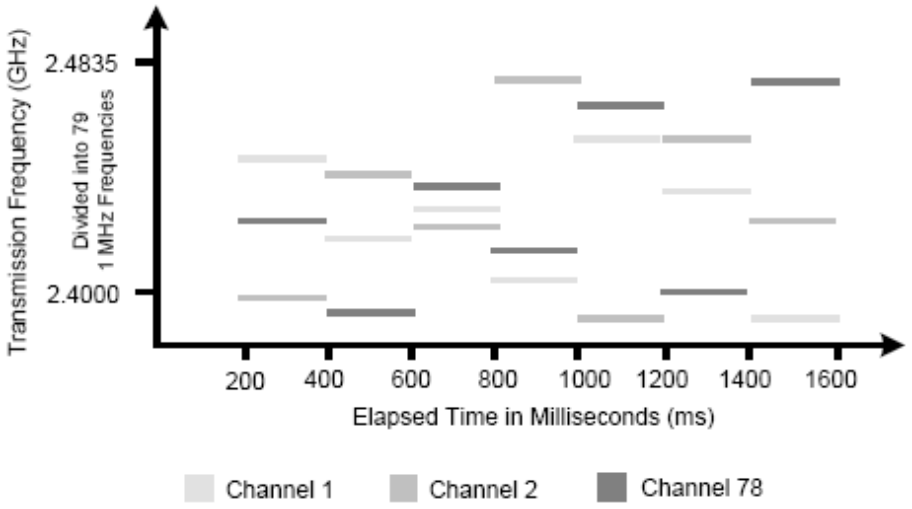


Fig. A1.16 Sistemas de salto en frecuencia co - alojados

Ahora, considerando el mismo sistema de salto en frecuencia, el tiempo de vida solo es una parte de la historia. Cuando un radio de salto en frecuencia salta de una frecuencia A a una frecuencia B, este debe cambiar la frecuencia de transmisión en una de dos formas. El radio debe cambiar a un diferente circuito sintonizado a la nueva frecuencia, o este debe cambiar algún elemento del circuito actual de tal forma que se sintonice a la nueva frecuencia. En cada caso, el proceso de cambiar a la nueva frecuencia debe estar completo antes de que la transmisión se pueda reanudar, y estos cambios toman tiempo debido a las latencias eléctricas inherentes en el circuito. Hay una pequeña porción de tiempo durante este cambio de frecuencia en el que el radio no está transmitiendo llamado *tiempo de salto*. El tiempo de salto está medido en microsegundos (μs) y con tiempos de vida relativamente largos de 100 a 200 ms, el tiempo de salto es insignificante. Un sistema típico de 802.11 con FHSS salta entre canales en 200 – 300 μs.

Con tiempos de vida muy cortos de 500 – 600 μs, como los que se usan en sistemas de Bluetooth, el tiempo de salto puede ser muy significante. Si vemos el efecto del tiempo de salto en términos de tasa de transferencia de datos, se ha descubierto que

un mayor tiempo de salto en relación al tiempo de vida, reduce la tasa de datos de bits que están siendo transmitidos.

Esto se traduce a grandes rasgos en *mayor tiempo de vida = mayor rendimiento*.

El FCC define el máximo tiempo de vida de un sistema FHSS en 400 ms por frecuencia portadora en cualquier periodo de tiempo de 30 segundos.

La organización IEEE en el estándar 802.11 define que en los sistemas FHSS habrá al menos 6 MHz de separación de frecuencias portadoras entre saltos. Por lo tanto, un sistema FHSS transmitiendo en 2.410 GHz debe saltar al menos a 2.404 si decrementa en frecuencia o 2.416 si aumenta en frecuencia."[102]

El estándar 802.11 describe sistemas que pueden operar solo a 1 o 2 Mbps aún cuando con DSSS se puede operar a más altas velocidades. Cuando este estándar era el único operando, varios fabricantes ofrecían velocidades de hasta 10 Mbps usando DSSS, sin embargo muchas veces esos equipos no podían comunicarse automáticamente con otros dispositivos originales de 802.11.

802.11b

"Aunque el estándar 802.11 fue capaz de permitir que sistemas DSSS y FHSS interoperaran satisfactoriamente, la tecnología creció más allá del estándar. Poco después de la aprobación e implementación del 802.11, las WLAN con DSSS estaban intercambiando datos hasta 11 Mbps. Pero, sin un estándar que guiara la operación de tales dispositivos, llegaron a haber problemas con la interoperabilidad e implementación. De este modo se creo el estándar 802.11b.

El IEEE 802.11b, referido como de alta tasa (*High – Rate*) y *Wi – Fi*, especifica sistemas de secuencia directa DSSS que operan a 1, 2, 5.5 y 11 Mbps. El estándar 802.11b

[102] Cfr. *Certified Wireless Network Administrator. Official Study Guide*. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A. 2003. P. 67 – 73.

no describe ningún sistema FHSS y los dispositivos que cumplen con el 802.11b, por defecto cumplen también con el 802.11, lo que significa que son reversamente compatibles y soportan tasas de datos de 2 y 1 Mbps.

Las altas tasas de datos de los dispositivos 802.11b es el resultado de usar una técnica de codificación diferente. Aunque el sistema sigue siendo un sistema de secuencia directa, la forma en que los chips son codificados (CCK en lugar de Código Barker) a través de la forma en que la información es modulada (DQPSK a 2, 5.5 y 11 Mbps y DBPSK a 1 Mbps) permite una mayor cantidad de datos para ser transferidos en la misma trama de tiempo. El estándar 802.11b para poder conseguir la compatibilidad con 802.11 usa la banda de frecuencias ISM 2.4 GHz entre 2.4000 y 2.4835 GHz."[103]

Como 802.11b basa su desarrollo en sistemas DSSS hablaremos ampliamente de estos sistemas. DSSS es un método para enviar datos en el que los sistemas de transmisión y recepción están ambos en un conjunto de frecuencias de 22 MHz de amplitud. La amplitud del canal permite a los dispositivos transmitir información a una mayor tasa de datos que los sistemas FHSS.

"DSSS combina una señal de datos en una estación de envío con una secuencia de bits con mayor tasa de datos, a lo cual se le conoce como *código de chip* (*chipping code*) o *ganancia de procesamiento*. Una alta ganancia de procesamiento incrementa la resistencia de la señal a la interferencia. El grupo de trabajo del IEEE 802.11 ha puesto los requerimientos mínimos de ganancia de procesamiento en 11.

El proceso de secuencia directa comienza con una portadora siendo modulada con una secuencia de código. El número de *chips* en el código determinará cuanto esparcimiento ocurrirá y el número de chips por bit y la velocidad del código (en chips por segundo) determinará la tasa de datos. A diferencia de los sistemas de salto en frecuencia que usan secuencias de saltos para definir los canales, los sistemas de secuencia directa usan una definición más convencional de canales. Cada canal es una

[103] Ibídem. P. 219 – 220.

banda continua de frecuencias de 22 MHz de extensión y frecuencias portadoras de 1 MHz son usadas como en FHSS. El canal 1 opera de 2.401 GHz a 2.423 GHz (2.412 GHz ± 11 MHz); el canal 2 opera de 2.406 a 2.429 GHz (2.417 GHz ± 11 MHz) y así sucesivamente. La fig. A1.17 muestra lo anteriormente citado.

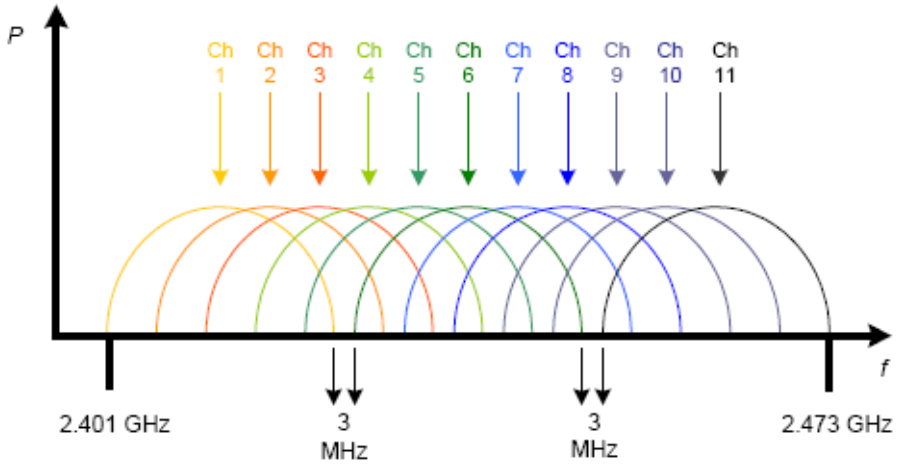


Fig. A1.17 Asignación de los canales en DSSS y su relación espectral

La tab. A1.2 muestra la lista completa de canales usados en los Estados Unidos y Europa. Es fácil observar que los canales 1 y 2 se superposicionan por una cantidad significante. Cada una de las frecuencias enlistadas en esta tabla, son consideradas frecuencias centrales. De esta frecuencia central, 11 MHz son sumados y substraídos para obtener el ancho del canal usable de 22 MHz. De este modo se ve que los canales adyacentes se superposicionarán significativamente.

El uso de sistemas DSSS con canales superposicionados en el mismo espacio físico causará interferencia entre los sistemas. Los sistemas DSSS con canales superposicionados no deben estar co – alojados porque casi siempre habrá una drástica o completa reducción en el rendimiento. Ya que las frecuencias centrales están a 5 MHz de distancia y los canales son de 22 MHz de extensión, los canales pueden ser co – alojados solo si el número de canal está a 5 de distancia, os canales 1 y 6 no se superposicionan, los canales 2 y 7 tampoco, etc. Existe un máximo de tres sistemas posibles co – alojados ya que los canales 1,6 y 11 son los únicos que teóricamente son canales no superposicionados. Esto se muestra en la fig. A1.18.

Número de Canal	Frecuencias de Canales FCC en GHz	Frecuencias de Canales ETSI en GHz
1	2.412	N/A
2	2.417	N/A
3	2.422	2.422
4	2.427	2.427
5	2.432	2.432
6	2.437	2.437
7	2.442	2.442
8	2.447	2.447
9	2.452	2.452
10	2.457	2.457
11	2.462	2.462

Tab. A1.2 Asignación de frecuencias de canales en DSSS

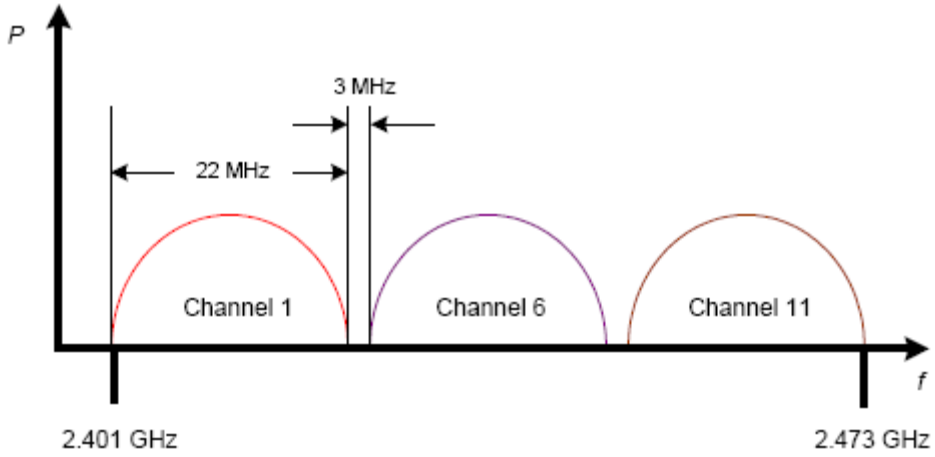


Fig. A1.18 Canales no superposicionados en DSSS

Como los sistemas de salto en frecuencia, los sistemas de secuencia directa son también resistentes a la interferencia de banda angosta debido a las características del espectro disperso. Una señal DSSS es más susceptible a la interferencia de banda angosta que una señal FHSS porque la banda DSSS es mucho más pequeña (22 MHz de extensión en lugar de la banda de 79 MHz usada por FHSS) y la información es transmitida a través de toda la banda simultáneamente en lugar de una frecuencia en un

tiempo. Con FHSS, la agilidad de la frecuencia y la extensión de la banda de frecuencia aseguran que la interferencia está solamente influenciando por un periodo corto de tiempo, corrompiendo solamente una pequeña parte de los datos. Una ventaja de FHSS sobre DSSS es la habilidad de tener muchos más sistemas de salto en frecuencia co – alojados que sistemas de secuencia directa. Desde que los sistemas de salto en frecuencia usan la agilidad de frecuencias y hacen uso de 79 canales discretos, los sistemas de salto en frecuencia tienen una ventaja de co – alojamiento sobre los sistemas de secuencia directa, los cuales solo tienen un máximo de 3 sistemas co – alojados.

Sin embargo cuando calculamos el hardware y los costos para que un sistema FHSS obtenga el mismo rendimiento que un sistema DSSS, la ventaja rápidamente desaparece. Debido a que DSSS puede tener 3 puntos de acceso co – alojados el máximo rendimiento para esta configuración deberá ser: 3 puntos de acceso x 11 Mbps = 33 Mbps. Para adquirir aproximadamente el mismo ancho de banda un sistema 802.11 con FHSS requerirá: 16 puntos de acceso x 2 Mbps = 32 Mbps

Como se puede ver, existen ventajas en la superposición en cada tipo de sistema. Si el objetivo es bajo costo y alto rendimiento, claramente la tecnología DSSS gana. Si el objetivo es mantener diferentes usuarios segmentados usando diferentes puntos de acceso en un ambiente densamente co – alojado, FHSS puede ser una alternativa viable."[104] La comparación de sistemas se puede observar en la fig. A1.19.

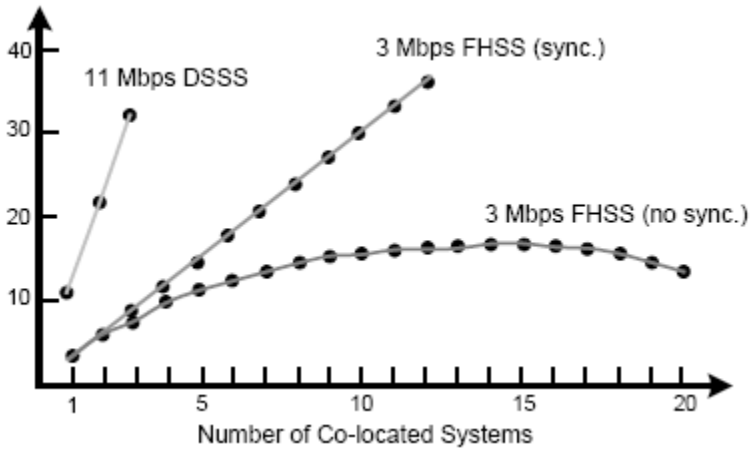


Fig. A1.19 Comparación de sistemas co – alojados

[104] Ibídem. P. 74 – 80.

802.11a

Debido a la creciente necesidad de una mayor tasa de datos para los usuarios, fue necesario crear un estándar diferente para redes WLAN, ya que hasta este punto la velocidad de estas redes era muy limitada comparada con las redes cableadas. Poco tiempo después de que el estándar 802.11b fue ratificado, el IEEE dio a conocer el estándar 802.11a. "Este estándar describe la operación de dispositivos de redes locales inalámbricas que trabajan en la banda UNII de 5 GHz. El hecho de que estos dispositivos trabajen en las bandas UNII los hace totalmente incompatibles con todos los otros dispositivos de la serie de estándares del 802.11.

Usando las bandas UNII, la mayoría de los dispositivos son capaces de adquirir tasas de datos de 6, 9, 12, 18, 24, 36, 48 y 54 Mbps. Algunos de los dispositivos que usan estas bandas han adquirido tasas de datos de hasta 108 Mbps mediante el uso de tecnología propietaria, como lo es el *doblaje de tasa*. Las altas tasas de algunos de estos dispositivos son el resultado de nuevas tecnologías no especificadas en el estándar 802.11a. El IEEE 802.11a especifica solamente tasas de datos de 6, 12 y 24 Mbps. Un dispositivo de WLAN debe soportar al menos estas tasas de datos en las bandas UNII en orden de ser un dispositivo que cumpla con el 802.11a. La tasa de datos máxima especificada en el estándar 802.11a es de 54 Mbps y los límites de la potencia de transmisión cambian de acuerdo a la región en que se este trabajando.

El estándar 802.11a especifica el uso de tecnología de Multiplexaje por División de Frecuencias Ortogonales (OFDM por sus siglas en inglés). OFDM es el secreto de cómo el 802.11a obtiene tasas de datos de hasta 54 Mbps. OFDM crea ocho canales no superposicionados de 20 MHz de extensión a través de las dos bandas más bajas de la banda UNII de 5 GHz (cuatro canales en cada uno de las dos bandas más bajas). Cada uno de estos ocho canales es subdividido en 52 subportadoras, cada una aproximadamente de 300 KHz de extensión. Cada subportadora es transmitida en paralelo, significando que estas son enviadas y recibidas simultáneamente. Una estación de recepción procesa estas 52 señales de llegada, cada una representando una fracción de los datos totales transmitidos, y esto hace que se complete la transmisión.

Debido a la gran cantidad de transmisión siendo transmitida a estas altas tasas, algunos patrones de corrección de errores fueron requeridos para prevenir la pérdida de datos. Un ejemplo de tales mecanismos es la corrección de errores hacia delante (FEC de sus siglas en inglés) el cual fue añadido al 802.11a. El FEC esencialmente envía dos copias de la información en cada transmisión, una primaria y una secundaria. Si la transmisión primaria se daña en el viaje, la estación receptora puede recrear los datos perdidos mediante el corrimiento de la transmisión secundaria a través de un conjunto de algoritmos. El impacto en el rendimiento es casi insignificante debido a las altas tasas de datos."[105]

La técnica OFDM remonta sus orígenes a la técnica de Multiplexaje por División de Frecuencias (FDM por sus siglas en inglés). La técnica FDM transmite múltiples señales simultáneamente sobre una sola ruta de transmisión y la información viaja a través de una portadora que es modulada por los datos. FDM nos ayuda exitosamente a aumentar el ancho de banda y a reducir la interferencia intersímbolo ocasionada por las multitrayectorias. Sin embargo al aumentar el ancho de banda nos encontraremos que más del 50 % del espectro disponible para transmisión se desperdicia debido a que es necesario aislar cada una de las frecuencias mediante el uso de bandas de guarda. De este modo fue necesario crear una nueva técnica de transmisión que permitiera un uso más aprovechable del espectro.

OFDM distribuye la información sobre múltiples portadoras que se encuentran espaciadas en frecuencias precisas mediante el uso de modulación QAM ya antes vista en este trabajo. Debido a que las frecuencias se encuentran espaciadas con alta precisión, cada portadora puede ser fácilmente identificable, de modo que los sistemas de demodulación no son interferidos por otras frecuencias de la que se encuentra trabajando. Todo lo anterior proporciona la ortogonalidad necesaria a FDM para poder eliminar las bandas de guarda y así incrementar el uso del espectro que esta disponible para transmisión.

[105] Ibídem. P. 220 – 222.

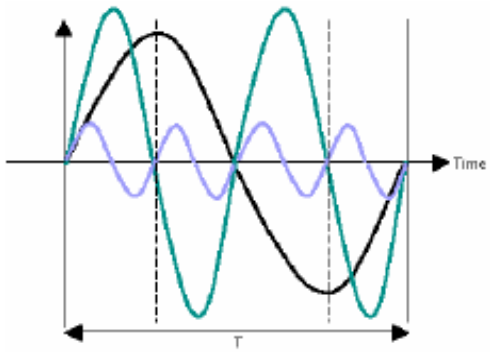


Fig. A1.20 Ortogonalidad de las diferentes frecuencias portadoras

"La fig. A1.20 muestra la ortogonalidad de las diferentes frecuencias portadoras, la frecuencia base es igual a 1 / T, donde T es igual al periodo del símbolo. Las líneas negras representan una suma de las diferentes frecuencias portadoras."[106]

Cabe señalar que en Estados Unidos la FCC aprobó el uso de 300 MHz de las bandas UNII en las que trabaja el 802.11a y que están divididas en dos porciones, primeramente de los 5.15 GHz a 5.35 GHz obteniendo 200 MHz y después entre 5.725 GHz y 5.825 GHz obteniendo los otros 100 MHz. Una vez que el estándar 802.11a usa la banda de los 5 GHz es necesario contemplar que al ser una frecuencia más alta que la que usan el resto de los estándares 802.11, es necesario transmitir con una mayor potencia para alcanzar las distancias que operan los otros estándares, ya que la relación potencia y distancia es inversa.

OFDM es solo una parte de la forma en que opera 802.11a, en realidad el método que usa es el de OFDM Codificado (COFDM, por sus siglas en inglés). COFDM usa 48 de los 52 canales para enviar los datos y los otros cuatro canales restantes se usan para el control de errores que ya hemos mencionado. Del mismo modo como hemos visto antes, las tasas de datos varían dependiendo del esquema de modulación que se este empleando, así que con un esquema de 64 QAM, el cual produce 8/10 bits por ciclo se puede alcanzar hasta 1.125 Mbps por cada canal de 300 KHz, de este modo si lo multiplicamos por los 48 canales disponibles en COFDM obtenemos una tasa de datos de 54 Mbps.

[106] Méndez, Luis. Op. Cit. P. 27.

APÉNDICE 2
TCP/IP

INTRODUCCIÓN A TCP/IP

Las redes de computadoras no servirían de mucho si no existieran los protocolos de comunicaciones y si estos no estuvieran estandarizados. En las siguientes líneas analizaremos como es que se definen los protocolos, en particular TCP/IP. También veremos resumidamente la forma en que TCP/IP está compuesto, así como el modo en que interactúan las diferentes capas que tiene este conjunto de protocolos.

"Un *protocolo de red* es un sistema de reglas comunes que ayudan a definir el complejo proceso de la transferencia de datos. Los datos viajan de una aplicación en una computadora, por el hardware de red, a través del medio de transmisión al destino correcto, y hacia arriba por el hardware de red de la computadora destino a la aplicación de recepción.

Los protocolos de TCP/IP definen el proceso de comunicación de red y definen como una unidad de datos debe verse y que información debe contener de tal modo que la computadora receptora pueda interpretar el mensaje correctamente. TCP/IP y sus protocolos relacionados forman un sistema completo que define como los datos deben ser procesados, transmitidos y recibidos en una red TCP/IP. Un sistema de protocolos relacionados, como lo son los protocolos de TCP/IP, es llamado un *conjunto de protocolos*. Es importante hacer notar la siguiente distinción en cuanto a TCP/IP:

- Un *estándar TCP/IP* es un sistema de reglas que definen la comunicación en redes TCP/IP.

- Una *implementación TCP/IP* es un componente de software que realiza las funciones que habilitan a la computadora para participar en una red TCP/IP.

El propósito del estándar TCP/IP es asegurar la compatibilidad de todas las implementaciones de TCP/IP sin importar la versión o el fabricante."[107]

"El nombre TCP/IP proviene de dos de los protocolos más importantes de la familia de protocolos *Internet*, el *Transmisión Control Protocol* (*TCP*) y el *Internet Protocol* (*IP*).

La principal virtud de TCP/IP estriba en que está diseñado para enlazar computadoras de diferentes tipos, incluyendo PCs, mini y mainframes, que ejecuten sistemas operativos distintos, sobre redes de área local y redes de área extensa y, por tanto permite la conexión de equipos distantes geográficamente. Otro gran factor que ha permitido su expansión es la utilización de TCP/IP como estándar de Internet.

TCP/IP fue desarrollado en 1972 por el Departamento de Defensa de los Estados Unidos, ejecutándose en ARPANET. Posteriormente, una red dedicada exclusivamente a aspectos militares denominada *MILNET* se separó de ARANET. Fue el germen de lo que después constituiría Internet."[108]

"Dos cualidades importantes de TCP/IP que se proveyeron al medio ambiente descentralizado ARPANET son:

- Verificación de nodo final.- Las dos computadoras que están actualmente comunicándose (llamadas nodos finales porque están a cada extremo de la cadena pasando el mensaje) son responsables de las contestaciones y la verificación de la transmisión. Todas las computadoras básicamente operan como iguales, y no hay un esquema central de comunicaciones supervisadas.

[107] Casad, Joe. Op. cit. P. 8 – 9.
[108] Raya, Jose Luis. Op. Cit. P. 59.

- Ruteo dinámico.- Los nodos están conectados a través de múltiples trayectorias, y los ruteadores eligen una trayectoria para los datos basados en las condiciones presentes."[109]

CARACTERÍSTICAS BÁSICAS DE TCP/IP

"TCP/IP enlaza las redes y la Internet juntos, sin importar el hardware y software utilizados para establecer esas redes. TCP/IP corre y se conecta a casi todo. Esta versatilidad es la razón por la cual TCP/IP es el protocolo de red más popular del mundo.

Desde el inicio, TCP/IP se diseñó para enlazar las computadoras de diferentes vendedores, como IBM y Hewlett Packard. Otros protocolos de red no son tan flexibles. Con TCP/IP, puede tenerse una computadora y saber que puede comunicarse con todas las otras computadoras. Debido a que todas las implementaciones de TCP/IP deben trabajar conjuntamente o *interoperar*, sin importar quién las creó, se puede tener varias implementaciones entre las cuales escoger. Los diversos productos pueden diferir en precio, número de opciones, rendimiento o en un sinnúmero de otras formas.

Al año 2001 más de 50 millones de computadoras corrían ya TCP/IP, en la actualidad ese número se estima que es mucho mayor. Muchas de estas son capaces de correr varios sistemas operativos. Por ejemplo, se puede instalar diferentes versiones de Microsoft Windows, Linux y Unix, todos en una misma red o hasta en la misma computadora. Se puede correr TCP/IP en cualquiera de los sistemas operativos. Aunque el sistema operativo Unix fue el primero en *incorporar* TCP/IP, la mayoría de los sistemas vienen con éste incluido hoy día."[110]

TCP/IP incluye muchas cualidades muy importantes, sin embargo se pueden resumir en básicamente seis cualidades, las cuales analizaremos a fondo a continuación.

[109] Casad, Joe. Op. Cit. P. 11.
[110] Cfr. Leiden, Candance. *TCP/IP para Dummies*. Ed. ST Editorial, Inc. Edic. 4ª. Panamá 2001. P. 11 – 12.

- *Direccionamiento lógico*.- Un adaptador de red tiene una única y permanente dirección física. La dirección física es un número que ha sido dado a la tarjeta en la fábrica. En una LAN, los protocolos consistentes de capas bajas entregan datos a través de la red física usando la dirección física del adaptador. Existen muchos tipos de redes y cada uno tiene diferentes formas de entregar los datos.

  En redes muy grandes, cada adaptador de red no puede escuchar a todos los mensajes. Conforme el medio de transmisión se vuelve más congestionado con computadoras, un esquema de direccionamiento físico no puede funcionar eficientemente. Los administradores de redes a menudo segmentan las redes usando dispositivos como los ruteadores para reducir el tráfico en la red. En redes ruteadas, los administradores necesitan una manera de subdividir la red en subredes mas pequeñas e imponer un diseño jerárquico de tal modo que el mensaje pueda viajar eficientemente a su destino. TCP/IP provee esta capacidad de subred a través del direccionamiento lógico. El *direccionamiento lógico* es una dirección configurada a través del software de red. En TCP/IP, una dirección lógica de una computadora es llamada *dirección IP*. Una dirección IP puede incluir:

  - ✓ Un número ID de red que identifica a la red.
  - ✓ Un número ID de subred que identifica a la subred en la red.
  - ✓ Un número ID de cliente que identifica a la computadora en la red.

  El sistema de direccionamiento por IP también permite al administrador imponer un esquema de numeración sensible en la red así que la progresión de la dirección refleja la organización interna de la red. En TCP/IP un direccionamiento lógico es resuelto al y desde la dirección física específica del hardware correspondiente usando los protocolos ARP y RARP.

- *Ruteo*.- El ruteador es un dispositivo especial que puede leer la información del direccionamiento lógico y encaminar los datos a través de la red a su destino. En el nivel más simple, un ruteador divide una subred local de una red más grande.

Los datos direccionados a otra computadora o dispositivo en la subred local no cruzan el ruteador y por lo tanto no se congestionan las líneas de transmisión de la red mayor. Si los datos son direccionados a una computadora fuera de la subred, el ruteador reenviará los datos acordemente.

TCP/IP incluye protocolos que definen como los ruteadores encontrarán un camino a través de la red. Otros dispositivos de red como los puentes, switches y otros pueden también filtrar y reducir el tráfico en la red. Debido a que estos dispositivos trabajan con las direcciones físicas más que con las lógicas, no pueden realizar las complejas funciones de ruteo.

- *Resolución de nombres*.- Aunque la dirección numérica IP es probablemente mas amigable al usuario que la dirección física del adaptador de red, la dirección IP está diseñada para la conveniencia de las computadoras más que para los usuarios. Por lo tanto, TCP/IP provee de una estructura paralela de nombres alfanuméricos orientados al usuario, llamados nombres de dominio o nombres DNS. Este mapeo de nombres de dominio a una dirección IP es llamado *resolución de nombres*. Computadoras especiales llamadas *servidores de nombres* guardan tablas las cuales muestran como traducir esos nombres de dominios a una dirección IP.
  Las direcciones de computadoras comúnmente asociadas con correos electrónicos o el World Wide Web están expresadas como nombres DNS. El sistema de servicios de nombres de TCP/IP provee una jerarquía de servidores de nombres que proporcionan mapeo de nombres por dominio / direcciones IP para computadoras DNS registradas en la red. Esto significa que cada día los usuarios raramente tienen que escribir o descifrar una IP actual.

  DNS es el sistema de resolución de nombres para Internet y es el método más común de resolución de nombres. Sin embargo, algunas redes TCP/IP también pueden soportar otros métodos para resolver nombres alfanuméricos a direcciones IP. Otro esquema común de resolución de nombres es el Servicio de Nombres de Internet de Windows (WINS) para resolver nombres NetBIOS de Microsoft Windows a direcciones IP.

- *Control de errores y control de flujo*.- El conjunto de protocolos de TCP/IP provee cualidades que aseguran la entrega confiable de datos a través de la red. Estas características incluyen chequeo de datos de errores de transmisión (para asegurarse que los datos que se recibieron están exactamente como fueron enviados) y acuse de recepción satisfactoria de un mensaje en la red. La capa de transporte de TCP/IP define muchas funciones de chequeo de errores, control de flujo y acuse de recibo a través del protocolo TCP. Protocolos de nivel más bajo de la capa de acceso a la red de TCP/IP también juegan parte importante en el sistema completo de control de errores.

- *Soporte a aplicaciones*.- Muchas aplicaciones de red pueden estar corriendo en la misma computadora. El software de protocolo debe proveer algunos significados para determinar cual paquete entrante pertenece a cada aplicación. En TCP/IP, esta interfase de la red a las aplicaciones es lograda a través de un sistema de canales lógicos llamados *puertos*. Cada puerto tiene un número que es usado para identificar el puerto.

  El conjunto de TCP/IP también incluye un número de aplicaciones diseñadas para ayudar con varias tareas de red.

TCP/IP está actualmente entrando en una nueva fase. Nuevas tecnologías como las redes inalámbricas, las redes privadas virtuales, y la traducción de direcciones de red están agregando nuevas complejidades que los creadores de TCP/IP no hubieran imaginado."[111]

MODELO DE CAPAS DE TCP/IP Y MODELO OSI

"TCP/IP es un sistema (o conjunto) de protocolos, y un protocolo es un sistema de reglas y procedimientos. Para la mayor parte, el hardware y software de las computadoras en comunicación lleva a cabo las reglas de las comunicaciones TCP/IP.

[111] Casad, Joe. Op. Cit. P. 12 – 17.

Un sistema de protocolos como lo es TCP/IP debe ser responsable de las siguientes tareas:

- Dividir los mensajes en trozos manejables de datos que pasarán eficientemente a través del medio de transmisión.
- Hacer la interfase con el hardware del adaptador de red.
- Direccionamiento: la computadora transmisora debe ser capaz de dar una trayectoria a los datos hacia la computadora receptora. La computadora receptora debe ser capaz de reconocer un mensaje que está supuesto a recibir.
- Rutear datos a la subred de la computadora destino, aún si la subred fuente y la subred destino son redes físicas diferentes.
- Realizar el control de errores, control de flujo y acuse de recibo: para una comunicación confiable, las computadoras transmisora y receptora deben ser capaces de identificar y corregir fallas en la transmisión y controlar el flujo de los datos.
- Aceptar datos de las aplicaciones y pasarlos a la red.
- Recibir datos de la red y pasarlos a una aplicación.

Para cumplir las tareas anteriores, los creadores de TCP/IP lo colocaron en un diseño modular. El sistema de protocolos de TCP/IP está dividido en componentes separados que funcionan teóricamente independientemente uno de otro. Cada componente es responsable de una parte del proceso de comunicación. La ventaja de este diseño modular es que permite a los fabricantes adaptar fácilmente el software de protocolo a hardware específico y sistemas operativos.

El sistema de protocolos TCP/IP está subdividido en componentes colocados en capas, cada uno de los cuales realiza funciones específicas. Este modelo, o pila, viene de los primeros días de TCP/IP, y a veces es llamado modelo de TCP/IP. Las capas oficiales del protocolo TCP/IP y sus funciones están descritas en la siguiente lista.

- *Capa de acceso a la red*.- Provee una interfase con la red física. Da formato a los datos para el medio de transmisión y direcciona los datos a la subred basada en la

dirección física del hardware. Provee control de errores para los datos entregados en la red física.

- *Capa de Internet*.- Provee direccionamiento lógico independiente del hardware de tal modo que los datos pueden pasar a través de subredes con diferentes arquitecturas físicas. Provee ruteo para reducir el tráfico y soportar el reparto a través de la interred. Relaciona direcciones físicas con direcciones lógicas.

- *Capa de transporte*.- Provee servicios de control de flujo, control de errores y acuse de recepción para la interred. Sirve como una interfase para las aplicaciones de red.

- *Capa de aplicación*.- Provee aplicaciones para la detección de problemas de la red, transferencia de archivos, control remoto, y actividades de Internet. También soporta las Interfases de Programación de Aplicación (APIs) de la red que habilitan la escritura de programas para un ambiente operativo en particular para acceder a la red."[112]

En este punto es necesario señalar que algunos autores separan las capas de TCP/IP en cinco niveles funcionales, siendo estas: capa de aplicación, capa de transporte, capa de Internet, capa de red y capa física. Sin embargo TCP/IP no considera oficialmente el nivel físico como componente específico de su modelo y tiende a agrupar el nivel físico con el nivel de red.

"Cuando el software del protocolo TCP/IP prepara una pieza de datos para su transmisión a través de la red, cada capa en la máquina transmisora suma una capa de información a los datos que será relevante a la capa correspondiente en la máquina receptora.

Por ejemplo, la capa de Internet de la computadora transmisora agrega una cabecera con información que es importante para la capa de Internet de la computadora

[112] Ibídem. P. 22 - 24

que recibe el mensaje. Este proceso es a veces conocido como encapsulamiento. En el receptor estas cabeceras son removidas conforme los datos van pasando en la pila del protocolo. A modo de recordatorio, el modelo OSI representa un intento de la ISO de estandarizar el diseño de sistemas de protocolos para red para fomentar la interconectividad y el acceso abierto a estándares de protocolos para desarrolladores de software.

TCP/IP ya estaba en vías de desarrollo cuando la arquitectura del estándar OSI apareció y, estrictamente hablando, TCP/IP no cumple con el modelo OSI. Sin embargo, los dos modelos tienen finalidades similares, y había suficiente interacción entre los diseñadores de estos estándares que estos emergieron con cierta compatibilidad. El modelo OSI ha influenciado mucho el desarrollo y crecimiento de implementaciones de protocolos, y es muy común ver la terminología OSI aplicada a TCP/IP.

La fig. A2.1 muestra la relación entre el estándar de cuatro capas de TCP/IP y el modelo OSI de siete capas. Note que el modelo OSI divide las actividades de la capa de aplicación en tres capas más: aplicación, presentación y sesión. OSI separa las actividades de la capa de interfase de red en una capa de enlace y una capa física. Esta subdivisión agregada suma algo de complejidad, pero también agrega flexibilidad a los desarrolladores mediante el uso de las capas del protocolo en servicios más específicos.

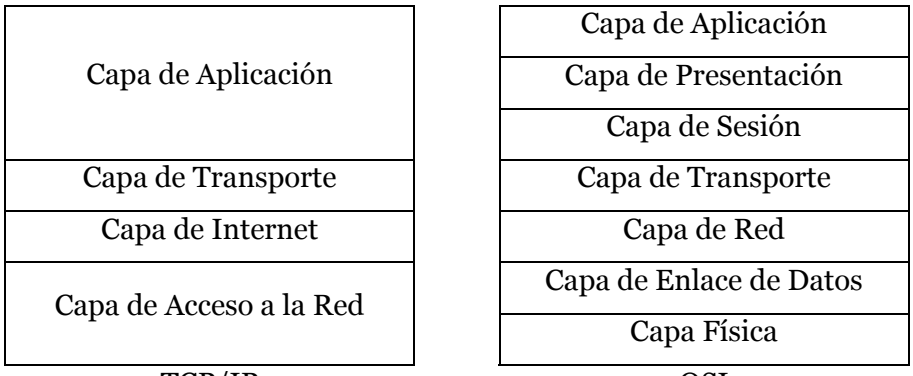


Fig. A2.1 Comparación del Modelo TCP/IP con OSI

Es importante recordar que los modelos TCP/IP y OSI son estándares, no implementaciones. Las verdaderas implementaciones de TCP/IP, no siempre trazan

fielmente los modelos mostrados anteriormente y la correspondencia representada en la fig. A2.1 es además cuestión de algunas discusiones dentro de la industria."[113]

PILA DE PROTOCOLOS DE TCP/IP

El modelo de TCP/IP cuenta con una pila de protocolos bastante amplia para poder funcionar adecuadamente. Sin embargo, no todos los protocolos existen en las diferentes implementaciones del protocolo que existen de diversos fabricantes. Es por lo anterior que en este apartado solo veremos los protocolos más importantes con que cuenta TCP/IP, sobre todo de las capas inferiores a la de aplicación, ya que es en esta capa donde se encuentra más variedad de protocolos y no se pueden analizar todos. Los protocolos que mencionaremos a continuación pertenecen a diferentes capas, empezando por los protocolos ARP y RARP que se encuentran en la parte más baja del modelo de TCP/IP, hasta llegar a dar una breve descripción de algunos protocolos de la capa de aplicación.

- *ARP* (*Protocolo de resolución de direcciones*).- "Se han desarrollado varios protocolos de enrutamiento físico. Entre ellos están el protocolo intrarredes denominado *protocolo de resolución de direcciones* (ARP: address resolution protocol), varios protocolos de puerta interior (IGP) y un protocolo de pasarela exterior (EGP)."[114] De estos solo ARP es concebido en todas las implementaciones de TCP/IP.

  "ARP es un protocolo que se utiliza para convertir las direcciones IP en direcciones físicas que puedan ser utilizadas por los manejadores.

  Para poder realizar esta conversión, existe en cada computadora un módulo ARP que utiliza una tabla de direcciones ARP, que en la mayoría de las computadoras trata

[113] Ibídem. P. 24 - 26
[114] Hallsal, Fred. Op. Cit. P. 531.

como si fuera una memoria intermedia (*caché*), de forma que la información que lleva mucho tiempo sin utilizarse se borra.

Si encuentra la correspondencia entre la dirección IP y la dirección física se procede a la transmisión. Si no la encuentra en la tabla, se genera una petición ARP que se difunde por toda la red. Si alguna de las computadoras de la red reconoce su propia dirección IP en la petición ARP, envía un mensaje de respuesta indicando su dirección física y se graba en la *tabla de direcciones ARP*."[115]

"ARP es un protocolo, un servicio y una aplicación, aunque puede no utilizar la aplicación. El RFC 826 se refiere al protocolo ARP".[116]

- *RARP* (*Protocolo de resolución de direcciones reverso*).- "RARP (Reverse address resolution protocol) se utiliza cuando, al producirse el arranque inicial, las computadoras no conocen su dirección IP.

  Requiere que exista en la red, al menos, un servidor RARP. Cuando una computadora desea conocer su dirección IP envía un paquete que contiene su propia dirección física.

  El servidor RARP, al recibir el paquete, busca en su tabla RARP la dirección IP correspondiente a la dirección física inicial indicada en el paquete y envía un paquete a la computadora origen con esta información. A diferencia del protocolo ARP que se incorpora normalmente en todos los productos TCP/IP, el protocolo RARP sólo se incorpora en unos pocos productos."[117]

  "RARP se encuentra especificado en el RFC 903. Además de ser un protocolo, RARP es también un servicio."[118]

[115] Raya, Jose Luis. Op. Cit. P. 91.
[116] Cfr. Leiden, Candance. Op. Cit. P. 63.
[117] Raya, Jose Luis. Op. Cit. P. 91 – 92.
[118] Cfr. Leiden, Candance. Op. Cit. P. 64.

- *ICMP* (*Protocolo de mensajes de control de Internet*).- "ICMP (Internet control message protocol) es un protocolo de mantenimiento/gestión de red que ayuda a supervisar la red.

  Se utiliza para poder encontrar una ruta a través de la cual los datagramas viajen por la red y alcancen su destino.

  El objetivo principal de ICMP es proporcionar la información de error o control entre nodos. La implementación de ICMP es obligatoria como un subconjunto lógico del protocolo IP.

  Los mensajes de error de este protocolo normalmente los genera y los procesa TCP/IP y no el usuario.

  Existen cuatro tipos de mensajes ICMP:

  - ✓ Mensajes de destino no alcanzable.
  - ✓ Mensajes de control de congestión.
  - ✓ Mensajes de redireccionamiento.
  - ✓ Mensajes de tiempo excedido."[119]

  "El protocolo ICMP está descrito en los RFCs 1256 y 2463. Además de ser un protocolo, ICMP también es un servicio y una aplicación, aunque la aplicación se llame *ping*."[120]

- *IP* (*Protocolo de Internet*).- "IP (Internet protocol) se encarga de seleccionar la trayectoria a seguir por los datagramas, es decir, por dónde se deben encaminar los datagramas salientes pudiendo llevar a cabo tareas de fragmentación y reensamblado.

[119] Raya, Jose Luis. Op. Cit. P. 93.
[120] Cfr. Leiden, Candance. Op. Cit. P. 64.

Este protocolo, que no es fiable ni está orientado a conexión, no garantiza el control de flujo, la recuperación de errores no que los datos lleguen a su destino. IP no se encarga de controlar que sus datagramas, que envía a través de la red, puedan perderse, llegar desordenados o duplicados. Para ello, estas opciones tendrán que ser contempladas por protocolos del nivel de transporte.

Los datagramas IP contienen una cabecera con información para el nivel IP y datos. Estos datagramas se encapsulan en tramas que, dependiendo de la red física utilizada, tienen una longitud determinada.

Cuando los datagramas viajan de unos equipos a otros pueden atravesar diferentes tipos de redes. El tamaño máximo de estos paquetes puede variar de una red a otra dependiendo del medio físico que se emplee par su transmisión. A este tamaño máximo se le denomina MTU (*Unidad Máxima de Transmisión*) y ninguna red puede transmitir ningún paquete cuya longitud exceda el MTU de dicha red.

Debido a este problema, es necesario reconvertir los datagramas IP en el formato requerido por cada una de las redes que va atravesando. Esto es lo que se denomina *fragmentación* y *reensamblado*.

La fragmentación divide los paquetes en fragmentos de menor longitud y el reensamblado realiza la operación contraria."[121]

"El núcleo de IP trabaja con las direcciones de la Internet. Cada computadora en una red TCP/IP debe tener una dirección numérica. La IP en la computadora entiende cómo y dónde enviar mensajes a estas direcciones. Todo lo anterior es cierto para ambas versiones 4 de IP y la nueva versión 6 (IPv6, originalmente llamada *IPng* para próxima generación). IPv6 es simplemente más grande y mejor.

Este protocolo se puede ver mejor en los RFCs 791 y 2460."[122]

[121] Raya, Jose Luis. Op. Cit. P. 93 – 94.
[122] Cfr. Leiden, Candance. Op. Cit. P. 61.

- *TCP* (*Protocolo de control de transporte*).- TCP es un protocolo orientado a conexión muy importante para mi trabajo, ya que esté está basado en el protocolo TCP. No diremos más de este ya que ya lo hemos analizado a fondo en la tesis. Si aún requiere más información sobre este protocolo consulte el RFC 793.

- *UDP* (*Protocolo de datagrama de usuario*).- "UDP proporciona comunicación sin conexiones entre los programas de aplicación. Permite que un programa que se encuentra en una máquina envíe datagramas a programas en otras máquinas y reciba respuestas"[123]. El RFC que describe a este protocolo es el 768.

- *FTP* (*Protocolo de transferencia de archivos*).- "Es el más utilizado de todos los protocolos de aplicación y uno de los más antiguos. Se utiliza para la transferencia de archivos proporcionando acceso interactivo, especificaciones de formato y control de autentificación."[124]

  "Tome en cuenta que FTP es también el nombre de una aplicación y un servicio. Para más información referirse a los RFCs 959 y 2640."[125]

- *HTTP* (*Protocolo de transferencia de hipertexto*).- "Es uno de los protocolos más recientes. Se utiliza para manejar la consulta de hipertexto y el acceso de datos con World Wide Web (WWW). El tráfico generado por este protocolo ha pasado, debido a la influencia de Internet, a ser muy grande."[126]

  "HTTP transfiere el lenguaje marcado de hipertexto (HTML) y otros componentes de los servidores en la Web (o intranet o extranet) al cliente explorador. Este protocolo se puede analizar mejor en el RFC 2616."[127]

[123] Comer, Douglas E. Op. Cit. P. 173.
[124] Raya, Jose Luis. Op. Cit. P. 95.
[125] Cfr. Leiden, Candance. Op. Cit. P. 64.
[126] Raya, Jose Luis. Op. Cit. P. 95 – 96.
[127] Cfr. Leiden, Candance. Op. Cit. P. 68.

- *NFS* (*Sistema de archivos de red*).- "NFS ha sido desarrollado por *Sun Microsystems Incorporated* y autoriza a los usuarios el acceso en línea a archivos que se encuentran en sistemas remotos (accede a un archivo remoto como si se tratara de un archivo local). La mayoría del tráfico *NFS* es ahora un caso especial del protocolo *RPC*.

- *NTP* (*Protocolo de tiempo de red*).- NTP permite que todos los sistemas sincronicen su hora con un sistema designado como servidor de horario."[128] "El tiempo es importante en todo tipo de aplicaciones, ya que brinda desde fechas de creación de documentos hasta información de fecha/hora de ruta de red. Este protocolo está especificado en el RFC 1305."[129]

- *SMTP* (*Protocolo de transferencia de correo sencillo*).- "SMTP es un protocolo de correo electrónico. Especifica el formato exacto de los mensajes que un cliente debe enviar desde una computadora al servidor de otro, pero no especifica como debe almacenarse el correo ni con que frecuencia se debe intentar el envío de los mensajes."[130]

  "Cuando los usuarios de computadoras ignorantes de SMTP necesitan ingresar al mundo externo, una pasarela especial de SMTP debe establecerse para esa comunicación. El RFC que está disponible para este protocolo es el 821."[131]

- *SNMP* (*Protocolo de administración de red simple*).- "SNMP sirve para administrar los sistemas de forma remota. También se puede utilizar para supervisar el tráfico de la red desde una o varias estaciones de trabajo llamadas administradores SNMP.

  Los elementos de la red que puede administrar y monitorizar son dispositivos como computadoras, puertas de enlace, encaminadotes (routers), mainframes, minicomputadoras, etc."[132]

[128] Raya, Jose Luis. Op. Cit. P. 96.
[129] Cfr. Leiden, Candance. Op. Cit. P. 67 – 68.
[130] Raya, Jose Luis. Op. Cit. P. 96.
[131] Cfr. Leiden, Candance. Op. Cit. P. 66.
[132] Raya, Jose Luis. Op. Cit. P. 96 – 97.

"Si se desea escribir una aplicación de administración de red, necesita leer mensajes de SNMP, llamados *unidades de datos de protocolo* (PDUs), para verificar cuando los dispositivos están conectados o no a la red y si todo está bien. El RFC que contiene más información de este protocolo es el 2572."[133]

- *TELNET*.- "Telnet permite que un usuario, desde una Terminal, acceda a los recursos y aplicaciones de otras computadoras. Una vez que la conexión queda establecida, actúa como intermediario entre ambas computadoras. Se fundamenta en tres principios:

  - ✓ El *concepto de terminal virtual de red* (NVT). Corresponde a la definición de cómo han de ser los datos, caracteres de control y las secuencias de los mandatos que han de circular por la red para permitir una heterogeneidad de los sistemas.
  - ✓ La *simetría entre terminales y procesos*. La comunicación puede ocurrir entre dos terminales, dos procesos o entre una terminal y un proceso.
  - ✓ *Permite que el cliente y el servidor negocien sus opciones*. La conexión comienza con una fase de negociación de opciones en la que se utilizan cuatro comandos: WILL, WONT, DO y DONT."[134]

  "Además de ser un protocolo, telnet es un servicio y una aplicación. Se puede observar mejor este protocolo mediante los RFCs 854 y 855."[135]

- *TFTP* (*Protocolo de transferencia de archivos trivial*).- "TFTP es un protocolo destinado a la transferencia de archivos aunque sin permitir tanta interacción entre cliente y servidor como la que existe en FTP. En lugar de utilizar el protocolo TCP, utiliza UDP. Sus reglas son muy sencillas, en el envío del primer paquete se establece una interacción entre el cliente y el servidor. Se empieza una numeración de los bloques. Cada paquete de datos contiene una cabecera que especifica el bloque que

[133] Cfr. Leiden, Candance. Op. Cit. P. 65 – 66.
[134] Raya, Jose Luis. Op. Cit. P. 98.
[135] Cfr. Leiden, Candance. Op. Cit. P. 65.

contiene. Un bloque de menos de 512 bytes indica que es el último y corresponde al final del archivo."[136]

El RFC que contiene más información sobre este protocolo es el 1350.

Los protocolos acabados de mencionar, son los más usuales, sin embargo existen muchos otros que no describiremos, como lo son:

- *BGP* (*Protocolo de pasarela de borde*) RFCs 1771 y 2545
- *BOOTP* (*Protocolo de arranque*) RFC 2132
- *DHCP* (*Protocolo de configuración de anfitrión dinámico*) RFC 2131
- *IMAP4* (*Protocolo de acceso de mensajes de Internet versión 4*) RFC 2060
- *IPP* (*Protocolo de impresión de Internet*) RFCs 2565 a 2568
- *IPSec* (*Protocolo de seguridad de IP*) RFC 2401
- *LDAP* (*Protocolo de acceso de directorio liviano*) RFC 2251
- *OSPF* (*Abrir ruta más corta primero*) RFC 2328
- *POP3* (*Protocolo de oficina de correos versión 3*) RFCs 1939 y 2449
- *PPTP* (*Protocolo de túnel de punto a punto*) RFCs 2637 y 226
- *RIP* (*Protocolo de información de ruta*) RFCs 1723 y 2080
- *RSVP* (*Protocolo de reservación de recurso*) RFCs 2205 y 2379
- *S – HTTP* (*Protocolo de transferencia de hipertexto seguro*) RFC 2660
- *Y otros*

INTERACCIÓN ENTRE LAS CAPAS DEL MODELO TCP/IP

Muchas veces si podemos entender la interacción entre protocolos podemos entender la interacción entre las capas de los modelos de red, ya que la mayoría de estas interacciones se llevan a cabo entre protocolos de diferentes capas o niveles de dichos modelos.

[136] Raya, Jose Luis. Op. Cit. P. 99.

Como acabamos de ver en el apartado anterior, el modelo de TCP/IP cuenta con una amplia variedad de protocolos que interactúan entre sí. Para poder comprender más adelante el funcionamiento de un protocolo en particular, para nuestro caso TCP y UDP, es necesario considerar la interacción de estos protocolos con otros protocolos de la pila de TCP/IP.

"La razón principal por la que la tecnología TCP/IP sigue siendo tan difícil de entender es que su documentación a menudo expone cada protocolo en forma independiente, sin considerar cómo operan en conjunto diversos protocolos. La documentación de un protocolo describe por lo regular la forma en que debe operar dicho protocolo; expone la acción de éste y su respuesta a mensajes sin tomar en cuenta al resto del sistema. Sin embargo, el aspecto más difícil de entender de los protocolos reside en su interacción. Cuando consideramos la operación conjunta de todos los protocolos, las interacciones producen efectos complicados y a veces inesperados. Los detalles menores que podrían parecer irrelevantes se convierten de repente en esenciales. Las interacciones entre protocolos a menudo dictan la forma en que deben ser implementados. Las estructuras de datos deben ser seleccionadas teniendo en mente a todos los protocolos."[137]

Generalidades de procesos

"La mayoría del software TCP/IP opera en computadoras que emplean un sistema operativo para administrar recursos, como los dispositivos periféricos. Los sistemas operativos proporcionan el soporte para el procesamiento concurrente. Además los sistemas operativos administran la memoria principal que mantiene en ejecución a los programas, así como el almacenamiento secundario donde residen los sistemas de archivos. El software TCP/IP reside por lo regular en el sistema operativo, donde puede ser compartido por todos los programas de aplicación que se ejecutan en la máquina. Desde luego cada llamada a TCP/IP debe operar en forma independiente, de modo que los datos transferidos por un programa no afecten a los transferidos por otro.

[137] Comer, Douglas E. Op. Cit. P. 2.

Los sistemas operativos ofrecen varias abstracciones que son necesarias para entender la implementación de los protocolos TCP/IP. Quizás el concepto más importante sea el de un *proceso* o *subproceso de control*. De manera conceptual, un proceso es un cálculo que se desarrolla en forma independiente de otros cálculos. Los sistemas operativos proporcionan mecanismos para crear nuevos procesos y terminar procesos existentes. Debido a que los procesos se ejecutan de manera independiente, pueden continuar a velocidades diferentes. Los procesos llegan a realizar operaciones que hacen que se *bloqueen* o se *suspendan*. Los procesos son especialmente útiles en el manejo de los algoritmos de tiempo de espera y retransmisión que se encuentran en muchos protocolos. El programador que diseña el software asigna a cada proceso una *prioridad*. Es sistema operativo acepta las prioridades al conceder a los procesos el uso de la CPU.

En el software de protocolos, el sistema de prioridad resulta valioso, ya que permite al programador dar prioridad a un proceso sobre otro. El diseñador debe asignar mayor prioridad al proceso que implementa el software de protocolos, obligándole a tener prioridad sobre los procesos de aplicación. Debido a que el sistema operativo maneja todos los detalles de la calendarización de los procesos, éstos no necesitan contener código que la maneje. El sistema operativo proporciona mecanismos que permiten que los procesos se comuniquen, como lo son los *semáforos de conteo*, *puertos* y *transferencia de mensajes*.

- Un *semáforo de conteo* es un mecanismo general que permite que un programa sincronice la ejecución de cálculos independientes. Los semáforos pueden usarse para proporcionar *exclusión mutua*, además de que también proporcionan sincronización al acceso a las colas.

- Un *puerto* ofrece un punto de encuentro a través del cual los procesos pueden transferir datos. Una vez creado un puerto, los procesos llaman a procedimientos para depositar o retirar elementos.

- La *transferencia de mensajes* permite a un proceso enviar un mensaje directamente a otro proceso. Es responsabilidad del programador construir el sistema de tal manera que no se pierdan mensajes. La mayoría de los protocolos especifican un tiempo máximo de espera para los acuses de recibo.

Interrupciones por el sistema de red

El hardware de interfaz de red transfiere los paquetes entrantes de la red a la memoria de la computadora e informa al sistema operativo que llegó un paquete. Para hacer esto, la interfaz de red emplea regularmente el mecanismo de interrupciones. Una interrupción hace que la CPU suspenda temporalmente su proceso normal y pase al código denominado *controlador de dispositivos*. El software del controlador de dispositivos se hace cargo de los detalles menores. El controlador de dispositivos también informa al software de protocolos que llegó un paquete y debe ser procesado. Una vez que el controlador de dispositivos termina sus tareas, regresa desde el punto de interrupción hasta el punto en que estaba ejecutándose la CPU cuando ocurrió la interrupción.

El código del controlador de dispositivos maneja la interrupción y reinicia el dispositivo para aceptar el siguiente paquete. Además el controlador de dispositivos proporciona una interfaz conveniente para que los programas envíen o reciban paquetes. En particular, permite que un proceso bloquee (espere) un paquete entrante.

Para dar lugar a la llegada aleatoria de paquetes, el sistema requiere tener la capacidad de leer paquetes desde cualquier interfaz de red. Hay varias formas de resolver este problema de espera de una interfaz aleatoria. Algunos sistemas operativos usan el mecanismo de *interrupción de software* de la computadora.

En redes como Ethernet, que incluyen en cada paquete información sobre el tipo de paquete, la rutina de interrupción usa el campo que indica el tipo del paquete entrante para determinar qué protocolo fue utilizado en el paquete.

Flujo de paquetes a protocolos de la capa de Internet

Debido a que la entrada ocurre al momento de una interrupción, el código del controlador de dispositivos no puede llamar procesos en forma arbitraria para procesar cada paquete; es necesario diseñar el sistema de manera que regrese rápidamente de una interrupción. Por lo tanto, el procedimiento de interrupción no llama directamente a IP o ICMP. Además, debido a que el sistema emplea un proceso independiente para implementar IP, el controlador de dispositivos no puede llamar a IP en forma directa. En su lugar, para sincronizar la comunicación con el sistema usa una cola junto con las primitivas funciones de transferencia de mensajes. Cuando llega un paquete que contiene un datagrama IP, el software de interrupción debe colocar el paquete en la cola y notificar al proceso IP que llegó un datagrama. Cuando el proceso IP no tiene paquetes por manejar, espera el arribo de otro datagrama. Existe una cola de entrada asociada con cada dispositivo de red; un solo proceso IP extrae los datagramas de todas las colas y los procesa.

Transferencia de datagramas IP a protocolos de transporte

Una vez que el proceso IP acepta un datagrama entrante, debe decidir adónde enviarlo para su procesamiento posterior. Si el datagrama contiene un segmento TCP, debe ir al módulo TCP; si lleva un datagrama UDP, debe ir al módulo UDP, etc. Debido a la complejidad de TCP, la mayoría de los diseños emplea un proceso independiente para manejar los segmentos TCP entrantes. Una consecuencia de tener procesos IP y TCP por separado es que deben usar un mecanismo de comunicación entre procesos cuando interactúen. Una vez que TCP recibe un segmento, usa los números de puerto de protocolo TCP para localizar la conexión a la que el segmento pertenece. Si el segmento contiene datos, TCP los agregará a un búfer asociado a la conexión y devuelve un acuse de recibo al invocador. Si el segmento entrante lleva un acuse de recibo de los datos de salida, el proceso de entrada TCP también debe comunicarse con el proceso temporizador TCP para cancelar la retransmisión pendiente.

La estructura del proceso usada para manejar datagramas UDP entrantes es muy diferente a la que se utiliza para TCP. Debido a que UDP es mucho más simple que TCP, el módulo de software UDP no ejecuta un proceso independiente. En su lugar, consta de procedimientos convencionales que ejecutan el proceso IP para manejar un datagrama UDP entrante. Estos procedimientos examinan el número de puerto de destino del protocolo UDP y lo utilizan para seleccionar una cola (puerto) del sistema operativo para los datagramas entrantes. El proceso IP deposita el datagrama UDP en el puerto correspondiente, de donde un programa de aplicación puede extraerlo.

Transferencia de datagramas a programas de aplicación

UDP realiza un demultiplexado de los datagramas de usuario entrantes, con base en el número de puerto del protocolo, y los coloca en colas del sistema operativo. Mientras tanto, TCP separa los flujos de datos entrantes y coloca los datos en búferes. Cuando un programa de aplicación necesita recibir ya sea un datagrama UDP o datos de un flujo TCP, debe acceder al puerto UDP o al búfer TCP. Debido a que cada programa de aplicación se ejecuta como un proceso independiente, debe usar las primitivas funciones de comunicación del sistema para interactuar con los procesos que implementan protocolos. Para los datos TCP entrantes, el sistema usa semáforos para controlar el acceso a los datos contenidos en un búfer TCP. La fig. A2.2 muestra todo el flujo de la información desde la red hasta las aplicaciones a través de IP.

Transferencia de mensajes a la salida

Los paquetes salientes se originan por una de dos razones. Ya sea que un programa de aplicación transfiera datos a uno de los protocolos de alto nivel que, a su vez, envía un mensaje (o datagrama) a un protocolo de nivel inferior y finalmente genera una transmisión en una red; o el software de protocolos del sistema operativo transmite información. En ambos casos es necesario enviar a la salida una trama de hardware a través de una interfaz de red en particular.

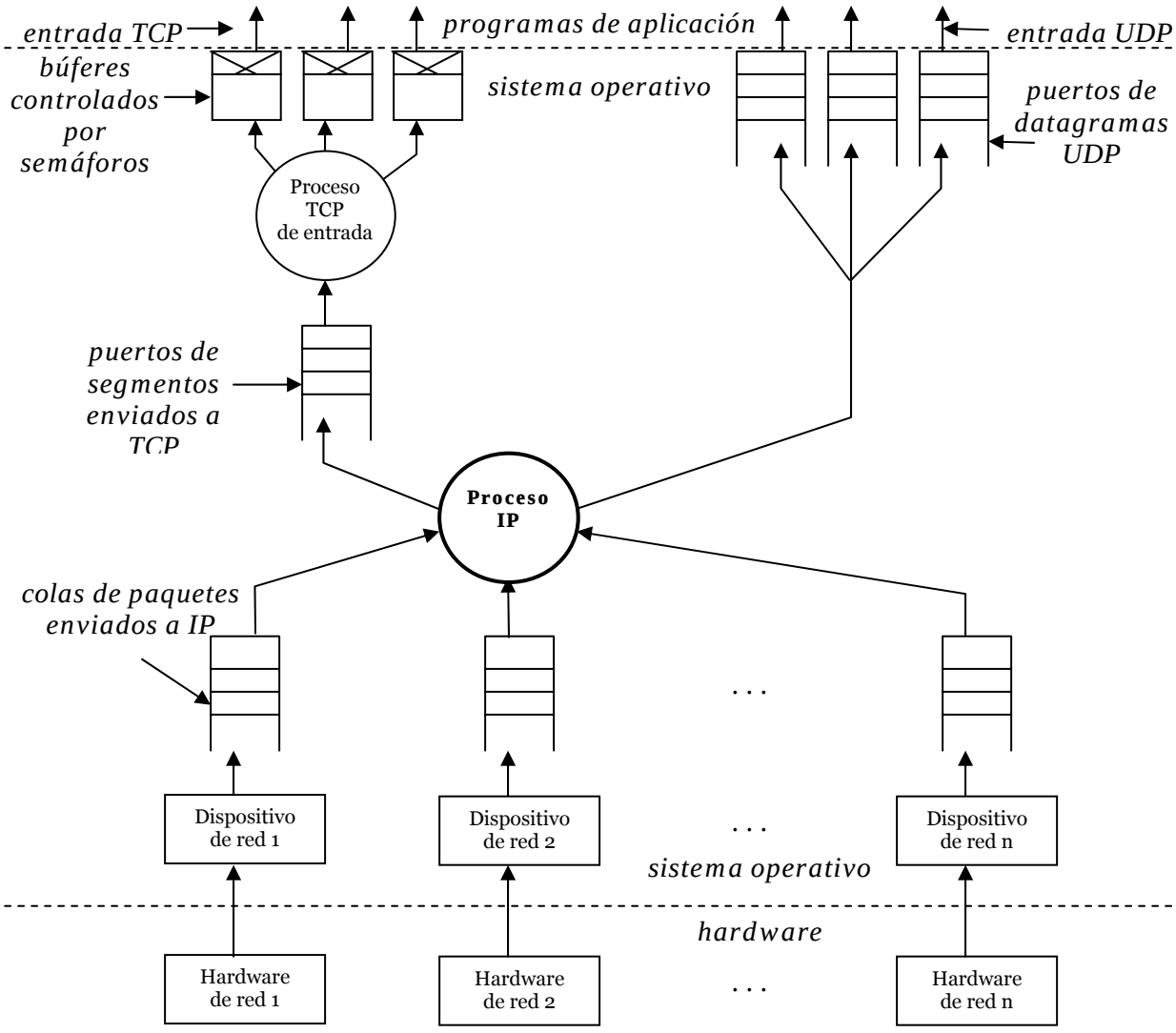


Fig. A2.2 Flujo de información de entrada

Para ayudar a aislar la transmisión de paquetes de la ejecución de los procesos que implementan programas de aplicación y protocolos, el sistema tiene colas de salida para cada interfaz de red. Las colas asociadas con los dispositivos de salida proporcionan una pieza importante del diseño. Permiten que los procesos generen un paquete, lo pongan en una cola de salida y continúen la ejecución sin esperar a que sea enviado. Mientras tanto, el hardware puede seguir transmitiendo paquetes en forma simultánea. Si el hardware está ocioso cuando llega un paquete, el proceso que realiza la salida coloca su paquete en la cola y llama a una rutina del controlador de dispositivos para iniciar el hardware. Cuando termina la operación de salida, el hardware interrumpe la CPU. El

manejador de interrupciones, que forma parte del controlador de dispositivos, saca de la cola el paquete que acaba de ser enviado. Si quedan paquetes adicionales en la cola, el manejador de interrupciones reinicia el hardware para enviar el siguiente paquete. Después el manejador de interrupciones regresa de la interrupción, permitiendo que continúe el proceso normal.

Por lo tanto, desde el punto de vista del proceso IP, la transmisión de paquetes ocurre automáticamente en segundo plano. En tanto queden paquetes en la cola, el hardware continuará transmitiéndolos. El hardware sólo necesita iniciarse cuando IP deposita un paquete en una cola vacía. Desde luego, cada cola de salida tiene una capacidad finita y puede llenarse si el sistema genera paquetes más rápido de lo que el hardware de red puede transmitirlos. Estos casos son raros, pero en casi de que ocurran, los procesos que generan los paquetes deben tomar una decisión: descartar el paquete o bloquearse hasta que el hardware termine de transmitir un paquete y libere más espacio.

Flujo de datagramas de la capa de transporte a la salida de red a través de IP

Al igual que la entrada de TCP, la salida también es compleja. Es necesario establecer conexiones, colocar los datos en segmentos y retransmitir los segmentos hasta que llegue un acuse de recibo. Una vez colocado un segmento en un datagrama, puede ser transferido a IP para su enrutamiento y entrega. Para manejar la complejidad, el software utiliza dos procesos TCP. El primero, llamado *tcpout*, maneja la mayoría de los detalles de la segmentación y transmisión de datos. El segundo, denominado *tcptimer*, maneja un temporizador, programa los tiempos de espera de retransmisión e indica a tcpout cuando debe retransmitir un segmento. El proceso tcpout utiliza un puerto para sincronizar la entrada de varios procesos. TCP está orientado a flujos, esto significa que un programa de aplicación puede enviar unos cuantos bytes de datos a la vez. En consecuencia, los elementos en el puerto no corresponden a paquetes o segmentos individuales. En su lugar, un proceso que emite datos coloca a éstos en un búfer de salida y coloca un solo mensaje en el puerto para informar a TCP que se escribieron más datos.

El proceso temporizador deposita un mensaje en el puerto cada vez que expira un temporizador y que TCP necesita retransmitir un segmento. Una vez que TCP produce un datagrama, lo pasa a IP para su entrega. Aunque es posible que dos aplicaciones en una máquina dada se comuniquen, en la mayoría de los casos, el destino de un datagrama es otra máquina. IP elige una interfaz de red a través de la cual debe enviarse el datagrama y lo transfiere al proceso de salida de red correspondiente."[138]

La fig. A2.3 muestra todo el flujo de la información desde las aplicaciones hasta la red a través de IP.

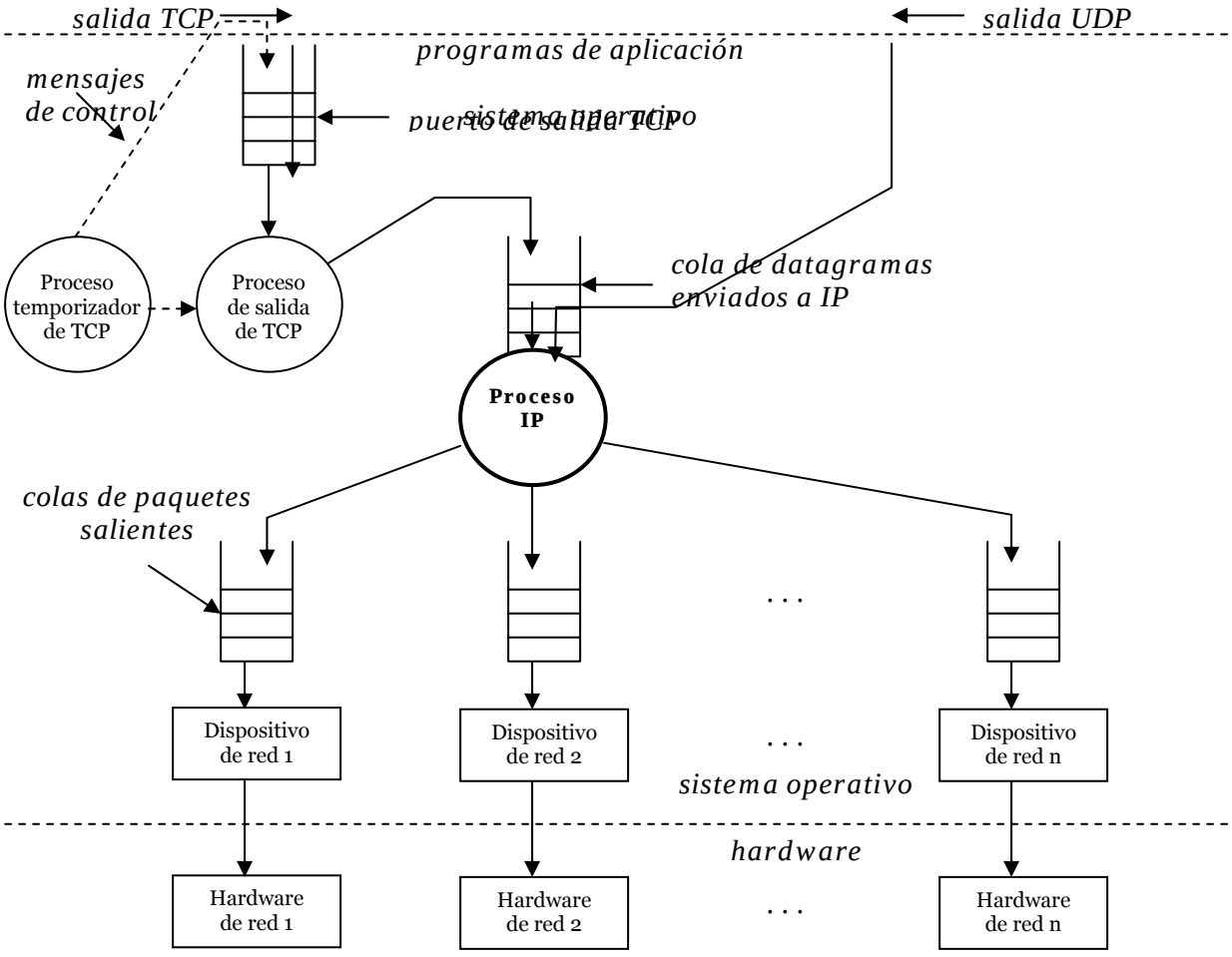


Fig. 1.22 Flujo de información de salida

[138] Ibídem. P. 7 – 21.

BIBLIOGRAFÍA Y REFERENCIAS

BIBLIOGRAFÍA

- Bandel, David. *Edición Especial Linux*. Ed. Prentice Hall. Edic. 6ª. Madrid, España, 2001.
- Casad, Joe. *Sams Teach Yourself TCP/IP in 24 Hours*. Ed. Sams Publishing. Edic. 3ª. U.S.A., 2003.
- *Certified Wireless Network Administrator*. Official Study Guide. Ed. Mc-GrawHill. Edic. 2ª. California, U.S.A., 2003.
- Comer, Douglas E. *Interconectividad de Redes con TCP/IP. Vol II*. Ed. Pearson Educación. Edic. 3ª. México, 2000.
- Coulouris, G. *Sistemas Distribuidos. Conceptos y diseño*. Ed. Pearson Educación. Edic. 3ª. Madrid, España, 2001.
- Crow, Brian. *IEEE 802.11 Wireless Local Area Networks*. Ed. IEEE Communications Magazine. Septiembre 1997.
- Deitel, Harvey. *Cómo programar en C++*. Ed. Pearson Educación. Edic. 4ª. México, 2003.
- *Diccionario de Términos de Computación*. Ed. Grupo Editorial Tomo, S. A. de C. V. México D.F., 2000.
- Halsall, Fred. *Comunicación de Datos, Redes de Computadores y Sistemas Abiertos*. Ed. Pearson Educación. Edic. 4ª. México, 1998.
- *IEEE Std. 802.11GTM* – 2003.
- Leiden, Candance. *TCP/IP para Dummies*. Ed. ST Editorial, Inc. Edic. 4ª. 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.
- Navarro, Anna. *Diccionario de Términos de Comunicaciones y Redes*. Ed. Pearson Educación, S. A. Madrid, España, 2003.
- Parker, Tim. *Teach Yourself TCP/IP in 14 days*. Ed. Sams Publishing. Edic. 2ª. U.S.A.
- Raya, Jose Luis. *TCP/IP para Windows 2000 Server*. Ed. Alfaomega. Colombia, 2001.
- Reid, Neil. *802.11 (Wi-Fi) Manual de Redes Inalámbricas*. Ed. Mc Graw Hill. México, 2004.
- The Vint Proyect. *The NS Manual*. 2001.

SITIOS DE INTERNET

- http://eia.udg.es/~atm/tcp-ip/tema_4_1_1.htm
- http://es.wikipedia.org/wiki/IEEE_802.11
- http://es.wikipedia.org/wiki/Red_de_computadoras
- http://evanjones.ca/basic-80211-stats.html
- http://galeon.hispavista.com/redeslanabedulmo/historia.html
- http://usuarios.lycos.es/janjo/janjo1.html
- http://www.cs.cornell.edu/skeshav/real/overview.html
- http://www.cse.iitk.ac.in/%7Ebhaskar/tens/
- http://www.cse.iitk.ac.in/users/sidd/projects/ns/NS_Modification.html
- http://www.cygwin.com/
- http://www.ece.rice.edu/%7Ejpr/ns-802_11b.html
- http://www.eveliux.com/articulos/estandareswlan.html
- http://www.galeon.com/tutorial/inalambrico.htm#_Toc68830816
- http://www.icir.org/
- http://www.intelligraphics.com/articles/80211_article.html
- http://www.isi.edu/conser/overview.html
- http://www.isi.edu/nsnam/ns/
- http://www.isi.edu/nsnam/ns/CHANGES.html
- http://www.isi.edu/nsnam/ns/ns-build.html
- http://www.isi.edu/nsnam/ns/ns-tests.html
- http://www.isi.edu/nsnam/vint/index.html
- http://www.isi.edu/nsnam/vint/proyect_overview.html
- http://www.isi.edu/saman/index.html
- http://www.mat.upc.es/~calveras/PAPERS/TELECOM97DOC.pdf
- http://www.monografias.com/trabajos5/redes/redes.shtml

- http://www.stetson.edu/~helaarag/Spie99.pdf
- http://www.winlab.rutgers.edu/%7Ebhaskar/ns-modifications.html
- http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin
- Robinson, Joshua. 802.11 MAC code in NS – 2 (version 2.28). http://www.ece.rice.edu/~jpr/ns/docs/802_11.html