Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

MÉTODOS PARA EL MEJORAMIENTO DEL DESEMPEÑO DEL PROTOCOLO TCP/IP SOBRE REDES INALÁMBRICAS

Iván Alfredo Morales Salazar
Asesorado por el Ing. Luis Eduardo Durán Córdova

Guatemala, marzo de 2007

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

MÉTODOS PARA EL MEJORAMIENTO DEL DESEMPEÑO DEL PROTOCOLO TCP/IP SOBRE REDES INALÁMBRICAS

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA

FACULTAD DE INGENIERÍA

POR

IVÁN ALFREDO MORALES SALAZAR

ASESORADO POR EL INGENIERO LUIS EDUARDO DURÁN CÓRDOVA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELÉCTRICO

GUATEMALA, MARZO DE 2007

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Gustavo Benigno Orozco Godinez
EXAMINADOR	Ing. Otto Fernando Andrino Gonzalez
EXAMINADOR	Ing. Gustavo Adolfo Villeda Vasquez
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

MÉTODOS PARA EL MEJORAMIENTO DEL DESEMPEÑO DEL PROTOCOLO TCP/IP SOBRE REDES INALÁMBRICAS,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Eléctrica, con fecha 18 de abril de 2005.

Iván Alfredo Morales Salazar

ACTO QUE DEDICO A:

DIOS	Que me dió la vida y que la ha guiado aún cuando no lo conocía llenandola de bendiciones y que permite que llegue al punto de poder compartir los conocimientos adquiridos para beneficio de la sociedad y de mi familia.
MIS PADRES	Héctor Alfredo y Dina Clemencia Concepción, por su amor incondicional, su ejemplo, guía y apoyo incondicional en todo momento. Este triunfo es para ustedes también.
MI ESPOSA	Heydi Xiomara Esquivel Lemus, en la que encontré compañera ideal y perfecta para mi vida y que me ha dado todo su amor, apoyo y comprensión.
MIS HIJOS	Andrea Mariel y Javier Iván, fuente de inspiración y esperanza en el futuro. Que ojalá este logro alcanzado sirva de ejemplo para su vida.
MI HERMANO	Héctor Raúl, por todo el cariño, amor y compañía que siempre me ha dado.
A MI FAMILIA	Abuelos Q.E.P.D., tíos, primos, cuñados, suegra y sobrinos, todo un manatial de cariño y amor que complementan mi existir y con quienes ha sido una dicha compartir.
A LA USAC	Por haberme brindado el conocimiento que con este acto se confirma y que me permite aportar lo mejor de mi persona a Guatemala.
A MIS MAESTROS	A todos ellos gracias, por compartir sus conocimientos y su paciencia, y en especial a Jorge Luis Galindo Arandi, quien me enseño algo más que matemáticas, me enseño a jugar ajedrez y a pensar y meditar en lo profundo de las cosas.
A TODOS MIS COMPAÑEROS Y AMIGOS	En especial a los muchados de la Delta-Estrella, Geovanni, Luis, Vicente, José Carlos, Edgar, Eduardo, Héctor y Carlos Alvarez, con quienes comparto una gran amistad, ideales y metas.

AGRADECIMIENTOS A:

A MI ASESOR Luis Eduardo Durán Córdova, por su tiempo, guía y amistad.

A MIS PADRINOS Por compartir y confirmar con su presencia este triunfo.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

ACK *Acknowledgement* o acuse de recibo o también reconocimiento.

ASCII *American standard code for information interchange*, código estadounidense estándar para el intercambio de información.

ASN.1 *Abstract syntax notation* 1, notación de sintaxis abstracta nº 1. Es una norma de representación de datos.

BSD *Berkeley Software Distribution*, distribución de software de la Universidad de Berkeley.

CDMA *Code división múltiple access*, múltiple acceso por división de código.

Caching Palabra en inglés que identifica procedimiento para el almacenamiento de copias locales de información, en medios más veloces que los originales.

Conexión Proceso en el cual dos equipos establecen comunicación y acuerdan los parámetros de esta.

CSMA/CA *Carrier sense multiple access/colission avoidance*, múltiple acceso por detección de portadora con evasión de colisiones.

DARPA *Defense advanced research project agency*, agencia para proyectos avanzados de investigación de la defensa de los Estados Unidos.

DNS *Domain name system*, sistema de nombre de dominio.

DVB *Digital video broadcasting*, emisión de video digital.

EBCDIC *Extended binary coded decimal interchange code*, código binario extendido para el intercambio de códigos decimales.

EDGE *Enhanced data rates for GSM evolution*, Tasas de transmisión mejoradas para la evolución del GSM.

FDMA *Frecuency division multiple access*, múltiple acceso por división de frecuencia.

Flujo Es una concatenación de paquetes relacionados. En el caso del TCP un flujo se refiere a una conexión entre dos puntos o direcciones IP y una pareja de puertos, en el caso del UDP se refiere a un paquete.

FTP *File transfer protocol*, protocolo de transferencia de archivos.

GPRS *General packet radio service*, servicio general de paquetes por radio.

HDLC *High level data link*, protocolo de alto nivel para enlaces de datos.

HTTP *Hypertext transfer protocol*, protocolo de transferencia de hipertexto.

IEEE *Institute of electrical and electronics engineers*, instituto de ingenieros Eléctricos y Electrónicos.

IMAP *Internet message access protocol*, protocolo de acceso a mensajes de Internet.

ISO *International standard office*, oficina internacional de estándares.

MACA *Multi access colission avoidance*, múltiple acceso con evasión de colisiones.

MIB *Management information base*, base de datos de información de administración. Es un esquema o un modelo, que contiene la orden jerárquica de todos los objetos o variables manejados por el protocolo SNMP.

MTU *Maximum tranfer unit*, unidad máxima de transferencia.

OSI *Open systems interconnect*, interconexión de sistemas abiertos.

Paquete Es un fragmento de información con características propias y particulares con respecto a otros.

Pipelining Término en inglés, que identifica un procedimiento para el encolado de solicitudes de información o paquetes.

Prefetch Término en inglés, que identifica el procedimiento para la busqueda anticipada de bloques de información, objetos o páginas web.

POP *Post office protocol*, protocolo de oficina postal.

PPP *Point to point protocol*, protocolo de comunicación de punto a punto.

QOS *Quality of service*, calidad de servicio.

RFC *Request for comments*. Conjunto de notas técnicas y organizativas donde se describen los estándares o recomendaciones de Internet.

Sesión Conexión permanente, hasta que se termine de enviar todo un bloque de información y los equipos estén de acuerdo en cerrar dicha conexión.

SDMA *Space division multiple access*, multiple acceso por división de espacio.

SMTP *Simple mail transfer protocol*, protocolo simple para la transferencia de correo.

SNMP *Simple network management protocol*, protocolo simple de administración de redes.

Spofing Término en inglés que identifica un procedimiento para engañar a un equipo sobre el comportamiento de un protocolo.

SSH Secure Shell o Shell seguro. Término en inglés que identifica la conexión a una terminal remota de computadora en forma encriptada.

TCP/IP *Transport control protocol/internet protocol*, protocolo de control de transporte/protocolo entreredes.

Telnet Término en inglés, que identifica la conexión a una Terminal remota de computadora en forma no segura o no encriptada.

TDM *Time division multiplexing*, multiplexeado por división de tiempo.

UDP *User datagram protocol*, procotolo de datagramas de usuario.

Unicode Es un estándar industrial cuyo objetivo es proporcionar el medio por el cual un texto en cualquier forma e idioma pueda ser codificado para el uso informático.

Wifi *Wireless fidelity*, término de mercadotecnia para identificar los productos que cumplen con el estándar 802.11x de la IEEE.

Wimax *Worldwide interoperability for microwave access*, interoperabilidad mundial para acceso por microondas. Término de mercadotecnia para identificar los productos que cumplen con el estándar 802.16x de la IEEE.

WLAN *Wireless lan*, red de área local inalámbrica.

RESUMEN

La necesidad de la mejora del protocolo TCP sobre redes inalámbricas, viene de la experiencia del usuario final en cuanto a la percepción de la calidad de servicio (QOS). Ésto es importante, tomando en cuenta que algunos protocolos de aplicación pueden verse severamente afectados por el desempeño del TCP, como lo son: las que utilizan sesiones interactivas (telnet y ssh) o la navegación (http). Este trabajo se enfoca en la descripción de los problemas de desempeño de TCP sobre medios inalámbricos; los escenarios donde suceden dichas faltas de desempeño y los métodos que se pueden utilizar para compensar o solucionar dichos problemas, así el trabajo se divide en 5 capítulos, con el siguiente contenido:

En el primer capítulo se hace una reseña de los términos utilizados y de la información esencial sobre el protocolo TCP/IP, y luego se da una explicación más detallada de los aspectos importantes sobre el protocolo TCP específicos.

En el segundo capítulo se esbozan los temas que tienen relevancia, en el trabajo de graduación sobre la tecnología subyacente en las redes inalámbricas y los escenarios más comunes de transmisión de datos sobre redes inalámbricas, donde se puede utilizar el TCP/IP.

En el tercer capítulo se trata con detalle de explicar las razones del por qué existe el mencionado mal desempeño, y sobre que escenarios se presenta en las redes inalámbricas.

En el cuarto capítulo se enumeran y describen en forma general los métodos para compensar o solucionar los problemas de desempeño que se describieron en el capítulo anterior.

En el último capítulo se presenta un caso práctico, con el objetivo de hacer una demostración, con mediciones antes y después de la aplicación de algunos métodos, de las mejoras de desempeño que se pueden obtener mediante la propuesta contenida en este trabajo.

OBJETIVOS

- **General**

El objetivo general de esta investigación es comprender las limitaciones del protocolo TCP/IP sobre redes inalámbricas, y establecer los métodos para solucionarlas.

- **Específicos**

1. Conocer el protocolo TCP/IP y su relación con el modelo de referencia OSI.
2. Conocer la tecnología que fundamenta las redes inalámbricas y los diferentes tipos de éstas.
3. Conocer los problemas e inconvenientes que se presentan cuando se utiliza el protocolo TCP sobre redes inalámbricas.
4. Proponer las posibles soluciones a los problemas de desempeño del TCP sobre redes inalámbricas y las situaciones donde se pueden aplicar cada una de estas soluciones.
5. Establecer un escenario real donde se puedan poner en práctica la soluciones descritas.

INTRODUCCIÓN

Debido al amplio uso y popularidad del protocolo TCP/IP, se necesitan métodos, para mejorar su desempeño en redes con ciertas particularidades como las inalámbricas donde se presentan limitaciones en el ancho de banda, asimetría en los canales de bajada y subida, medios ruidosos y retardos grandes o variables. Todas estas causas afectan severamente el desempeño del TCP/IP y representan las condiciones típicas de las redes inalámbricas, tanto para las de radiocomunicación terrestre, como la satelital que cada vez se usan más en Guatemala.

Existe alguna documentación sobre el tema y sobre sus soluciones, pero está bastante dispersa y, prácticamente, analiza sólo algunos aspectos específicos. Adicionalmente toda esta información está en inglés. Este trabajo propone un tratamiento completo desde todos los ángulos posibles y enteramente en español.

El enfoque de este trabajo, se centra en exponer con detalle a nivel de protocolo, los problemas y las soluciones, para obtener el mejor desempeño en redes inalámbricas que utilizan el TCP, para la comunicación de datos, como: las redes de transmisión de datos para WLAN o *wireless lan* (802.11x y 802.16x), redes de datos sobre conexiones celulares inalámbricas (GPRS, EDGE, CDMA1x etc) y redes satelitales.

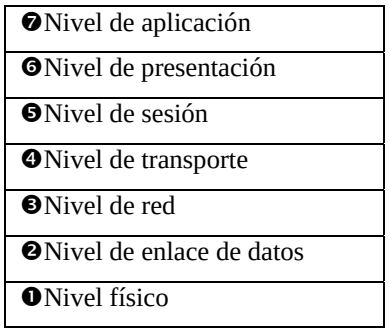
1. MODELO OSI Y EL PROTOCOLO TCP/IP

1.1 Modelo de referencia OSI y sus siete capas

Para evitar la duplicación de esfuerzos en la definición de las capacidades de cualquier sistema de comunicaciones la Organización Internacional de Normas (ISO, *Internacional Standars Organization*), desarrolló un modelo de referencia, para describir la estructura y funcionamiento de los protocolos de comunicación de datos. Este modelo de arquitectura, llamado Modelo de Referencia de Interconexión de Sistemas Abiertos (OSI, *Open Systems Interconection*), proporciona una referencia común para analizar los sistemas de comunicaciones, teniendo una terminología que es bien comprendida y ampliamente utilizada.

El modelo de referencia OSI, contiene siete niveles o capas que definen las funciones de los protocolos de comunicación de datos. Cada nivel del modelo OSI representa una función que se ejecuta cuando transfiere información entre aplicaciones cooperativas a través de la red en que intervienen. A estas siete capas se les llama con frecuencia “pila de protocolos” y cada nivel o bloque de la pila tiene un nombre descriptivo que hace referencia a su funcionalidad, tal y como se ve en la figura 1.

Figura 1. **Modelo OSI**

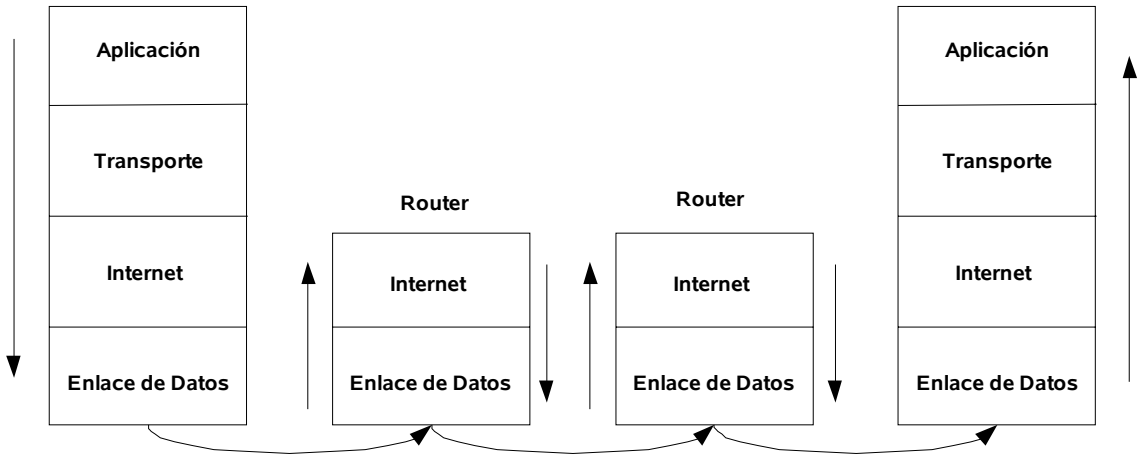


Un nivel no define un solo protocolo, más bien define una función de datos que puede ejecutarse por cualquier cantidad de protocolos. Así, un nivel puede contener varios protocolos, cada uno de los cuales proporciona un servicio apropiado para la función de ese nivel. Por ejemplo un protocolo de correo electrónico y uno de transferencia de archivos, son parte de los servicios hacia el usuario y pertenecen al nivel de aplicación.

Cada protocolo se comunica con su compañero o par en el mismo nivel de un sistema remoto. Dicha comunicación entre pares debe estar estandarizada, para ser exitosa. Se podría pensar que un cierto nivel solo le interesa comunicarse con su par en otro punto remoto y en cierto sentido es así, pero también cada nivel debe proporcionar servicios a los niveles inferior y superior a el, permitiendo el paso de información a través de el, sin que implique saber como funcionan estos, por lo que debe estar involucrado en el envío de información de una aplicación local a su aplicación remota equivalente.

Los niveles superiores se basan en los inferiores para transferir la información a través de la red. La información desciende por la pila de un nivel al siguiente, hasta que se transmite a través de la red por los protocolos de nivel físico. En el extremo remoto, la información asciende por la pila a la aplicación receptora. Todo este proceso se puede ver en la figura 2. Aislar las funciones de comunicaciones de la red en diferentes niveles minimiza el impacto del cambio tecnológico en todo el grupo de protocolos, por lo que se pueden agregar nuevas aplicaciones, sin cambiar la red física y también se puede instalar nuevo hardware de red sin volver a escribir el software de aplicación, ni todos los protocolos de los diferentes niveles.

Figura 2. **Paso de información de una pila local a otra remota**



A continuación se presenta una breve explicación de cada nivel:

1.1.1 Nivel de aplicación

Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico gestores de bases de datos y servidor de ficheros. La capa de aplicación incluye todos los procesos en los que los usuarios interactúan directa o indirectamente. Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.

Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactuan con el nivel de aplicación pero ocultando la complejidad subyacente. Así por ejemplo un usuario no manda una petición "HTTP/1.0 GET index.html" para conseguir una página en html, ni lee directamente el código html/xml.

Entre los protocolos de aplicación (refiriéndose a protocolos genéricos, no a protocolos de la capa de aplicación de OSI) más conocidos destacan:

- HTTP (*hypertext transfer protocol*) el protocolo de la Web.
- FTP (*file transfer protocol*) transferencia de ficheros.
- SMTP (*simple mail transfer protocol*) envío y distribución de correo electrónico.
- POP (*post office protocol*)/IMAP: reparto de correo al usuario final.
- SSH (*Secure Shell*) principalmente terminal remoto, aunque en realidad cifra casi cualquier tipo de transmisión.
- Telnet, otro terminal remoto, ha caído en desuso por su inseguridad intrínseca, ya que las claves viajan sin cifrar por la red.
- SNMP (*simple network management protocol*), protocolo que facilitan el uso y administración de la red
- DNS (*domain name system*), sistema de nombre de dominio.

1.1.2 Nivel de presentación

Esta capa o nivel es la primera, en trabajar más el contenido de la comunicación que cómo se establece la misma. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlas. Ej. ASN.1, MIME, etc.

Proporciona las rutinas para que la información se presente en un formato estándar y realizar las conversiones de representación de datos necesarias para la correcta interpretación de los mismos, de manera que las aplicaciones puedan intercambiar información y estén de acuerdo en como representarla, permitiendo que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres (ASCII, unicode, EBCDIC), números (*little-endian* tipo intel, *big-endian* tipo motorola), sonido o imágenes, los datos lleguen de manera reconocible.

Para conseguir este objetivo se describió una posible notación de sintaxis abstracta (ASN.1), que en realidad en la práctica solo se utiliza internamente en los MIB de SNMP (protocolo de gestión de red, para supervisar equipos de comunicaciones a distancia).

1.1.3 Nivel de sesión

Es el nivel, que proporciona los mecanismos para controlar el diálogo entre las aplicaciones cooperativas de los sistemas finales. En muchos casos, los servicios de la capa de sesión son necesarios solo parcialmente, o incluso, son totalmente prescindibles, dependiendo de la aplicación.

Esta capa ofrece varios servicios que son cruciales para la comunicación, como son:

- Control de la sesión a establecer entre el emisor y el receptor (quién transmite, quién escucha y seguimiento de ésta).
- Control de la concurrencia (que dos comunicaciones a la misma operación crítica no se efectúen al mismo tiempo).
- Control del diálago: Éste puede ser simultáneo en los dos sentidos (*full-duplex*) o alternado en ambos sentidos (*half-duplex*).
- Agrupamiento: El flujo de datos se puede marcar para definir grupos de datos.
- Mantener puntos de verificación (*checkpoints*), que sirven para que, ante una interrupción de transmisión por cualquier causa, la misma se pueda reanudar desde el último punto de verificación en lugar de repetirla desde el principio.

Por lo tanto, el servicio provisto por esta capa es la capacidad de asegurar que, dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción.

La capa de sesión permite a los usuarios de máquinas diferentes establecer sesiones entre ellos. Una sesión permite el trasporte ordinario de datos, como lo hace la capa de transporte, pero también proporciona servicios mejorados que son útiles en algunas aplicaciones. Se podría usar una sesión para que el usuario se conecte a un sistema remoto de tiempo compartido o para transferir un archivo entre dos máquinas. Ej. SIP, SMB, etc.

1.1.4 Nivel de transporte

El nivel de transporte garantiza que el receptor obtenga la información tal y como se envió, maneja la comunicación de extremo a extremo, la detección y corrección de errores y el control de flujo, aun cuando el receptor y el emisor no estén directamente conectados.

Proporciona un control de alto nivel para la transferencia de datos, y es capaz de detectar y remover paquetes duplicados, velar por el sincronismo en la información y coordinar el reenvío de un paquete si este no ha llegado correctamente a su destino. Puede asignar un número único de secuencia al paquete que va a ser transmitido, para que este sea revisado en el destino por el otro nivel de transporte.

Ejemplos de protocolos de nivel de transporte son TCP (*transport control protocol* o protocolo de control de transporte) y UDP (*user Datagram Protocol*, o Protocolo de datagramas de usuario).

1.1.5 Nivel de red

El nivel o capa de red, según la normalización OSI, es una capa compleja que proporciona conectividad, direccionamiento y selección de ruta entre dos sistemas de equipos que pueden estar ubicados en redes geográficamente distintas.

Un ejemplo de capa de red es el protocolo de Internet (o protocolo IP), el IP encamina datos entre sistemas. Los datos pueden atravesar un enlace único o pueden reenviarse por varios enlaces de una internet. Los datos se transportan en unidades llamadas datagramas.

Un datagrama tiene una cabecera de IP que contiene información de direcciones de la capa 3. Los encaminadotes (*routers*) examinan la dirección de destino de la cabecera de IP, para dirigir los datagramas al destino.

La capa de IP se denomina no orientada a conexión ya que cada datagrama se encamina de forma independiente e IP no garantiza una entrega fiable, ni en secuencia, de los mismos. IP encamina su tráfico sin tener en cuenta la relación entre aplicaciones a la que pertenece un determinado datagrama. Ej. IP, X.25, NETBEUI, etc.

1.1.6 Nivel de enlace de datos

Este nivel maneja la comunicación directa con el equipo físico o hardware y el envío de datos de manera confiable a través del enlace físico. Este nivel recibe peticiones del nivel de red y utiliza los servicios del nivel físico.

El objetivo del nivel de enlace es conseguir que la información fluya, libre de errores, entre dos máquinas que estén conectadas directamente. Para lograr este objetivo tiene que montar bloques de información (llamados tramas en este nivel), dotarles de una dirección de nivel de enlace, gestionar la detección o corrección de errores, y ocuparse del control de flujo entre equipos (para evitar que un equipo más rápido desborde a uno más lento).

Cuando el medio de comunicación está compartido entre más de dos equipos es necesario arbitrar el uso del mismo. Esta tarea se realiza en el subnivel de acceso al medio (MAC, médium access control).

Dentro del grupo de normas IEEE 802, el subnivel de enlace lógico (LLC, logical link layer) se recoge en la norma IEEE 802.2 y es común para todos los demás tipos de redes (Ethernet o IEEE 802.3, IEEE 802.11 o Wi-Fi, IEEE 802.16 o WiMAX, etc.); todas ellas especifican un subnivel de acceso al medio así como un nivel físico distintos.

Algunos ejemplos adicionales de tipo de protocolos de nivel de enlace serían PPP (*point to point protocol* o protocolo punto a punto), HDLC (*high level data link control* o protocolo de enlace de alto nivel).

En la práctica el subnivel de acceso al medio suele formar parte de la propia tarjeta de comunicaciones, mientras que el subnivel de enlace lógico está en el programa adaptador de la tarjeta (*driver* en inglés).

1.1.7 Nivel físico

Este define las características del hardware, eléctricas, mecánicas y de procedimiento necesarias para enviar la señal de transmisión de datos, incluyendo los niveles de voltaje, señalización, conectores y cableado.

Este nivel del modelo de referencia OSI es el que se encarga de las conexiones físicas de la computadora hacia la red, en este nivel se especifican los estándares de cable de par trenzado, coaxial, de fibra óptica, o equipo inalámbrico que se deben usar para conectar una red, la topología de la misma, los niveles de voltaje para 0 y 1, la forma de modulación de la señal y otras características eléctricas, así como la forma en que las antenas de microondas deben estar orientadas para comunicarse, y las características de propagación de ondas radiales en el caso de conexiones inalámbricas. También tiene atribuidos aspectos como la codificación de línea, el establecimiento de la velocidad de transmisión y la técnica usada para la misma.

1.2 Introducción al protocolo TCP/IP

A mediados de la década de los 60, el Departamento de la Defensa de los Estados Unidos, decidió promover la investigación y desarrollo de una estructura computacional nacional, que respondiera los requerimientos de funcionamiento bajo circunstancias extremas. Para este objetivo se constituyó un grupo financiado por la *DARPA (defense advanced research project agency*), cuya misión fue crear una red que interconectara computadoras en forma segura y confiable, con múltiples trayectorias y con la capacidad de conectar equipos heterogéneos, con diferente hardware y sistema operativo. Como resultado de estas investigaciones y al cabo de cerca de 20 años de investigación se establecieron los protocolos que hoy se conocen como TCP/IP.

El nombre TCP/IP se refiere a un conjunto de protocolos de comunicación de datos, pero este nombre en sí se deriva de dos de los protocolos más importantes que pertenecen al conjunto, como son: el protocolo de control de transmisión (TCP) y el protocolo de internet (IP).

Los protocolos TCP/IP han ganado popularidad, no solo por su utilización como base de Internet, sino también por sus méritos técnicos entre los que se encuentran:

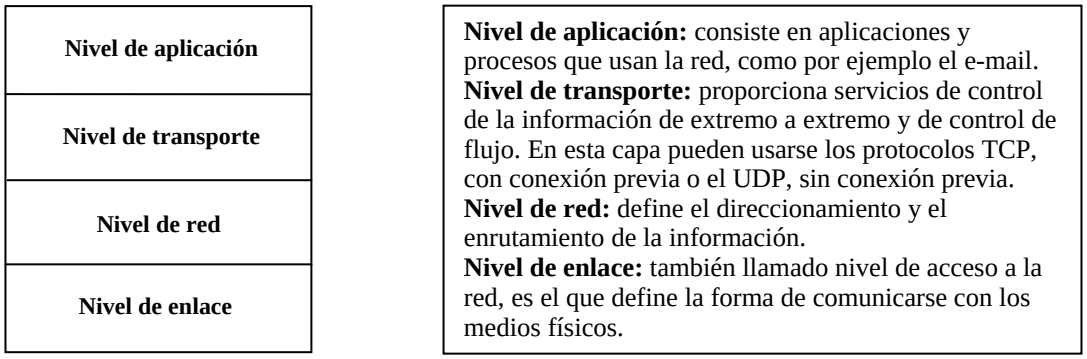
- Estándares de protocolo abiertos (RFCs), disponibles públicamente y desarrollados independientemente de cualquier hardware, sistema operativo o fabricante específicos.
- Independencia del hardware específico de red, permitiendo utilizarse sobre redes Ethernet, *Token Ring*, líneas telefónicas, ATM, *Frame Relay*, X.25 y casi cualquier otra clase de medio físico de transmisión, como fibra, cobre, inalámbrico, satelital.

- Un esquema común de direccionamiento que permite que cualquier dispositivo con TCP/IP se identifique de modo único de cualquier otro dispositivo, aun en redes tan grandes como Internet.
- Protocolos de alto nivel estándares para servicios de usuario consistentes (ftp, e-mail, web, etc), y ampliamente disponibles.

1.2.1 Descripción del protocolo y sus capas

El TCP/IP conocido hoy fue implementado originalmente sobre computadoras corriendo el Unix de Berkeley o BSD, aunque actualmente funciona sobre todos los sistemas operativos conocidos, sigue teniendo la misma estructura de 4 capas o niveles como en la Figura 3, en la cual también se muestra su correspondencia con el modelo OSI. La versión de la cual se hablará en este apartado es principalmente de la implementación estándar mas difundida del TCP, llamada TCP Tahoe, sin embargo como se verá en capítulos posteriores existen otras implementaciones con funcionalidades, métodos y capacidades ligeramente distintos (TCP Reno, TCP Vegas, etc).

Figura 3. **Pila de protocolos TCP/IP**



Los programadores del protocolo definieron según el tipo de aplicación que se utilizarían, la capa de transporte que emplearían TCP (con conexión) o UDP (sin conexión), según fuese mas conveniente, para dicha aplicación, tal y como se ve en la Figura 4.

Tabla I. **Aplicaciones comunes del TCP/IP**

TCP	UDP
e-mail (SMTP)	Sistema de archivos en red (NFS)
Transferencia de archivos (FTP)	Administración de redes (SNMP)
Web (HTTP)	Ejecución de procedimiento remotos (RPC)
Conexión remota de terminal (Telnet)	Sistema de nombre de dominio (DNS)
Sistema de representación gráfica (XWindows)	Voz sobre IP (SIP, H.323)

En la Figura 4 se puede observar, como en el campo del TCP es donde se encuentran la mayoría de protocolos comúnmente usados y es, precisamente en su orientación a la conexión, previa a la comunicación donde se presentan problemas a la hora de ser utilizado sobre redes inalámbricas, por esta razón, para el presente trabajo tiene relevancia solo el nivel de Transporte y en especial el protocolo TCP.

1.3 El protocolo TCP

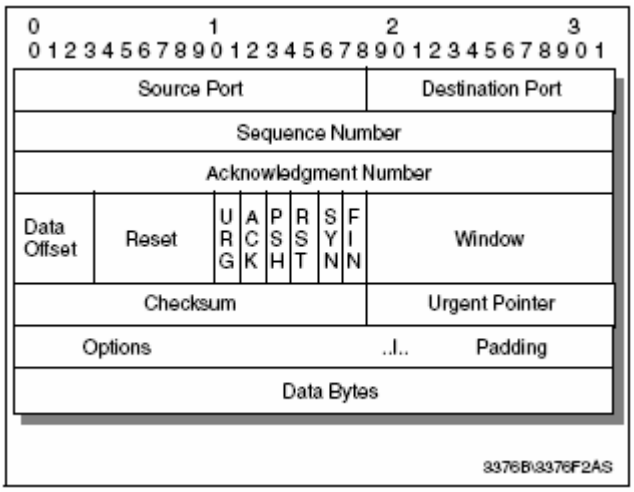
Las aplicaciones que requieren que el protocolo de transporte proporcione un envío confiable de datos usan TCP porque verifica, que los datos sean enviados a través de la red de modo correcto y en la secuencia adecuada. TCP es un protocolo confiable, *full duplex*, orientado a la conexión, de cadena o flujo de bytes.

La unidad de información intercambiada entre dos capas TCP que están comunicándose se llama segmento. Estos segmentos consisten de una cabecera TCP fija

de 20 bytes, una parte opcional, seguida de cero o más bytes de datos. El software de TCP decide el tamaño de los segmentos, respetando dos límites, el primero lo impone el IP y es que el segmento completo con cabecera y datos no puede sobrepasar los 65,535 bytes, el segundo es que cada red tiene lo que se llama MTU (*maximun tranfer unit*) o unidad máxima de transferencia.

El MTU normalmente es de unos cuantos miles de bytes y es el que generalmente impone el límite de tamaño al segmento TCP. El TCP puede negociar el tamaño de segmento máximo o MSS (*maximun segment size*), a través del campo de opciones del protocolo. En la Figura 5, se pueden ver todos los campos que conforman un segmento TCP.

Figura 4. **Formato del segmento TCP**



Fuente: TCP/IP tutorial and technical overview, IBM Redbooks, página 213.

El TCP visualiza la información que envía como una cadena o flujo de bytes continua, no como paquetes independientes, por lo tanto, el TCP se ocupa de mantener la secuencia de cuáles bytes son enviados y recibidos. Los campos de número de secuencia y de número de reconocimiento, llevan la cuenta de los bytes.

Por otro lado el estándar TCP, no requiere que cada sistema comience a numerar los bytes con algún número específico, sino mas bien el ISN (*initial sequence number*) o número inicial de secuencia, se elige al azar. Los dos extremos de la conexión sincronizan los sistemas de numeración de bytes en el intercambio inicial de segmentos SYN, durante la negociación inicial de la comunicación.

El TCP se encarga también de enviar la información recibida desde el protocolo IP hacia la aplicación correcta, utilizando los campos de 16 bits de cada uno, llamados Puerto fuente y Puerto destino, en un proceso que se llama multiplexación.

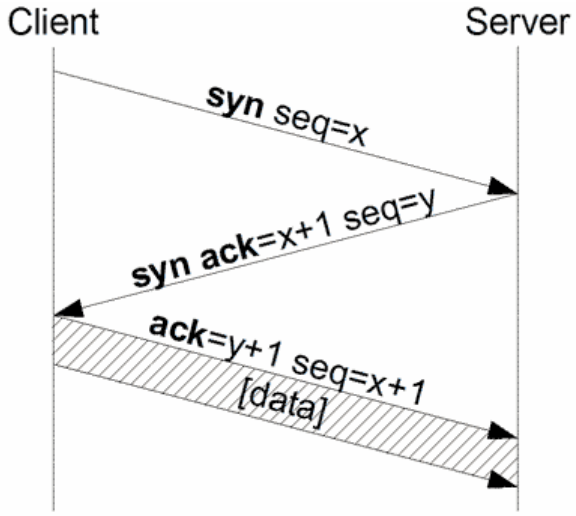
Los mecanismos que utiliza el TCP para proporcionar un flujo confiable de datos son:

1. TCP brinda confiabilidad con un mecanismo llamado reconocimiento positivo con retransmisión (PAR, *positive acknowledgement with retransmition*). Dicho sistema, envía la información de nuevo a menos que oiga por parte del sistema remoto que llegó bien. Cada segmento contiene una suma de verificación (*checksum*), que usa el receptor para verificar que la información no sufrío daños. Si el segmento de información está dañado, el receptor lo descarta. Después de un lapso apropiado de suspensión o *timeout*, el módulo TCP emisor retransmite cualquier segmento para el cual no haya recibido reconocimiento positivo. El estándar no requiere un reconocimiento individual por cada segmento, sino mas bien el número de reconocimiento es el PAR de todos los bytes hasta ese número.

2. El TCP está orientado a la conexión, por lo que establece una conexión lógica extremo a extremo entre los dos anfitriones que se comunican. La información de control, llamada negociación o *handshake*, se intercambia entre los dos extremos para establecer un diálogo antes de que se transmitan los datos.

Para establecer la negociación se utilizan ciertos bits en el campo llamado banderas. La implementación en sí de la negociación en el TCP, utiliza un método llamado negociación en tres sentidos (*three way handshaking*), porque se intercambian tres segmentos antes de comenzar el envío de la información propiamente dicha. En la Figura 6, se puede ver un esquema del proceso de la negociación en tres sentidos. El equipo A inicia la conexión enviando un segmento al equipo B con el bit SYN (sincronización de números de secuencia) encendido. El segmento dice al equipo B qué número de secuencia usará como número inicial de sus segmentos, usado este para mantener la información con un orden apropiado. El equipo B responde al A con un segmento que tiene encendidos los bits ACK (reconocimiento) y SYN. El segmento de B reconoce la recepción del segmento de A, e informa a A con cual número de secuencia comenzará el equipo B. Por último, el equipo A envía un segmento que reconoce la recepción (ACK) del segmento de B, dado que A tiene evidencia que B esta vivo y listo para recibir datos, o sea una vez establecida la conexión, comienza a transferir la información real. Cuando las capas TCP de A y B terminaron la transferencia de información, intercambian una negociación en tres sentidos con segmentos que contienen el bit FIN (No hay mas información), para cerrar la conexión.

Figura 5. **Negociación en tres sentidos**



Fuente: www.wikipedia.org

3. El TCP utiliza también un procedimiento que se conoce como Control de flujo por el método de ventanas corredizas, el cual evita saturar de información los *buffers* del equipo remoto, con la posibilidad de que se pierdan paquetes. Este procedimiento se implementa usando, el campo llamado ventana y el bit de reconocimiento ACK, que cumple la doble función de: reconocimiento positivo descrito antes y el control de flujo, por el medio del cual le indica al emisor cuanta información se ha recibido y cuanta puede todavía recibir el receptor. La ventana indica al emisor que puede seguir enviando segmentos mientas el número total de bytes que envíe sea menor que la ventana de bytes que puede aceptar el receptor. El receptor controla el flujo de bytes del emisor cambiando la medida de la ventana. Una ventana en cero dice al emisor que deje de transmitir hasta que reciba un valor de ventana distinto. El uso de la ventana corrediza junto con el reconocimiento ACK da lugar a lo que llama reconocimientos acumulativos.

4. Tiempo límite para retransmisión o RTO (*retransmition time out*). Cada vez que se envía un segmento, El TCP arranca un temporizador y espera un acuse de recibo. Si te termina el tiempo antes de que se acusen de recibidos los datos del segmento, el TCP asume que dicho segmento se perdió o daño y lo retransmite. El TCP calcula el RTO, utilizando un algoritmo adaptable de retransmisión, que en esencia implica que el TCP monitorea el desempeño de cada conexión y deduce de estos un tiempo razonable para la terminación del temporizador de retransmisión.

1.4 Algoritmos de Congestión del protocolo TCP

Existen dos indicaciones de la pérdida de paquetes:

1. Que se expire el *timeout timer* antes de recibir un reconociendo de respuesta
2. Que se reciba una duplicación de reconocimientos con el mismo número de secuencia

Cuando una pérdida es detectada, el TCP retransmite el segmento perdido. Normalmente se supone que las pérdidas por daños en la red son muy pequeños (menos del 1%) en redes alámbricas, por lo que se supone que las todas las pérdidas de paquetes corresponden a la congestión en alguna parte de la red, entre el emisor y el receptor. El TCP utiliza algoritmos de control de congestión para prevenir que el emisor sature los *buffers* del receptor y/o la capacidad de la red, adaptando la tasa de transmisión a las condiciones de estos. Para el efecto se mantienen las siguientes variables:

1. *Advertised window*, ventana anunciada de recepción, que es el control de flujo anunciado por el receptor.
2. *Congestion window*, ventana de congestión, que es el control de flujo impuesto por el emisor.
3. *Timeout timer*, reloj de expiración de recepción, que mantiene cuanto tiempo se debe esperar el retorno de un reconocimiento antes de dar por perdido el paquete.
4. *Slow start threshold size* o tamaño del umbral de arranque lento, que es el límite entre el uso del algoritmo *slow start* y el de *congestion avoidance*.

A continuación con una breve descripción de los algoritmos de congestión usados en las versiones más comunes de TCP:

1.4.1 Arranque lento, *slow start*

Las primeras implementaciones del TCP, comenzaban a enviar múltiples segmentos hasta el máximo anunciado como *advertised window*, pero en caso de que existan *routers* intermedios entre el emisor y receptor, estos pueden no poder manejar todos los segmentos, debido a limitaciones propias, por lo que se pueden eliminar paquetes, teniendo como resultado la retransmisión y la degradación en la comunicación.

Para evitar este efecto se implementó un algoritmo llamado *slow Stara*, arranque lento, que funciona suponiendo que la tasa a la que se pueden enviar segmentos es la misma a la cual los reconocimientos se reciben del otro extremo. Cuando la conexión se establece, la ventana de congestión se inicializa a un segmento, cuyo tamaño es dado por el receptor. Cada vez que una reconociendo se recibe, la ventana de congestión se aumenta, primero en 1 luego en 2, luego en 4 y así sucesivamente, en forma pseudo exponencial, hasta que se pierda algún acuse de recibo en la red, indicándole al transmisor que se llegó al límite de la capacidad de la red y/o el receptor. Este método se utiliza normalmente cuando se comienza una sesión y cuando se expira el *timeout timer*.

1.4.2 Retransmisión rápida, *fast retransmit*

Este algoritmo reduce el tiempo que el transmisor espera para retransmitir un segmento perdido. Si un paquete se pierde, pero los subsecuentes arriban, el receptor genera un paquete ACK, para cada segmento visto.

Como el paquete ACK especifica el último segmento recibido correctamente, estos ACKs serán todos duplicados. Si el TCP del transmisor, recibe tres reconocimientos duplicados con el mismo número de secuencia osea cuatro reconocimientos en total con el mismo numero de secuencia, el transmisor puede estar razonablemente confiado de que dicho segmento se perdió y no de que no va a llegar fuera de orden, entonces el trasmisor retransmite el paquete sin esperar que se expire el temporizador de retransmisión. La única desventaja de este procedimiento es que no puede detectar más de un paquete perdido en una secuencia, por lo que el procedimiento puede correctamente detectar la primera perdida, pero las perdidas adicionales solo serán detectadas hasta que la primera haya sido corregida.

1.4.3 Congestión evitable, *congestion avoidance*

Cuando se detecta la pérdida de un segmento debido a la recepción de ACKs duplicados, este algorítmo reduce a la mitad (50%) la ventana de congestionamiento, hasta un mínimo de un segmento y para los segmentos que permanezcan en la ventana permitida, anula exponencialmente el temporizador para la retransmisión.

En la práctica si bien estos algorítmos, son diferentes y tienen objetivos diferentes, se implementan juntos, usando uno u el otro, o uno después del otro según sea necesario.

2. CONCEPTOS GENERALES SOBRE REDES INALÁMBRICAS

2.1 ¿Qué es una red inalámbrica?

Las redes inalámbricas se pueden definir, como el conjunto de equipos (de capa 1) que usan ondas electromagnéticas, para transmitir información a través del medio físico aire o sobre el vacío. Es decir en estas redes el medio de transmisión es el espacio vacío.

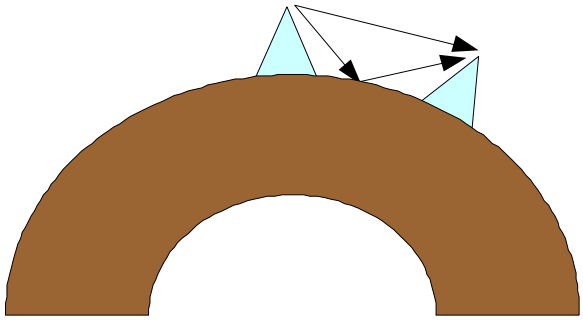
2.2 Tipos de redes inalámbricas

Para efectos de esta trabajo, nos interesan las redes inalámbricas utilizadas para la transmisión de datos del tipo WLAN (*wireless lan* o 802.11x), las redes de datos sobre conexiones celulares inalámbricas (GPRS, EDGE, CDMA1x etc) y las redes de datos satelitales. Las redes inalámbricas se pueden clasificar de diferentes formas, pero para efectos de esta tesis las clasificaremos de la siguiente forma:

2.2.1 Redes inalámbricas terrestres

Definidas como las compuestas por dos o mas parejas de transmisor-receptor sobre la corteza terrestre y que se comunican directamente entre sí a través del aire, pudiendo ser móviles o fijas. Este tipo de comunicación generalmente genera dos tipo de ondas, una directa y otras reflejadas. Las ondas pueden ser reflejadas en la tierra o en la ionosfera.

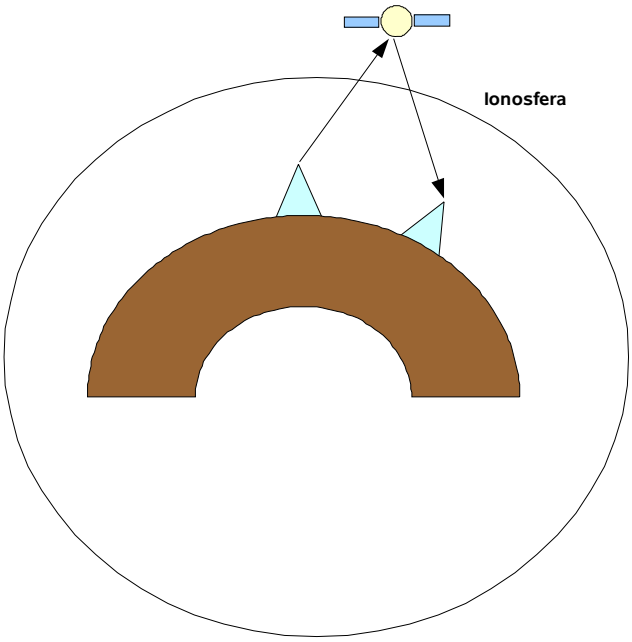
Figura 6. **Redes inalámbricas terrestres**



2.2.2 Redes inalámbricas satelitales

Definidas como las compuestas por dos o más parejas de transmisor-receptor sobre la corteza terrestre y que se comunican entre sí con la ayuda de un satélite artificial en órbita alrededor de la tierra.

Figura 7. **Redes inalámbricas satelitales**



2.3 Métodos de acceso al medio físico en redes inalámbricas

Existen varias formas de acceder al medio físico, en forma estática (FDMA, TDMA y CDMA) y en forma dinámica (CSMA y sus variantes y el SDMA).

2.3.1 FDMA

FDMA es un acrónimo acrónimo inglés que significa *frequency division multiple access*, que traducido al español es Tecnología de acceso múltiple por división de frecuencias. El FDMA separa el espectro en distintos canales, al separar el ancho de banda en pedazos (frecuencias) uniformes.

La tecnología FDMA es mayormente utilizada para la transmisión analógica, aunque es capaz de transportar información digital. Utiliza un mecanismo de control para asegurar que dos estaciones no transmitan en la misma frecuencia al mismo tiempo.

2.3.2 TDMA

La tecnología TDMA envía la informaciones digitales, cada una utilizando la señal de radio por un espacio de tiempo definido (*timeslot*), una detrás de otra. La tecnología TDMA puede tener varias veces la capacidad de un sistema analógico que utilice el mismo número de canales.

2.3.3 CDMA

Existen dos esquemas comunes de CDMA, uno es el de secuencia directa o DSSS y el otro es el salto en frecuencia o FHSS. En el DSSS, después de digitalizar la información, la transmite a través de todo el ancho de banda disponible. Varios canales de datos son sobrepuestos en el medio físico, y cada una tiene un código de secuencia único. Usando al tecnología CDMA, es posible comprimir entre 8 y 10 llamadas digitales para que estas ocupen el mismo espacio que ocuparía una llamada en el sistema analógico. En el FHSS, se transmite sobre un mismo ancho de banda pero saltando de una frecuencia a otra en una secuencia determinada, en rápida sucesión. Es posible que varios canales de datos se transmitan sobre el mismo medio físico, asignándoles a cada canal una combinación de salto diferente.

2.3.4 CSMA

Método comúnmente usado en las redes inalámbricas tipo *wireless* LAN y que significa acceso múltiple por detección de portadora, (*carrier sense multiple access*). Con este método se cuenta con uno o varios canales en los cuales se realiza una escucha del medio para saber si existe presencia de portadora de algún otro equipo o estación en los momentos en los que se quiere ocupar el canal. Lo que significa que se utiliza un medio de acceso múltiple y que la estación que desea emitir previamente escucha el canal antes de emitir. Si el canal está ocupado espera un tiempo aleatorio y vuelve a escuchar. Cuando detecta libre el canal puede actuar de dos formas distintas: emitiendo de inmediato o esperando un tiempo aleatorio antes de emitir. El fin es evitar colisiones, es decir que dos equipos no hablen al mismo tiempo. Por otro lado define el procedimiento que estos dos equipos deben seguir si llegasen a usar el mismo medio de forma simultánea.

En redes inalámbricas tipo *wireless* LAN, resulta a veces complicado llevar a cabo el primer paso dado que la transmisión y recepción se realiza por el mismo canal y por lo tanto no se puede escuchar a la vez que se trasmite lo que implica que no pueden detectarse colisiones (escuchar al medio para determinar si está libre o no). Por este motivo, surgen dos problemas que pueden ser detectados:

1. Problema del nodo oculto: la estación cree que el medio está libre cuando en realidad no lo está, pues está siendo utilizado por otro nodo al que la estación no "oye".
2. Problema del nodo expuesto: la estación cree que el medio está ocupado, cuando en realidad lo está ocupando otro nodo que no interferiría en su transmisión a otro destino.

Los dos distintos tipos de CSMA relevantes para esta tesis son:

- CSMA/CD
- CSMA/CA

CSMA/CD. Siglas que corresponden a *carrier sense multiple access with collision detection* (en español, "Acceso Múltiple con Escucha de Portadora y Detección de Colisiones"), es una técnica usada en redes Ethernet para mejorar sus prestaciones. El CSMA/CD supone una mejora sobre CSMA, pues la estación está a la escucha a la vez que emite, de forma que si detecta que se produce una colisión para inmediatamente la transmisión. La ganancia producida es el tiempo que no se continúa utilizando el medio para realizar una transmisión que resultará inútil, y que se podrá utilizar por otra estación para transmitir.

Si durante la transmisión de una trama se detecta una colisión, entonces las estaciones que colisionan abortan el envío de la trama y envían una señal de reinicio. Después de una colisión, las estaciones esperan un tiempo aleatorio (tiempo de *backoff*) para volver a transmitir una trama.

CSMA/CA. En redes informáticas, *carrier sense multiple access with collission avoidance* (CSMA/CA) es un protocolo de control de redes utilizado para evitar colisiones entre los paquetes de datos en redes inalámbricas. Es un método de acceso de red en el cual cada estación señala su intento para transmitir antes de que lo haga realmente. Esto evita que otros dispositivos envíen la información, así evitando que las colisiones ocurran entre las señales a partir de dos o más dispositivos. De esta forma permite a un emisor transmitir en cualquier momento en que el medio no esté ocupado.

En CSMA/CA, tan pronto como un nodo recibe un paquete que deba ser enviado, comprueba que el canal esta libre (que ningún otro nodo este transmitiendo en ese momento). Si el medio o canal está libre , entonces el paquete es enviado despues de esperar por un corto periodo de tiempo; pero si el canal esta ocupado, el nodo esperara por un periodo de tiempo aleatorio.

2.3.5 MACA

Para resolver estos problemas, la IEEE 802.11 propone MACA (*multiaccess collision avoidance* – evasión de colisión por acceso múltiple).

La modificación incluida en este protocolo, respecto a CSMA/CD, es que ahora las estaciones, antes de transmitir, deben enviar una trama RTS (*request to send*). Dicha trama, indica la longitud del paquete de datos a enviar.

Ante esto, el resto de estaciones actuarán de tal forma que, si “escuchan” un RTS, esperarán por el CTS (*clear to send*) y, si “escuchan” un CTS, esperarán el tiempo necesario para que se transmita la longitud indicada en dicho CTS.

2.3.6 SDMA

En las redes inalámbricas tradicionales, la estación base no tiene información sobre la posición de las estaciones remotas dentro de su radio de alcance y envía la señal en todas direcciones para lograr la mayor cobertura. Esto resulta en un desperdicio de la potencia de transmisión a lugares donde no existen estaciones receptoras y adicionalmente en una interferencia con las celdas adyacentes que usen las mismas

frecuencias. En sentido contrario, en la recepción la antena recibe señales provenientes de todas direcciones incluyendo el ruido y las señales interferidas.

Todo esto se puede solucionar utilizando sistemas de antenas inteligentes, que aprovechan la información sobre la localización espacial de las estaciones remotas dentro de la celda para dar lugar a la técnica llamada SDMA (*space division multiple access*, multiple acceso por división de espacio). En este método de acceso el patrón de radiación de la estación base, tanto en la transmisión como en la recepción es adaptado a cada estación remota usando múltiples antenas y procesamiento digital de las señales de manera de obtener la máxima ganancia en la dirección de dicha estación remota y es la combinación compleja de varias señales a la vez.

2.3.7 TDM Estadístico

Es un esquema de asignación del canal dinámico de comunicaciones donde la capacidad del canal se divide equitativamente en tiempo entre las terminales a las que se va a trasmitir, osea que se en un determinado momento hay 3 terminales la capacidad se divide dentro de 3 y si en otro momento hay 10 se dividen entonces dentro de 10.

2.3.8 DVB

El DVB utiliza un esquema de transmisión basado en celdas de tamaño fijo de 88 bytes, cada una de las cuales pueden contener información de diferentes terminales y se envían todas en secuencia por el mismo canal.

2.4 Principales escenarios en redes inalámbricas

Aunque existen muchos tipos de redes inalámbricas para transmisión de datos se presentarán a continuación los más comunes, con sus características relevantes, como una manera de ejemplificar las situaciones donde se desempeña el TCP:

2.4.1 Enlaces satelitales

Las redes satelitales presentan las siguientes características:

- Tienen retardos altos con un mínimo de 500ms debido a la distancia que recorre la información, dado que la mayoría de los satélites utilizados son geostacionarios a 36,000Kms de la tierra sobre el Ecuador. El retardo puede elevarse más allá de los 500ms debido al procesamiento de los paquetes y canales de comunicación y al esquema de acceso al canal.
- Son muy afectos a las condiciones climáticas adversas, como la lluvia y las tormentas solares, lo que da como resultado tasas de error altas.}
- Se utilizan normalmente TDM estadístico o DVB (Celdas de 88 bytes), para el canal de bajada y FDMA y/o TDMA en el canal de subida como esquemas de acceso al canal.
- Se utilizan normalmente *Frame Relay* o alguna variedad de HDLC como protocolos de la capa de enlace.
- La asignación de los canales individuales puede ser estática, dinámica o a través de la reserva previa.
- Las velocidades de transmisión van desde 64Kbps hasta 45Mbps.
- Los canales de bajada y subida normalmente son asimétricos.

2.4.2 Enlaces WiFi

Los enlaces WiFi presentan las principales características:

- Si bien sus retardos son relativamente bajos (menores de 40ms), tienden a sufrir mucho de problemas de interferencia y por ende de pérdida eventual de paquetes.
- Utilizan principalmente el CDMA, solo o en combinación del SDMA, como métodos de acceso al medio físico.
- Se utiliza el protocolo 802.11 (CSMA/CA o MACA), como protocolo de acceso.
- Las velocidades de transmisión van desde 1Mbps hasta los 54Mbps.
- Se basa en el concepto de un único canal por el cual todos compiten para transmitir.
- Su rango de cobertura va de los 45 a 90mts en áreas cerradas y puede aumentar a varios kilómetros en áreas abiertas con antenas especiales de alta ganancia.

2.4.3 Enlaces WiMAX

Los enlaces WiMAX presentan las principales características:

- Si bien sus retardos son relativamente bajos (menores de 40ms), tienden a sufrir mucho de problemas de interferencia y por ende de pérdida eventual de paquetes.
- Utilizan principalmente el CDMA en combinación del SDMA, como métodos de acceso al medio físico.
- Se utiliza el protocolo 802.16 que es un protocolo de planificación donde se asigna que estación puede transmitir en que momento y cuanta información puede enviar.

- Realmente WIMAX es una tecnología de banda ancha basada en el Standard IEEE 802.16, es decir toda la especificación de WiMAX esta basada en este estandar.
- Las velocidades de transmisión pueden llegar en ciertas condiciones hasta los 70 Mbps.
- Su rango de cobertura es un compromiso con la velocidad, si una aumenta la otra disminuye y puede llegar teóricamente hasta los 100Kms.
- Se diferencia principalmente del WiFi en el uso de códigos de corrección de errores (FEC y *turbo codes*) en conjunto con la retransmisión automática de tramas a nivel capa de enlace (ARQ).

2.4.4 Redes celulares GSM con GPRS y EDGE

Los enlaces GSM de datos presentan las principales características:

- Tienen retardos típicos altos entre 600ms y 1 segundo.
- Utiliza una combinación de FDMA y TDMA.
- Tiene una velocidad de transmisión máxima teórica de 171.2Kbps para GPRS y 473.6 Kbps para EDGE, para la bajada y 13.4Kbps en ambos casos.
- Utiliza un esquema de asignación dinámica o estática según las necesidades de transmisión de los terminales remotos.
- Su rango de cobertura es normalmente la misma de las redes GSM de voz, aproximadamente unos 35Kmts.

2.4.5 Redes celulares CDMA con CDMA1x y EvDO

Los enlaces CDMA de datos presentan las principales características:

- Tienen un retardo típico entre 150 y 200ms.
- Utilizan CDMA.
- Tienen una tasa de transmisión máxima teórica de 144Kbps para CDMA1x y de 2.5Mbps para EvDO, para la bajada y de 14.4Kbps para la subida en CDMA1x y 154Kbps para la subida en EvDO.
- Utiliza un protocolo de planificación para la asignación de los *timeslots* en el canal de bajada.

3. PROBLEMAS DE DESEMPEÑO DEL TCP EN REDES INALÁMBRICAS

3.1 Problemas debido a los medios inalámbricos

3.1.1 Limitaciones de ancho de banda

Muchos servicios inalámbricos sufren de la falta de disponibilidad de espectro radioeléctrico, debido a condiciones de mercado o legislación. En otros casos los sistemas inalámbricos fueron diseñados originalmente con un propósito diferente, como en el caso de las redes celulares, que fueron diseñadas para transmitir voz, no datos y cuyos canales son los suficientes para dicha tarea y por lo tanto son bastante pequeños (8 o 13kbps). Esto da como resultado desempeños o capacidades de transmisión modestos o pobres.

3.1.2 Asimetría en los canales de bajada y subida

Es bastante común encontrar que por razones de diseño como utilización, costo, potencia y/o disponibilidad de frecuencias, los sistemas inalámbricos presentan asimetrías en los canales de bajada y subida. Por ejemplo es común que los patrones de consumo del ancho de banda en los sistemas de distribución de Internet/Intranet, el de bajada sea bastante mayor que el de subida, por lo que dichos sistemas son dimensionados para ajustarse a estos patrones.

La asimetría común en los sistemas inalámbricos provoca problemas en el desempeño, debido a que la tasa de envío de información está limitada por la tasa a la que los reconocimientos ACKs arriban al receptor. Experimentalmente una relación de 47:1 es más o menos el máximo de asimetría que se puede permitir para tener una máxima transferencia. Así por ejemplo un canal de subida de 33Kbps será desbordado, afectando el desempeño, si se necesita bajar más de 1.5Mbps en el canal de bajada.

3.1.3 El medio ambiente

Sin entrar en detalles, los sistemas inalámbricos son susceptibles de interferencias, atenuación o absorción provocadas por diferentes fuentes:

- La frecuencia, la temperatura y la geografía las cuales definen las características de propagación
- Las interferencias por las tormentas solares
- La interferencia por descargas electroatmosféricas o rayos
- La interferencia por el ruido generado por motores de combustión y/o eléctricos
- La interferencia por campos electromagnéticos fuertes provocados por líneas de alta tensión
- La interferencia de otros sistemas de comunicación cercanos, en las mismas frecuencias o en frecuencias cercanas con amplificadores con sistemas de estabilización de frecuencia pobres.
- La interferencia de los propios amplificadores del sistema saturados o al límite de saturación.
- La interferencia provocada por canales contiguos en un mismo sistema.
- La interferencia de las ondas reflejadas en lagos, montañas, edificios o paredes
- La absorción o atenuación por la polución
- La absorción o atenuación por lluvia o humedad

Todas estas fuentes pueden provocar la pérdida o daño de paquetes lo que afecta el desempeño del TCP, tanto por la retransmisión de paquetes como por la interpretación de la pérdida de estos, lo que provoca saturación de la red y el subsecuente uso los algoritmos de control de congestión.

3.1.4 Método de asignación de capacidad y acceso al medio físico

El método para asignar la capacidad a través del medio físico es otra area que tiene un potencial de reducir el rendimiento o la eficiencia. Las dos formas más comunes de asignar la capacidad reflejan dos filosofías diferentes de cómo administrar los enlaces: orientados a conexión y sin conexión.

El método orientado a la conexión asigna un nivel fijo de capacidad antes de la transmisión y retira dicha asignación de la capacidad al concluir la transmisión. Lo sobresaliente de este método es que se requiere cuando menos una demora de viaje redondo para asignar y cancelar la asignación de capacidad, sobre alguno de los canales disponibles (FDMA, TDMA, CDMA, etc.). Para conexiones prolongadas, tales como la transferencia de archivos, el costo de establecer y cancelar enlaces es reducido. Para conexiones cortas, tales como una solicitud URL o de *web* a través de una red, el tiempo requerido para el establecimiento y cancelación del enlace puede compensar las eficiencia logradas por otros medios.

El método sin conexiones evita el tiempo requerido para la conexión, por lo que es el método preferido para el tráfico general de datos, que es una mezcla de sesiones interactivas y conexiones prolongadas, por lo general mas de la primera que de la segunda. Sin conexiones se refiere en parte a la forma en que se asigna la capacidad y en parte a como se accesa al medio físico.

En un sistema sin conexiones, cuando un paquete está listo para ser transferido, simplemente se envía, sin necesidad de configurar nada previamente. Como normalmente varios emisores comparten uno o varios canales (FDMA, TDMA, CDMA, CSMA, etc), se presenta el problema de las colisiones debido al tráfico aleatorio. El efecto de las colisiones es la retransmisión y la eventual pérdida de paquetes, tanto de datos, como de reconocimientos.

3.1.5 Retardos grandes y/o variables

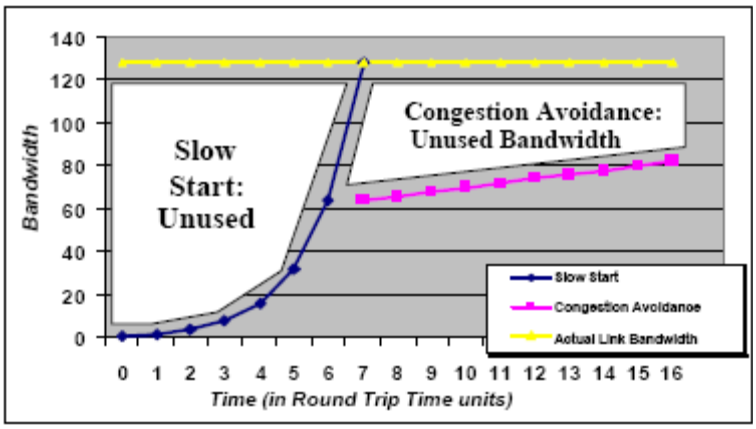
Estos se pueden presentar debido a las siguientes razones:

- Medios ruidosos que generan interferencias
- Debido a la difracción natural de las ondas electromagnéticas sobre medios diferentes del vacío, como el aire u otros gases presentes en el medio y/o la polución.
- El procesamiento natural de los protocolos de comunicación.
- Las colisiones que se generan en algunos esquemas de asignación o métodos de acceso al medio
- La distancia de transmisión, por ejemplo un satélite geoestacionario se encuentra a 36,000 kilómetros de distancia de la superficie terrestre, lo que representa un retardo mínimo en viaje redondo de 500ms en una vía. Aún enlaces terrestres de distancias grandes, pueden tener un retardo natural alto.

3.2 Limitaciones inherentes del TCP

El TCP fue diseñado con la flexibilidad en mente, para que pudiera funcionar sobre cualquier topología de red existente o por existir. La consecuencia de esta flexibilidad es que el TCP no se comporta en forma óptima sobre redes inalámbricas. A continuación se exponen las razones de ese pobre desempeño:

Figura 8. **Efecto de los protocolos de congestión**



Fuente: Documentación técnica de Flash Networks.

3.2.1 Inicio de sesión de tres vías y reconocimientos

Cada conexión TCP es establecida a través de un procedimiento llamado apretón de manos de tres vías, entre el emisor y el receptor, este procedimiento limita la velocidad a la cual se inicia el envío y recepción de los paquetes. Adicionalmente los paquetes enviados a través de un enlace TCP tienen un paquete de aceptación correspondiente que se envía de regreso para confirmar la recepción. Este sistema de reconocimiento asegura una comunicación confiable extremo a extremo bajo condiciones inciertas y de congestionamiento de la red.

La generación de tráfico de reconocimiento se convierte en un problema para transmisiones cortas a través de canales con elevados niveles de retardo o latencia. En el caso de los enlaces inalámbricos con retardo o latencia alto como los enlaces de larga distancia o los enlaces satelitales, estas sobrecargas fijas, significan que aún pequeños intercambios de datos pueden verse seriamente retardados y la tasa de transmisión verse también seriamente disminuida. Aunque estos factores pueden ser un factor mínimo para aplicaciones de transmisión solamente como la transferencia de archivos, sí puede limitar significativamente el rendimiento de aplicaciones típicamente interactivas como el *web*.

3.2.2 Inicio lento, *slow start*

Como el TCP no conoce de antemano el ancho de banda real de la red, el TCP utiliza un mecanismo llamado *slow start* para determinar la capacidad de de rendimiento del canal al realizarse la configuración para la conexión inicial y así evitar condiciones de congestión. El inicio lento envía un paquete a través del canal y espera una respuesta. Si se recibe una respuesta, el siguiente paquete se envía a una tasa de transmisión mayor. Este procedimiento se repite hasta que se determina la velocidad del enlace. En las implementaciones comunes del TCP, este proceso toma normalmente entre 7 y 15 RTT (*round trip times*), para llegar límite de velocidad del canal, por lo que el enlace inalámbrico se subutiliza y se puede percibir como lento. Adicionalmente la transacción completa puede terminar antes que se llegue a la velocidad máxima.

3.2.3 *Congestion avoindance*

El TCP responde a una perdida de paquetes asumiendo que se debe a la congestión de la red y reduce la tasa de transmisión en 50%, situación que en el caso de una red inalámbrica puede no ser cierta ya que una buena parte de la pérdida de paquetes se debe ruido en el canal o a otras causas transitorias. Si esta situación se presenta en forma temprana en el proceso de *slow start*, llegar a la velocidad máxima puede tomar más tiempo de lo normal o no llegar nunca.

3.2.4 *Advertised window* (RWIN)

El TCP cuenta con un mecanismo de ventana integrado que tiene el propósito de lograr el mayor rendimiento posible, mientras balancea el riesgo de volver a transmitir paquetes cancelados. Funciona permitiendo que un transmisor envíe un cierto número de paquetes antes de tener que esperar la aceptación del receptor. Típicamente, el tamaño de la ventana se ajusta para acomodar canales con bajo nivel de latencia (terrestres) con un desempeño de ruido no uniforme, pero no mantienen la transmisión durante demoras de procesamiento en cualquiera de los extremos del enlace. Las ventanas cortas pueden reducir considerablemente la capacidad de un canal. Para evitar que se bloquee el flujo de paquetes es necesario ajustar o afinar el tamaño de la ventana para la latencia o retardo conocida y el ruido esperado en el enlace.

Por otro lado el TCP solo permite enviar un máximo de bytes dado por la ventana de recepción, que define el receptor, dado que el valor usado para esta ventana es de 16 bits, la ventana máxima permitida en una implementación común de TCP es de 2^16 = 64Kbytes. Usando la conocida formula para calcular el ancho de banda máximo en función del RTT dada por:

Max bps = RWIN / RTT

Esta muestra que el ancho de banda efectivo depende del retardo en la transmisión, que para las redes inalámbricas, puede ser variable y alto, según la tecnología y las condiciones del medio. A continuación se incluye una tabla donde se muestra con valores el efecto combinado de la ventana y el retardo:

Tabla II: **Ancho de banda según RTT**

Enlace Wan	**Retardo típico por distancia**	**Rwin 64K**	**Rwin 16K**
La misma ciudad	15ms	32Mbps	8Mbps
Región	30ms	16Mbps	4Mbps
A traves de un Continente	100ms	5.2Mbps	1.3Mbps
Entre Contienentes	200ms	2.6Mbps	640Kbps
Satelital	800ms	850Kbps	213Kbps

3.2.5 Tiempo de reacción

La respuesta del TCP a situaciones de saturación o pérdida de paquetes depende del RTT, por lo que se ve afectado por el retardo inherente del medio inalámbrico que dependiendo de la tecnología usada puede variar entre unas decenas de milisegundos, hasta varios segundos, provocando que aplicaciones interactivas se sientan mas lentas de lo normal.

3.2.6 Muchas conexiones TCP concurrentes

En ambientes donde se tienen una gran cantidad de conexiones TCP concurrentes[1] en un enlace y estas son mayores que el producto de ancho de banda por el retardo, el TCP se comporta forzando la pérdida de paquetes hasta un 50%, mientras la utilización del enlace se mantenga alto, manteniendo la respuesta a los usuarios con grandes variaciones. Estas pérdidas de paquetes son fuente de degradación de la velocidad y respuesta y una gran pérdida de recursos de la red. Se pueden utilizar técnicas de encolado para compensar o retardar el efecto, sin eliminarlo.

[1] *TCP Behavior with Many Flows, Morris, R.*
IEEE International Conf erence on Network Protocols, October 1997, Atlanta, Georgia.
http://www.eecs.harvard.edu/networking/papers/icnp97-web.ps

4. MÉTODOS PARA SOLUCIONAR LOS PROBLEMAS DE DESEMPEÑO DEL TCP SOBRE REDES INALÁMBRICAS

Existen varios métodos, para mejorar el desempeño del TCP/IP sobre redes inalámbricas, que se pueden utilizar solos o en conjunto, como es usual. En este apartado se procede a explicar cada uno en forma general, para luego en siguiente capítulo proceder a presentar un caso práctico donde se aplican en conjunto varios de estos métodos.

4.1 Métodos que compensan los problemas de desempeño

Existen métodos que si bien no arreglan completamente los problemas, compensan o disminuyen los efectos para ciertos protocolos que corren sobre TCP/IP (protocolos de aplicación) y en ciertas circunstancias, mejorando notablemente la experiencia de usuario final. Es usual utilizar varios de estos a la vez.

4.1.1 Ajuste de los parámetros TCP

Estos cambios se realizan manipulando variables que definen características adicionales o parámetros ajustables de las implementaciones de TCP.

4.1.1.1 Ajuste de la ventana de recepción (TCP *receive window*)

Este valor determina cuantos bytes se pueden permitir al transmisor enviar y por lo tanto afecta la tasa de transmisión. Un valor adecuado de ventana de recepción mejora el desempeño del enlace.

4.1.1.2 Escalado de Ventana, *window scaling*

Procedimiento en el cual se permite que todos los números de secuencia, de reconocimiento y de tamaño de ventana sean interpretados como múltiplos de bytes y no como bytes individuales, lo que permite efectivamente mayores rangos de de estos y por lo tanto posibles tasas de transmisiones mayores.

4.1.1.3 *Time stamping*

Este procedimiento implica que el transmisor ponga una marca de tiempo en cada segmento TCP y que el receptor al recibirla la devuelva en el reconocimiento asociado, de manera de permitir una mejor medición de los RTT y por lo tanto una mejor protección contra la reutilización y mezcla de números de secuencia, cuando estos se han dado vuelta, dentro del tiempo que un segmento ha sido demorado en una cola.

4.1.1.4 Otros tipos de paquetes de reconocimiento ACK

Reconocimiento negativo, *negative acknowledgement* (NACK). En este el receptor notifica explícitamente al transmisor cuales segmentos han sido recibidos en forma incorrecta y por tanto necesitan retransmitirse.

Reconocimientos selectivos, *selective acknowledgement* (SACK). En este el receptor lista explícitamente los segmentos en un flujo que son reconocidos en forma positiva. Este permite manejar efectivamente la pérdida de más de un segmento por ventana.

4.1.1.5 Descubrimiento del MTU de la ruta, *path* MTU *discovery*

Este procedimiento permite obtener el valor óptimo de la máxima unidad de transferencia o MTU, para una ruta dada, lo que permite enviar la cantidad óptima de información por el canal, sin que se presente fragmentación.

4.1.1.6 Detección de hoyos, *black hole detection*

Procedimiento por el cual se puede detectar la pérdida de un segmento, después de haber recibido varios en una cola.

4.1.2 Compresión

La compresión comprende la utilización de software que utilizan algoritmos para analizar, el flujo de información buscando patrones y substituyendo dichos patrones por cadenas de caracteres, que ocupan menos espacio. La compresión puede realizarse sobre los datos (carga útil) y/o sobre las cabeceras TCP/IP. Estos métodos ayudan a disminuir la cantidad de información enviada y recibida y por lo tanto a disminuir la cantidad de transacciones TCP, que son susceptibles a los problemas de desempeño.

4.1.2.1 Compresión de cabeceras

Compresión de cabeceras TCP/IP de Van Jacobson. Esquema de compresión de datos creado por el científico Van Jacobson y descrito en el RFC 1144, fue diseñado especialmente para mejorar el rendimiento del TCP/IP sobre enlaces seriales de baja velocidad. El esquema de compresión de Van Jacobson reduce las cabeceras normales de TCP/IP de 40 bytes a 3-4 bytes en promedio. Realiza dicha compresión manteniendo información sobre el estado de la conexión TCP en ambos lados del enlace y solo enviando las diferencias en los encabezados que cambian, dando como resultado un gran ahorro de información enviada y es especialmente útil cuando los paquetes son pequeños como en las sesiones interactivas, como los *chats* o las sesiones de Telnet. El esquema de Compresión de Cabeceras de Van Jacobson, compresión VJ o simplemente Compresión de Cabeceras, es utilizado principalmente como una opción en las implementaciones del protocolo de enlace PPP, adicionalmente existe una versión de esta compresión para el protocolo SLIP llamada CSLIP.

El *Robust header compression* (ROHC), compresión robusta de cabeceras es un esquema estandarizado de comprimir las cabeceras IP, UDP, RTP y TCP definido en le RFC 3095. Este esquema de compresión difiere de otros como RFC 1144 (Van Jacobson) y el RFC 2508 por el hecho de tener un muy buen desempeño sobre enlaces con alta pérdida de paquetes, como es el caso de los enlaces inalámbricos. El ROHC tiene tres modos de operación: unidireccional, bidireccional optimista y bidireccional confiable. ROHC comprime los 40 o 60 bytes de las cabeceras TCP/IP o IP/UDP/RTP a típicamente solo de 1 a 3 bytes, utilizando un método de compresión llamado W-LSB.

4.1.2.2 Compresión de carga útil

Los protocolos de aplicación normales usados en el TCP/IP, no incluyen soporte para la compresión de la información en curso, por lo que si se quiere comprimir la información a enviar es necesario utilizar uno de las dos siguientes maneras:

- Comprimiendo toda la información previo al envío, como en el caso de la utilización del gzip en el HTTP 1.1 o de los formatos de archivo comprimidos para el caso del ftp.
- Comprimiendo el flujo de información mientras se transmite a través de un túnel o VPN (*virtual private network*). Ejemplos de estos son el OpenVPN que utiliza el protocolo de compresión LZO y el VTUN que utiliza LZO o Zlib. La ventaja de este método es que funciona para cualquier protocolo de aplicación.

4.1.3 *Caching*

Consiste en el almacenamiento de los objetos o archivos mas utilizados en un cierto de medio de almacenamiento, como puede ser un disco duro y/o memoria RAM, para su utilización mas inmediata en subsiguientes peticiones. No para todos los protocolos es factible de realizar *caching*, solo los que implican información mas o menos estática que se consulta repetidamente, normalmente la transferencia de archivos (FTP), la transferencia de páginas *web* (HTML) e imágenes y las consultas de resolución de nombres a IPs (DNS). Este método ayuda a compensar el tiempo de espera del cliente por la información solicitada y normalmente se pueden obtener entre el 30 y 50% de los objetos en el *cache*, pudiendo mejorar conforme la cantidad de objetos en el *cache* aumenta.

El protocolo http tiene un conjunto algo complicado de opciones y características que los clientes y los servidores pueden usar para controlar si los documentos u objetos son o no almacenados en el *cache* y cuando dichas copias pueden ser reusadas. Algunos sitios de *web* son amigables con el *caching* y otros no.

Existen dos tipos de *caching*: del lado del cliente y del lado del servidor. Los caches del lado del cliente, llamado a veces *forward caches*, atienden a usuarios locales y normalmente son utilizados por los ISPs, las universidades y las empresas, para atender a sus usuarios. Los caches del lado del servidor son también llamados *reverse-caches o web accelerators*, son puestos enfrente de los servidores de origen para reducir la carga de solicitudes.

Existen protocolos como el ICP o el HTCP, para coordinar el trabajo colaborativo de caches en sitios distribuidos o remotos.

4.1.4 *Spoofing*

En el *spoofing*, como se llama en ingles, se usan paquetes de aceptación falsos para engañar al transmisor y hacerle mandar más paquetes de los que se han actualmente entregado al receptor final.

Normalmente el *spoofing* de protocolo es mas importante y utilizado en el acceso a Internet satelital, debido al gran retardo que los paquetes deben recorrer desde la antena terrestre hasta el satélite y luego hasta la antena maestra y que la mayoría de protocolos no se comportan bien con dichos retardos grandes, situación agravada por el algoritmo de arranque utilizado con frecuencia en las implementaciones de TCP. El *spoofing* ayuda a disminuir el retardo para el flujo de paquetes utilizando normalmente las confirmaciones del protocolo de enlace (HDLC, LAPB, etc), para generar reconocimientos locales con retardo pequeño en cada extremo de la conexión.

El *spoofing* presenta varios problemas:

- No se apega a las especificaciones formales del TCP que garantizan la entrega completa de paquetes al receptor final.
- Si hay un fallo en la conexión, los paquetes que se han aceptado vía paquetes de aceptación falsos pueden perderse para siempre.

4.1.5 Manejo de la calidad de servicio

Cuando el Internet y sus protocolos asociados fueron creados, no se percibió la necesidad para un manejo de la calidad de servicio, de hecho el Internet como un todo es un sistema *best effort* o de mejor esfuerzo.

A pesar de esto dentro del formato de los paquetes TCP/IP existen cuatro bits de "tipo de servicio" y tres bits de "precedencia", pero en la práctica rara vez se usan y es por esto que es necesario implementar otros métodos y políticas para permitir un control de la calidad de servicio que se reciben los usuarios finales.

La calidad de servicio (QOS) en la transmisión de datos se refiere al rendimiento de extremo a extremo tal y como los percibe el usuario final. Los parámetros que se manejan en una política de QOS son: el retardo, la variación del retardo (*jitter*) y la pérdida de paquetes. Una red debiera garantizar un cierto nivel de calidad de servicio para un nivel de tráfico que sigue un conjunto especificado de parámetros.

Con el objeto de la administración de la calidad de servicio las aplicaciones se pueden clasificar por el tipo de servicio que necesitan para funcionar óptimamente:

- Inelástico: El cual implica que las aplicaciones necesitan de un cierto nivel de ancho de banda y retardo para funcionar. Por un lado, significa que más ancho de banda de lo necesario se desperdicia y menos hace que las aplicaciones sean inservibles. Por otro, lo contrario se da para el retardo, uno de los ejemplos para este tipo de aplicaciones son: VoIP y las videoconferencias.
- Elástico: En este caso, las aplicaciones no se ven afectadas por los cambios del ancho de banda, ni el retardo y pueden funcionar con valores bastante flexibles de estos.

La implementación de las Políticas de Calidad de Servicio se puede enfocar en varios puntos según los requerimientos de la red, lo cuales son:

- Control de tráfico por asignación de ancho de banda en forma diferenciada
- Control de tráfico por prioridades de acuerdo al tipo de tráfico
- Evitar y/o administrar la congestión
- Evitar y/o administrar las colisiones en la red

4.1.5.1 Control de tráfico por asignación de ancho de banda en forma diferenciada

Antes de comenzar es necesario definir dos términos que son fundamentales:

- Ancho de banda es la capacidad del canal de comunicaciones.
- Tráfico es la cantidad de datos que tratan de usar el ancho de banda disponible.

Controlar o manejar el tráfico reduce el consumo del ancho de banda. Este proceso es llamado en inglés como *traffic shaping*, *traffic management* o *bandwidth management*, consiste en medir y controlar el tráfico en las redes de datos con el objeto de optimizar o garantizar el desempeño, la latencia y/o el ancho de banda dentro de dicha red y permite que el tráfico fluya mas uniformemente, eliminando los picos. El manejo del ancho de banda trabaja con los conceptos de clasificación del tráfico, manejo de colas y el reparto equitativo. Esto es importante tomando en cuenta que cualquier enlace de datos tiene un límite de ancho de banda y si se genera más tráfico del que puede soportar da lugar a una congestión.

El tráfico que es menor o igual a la cantidad especificada por el manejador de ancho de banda es eliminado o retardado usando una de las siguientes técnicas:

- *Policing* (Descartado del exceso de paquetes)
- *Queuing* (Demora de los paquetes en tránsito)

En la práctica el manejo del ancho de banda normalmente se realiza creando colas de tráfico en forma automática o manual y controlando la tasa de transferencia de entrada y/o salidas a estas según uno de los siguientes parámetros:

- El IP fuente o destino
- Tamaño del paquete
- Interfase en el que se recibe el paquete
- Los puertos fuente o destino
- El tipo de tráfico (transferencia de archivos, P2P, voz, video, etc)
- Una combinación de los anteriores

Normalmente para que sea efectivo es necesario detectar los flujos de tráfico en la entrada o salida de la red, identificando y separando dichos flujos y asignándoles diferentes tasas de transferencia, según su tipo.

Los beneficios que presenta el manejo del ancho de bando cuando gran cantidad de flujos de tráfico pasan por un cuello de botella lógico o físico son:

- Menos variación del retardo o *jitter*
- Reducción de la pérdida total de paquetes
- Menor retardo
- Menor congestión en *buffers* de transmisión y recepción

4.1.5.2 Control de tráfico por prioridades de acuerdo al tipo de tráfico

En este tipo de control de tráfico se marcan los paquetes según el tipo de tráfico en la fuente o por los equipos por donde transitan, poniéndolos en colas de tráfico, y asignándoles prioridades, pero sin controlar la tasa de transferencia o ancho de banda y despachando los paquetes en las colas en base a dicha prioridad. Esta asignación puede ser automática o manual. Algunas formas de control de tráfico por prioridades:

- A nivel físico:

    - DiffServ DSCP a nivel IP según RFC 2474
    - DiffServ IP precedence, según RFC 3168
    - DiffServ 802.1p a nivel de MAC (capa de enlace)

- A nivel de aplicación por:

    - El IP fuente o destino
    - Tamaño del paquete
    - Interfase en el que se recibe el paquete
    - Los puertos fuente o destino
    - El tipo de tráfico (transferencia de archivos, P2P, voz, video, etc)
    - Una combinación de los anteriores

4.1.5.3 Evitar y/o administrar la congestión

La administración de la congestión se enfoca en el manejo de la congestión una vez esta se haya presentado, tratando de mantener un flujo de datos a través de uno o múltiples colas sin que ellas se saturen. La administración de la congestión solo se puede aplicar a protocolos que internamente tienen mecanismos de control, como por ejemplo el protocolo TCP.

Evitar la congestión consiste principalmente en procedimientos para tratar de anticiparse y así que no se saturen los *buffers* de transmisión o recepción. Normalmente implican el aviso previo o el descartado de paquetes. Algunas formas de evitar la congestión son:

- ECN según RFC 3168
- RED y WRED
- Cisco AQM: Dynamic buffer limiting (DBL)

4.1.5.4 Evitar y/o administrar las colisiones en la red

Evitar las colisiones consiste en anticiparse a la colisión como el en caso del protocolo CSMA/CA o señalar la intención antes de hacerlo como en el caso de MACA.

La administración de las utiliza algún protocolo de planificación el cual indica a los transmisores el orden en el que pueden transmitir, la cantidad de tiempo y/o la cantidad de datos. Algunos ejemplos de este tipo de protocolo son:

- Karlnet TurboCell
- WiFi 802.11e
- WiMAX 802.16
- Los protocolos de asignación de canales en redes GSM o satelitales.

4.1.6 *Prefetch*

En el proceso normal del acceso de una página *web*, el navegador al recibir el URL a accesar sigue los siguientes pasos:

1. Primero resuelve el IP correspondiente al nombre del servidor que se escribe en el URL.

2. Luego trata de establecer una conexión con el servidor remoto y solicita la página *default* contenida en dicho URL, normalmente un archivo con extensión .htm o .html.
3. Al recibir dicho archivo realiza una escaneo, análisis sintáctico e interpretación de las etiquetas HTML contenidas en este, para su posterior presentación de la información al usuario.
4. Al realizar el paso anterior, si es necesario comienza a solicitar todos los objetos adicionales a los que se haga referencia en este, como archivos adicionales. Por ejemplo imágenes (gif, jpeg, flash, etc), sonido (.wav, mp3, etc) u otros.

Si se coloca algún dispositivo o software en el sitio central de acceso inalámbrico, se puede interceptar el archivo HTML realizar los pasos anteriores desde un principio y comenzar a solicitar los objetos o archivos, para que cuando al fín los solicite el navegador del cliente ya estén disponibles. Normalmente el *prefetch* se utiliza en combinación con alguna forma de *caching*. El *prefetch* tiene la ventaja que disminuye el tiempo de espera que normalmente se tiene para recibir la información desde su lugar de origen y que en el caso de las redes inalámbricas se suma también el dado por el retardo natural del medio y el ancho de banda y también por los retrasos provocados por las ineficiencias del protocolo TCP.

Otra forma de *prefetch* es la que se realiza del lado del cliente, llamada *link prefetching*, la cual aprovecha los tiempos muertos entre cargas de páginas para descargar páginas u objetos que el usuario pudiera visitar. Este método es mas efectivo cuando las páginas almacenadas en el servidor, al cual se accesa, contienen información sobre que se puede poner en un *cache*.

4.1.7 Mejoras al protocolo HTTP

El HTTP como protocolo sobre el cual funciona el servicio de *web* a evolucionado desde sus inicios hasta la versión 1.1 y con cada nueva versión a agregado nuevas características. A continuación se describen las versiones más destacadas:

- HTTP/0.9. En deshuso y nunca fue ampliamente utilizada. Solo soportaba un comando el GET y No soportaba cabeceras. Debido a que no soportaba POST el cliente no podía enviar mucha información al servidor.
- HTTP/1.0. El más utilizado, sobre todo en los servidores de *prox*y. Permite conexiones persistentes también llamadas *keep-alive*, las que permiten más de una pregunta-respuesta, por conexión TCP; aunque una a la vez cuando se negocia explícitamente y normalmente trabaja bien cuando no se utilizan servidores de *proxy*.
- HTTP/1.1. Ésta es la última versión y en ella las conexiones persistentes están habilitadas por defecto y trabaja muy bien con los servidores de *proxy*. Tres de las características que distinguen a la versión 1.1 son: *pipelining*, conexiones persistentes y compresión, que a continuación se describen en forma breve:

4.1.7.1 *Pipelining*

El *pipelining* apareció en la versión 1.1 y permite tanto a los clientes, como enviar múltiples solicitudes, sin esperar por la respuesta. Los servidores también pueden enviar múltiples respuestas, sin cerrar la conexión TCP. Este método requiere soporte tanto del cliente como del servidor. Para los servidores se requiere que aunque no envié múltiples respuestas sí acepte múltiples solicitudes para considerarse compatible con HTTP 1.1. Este método de trabajo ayuda en las redes inalámbricas por las siguientes razones:

- Como permite enviar múltiples solicitudes al mismo tiempo, permite a los servidores prepararse para la carga y potencialmente transferir las respuestas más rápidamente al cliente.
- Se tienen menos envíos y recepciones y por lo tanto tiempos de carga de páginas más rápidos, lo que ayuda especialmente a las conexiones con retardos altos como en los enlaces satelitales, al no ser necesarias una solicitud por cada archivo.
- Como permite enviar varias preguntas y respuestas en el mismo paquete TCP, reduce la cantidad de paquetes TCP necesarios para una transacción y por lo tanto también reduce la carga de la red inalámbrica.

4.1.7.2 Conexiones persistentes

En el HTTP/0.9 y HTTP/1.0, un cliente envía una solicitud al servidor y entonces el servidor le envía una respuesta devuelta, luego de esto la conexión TCP se cierra. El HTTP/1.1, permite conexiones persistentes, lo que permite al cliente enviar una solicitud y recibir una respuesta y luego enviar solicitudes adicionales y recibir respuestas adicionales, una a una sin cerrar la conexión TCP. Como la sesión TCP no es desconectada hasta la última solicitud la sobrecarga debido a las conexiones TCP es mucho menor.

Hay una extensión para permitir la persistencia con el HTTP/1.0 pero esta posibilidad está muy limitada debido a que el HTTP/1.0 no tiene una forma no ambigua de delimitar los mensajes o solicitudes. Esta extensión para persistencia utiliza el encabezado llamado *Keep-Alive*, mientras el HTTP/1.1 usa el encabezado *Connection*, por lo tanto el HTTP/1.1 puede escoger soportar solo la forma de persistencia propia del HTTP/1.1 o las dos HTTP/1.0 y HTTP/1.1.

Algunos clientes y servidores con soporte para HTTP/1.1 no implementan las conexiones persistentes o las tienen deshabilitadas en su configuración por defecto.

4.1.7.3 Compresión

El RFC 2616 describe un método de compresión de las conexiones HTTP entre el servidor Web y el navegador del cliente. Esta tecnología asume que el servidor de web es capaz de comprimir y codificar el contenido que envía y el navegador del cliente es capaz de realizar el proceso inverso, lo cual es cierto para la mayoría de los navegadores importantes como Netscape, Mozilla, Firefox, Internet Explorer, Opera y otros e implica que no se necesita de software adicional en el lado del cliente. La compresión de HTTP utiliza algoritmos de compresión de uso público, principalmente el gzip y está incluido en la especificación HTTP/1.1. de el protocolo.

4.1.8 Protocolos de la capa de enlace

Se pueden utilizar dos procedimientos a nivel de la capa de enlace para minimizar o en ciertos casos eliminar la pérdida de paquetes debido a los medios de transmisión hostiles como el caso de las redes inalámbricas. Los dos procedimientos que se pueden utilizar son:

- Detección y corrección de errores en la transmisión
- La retransmisión de la información dañada

El primer caso llamado normalmente corrección anticipada (FEC, *forward error correction*) consiste en enviar información redundante, permitiendo al receptor detectar y corregir los datos sin necesidad de pedir al trasmisor que los reenvíe de nuevo. El problema de este método consiste en los requerimientos adicionales de ancho de banda para acomodar la información extra. Normalmente se aplica en situaciones donde la retransmisión es muy costosa o imposible. Ejemplos de este tipo de algoritmos de corrección de errores son:

- Reed Solomon
- Viterbi
- *Turbo codes*
- Golay
- Hamming
- BCH, Bose, Ray-Chaudhuri, Hocquenghem

El segundo caso consiste en solicitar y retransmitir la información que se ha dañado o no se a recibido. A este procedimiento se le llama normalmente requerimiento automático de retransmisión (ARQ, *automatic repeat-request*). Existen varias formas de ARQ:

- ARQ de parada y espera
- ARQ de repetición selectiva
- ARQ por el método de ventana corrediza

Un ejemplo complejo de ARQ es el protocolo RLP que se describe a continuación:

El RLP (radio link protocol) es un protocolo para redes inalámbricas (típicamente celulares) que utiliza el protocolo ARQ, que fue inventado por Phil Karn en 1990.

Muchos de los enlaces inalámbricos están diseñados para proveer un máximo en condiciones ideales del 1% de pérdida, lo que es tolerable para la mayoría de los *codecs* de voz usados en las redes celulares. El RLP detecta las pérdidas de paquetes y realiza retransmisiones hasta bajar la pérdida hasta .01%, que es adecuado para aplicaciones TCP/IP. El RLP también implementa la fragmentación y el reensamblado de flujos y en algunas ocasiones la entrega en orden. Las formas nuevas de RLP también proveen de empaquetado y compresión, mientras las formas viejas de este utilizaban el PPP para dicho propósito.

El transporte RLP nunca conoce qué tan grande es un paquete, del enlace inalámbrico que puede transportar, en vez de eso el manejador del enlace inalámbrico determina el tamaño del segmento y llama al RLP para que forme un segmento en demanda para la transmisión. La mayoría protocolos utilizados en redes inalámbricas como el 802.11b o el TCP/IP usan segmentos de tamaño fijo, lo que los hace menos flexibles y inevitablemente presentan bloqueos o grandes pérdidas en caso de desvanecimientos fuertes de la señal en el medio inalámbrico.

El protocolo RLP puede utilizar ACK o NACK, pero debido a que en la mayoría de los enlaces inalámbricos el canal de retorno es muy caro y demandado como en las redes celulares, normalmente las implementaciones utilizan NACK cuando se reciben segmentos fuera de orden o que no se recibieron. Cuando el canal de transmisión está sin uso o inactivo, el RLP con NACK, puede eventualmente enviar el último segmento una segunda vez para llegar a obtener la meta de .01% de pérdida de paquetes. Esta transmisión duplicada se controla a través de un temporizador llamado "*flush timer*" puesto para que expire entre 300-500 milisegundos después de que la transmisión se ha terminado.

La mayoría de redes celulares, como la GSM y la CDMA, utilizan diferentes variaciones del RLP. En enero del 2006 la IEEE sacó la especificación 802.20 que utilize una de las más nuevas formas de RLP.

4.2 Métodos que solucionan el problema

Como el problema de desempeño radica en la forma en que se controla la congestión y el algoritmo que lo implementa, se han desarrollado dos formas diferentes para solucionarlo definitivamente, estos son:

1. Haciéndole ajustes al protocolo TCP y los algoritmos usados tradicionalmente para evitar la congestión. Estos métodos solo resuelven los problemas debido al inicio lento y al *congestion avoidance*, pero persisten el inicio de sesión de tres vías el y reconocimientos, el RWIN, el tiempo de reacción y el de muchas conexiones concurrentes.
2. Cambiando complemente el protocolo TCP, por otro completamente nuevo que incluye algoritmos diferentes y procedimientos diferentes, pero que presenta los mismos servicios a las capas inferiores y superiores. Estos tratan de darle solución a todos los problemas en conjunto.

La primera forma, incluye una serie de variantes del TCP, que normalmente en los sistemas operativos comunes, como Windows, Linux o Mac OS X, basta con cambiar un parámetro para usarlos.

En el caso de la segunda forma, normalmente es necesario insertar parejas de equipos o pilas de los nuevos protocolos dentro del camino normal del tráfico y que puedan hablar de un lado TCP estándar y del otro el protocolo nuevo, en una configuración conocida como de Proxy.

TCPEstándar ←→ NuevoProtocolo ← MedioFísico → NuevoProtocolo ←→ TCPEstándar

4.2.1 Otras implementaciones del TCP

Estas solo se enfocan en arreglar los problemas debido al algoritmo de congestión. Aunque hay muchas variantes del TCP, unas orientadas al aumento del desempeño sobre redes con mayor ancho de banda y/o enlaces con retardos grandes (*Fast*TCP, BIC-TCP, HSTCP, etc) y otras orientadas a enlaces con grandes pérdidas de paquetes, para efectos de este trabajo analizaremos solo las que tienen características importantes para las redes inalámbricas y/o están ampliamente disponibles en las implementaciones de redes inalámbricas, aunque no estén optimizados para ellas.

4.2.1.1 TCP Tahoe y TCP Reno

Unas de las primeras versiones de TCP llamadas Tahoe y Reno usan un algoritmo de control de congestión que incluye varios aspectos del esquema AIMD (*additive-increase-multiplicative-decrease*), o incremento aditivo y decrecimiento multiplicativo de las ventanas de transmisión, adicionalmente al esquema llamado *slow-start*, con el objetivo de evitar la congestión.

El TCP Reno que es la versión mas usada actualmente de TCP fue creado en 1990 para la versión 4.3 de BSD y agregó el concepto adicional de Recuperación Rápida o *fast recovery*. En el algoritmo de Recuperación Rápida cuando los segmentos han perdido, detectados por la recepción de 3 ACKs duplicados, la ventana de congestión es reducida a su valor umbral de arranque lento en vez de a su valor inicial mas pequeño posible de uno, como el caso de TCP Tahoe.

El algoritmo de *fast recovery* de Reno está optimizado para el caso en que hay pérdida de un solo paquete de una ventana de datos. El transmisor Reno transmite a lo sumo un paquete perdido por RTT. Reno mejora significativamente con respecto a Tahoe cuando un paquete de datos se pierde de una ventana de datos, pero puede sufrir problemas de desempeño cuando múltiples paquetes se pierden de una ventana de datos.

La ventaja más destacable es el hecho de no pasar a la fase de *slow start*, cada vez que se produce la pérdida de un segmento, haciendo más recomendable en los enlaces de banda ancha.

4.2.1.2 TCP Vegas

El TCP Vegas es un algoritmo de control de congestión que acentúa más la importancia del retardo de los paquetes sobre la tradicional forma de utilizar la pérdida de paquetes, como una forma de determinar la tasa a la que deben enviarse los paquetes.

Fue desarrollado en la Universidad de Arizona en 1994 y es usado como base del protocolo de la NASA SCPS.

El protocolo TCP Vegas aumenta el tamaño de la ventana hasta que ocurre la pérdida del paquete debida a congestión. Se fundamenta en el estudio del RTT que se estudia en todos los segmentos. Si RTT es grande se asume que la red está congestionada por lo que se disminuye el tamaño de la ventana. Si el RTT baja se determina que la red no está congestionada y puede aumentar la ventana.

El TCP Vegas trata de predecir de forma anticipada la pérdida de paquetes y la congestión antes de que ocurra, monitoreando los tiempos de retorno de los paquetes (RTT) y luego usando incrementos y decrementos aditivos en la ventana de congestión. Intenta mantener un ancho de banda estable y equitativo si hay suficientes *buffer*s en los *routers* de la red para soportarlo.

4.2.1.3 TCP Westwood+

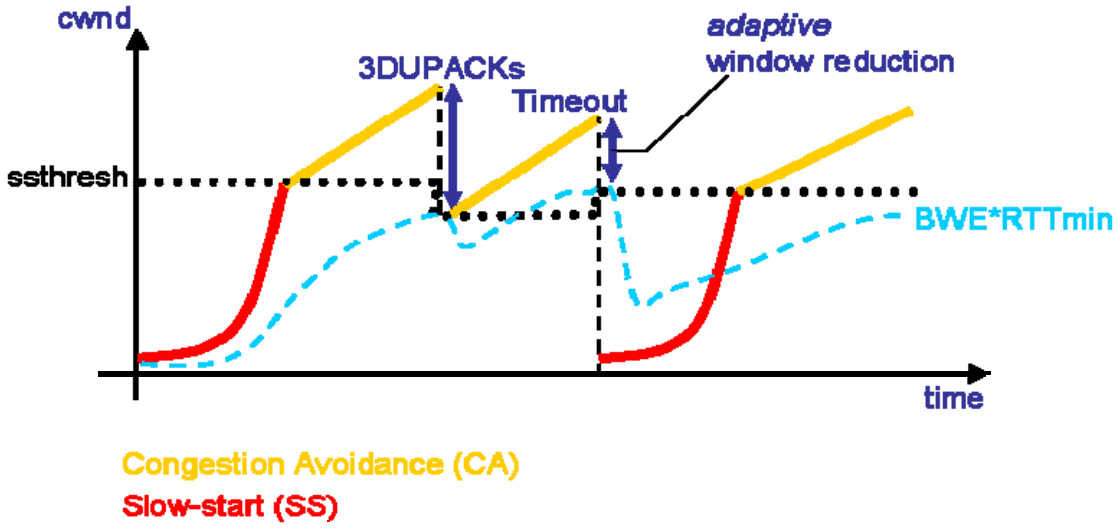
El TCP Westwood+ o TCPW+ es una modificación del lado del transmisor solamente del , TCP NewReno que optimiza el desempeño de la control de la congestión del TCP tanto sobre enlaces alámbricos, como inalámbricos , que presentan relación grande de ancho de banda contra retardo o con perdidas de paquetes o errores de transmisión potencialmente grandes y cargas de tráfico variables.

TCPW está basado en una estimación extremo a extremo de la disponibilidad de ancho de banda que realiza un análisis dinámico del flujo de reconocimientos en busca de información para mejorar los parámetros que controlan la congestión (el umbral de arranque lento o *slow start threshold* (ssthresh) y la ventana de congestión o *congestion window* (cwin), después de que se presenta un evento de congestionamiento, debido a la recepción seguida de tres reconocimientos duplicados o un *timeout*. El ancho de banda es estimado calculando apropiadamente la tasa de paquetes de reconocimiento recibidos. En contraste con el TCP Reno, que siempre disminuye la cwin a la mitad después de tres ACKs duplicados, el TCP Westwood+ ajusta en forma adaptiva los parámetros de congestión, tomando en cuenta el ancho de banda usado cuando la congestión se presentó.

El TCP Westwood+ utiliza un algoritmo llamado "*agile probing*", que es una modificación del *slow start* y adicionalmente utilize un esquema llamado PNCD

(persistent non congestion detection), para detectar la ausencia de congestión y así inducir la fase de *agile probing* para hacer expedita la utilización de anchos de banda mas grandes en forma dinámica.

Figura 9. **Comportamiento del TCP Westwood**



Fuente: documentación de TCP Westwood.

El TCP Westwood fue desarrollado en la Universidad de California en los Angeles UCLA por Saverio Mascolo en 1999 y refinado a su versión Westwood+ en años recientes.

4.2.1.4 TCP-Hybla

TCP Hybla elimina las penalizaciones que presenta el TCP con grandes retardos sobre enlaces terrestres o inalámbricos, basandose en una evaluación analítica de la dinámica de la ventana de congestión, que da como resultado las modificaciones necesarias para evitar los problemas de desempeño debido a los RTTs altos.

4.2.2 Conversión a otro protocolo

Estos tratan de resolver todos los problemas que presenta normalmente el TCP en los escenarios descritos. Entre los protocolos alternativos al TCP están algunos de dominio público (RLP, SCTCP y SCPS) y algunos propietarios (Packeteer XTP, Flash Networks BST, ICTCompress ITP y VenturiWireless VTP) y de los protocolos propietarios solo se hará una descripción general debido a la poca información técnica de ellos disponible.

4.2.2.1 SCTCP

SCTP *(stream control transmission protocol*) es un protocolo de comunicación de capa de transporte que fue definido por el grupo SIGTRAN de IETF en el año 2000. El protocolo está especificado en la RFC 2960, y la RFC 3286 brinda una introducción al mismo.

Como protocolo de transporte, podría considerárselo equivalente a TCP o UDP pues es capaz de operar en modo confiable o no confiable. En el modo confiable provee servicios similares a TCP, es decir, asegura la entrega confiable y ordenada de los mensajes, incluyendo control de congestión

Las ventajas de SCTP son:

- Capacidad de *multihoming*, en la cual uno (o dos) de los extremos de una asociación (conexión) pueden tener más de una dirección IP. Esto permite reaccionar en forma transparente ante fallas en la red.

- Entrega de los datos en trozos que forman parte de flujos independientes y paralelos (*multistreaming*) —eliminando así el problema de *head of the line blocking* que sufre TCP—, oséa esta orientado al flujo de mensajes y no al flujo de bytes como el TCP.
- Utiliza un inicio de sesión de cuatro vías, que a pesar de ser mas grande que el de tres del TCP, permite enviar datos incluidos desde el segundo mensaje, lo que hace que se puedan enviar datos antes que en el TCP.
- Es capaz de seleccionar y monitorizar caminos, seleccionando un camino primario y verificando constantemente la conectividad de cada uno de los caminos alternativos.
- Tiene un mecanismo de validación y asentimiento como protección ante ataques por inundación, proveyendo notificación de trozos de datos duplicados o perdidos.
- Solo utiliza SACKs.
- No permite sesiones parcialmente cerradas como en TCP.

SCTP fue diseñado inicialmente por el grupo SIGTRAN para transportar señalización telefónica SS7 sobre IP. La intención fue proveer en IP algunos de las características de confiabilidad de SS7. Por su versatilidad luego se ha propuesto utilizarlo en otras áreas, como por ejemplo para transportar mensajes de los protocolo DIAMETER o SIP.

4.2.2.2 SCPS

El SCPS (*space communication protocol standard*) son un conjunto de protocolos que se originaron en 1992 como una colaboración conjunta entre la NASA y la *US air force*, para desarrollar protocolos expresamente diseñados para los inóspitos ambientes en que se desarrollan las comunicaciones satelitales. Las metas de diseño fueron:

- La utilización de estándares de Internet y que fueran interoperables con los
- estándares de la IETF
- La mejor utilización del limitado ancho de banda disponible
- Utilización al máximo del canal
- Conservación de la potencia
- Prioritización del tráfico
- Que fuera tolerante a la conectividad intermitente
- Que soportara grandes asimetrías en los canales se ida y retorno

Esta compuesto de cuatro protocolos:

- SCPS-FP o SCPS *file protocol*, el cual es opcional
- SCPS-TP o SCPS *transport protocol*, este puede funcionar sobre el IP estándar o sobre el SCPS-NP
- SCPS-SP o SCPS *security protocol*, el cual es opcional
- SCPS-NP o SCPS *network protocol*, el cual es opcional

El protocolo SCPS-TP utiliza las siguientes modificaciones y extensiones al TCP:

- Escalado de la ventana, *window scaling* (RFC 1323)
- Medición del RTT, RTTM (RFC 1323)
- Protección Contra los Números de Secuencia Reutilizados (*protection against wrapped sequence numbers*, PAWS) (RFC 1323)
- Reconocimientos selectivos negativos (SNACK adaptado del RFC 1106)
- Compresión de cabeceras (adaptado del RFC 1144)
- El algoritmo de control de congestión puede ser TCP estándar, TCP Vegas u opcionalmente el no uso control de congestión, sino solo manejo de la tasa de transmisión.
- TCP *timestamps* (RFC 1323)
- *Corruption detection*

Algunas de estas características pueden usarse en conjunto con diferentes versiones de TCP, lo especial del SCPS, es su uso en conjunto y la utilización de algunos procedimientos nuevos. Una de las ventajas del SCPS-TP es que no es obligatorio que los dos extremos hablen SCPS, ya que es interoperable con el TCP estándar y si solo lo usa un extremo sus ventajas se verán solo en una dirección.

4.2.2.3 Packeteer XTP

Las características principales del protocolo BST son:

- Separa explícitamente la transferencia de datos de la información de control, usando paquetes diferentes para el control, que los usados para datos.
- Utiliza el concepto de circuitos virtuales
- Utiliza SACKs y NACKs para los reconocimientos y retransmisiones

- Control de tasa de transmisión, negociada entre las partes.
- Maneja prioridad y planificación de mensajes.
- Puede parametrizar el tráfico y negociar la calidad del servicio.
- No utiliza la perdida de paquetes como señal de congestionamiento.
- Utiliza un sistema de ventana deslizante dinámica, ajustada según el ancho de banda, el retardo y el número de conexiones simultaneas.
- Utiliza un método llamado *fast start* para abreviar las conexiones HTTP.
- Utiliza compresión del flujo de datos.
- Utiliza un procedimiento llamado *multicast* FAN-OUT, que permite convertir una conexión TCP *unicast* en una *multicast*.
- Usa *prefetch*

4.2.2.4 *Flash Networks* BST

Las características principales del protocolo BST son:

- El BST mantiene un túnel abierto entre el cliente y el servidor eliminando la necesidad del apretón de tres vías del TCP.
- Los parámetros sobre los que trabaja el protocolo se especifican desde un principio
- y tienen en cuenta las características específicas del enlace tales como el ancho de
- banda, el retardo, la tasa de errores y el MTU, lo que evita la necesidad del procedimiento de

Arranque Lento, permitiendo operar a máxima velocidad siempre y desde un principio.

- El BST reconoce los paquetes perdidos y no asume que se deban a congestión y simplemente reenvía los perdidos sin disminuir la velocidad.

- Utiliza un algoritmo de retransmisión selectivo de paquetes, el cual reconoce cuando un paquete se perdió o llegó en desorden, lo que evita su retransmisión innecesaria.
- El BST implementa políticas de control de ancho de banda, tanto para el servidor como para el receptor, para asegurar el uso eficiente y justo de este para todos los flujos de tráfico.
- Utiliza reconocimiento selectivos SACKs y reconocimiento negativos NACKs, para desminuir la sobrecarga del canal, sobre todo en situaciones de asimetría en el ancho de banda de bajada y subida.
- Maneja una ventana que no está relacionada con el RTT del enlace, de manera de utilizar al máximo la capacidad del enlace.
- Realiza compresión basada en Zlib para toda la información que se envía o recibe.
- Utiliza un protocolo propietario llamado *GetALL*, para el acceso a páginas Web el cual minimiza el dialogo necesario para obtener una página *web*.

4.2.2.5 ICT*Compress* ITP

Las características principales del protocolo ITP son:

- Funciona sobre UDP
- Utiliza información de los protocolos de aplicación para determinar el tratamiento que se dará en el transporte a cada segmento.
- Calcula los parámetros del enlace en el lado del receptor y luego los intercambia entre las partes.
- Usa NACKs
- Usa compresión predictiva y diferencial según cada protocolo
- Puede comprimir y descomprimir objetos de un documento de Office o de Flash en forma independiente del resto del documento, usando diferentes tipos de compresión.

4.2.2.6 Venturi Wireless VTP

Las características principales del protocolo VTP son:

- Control de flujo por control de tasa de transmisión, no usa ventana deslizante.
- Funciona sobre UDP.
- Tiene control de ancho de banda por aplicación y por usuario.
- Mantiene la sesión aún en condiciones de severa pérdida de paquetes.
- Utiliza multiplexación de solicitudes HTTP en un solo paquete.
- Utiliza *caching* de HTTP
- Mantiene en *cache* los objetos pre-comprimidos, para evitar la sobrecarga de tenerlos que comprimir cada vez que se van a enviar.
- Del lado del cliente maneja *prefetch*.
- Utiliza compresiones específicas dependiendo del tipo de documento, por ejemplo PPM, para documentos de Office y JavaScript y JPEG en imágenes.

5. CASO PRÁCTICO: ENLACE INALÁMBRICO LAN 802.11b DE LARGA DISTANCIA

5.1 Descripción del escenario de pruebas

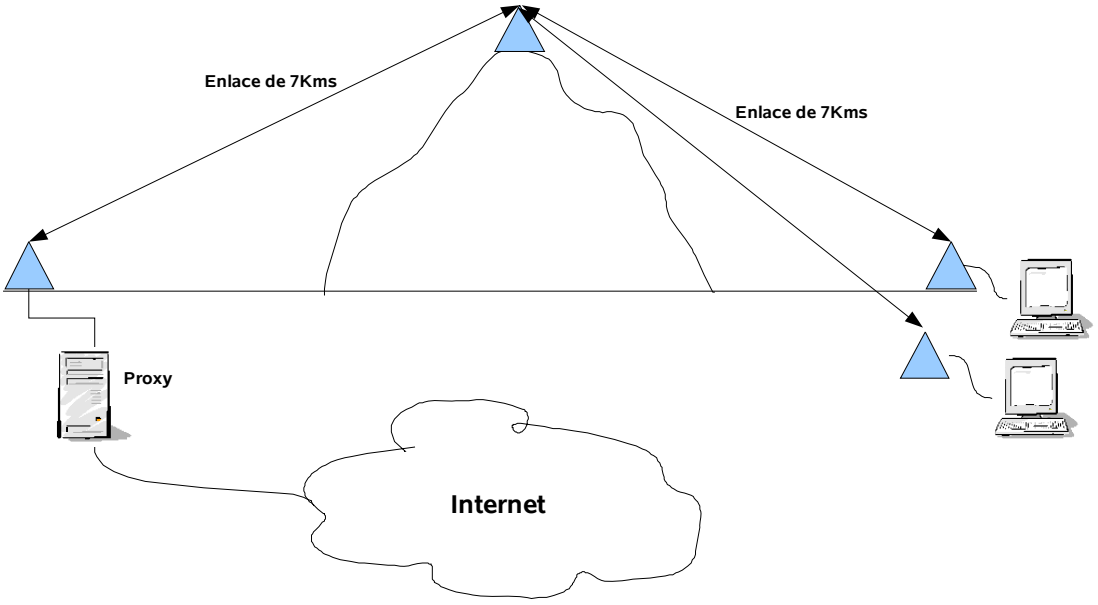
Por facilidad y costo de establecer un escenario de pruebas se implementó un enlace *wireless* LAN 802.11b a una distancia de aproximada de 14 kms, con un repetidor a medio camino en un sitio alto (Montebello), de manera de contar con dos ramas de 7 kms, en condiciones a propósito con pérdidas de paquetes, con la idea de poner a prueba los métodos descritos en capítulos anteriores. Se hará una descripción cualitativa de la implementación de los equipos de transmisión y antenas, pero no se entrarán en detalles técnicos sobre las funcionalidades de estos, ni sobre las características de propagación y del enlace, por no ser relevantes para el estudio.

Una comparación exhaustiva de todas las combinaciones posibles de métodos y su contribución individual o en grupo al mejoramiento del desempeño está fuera del ámbito de estudio de esta trabajo, más bien se tratará de aplicar la mayor cantidad de métodos posible y se harán mediciones y comparaciones antes y después de aplicar dichos métodos, con el objetivo de demostrar que el uso de los métodos descritos en la tésis tiene una real injerencia y mejora en el desempeño.

Las mediciones se pueden realizar en uno de tres escenarios, a saber:

1. Con tráfico controlado, oséa ciertos protocolos de prueba a ciertos sitios de prueba determinados. Visto de otra manera un escenario de pruebas controlado.
2. Con tráfico variado en condiciones de uso naturales de usuarios reales.
3. Escenario Mixto

Figura 10. **Diagrama del escenario de pruebas propuesto**



Los equipos a utilizar son:

En Montebello:

- Un radio de 500mWatts marca Hawking Tech
- Un router/AP Linksys WRT-54G con firmware original.
- Una antena de sector de 16dbi, 10grados de apertura vertical y 95grados de apertura horizontal de *Pacific Wireless*

En los puntos remotos:

- *Bridge* marca Senao SL-2611CB3 de 200mWatts(23dbm)
- Antena externa de 14dbi, 35grados de apertura vertical y horizontal, de *Pacific Wireless*.

- Como servidor una computadora Celeron de 2.66Mhz y 2Gb de RAM, con Linux Redhat *Enterprise* 4 (Kernel 2.6.9-19). Como cliente una laptop Toshiba Satellite, Celeron M de 1.7Mhz y 512Mbytes de RAM con Windows XP.

Observaciones:

- Se escogió Linux en el servidor debido a la facilidad de manipulación de los parámetros del TCP, en comparación de Windows y adicionalmente a la mayor opción de herramientas de medición.
- Tomando en cuenta que el TCP es un protocolo extremo a extremo, el software que corre el equipo en sitio de repetición no afecta, ni interviene en las pruebas y mediciones.
- Para evitar que las sesiones TCP se realicen directamente entre los servidores de Internet y los clientes, se instaló un *proxy* de *web*, de manera de partir las sesiones en dos (Una del cliente al *proxy* y otra del *proxy* al servidor en Internet), por lo que las pruebas se limitan al protocolo de *web* (HTTP) hacia Internet y a ftp y ssh entre el cliente y el servidor de *proxy*.
- Existe una gran variedad de herramientas de medición que se pueden utilizar para realizar las mediciones, pero haciendo un análisis previo de todas las opciones que se encontraron[2] se decidió utilizar las siguientes:

    - Ping: Este enviar un paquete de prueba icmp hacia un cierto IP y al recibir la respuesta calcular el tiempo de ida y vuelta.
    - Iperf: Este software genera tráfico entre un cliente y un servidor y permite obtener la capacidad real total del canal.
    - *Fasterfox*: Modulo que se agrega al navegador Mozilla Firefox y que permite contabilizar el tiempo de carga de una página dada.

[2] http://www.cs.columbia.edu/~hgs/internet/tools.html
http://dast.nlanr.net/NPMT/

- Tstat: Este es un software que captura los paquetes y genera gráficas y estadísticas sobre estos, a nivel de la capa de transporte reconstruyendo las sesiones TCP o flujos. De los 40 reportes de mediciones estándar que se pueden obtener con esta herramienta se usaron 16.

5.2 Métodos a implementar

Por la falta de disponibilidad de recursos y tiempo se escogieron los métodos cuyo efecto es susceptible de medir con facilidad y precisión para este escenario de pruebas y son:

1. Ajuste de parámetros TCP
2. TCP BIC y TCP Westwood
3. Utilización de un *cache* de páginas *web*

El TCP BIC es el estándar utilizada en las implementaciones del *kernel* 2.6.x que y si bien fue diseñado para mejorar el desempeño del protocolo TCP, no fue diseñado específicamente para escenarios inalámbricos, en cambio el TCP Westwood fue pensado principalmente el escenarios inalámbricos.

Como se puede ver a continuación son bastantes los parámetros del TCP, con los cuales se puede jugar, sin embargo, trabajaremos específicamente con solo 7 de estos que están marcados en negritas, los demás se dejarán tal y como aparecen en el siguiente listado:

net.ipv4.tcp_bic_beta = 819
net.ipv4.tcp_tso_win_divisor = 8
net.ipv4.tcp_moderate_rcvbuf = 1
net.ipv4.tcp_bic_low_window = 14
net.ipv4.tcp_bic_fast_convergence = 1

net.ipv4.tcp_bic = 0
net.ipv4.tcp_vegas_gamma = 2
net.ipv4.tcp_vegas_beta = 6
net.ipv4.tcp_vegas_alpha = 2
net.ipv4.tcp_vegas_cong_avoid = 0
net.ipv4.tcp_westwood = 1
net.ipv4.tcp_no_metrics_save = 0
net.ipv4.tcp_low_latency = 0
net.ipv4.tcp_frto = 0
net.ipv4.tcp_tw_reuse = 0
net.ipv4.tcp_adv_win_scale = 2
net.ipv4.tcp_app_win = 31
net.ipv4.tcp_rmem = 4096 87380 174760
net.ipv4.tcp_wmem = 4096 16384 131072
net.ipv4.tcp_mem = 49152 65536 98304
net.ipv4.tcp_dsack = 1
net.ipv4.tcp_ecn = 0
net.ipv4.tcp_reordering = 3
net.ipv4.tcp_fack = 1
net.ipv4.tcp_orphan_retries = 0
net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_rfc1337 = 0
net.ipv4.tcp_stdurg = 0
net.ipv4.tcp_abort_on_overflow = 0
net.ipv4.tcp_tw_recycle = 0
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_fin_timeout = 60
net.ipv4.tcp_retries2 = 15
net.ipv4.tcp_retries1 = 3
net.ipv4.tcp_keepalive_intvl = 75
net.ipv4.tcp_keepalive_probes = 9
net.ipv4.tcp_keepalive_time = 7200
net.ipv4.tcp_max_tw_buckets = 180000
net.ipv4.tcp_max_orphans = 16384
net.ipv4.tcp_synack_retries = 5
net.ipv4.tcp_syn_retries = 5
net.ipv4.tcp_retrans_collapse = 1
net.ipv4.tcp_sack = 1
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_timestamps = 1

La aplicación de los métodos a probar se puede realizar de una de las siguientes tres formas:

1. Sólo del lado del punto central
2. Sólo del lado de los puntos remotos
3. Tanto del lado del punto central como de los puntos remotos

Para limitar la extensión de las mediciones se optó por realizar los cambios solo en el punto central de acceso a Internet.

5.3 Mediciones antes de aplicar cambios

Se realizaron las siguientes pruebas:

5.3.1 Ping de 1000 paquetes de 32 bytes

Reply from 192.168.2.1: bytes=32 time=1ms TTL=64
Reply from 192.168.2.1: bytes=32 time=2ms TTL=64
Reply from 192.168.2.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.2.1:
Packets: Sent = 1000, Received = 954, Lost = 46 (4% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 266ms, Average = 12ms

5.3.2 Descarga desde el servidor por FTP de dos archivos

Se logró una tasa de transferencia de entre 72 y 100Kbytes por segundo.

Archivo de 37,088 Kbytes	7.36 min
Archivo de 52,781 Kbytes	8.52 min

5.3.3 Subida desde el cliente por FTP de los mismos dos archivos

Se logró una tasa de transferencia de entre 39.5Kbytes por segundo, pero no se logró terminar la transferencia de ninguno de los dos archivos a pesar que se probó en varias ocasiones.

5.3.4 Descarga desde el servidor por HTTP de los mismos dos archivos

Se logró una tasa de transferencia de entre 50 y 59Kbytes por segundo.

Archivo de 37,088 Kbytes	15.30 min
Archivo de 52,781 Kbytes	17.42 min

5.3.5 Descarga desde Internet de 10 páginas al azar

Página	Tiempo de Carga
www.disney.com	34.219s
www.univision.com	17.281s
www.ibm.com	12.657s
www.yahoo.com	10.563s
www.sharpusa.com	39.735s
www.konicaminolta.com	35.310s
www.alcatel-lucent.com	25.938s
www.terra.es	52.765s
www.osnews.com	25.422s
www.mozilla.org	10.609s

5.3.6 Prueba de capacidad de transmisión a través de la herramienta iperf

```
C:\iperf>iperf -c 192.168.2.1 -i 1
------------------------------------------------------------
Client connecting to 192.168.2.1, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1916] local 192.168.2.110 port 3011 connected with 192.168.2.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[1916]  0.0- 1.0 sec  16.0 KBytes   131 Kbits/sec
[1916]  1.0- 2.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  2.0- 3.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  3.0- 4.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916]  4.0- 5.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916]  5.0- 6.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  6.0- 7.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  7.0- 8.0 sec  16.0 KBytes   131 Kbits/sec
```

```
[1916]  8.0- 9.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  9.0-10.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916] 10.0-11.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 11.0-12.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 12.0-13.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 13.0-14.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 14.0-15.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 15.0-16.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 16.0-17.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  0.0-20.8 sec  64.0 KBytes  25.2 Kbits/sec

[root@proxy proxy]# iperf -c 192.168.2.110 -i 1
------------------------------------------------------------
Client connecting to 192.168.2.110, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.2.1 port 33069 connected with 192.168.2.110 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  64.0 KBytes   524 Kbits/sec
[  3]  1.0- 2.0 sec  32.0 KBytes   262 Kbits/sec
[  3]  2.0- 3.0 sec  32.0 KBytes   262 Kbits/sec
[  3]  3.0- 4.0 sec  48.0 KBytes   393 Kbits/sec
[  3]  4.0- 5.0 sec  64.0 KBytes   524 Kbits/sec
[  3]  5.0- 6.0 sec  56.0 KBytes   459 Kbits/sec
[  3]  6.0- 7.0 sec  24.0 KBytes   197 Kbits/sec
[  3]  7.0- 8.0 sec  48.0 KBytes   393 Kbits/sec
[  3]  8.0- 9.0 sec  48.0 KBytes   393 Kbits/sec
[  3]  9.0-10.0 sec  48.0 KBytes   393 Kbits/sec
[  3]  0.0-10.4 sec   472 KBytes   371 Kbits/sec
```

5.4 Mediciones después de aplicar cambios

5.4.1 Ping de 1000 paquetes de 32 bytes

```
Reply from 192.168.2.1: bytes=32 time=1ms TTL=64
Reply from 192.168.2.1: bytes=32 time=2ms TTL=64
Reply from 192.168.2.1: bytes=32 time=2ms TTL=64
```

Ping statistics for 192.168.2.1:
Packets: Sent = 1000, Received = 989, Lost = 11 (1% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 517ms, Average = 9ms

5.4.2 Descarga desde el servidor por FTP de dos archivos

Se logró una tasa de transferencia de entre 90 y 130Kbytes por segundo.

Archivo de 37,088 Kbytes	6.38 min
Archivo de 52,781 Kbytes	7.36 min

5.4.3 Subida desde el cliente por FTP de los mismos dos archivos

Se logró una tasa de transferencia de entre 26 y 28.5Kbytes por segundo, logrando completar las tranferencias.

Archivo de 37,088 Kbytes	23.35 min
Archivo de 52,781 Kbytes	32.05 min

5.4.4 Descarga desde el servidor por HTTP de los mismos dos archivos

Se logró una tasa de transferencia de entre 15 y 17 Kbytes por segundo.

Archivo de 37,088 Kbytes	39.07 min
Archivo de 52,781 Kbytes	55.30 min

5.4.5 Descarga desde Internet de 10 páginas al azar

Página	Tiempo de carga inicial	Tiempo de Carga desde el Cache
www.disney.com	50.110s	4.219s
www.univision.com	17.812s	3.172s
www.ibm.com	13.578s	2.969s
www.yahoo.com	16.310s	2.703s
www.sharpusa.com	17.516s	12.609s
www.konicaminolta.com	11.328s	0.922s
www.alcatel-lucent.com	9.828s	0.933s
www.terra.es	31.203s	7.688s
www.osnews.com	35.187s	4.234s
www.mozilla.org	9.281s	2.704s

5.4.6 Prueba de capacidad de transmisión a través de la herramienta iperf

```
C:\iperf>iperf -c 192.168.2.1 -i 1
------------------------------------------------------------
Client connecting to 192.168.2.1, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1916] local 192.168.2.110 port 4210 connected with 192.168.2.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[1916]  0.0- 1.0 sec  16.0 KBytes   131 Kbits/sec
[1916]  1.0- 2.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  2.0- 3.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916]  3.0- 4.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916]  4.0- 5.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916]  5.0- 6.0 sec  8.00 KBytes  65.5 Kbits/sec
[1916]  6.0- 7.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  7.0- 8.0 sec  16.0 KBytes   131 Kbits/sec
[1916]  8.0- 9.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  9.0-10.0 sec  16.0 KBytes   131 Kbits/sec
[1916] 10.0-11.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 11.0-12.0 sec  0.00 Bytes  0.00 bits/sec
[1916] 12.0-13.0 sec  0.00 Bytes  0.00 bits/sec
[1916]  0.0-16.2 sec  88.0 KBytes  44.5 Kbits/sec

[root@proxy ~]# iperf -c 192.168.2.110 -i 1
------------------------------------------------------------
Client connecting to 192.168.2.110, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
```

[3] local 192.168.2.1 port 34903 connected with 192.168.2.110 port 5001

[ID] Interval Transfer Bandwidth

[3] 0.0- 1.0 sec 72.0 KBytes 590 Kbits/sec

[3] 1.0- 2.0 sec 72.0 KBytes 590 Kbits/sec

[3] 2.0- 3.0 sec 48.0 KBytes 393 Kbits/sec

[3] 3.0- 4.0 sec 64.0 KBytes 524 Kbits/sec

[3] 4.0- 5.0 sec 72.0 KBytes 590 Kbits/sec

[3] 5.0- 6.0 sec 48.0 KBytes 393 Kbits/sec

[3] 6.0- 7.0 sec 56.0 KBytes 459 Kbits/sec

[3] 7.0- 8.0 sec 24.0 KBytes 197 Kbits/sec

[3] 8.0- 9.0 sec 64.0 KBytes 524 Kbits/sec

[3] 9.0-10.0 sec 104 KBytes 852 Kbits/sec

[3] 0.0-10.5 sec 632 KBytes 491 Kbits/sec

Las pruebas comenzaron a las 9 am y terminaron a las 2:30 pm, tomando en el primer periódo las pruebas sin los parámetros de TCP aplicados, con BIC TCP y sin *cache* de *web* y durante el segundo período con los parámetros TCP aplicados, TCP Westwood y con *cache* de *web* activado. Generando las siguientes gráficas:

Figura 11. ***IP packet lenght [byte] - incoming packets***

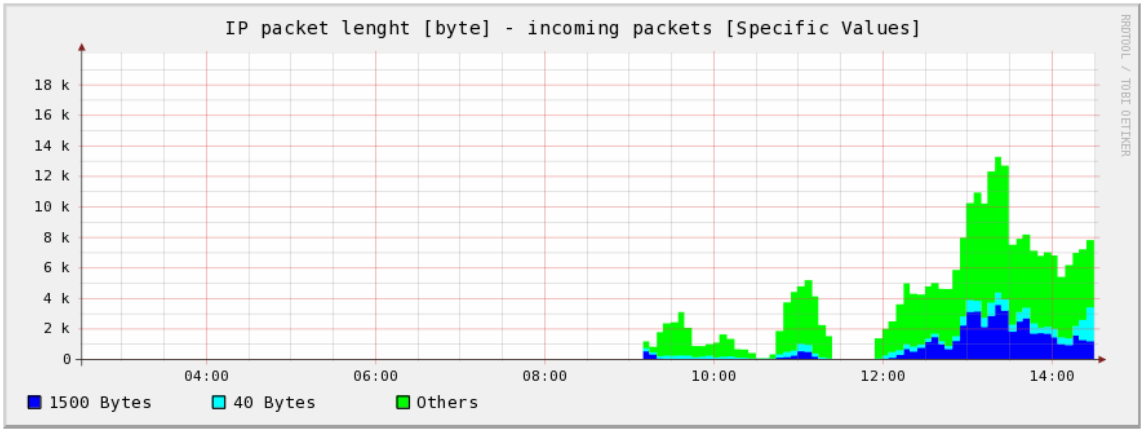


Figura 12. ***IP packet lenght [byte] - local packets***

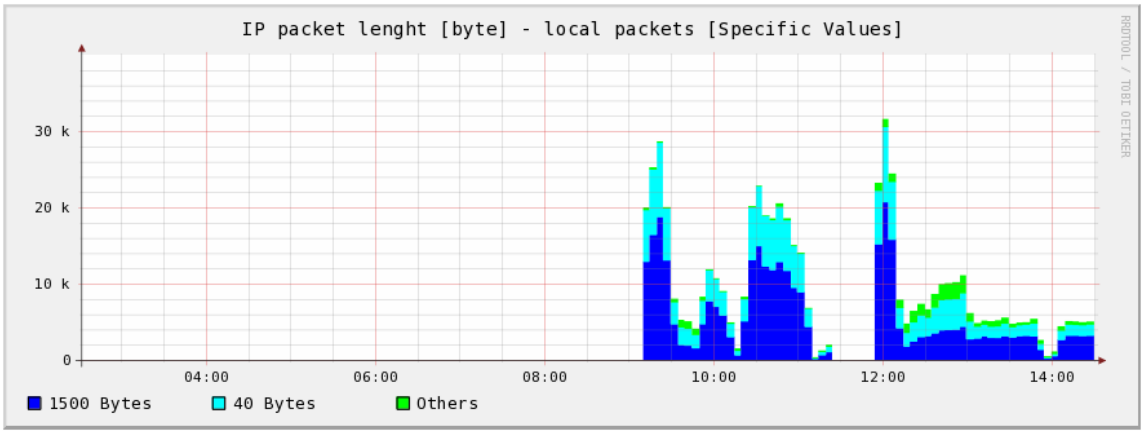


Figura 13. ***IP packet lenght [byte] - outgoing packets***

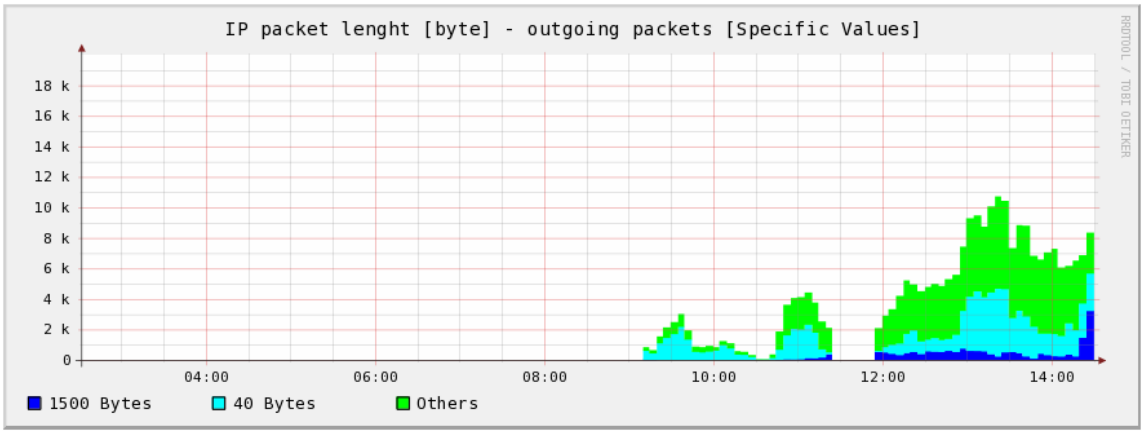


Figura 14. ***IP protocol - incoming packets***

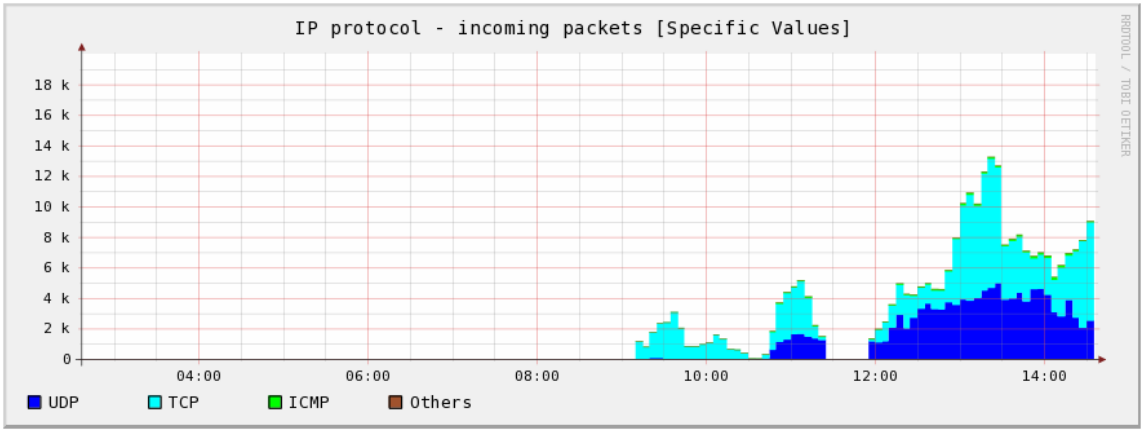


Figura 15. ***IP protocol - local packets***

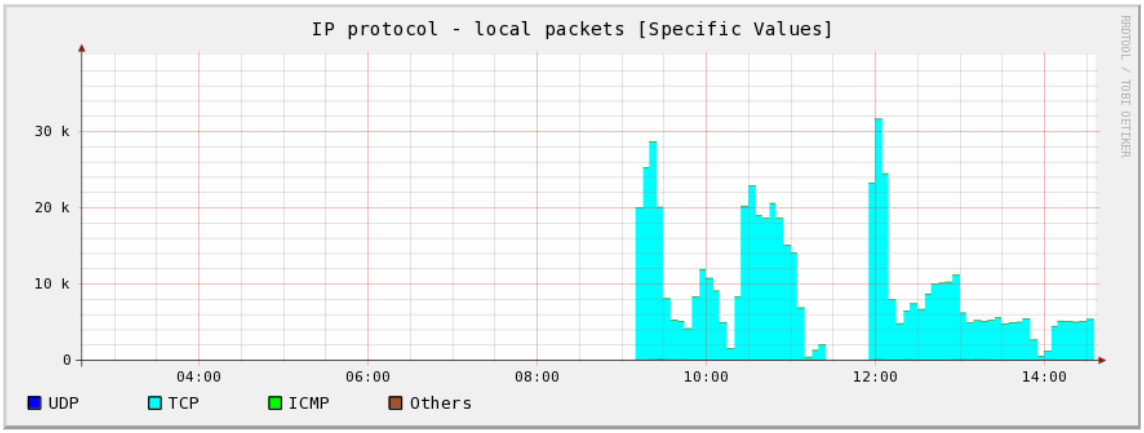


Figura 16. ***IP protocol - outgoing packets***

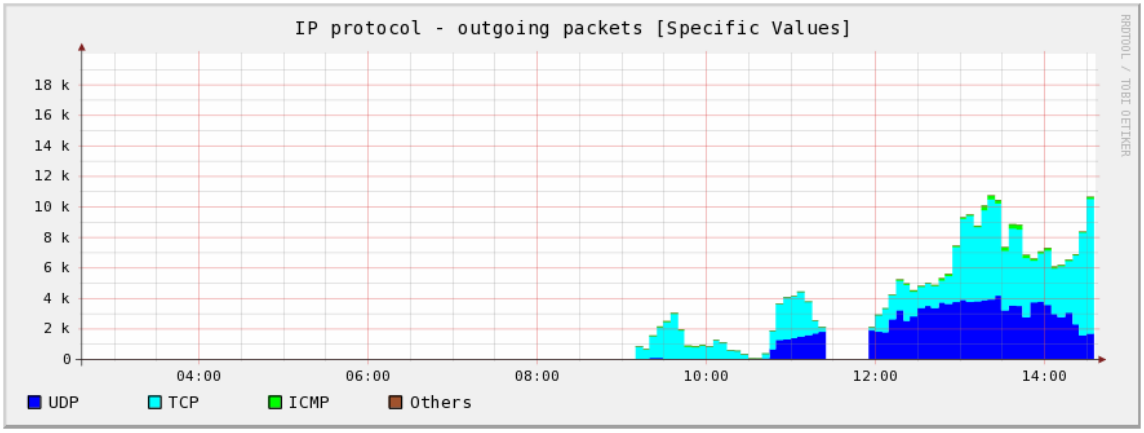


Figura 17. ***Number of tracked TCP/UDP/RTP/RTCP flow***

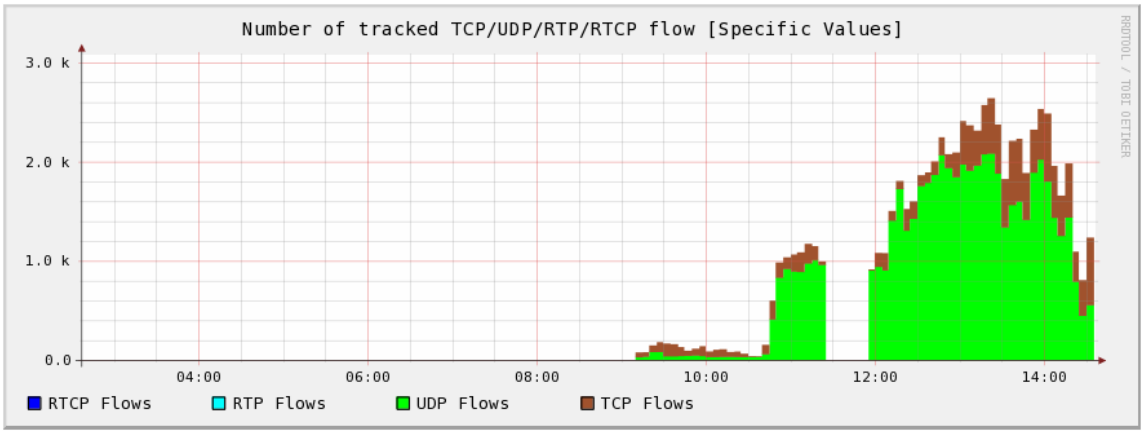


Figura 18. ***TCP Early interrupted flows***

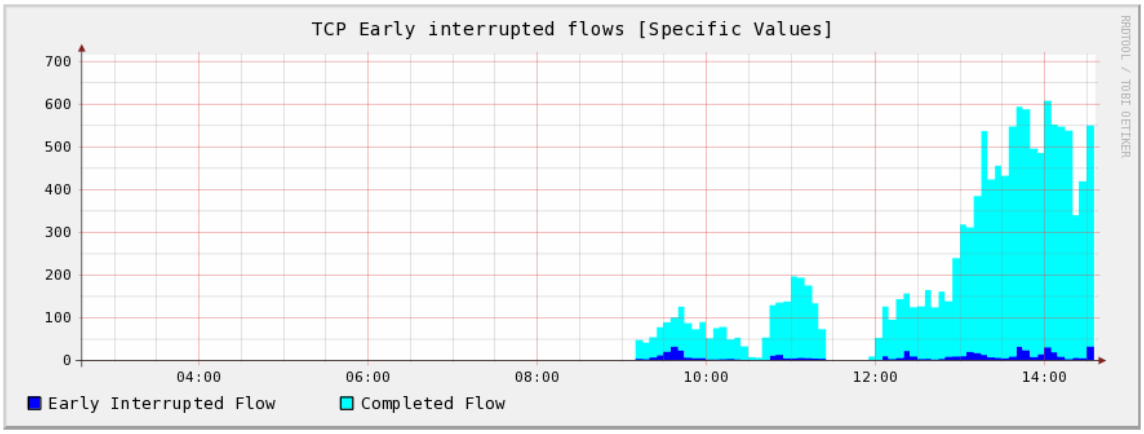


Figura 19. ***TCP option: SACK***

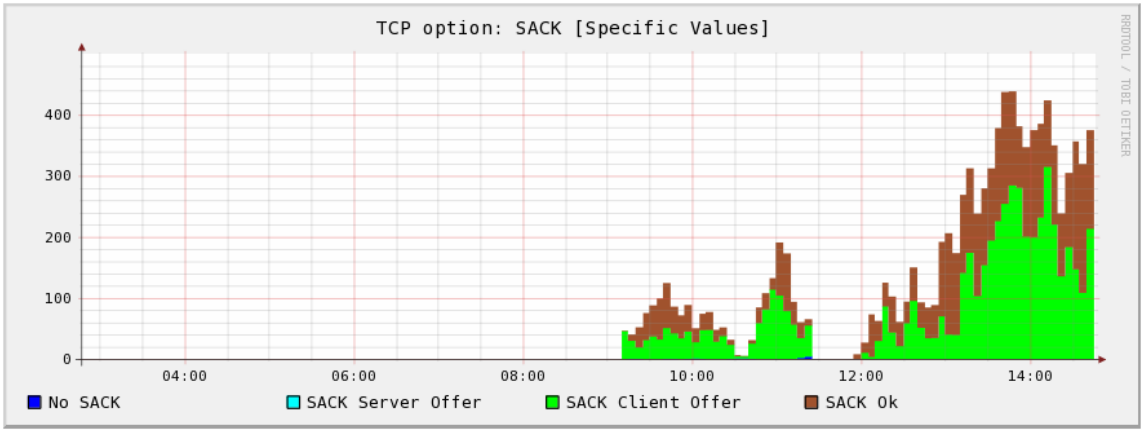


Figura 20. ***TCP option: TimeStamp***

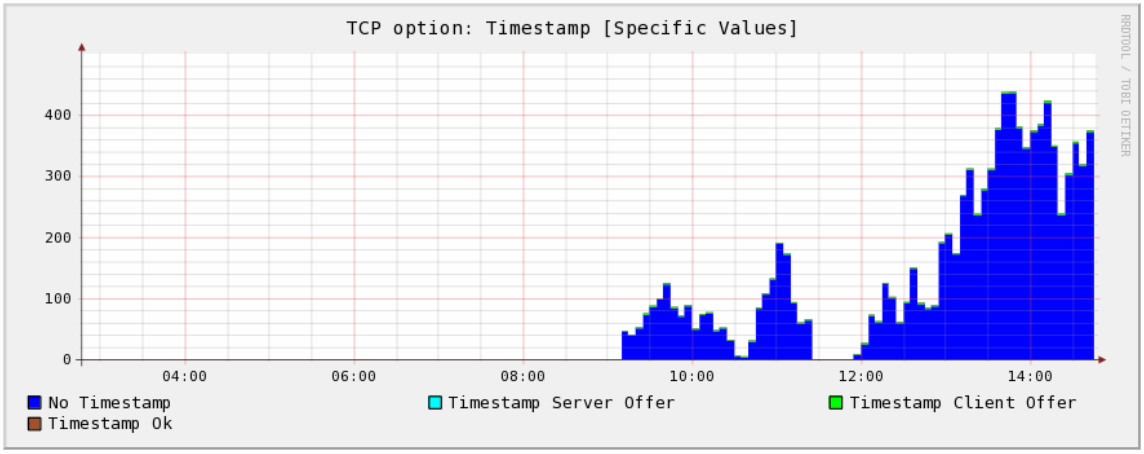


Figura 21. ***TCP option: WindowScale***

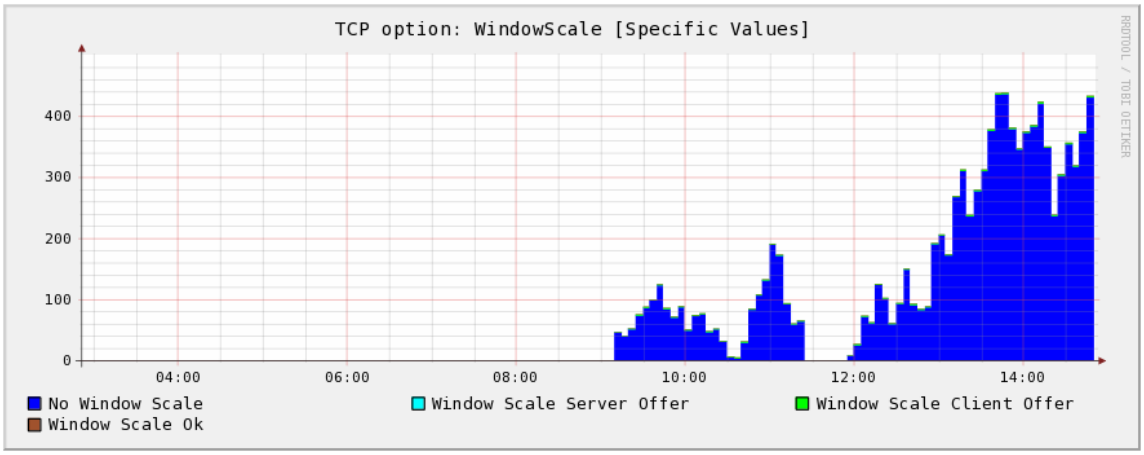


Figura 22. ***TCP throughput [Kbps] - client flows***

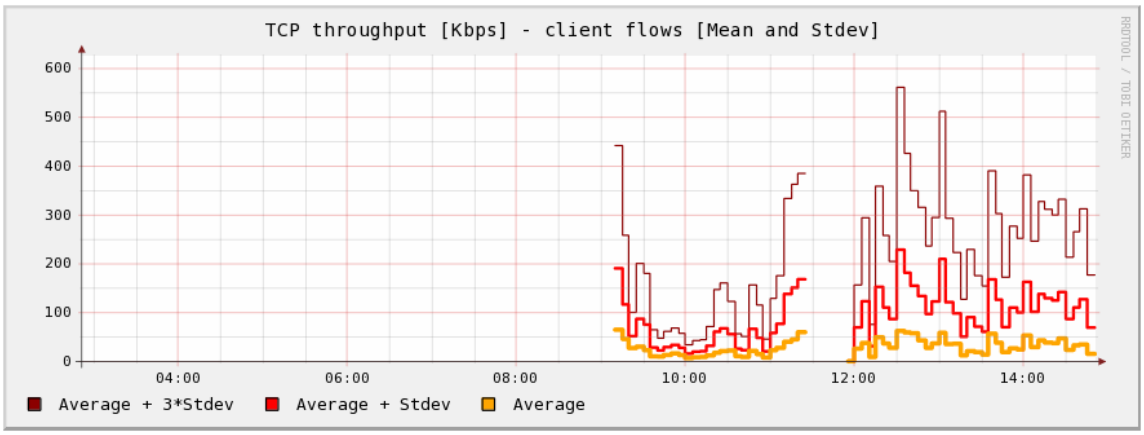


Figura 23. ***TCP throughput [Kbps] - server flows***

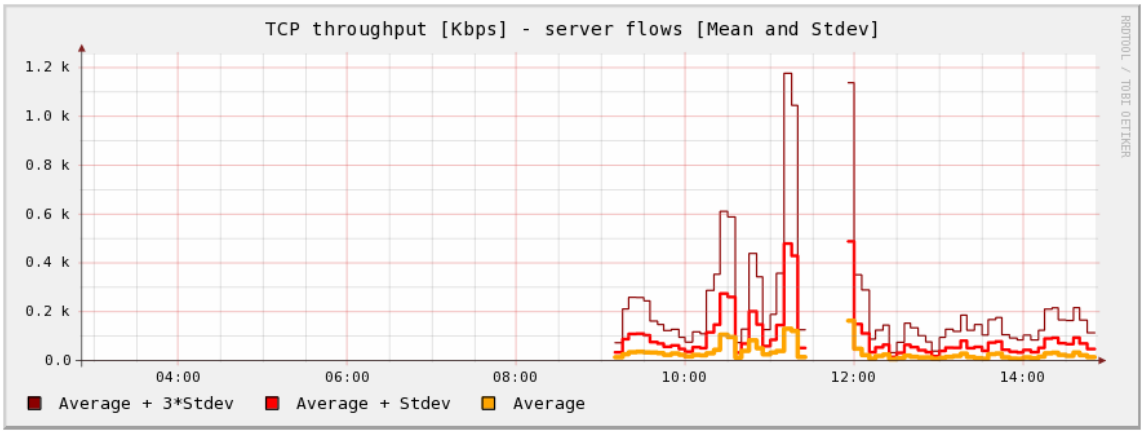


Figura 24. ***TCP total number of anomalies - incoming flows***

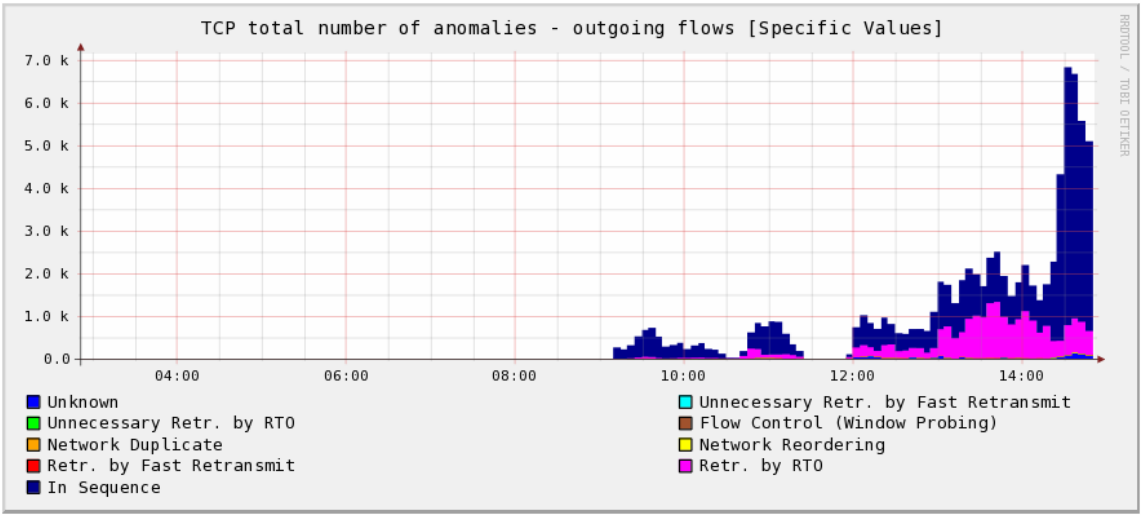


Figura 25. ***TCP total number of anomalies - local flows***

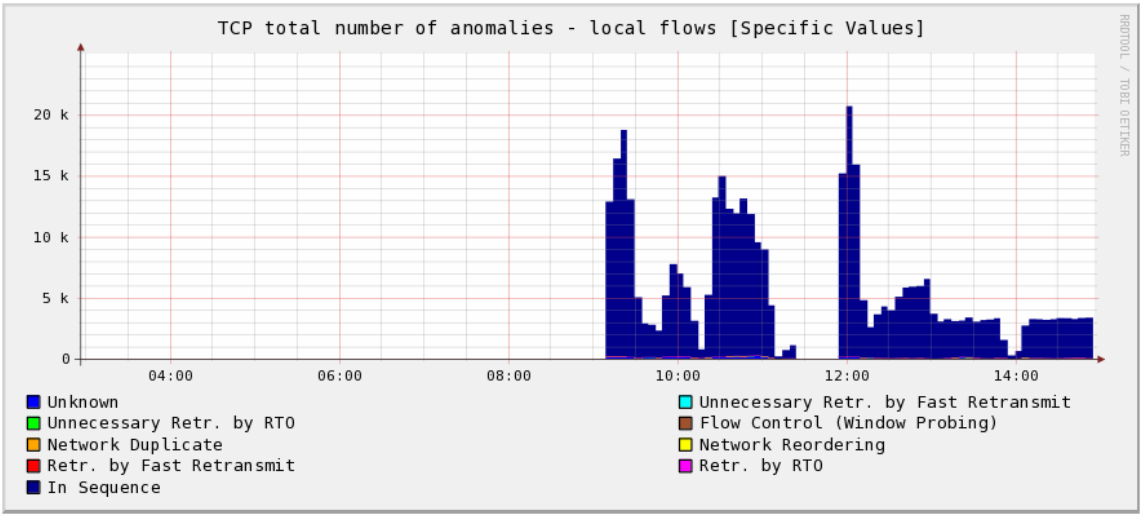
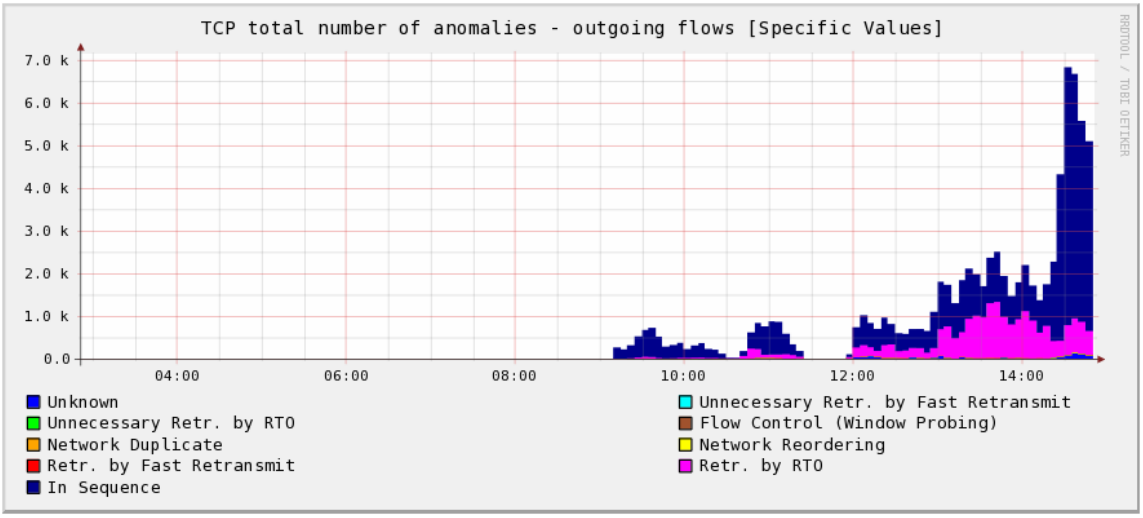


Figura 26. ***TCP total number of anomalies - outgoing flows***



5.5 Análisis sobre las mediciones tomadas

1. En las pruebas de ping, si bien se supone que los parámetros TCP no deberían de intervenir por alguna razón al cambiarlos se pudo notar una disminución en la pérdida de paquetes en el medio de 4% a 1%.
2. En las pruebas de transferencias de FTP del servidor al cliente se pudo notar una disminución en el tiempo de transferencia debido a una mejor tasa de transferencia.
3. En las pruebas de transferencias de FTP del cliente al servidor, solo cuando se activaron los parámetros de logró terminar las transferencias, aunque a una tasa menor que en el caso del servidor al cliente lo que comprueba la asimetría del canal observada en luego en las gráficas que se tomaron con la herramienta Tstat.
4. Las pruebas de transferencias de HTTP del servidor al cliente de los mismos archivos, al durar mas tiempo, sugieren que el protocolo FTP es mas eficiente para transferir archivos.
5. La carga de 10 páginas *web* desde Internet, con y sin los parámetros activados en promedio muestran una mejoría al activarlos, pero mas notable es el hecho del

positivo impacto de la utilización del *cache* de *web* en la carga posterior de las mismas páginas.

6. De las primeras tres gráficas se puede notar que se lograron paquetes de datos en general mas grandes, cuando se activaron los parámetros TCP de SACK, *window scaling*, *timestamping* y TCP Westwood.
7. De las gráficas 7 y 8 se puede notar que para las mismas pruebas, en el mismo orden y casi el mismo período de tiempo se logró un mayor número de flujos y se interrumpieron menos flujos.
8. De las gráficas 9, 10 y 11 se puede ver que dado que en la primera parte de la prueba el SACK, *window scaling* y *timestamp* estaban desactivados del lado del servidor y la en la segunda parte de la parte de la prueba estaban activados, solo el SACK se manifestó en activo en los dos casos, lo que permite concluir que la conexión está determinada con respecto a estos parámetros por el cliente que este caso era una computadora con Windows.
9. De lo anterior se puede ver que las ventajas obtenidas entonces se debieron principalmente al SACK y al TCP Westwood.
10. De las gráficas 12 y 13 se puede ver que el desempeño con y sin los parámetros activados fue mejor en un sentido que en el otro, lo que permite confirmar la asimetría observada en las pruebas realizadas con la herramienta Iperf.
11. De las gráficas 14, 15 y 16 se puede notar que la mayoría de los paquetes se mantuvieron en secuencia y que la única anomalía presente fue debida al RTO (*retransmision timeout*), la cual pueden estar justificadas debido a las retransmisiones debidas a los aumentos del RTT y adicionalmente a la pérdida de paquetes.
12. Se logró una sesión de terminal segura remota SSH mas estable y con mejor respuesta a los comandos y que no rompió la comunicación en ningún momento, después de activados los parámetros de TCP y el TCP Westood, a diferencia de antes de activarlos, que se rompió la comunicación en varias ocasiones.

CONCLUSIONES

1. Existen bastantes métodos disponibles para compensar o solucionar los problemas del desempeño del TCP sobre redes inalámbricas.

2. Aunque algunos métodos pueden requerir amplias modificaciones en los sistemas operativos y en las implementaciones TCP de éstos, es posible utilizar algunas modificaciones simples, para utilizar algunos de dichos métodos y, aún así, obtener mejoras substanciales.

3. De las pruebas realizadas en el caso práctico se puede ver que la utilización de los métodos y técnicas descritas en este trabajo de graduación pueden tener un impacto en el desempeño del TCP y, en la percepción del desempeño por parte de los usuarios y en las aplicaciones.

4. Construir un escenario de pruebas lo suficientemente controlado, y poder aislar y manejar cada una de las variables implicadas, tanto externas debidas al medio físico, como internas debido al protocolo en sí, es muy complejo, tanto como poder establecer la contribución individual que cada método podría aportar en el mejoramiento del desempeño total. Es posible hacer estimaciones y simplificaciones, y tener resultados bastante precisos.

5. Existen herramientas lo suficientemente buenas para medir los resultados del efecto, de los métodos utilizados para controlar y mejorar el TCP.

6. Existen cientos de escenarios de pruebas o escenarios reales de comunicaciones inalámbricas, y cada uno tendrá diferentes condiciones debidas a la tecnología inalámbrica usada, medios físicos diferentes, equipos con diferentes capacidades,

etc; sin embargo, se puede inferir que la utilización de los métodos descritos en este trabajo de graduación, pueden, en forma individual o en conjunto, proporcionar formas de mejorar el desempeño del TCP en cada uno de estos escenarios.

RECOMENDACIONES

1. Utilizar los métodos descritos en este trabajo de graduación para mejorar el desempeño del protocolo TCP en las redes inalámbricas de transmisión de datos.

2. Utilizar sistemas operativos de fuente abierta, que permiten más fácilmente aplicar los métodos descritos.

BIBLIOGRAFÍA

1. Allman, M. y otros, **RFC2488** ***Enhancing TCP Over Satellite Channels using Standard Mechanisms***, http://www.ietf.org/rfc/rfc2488.txt, 2005-2006.

2. *American Radio Relay League*, ***The ARRL Handbook for Radio Communications***, 18a Edición, Estados Unidos, 2004.

3. Amzak, O. y otros, ***Boosted Session Transport (BST) Protocol for Improved Performance in Satellite, Wireless and Mobile Networks***, http://www.potaroo.net/ietf/all-ids/draft-azmak-bst-00.txt, 2006.

4. Bates, Regis J. ***Broadband Telecomunicatons Handbook***, Segunda edición, Estados Unidos, Mcgraw-Hill Telecom, 2001.

5. Border, J. y otros, ***RFC3135 Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations***, http://www.ietf.org/rfc/rfc3135.txt, 2005-2006.

6. Comer, Douglas E. **Redes globales de información con Internet y TCP/IP, principios básicos, protocolos y arquitectura**. Tercera edición, Mexico, Prentice-Hall. 1996.

7. Carter, Michael. ***A Review of Transport Protocols as Candidates For Use in a Tactical Enviroment***. *Information Networks Division, Defense Science and Technology Organization, Australian Government, Department of Defense*. 2005.

8. Edwards, E y otros. ***Performance of SCPS in Complex Networks***. *Xiphos Technologies Inc*. 2006.

9. Elaarag, Hala. ***Improving TCP Performance over Mobile Networks***. *ACM Computing Surveys*, Estados Unidos, Volumen 34, No 3, 2002. 357-374pp.

10. ICTCompress, ***The Accelenet Advantage***, http://www.ictcompress.com/PDF/AcceleNet%20White%20Paper%202004.pdf, 2006.

11. IBM Redbooks, ***TCP/IP Tutorial and Technical Overview***, http://www.redbooks.ibm.com/abstracts/gg243376.html, 2006.

12. Jacobson, V. y otros, ***RFC1323 TCP Extensions for High Performance***, http://www.ietf.org/rfc/rfc1323.txt, 2005-2006.

13. Jones, M. Tim, ***Better networking with SCTP***, http://www.ibm.com/developerworks/linux/library/l-sctp/index.html, 2006.

14. Mellia, Marco y otros. ***Measuring IP and TCP behavior with Tstat***, Dipartamento di Elettronica, Politecnico di Torino. 2002.

15. Mellia, Marco y otros. ***TCP Anomalies: idenfication and analysis***. Dipartamento di Elettronica, Politecnico di Torino. 2002.

16. Morris, R. ***TCP Behavior with Many Flows***, IEEE International Conference on Network Protocols, 1997.

17. Orr, Michael, ***Internet via Satellite; Problem and Solutions***, *Flash Networks*, 1999.

18. Packeteer, ***Protocol Aceleration***, http://www.packeteer.com/technology/acceleration.cfm, 2006.

19. Parker, Tim. **Aprendiendo TCP/IP en 14 días**, Segunda edición. Mexico, Sams Publishing, 1996.

20. Tanenbaum, Andrew S. **Redes de Computadoras**. Tercera edición, Mexico, Prentice-Hall, 1997.

21. Tachyon, **Descripción Técnica**, http://www.tachyon.net, 2006.

22. Vacca, John R. ***Wireless Broadband, Networks Handbook***, Estados Unidos, Osborne-Mcgraw-Hill, 2001.

23. Venturi *Wireless*, ***Technology***, http://www.venturiwireless.com/solutions/technology.html, 2006.

APÉNDICE

Instalación, configuración y uso básico de la herramienta TSTAT

El software TSTAT se puede obtener en forma de codigo fuenta pra compilar de http://tstat.tlc.polito.it/ , en el caso de esta tésis se obtuvo precompilado y empaquetado para usarse en RedHat Linux de http:// .

Para su funcionamiento es necesario configurar varios archivos ubicados en /etc/tstat :

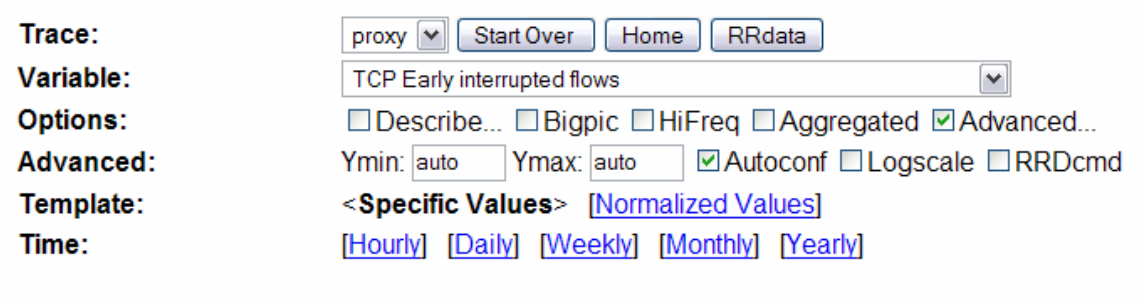
```
[root@proxy tstat]# more global.conf
# tstat global conf

# interface to listen
tstat_if="eth1"

# rrd configuration
tstat_rrdconf="/etc/tstat/rrd.conf"

# network configuration, please adjust to fit your network
# see /etc/tstat/net.conf.sample
tstat_netconf="/etc/tstat/net.conf"

# trace name
tstat_name="proxy"
```

```
[root@proxy tstat]# more net.conf
192.168.2.0
255.255.255.0
```

Para arrancar el proceso en segundo plano se utiliza el comando « service tstat start ».

Para ver las gráficas que se generan se tiene que accesar a la dirección http://IPServidor/cgi-bin/tstat_rrd.cgi.

Un ejemplo de la interfase para solicitar las gráficas:



Utilización del comando iperf

1. Para activar el iperf en modo servidor : iperf –s –D
2. Para activar el iperf en modo cliente : iperf –c “IP Servidor “ –i 1