
**CENTRO DE CIENCIAS BÁSICAS**

**MAESTRÍA EN CIENCIAS EXACTAS Y SISTEMAS DE LA INFORMACIÓN ESPECIALIDAD EN REDES**

# "Un Estudio Comparativo del Desempeño de Sistemas Operativos bajo el protocolo TCP en IPv6 "

**PRESENTA**

**I.S.C. Omar Alvarado González**

**DIRECTOR DE TESIS**

**Dr. Juan Manuel Gómez Reynoso**

**SINODALES**

**M.C. Eduardo Bautista Villalpando**

**M.C. Arturo Elías Ramírez**

**Aguascalientes, Ags., Junio 2009**

# DEDICATORIAS

El presente trabajo lo dedico a mis padres Humberto A. Alvarado Martínez y Blanca G. González Quintero a mi hermano Humberto Alvarado González, gracias por todos los momentos que hemos compartido juntos, pues siempre han estado conmigo y apoyado.

A mi director de tesis Juan Manuel Gómez Reynoso, por el tiempo y conocimientos compartidos, en ésta nueva etapa de mi preparación profesional.

A mis asesores Arturo Elías Ramírez y Eduardo Bautista Villalpando, por impulsarme a buscar más conocimiento.

Y en especial a Dios, por brindarme vida y salud.

Por este conducto autorizamos al tesista:

I.S.C. Omar Alvarado Gonzál

La impresión de su documento final de tesis, ya que cumple con los requisit
de contenido y forma exigidos en la Universidad Autónoma de Aguascalientes

Director

Juan Manuel Gómez Reynoso, Ph.D.

Sinodal

M.C. Eduardo Bautista Villalpando

Sinodal

M.C. Arturo Elías Ramírez

# RESUMEN

La tecnología es para las organizaciones un aliado que les ayuda a enfrentar y solucionar problemas, así como reaccionar de manera oportuna en el ambiente comercial, brindándoles oportunidades de negocio, para así reaccionar ante los cambios y necesidades de negocio y no ser desplazadas del mercado. Las organizaciones frecuentemente deben decidir la adquisición de tecnología entre dos o más que compiten. Generalmente, existen pocos estudios formales que hagan esto. Existen distintas clasificaciones de tecnologías de entre las cuales una muy importante son los sistemas operativos. Estudios previos como el de Zeadally y Raicu (2000) han evaluado Sistemas Operativos como Windows NT y Solaris comparando el desempeño de equipos con características y configuraciones similares. Principalmente, evaluaron los protocolos IPv6 e IPv4 midiendo los tiempos de respuesta en el envío de mensajes menores a 256 bytes y mayores a 512 bytes. Por otro lado, la tecnología avanza a gran rapidez por lo que es necesario realizar investigaciones basadas en estudios previos. Es por eso que se analiza en el presente estudio el desempeño de los Sistemas Operativos Windows Server 2008, comparándolo con respecto a FreeBSD 7.1 y CentOS 5.1 bajo el protocolo TCP en IPv6. Debido a lo anterior, las organizaciones requieren estudios, en los que se lleven a cabo estudios relacionadas a la tecnología que utilizan en sus procesos.

Los Sistemas Operativos (SOs) son el administrador principal de todo equipo computacional, de ahí la importancia de saber elegir la apropiada de acuerdo a los requerimientos y necesidades de cada organización. Por lo tanto, los Sistemas Operativos pueden ser un diferenciador para que las empresas respondan a las presiones del mercado y actuar de forma correcta. Mas, sin embargo, ¿Cuál es el desempeño que tiene el protocolo de Internet versión 6 sobre el sistema operativo CentOS 5.1, FreeBSD 7.1, comparado con el sistema operativo Windows server 2008?

La presente investigación, muestra las respuestas a la anterior pregunta, que será de beneficio para las organizaciones que están en lo último de la tecnología para eficientar sus labores.

Para ello se realizó un estudio comparativo del desempeño de sistemas operativos bajo el protocolo TCP en IPv6. Para la captura de datos se utilizaron dos equipos PC, los cuales tienen las mismas características de hardware. Con la finalidad de hacer la comparación de desempeño, en un equipo se instaló los SOs CentOS 5.1, FreeBSD 7.1, uno a la vez; y en el otro el SO Windows 2008. En dichos equipos se midió el round-trip time, así como el tiempo de respuesta para los siguientes escenarios:

1. Configuración LAN respecto a la Variable Tiempo
2. Configuración LAN respecto a la Variable Bytes
3. Configuración Loopback respecto a la Variable Tiempo
4. Configuración Loopback respecto a la Variable Bytes

En la configuración LAN Respecto a la variable Tiempo, el SO que más rápido transmite paquetes fue CentOS 5.1, seguido por FreeBSD 7.1, y por último Windows Server 2008.

Los resultados de la configuración LAN respecto a la variable paquetes enviados por Byte, evidenciaron lo siguiente: el SO que mas rápido transmite paquetes fue Windows Server 2008, seguido por CentOS 5.1 y por último FreeBSD 7.1.

El Resultado observado para la configuración LOOPBACK respecto a la variable tiempo, indica que el SO que más rápido transmite fue Windows, seguido por CentOS 5.1 y por último FreeBSD 7.1.

Para el resultados de la configuración LOOPBACK respecto a las variable paquetes enviados por byte, el SO que mas rápido transmite paquetes fue CentOS 5.1, seguido por Windows Server 2008 y por último FreeBSD 7.1.

**Contenido**

I. Introducción……..1
II. Contexto General de la Investigación……..3
1. La Tecnología en las Estrategias Competitivas de las Organizaciones..... 4
2. Aspectos relevantes de Tecnologías Relacionadas a la Investigación……17
2.1.- Definición y Concepto de Protocolo ..... 17
2.2.- Funciones de los Protocolos ..... 17
2.3.- Organización y modelado de los protocolos ..... 18
2.4.- Modelado de 7 Capas de OSI de la ISO ..... 18
2.5.- Importancia de OSI ..... 19
2.6.- Arquitectura de TCP/IP ..... 20
2.7.- Capas del Modelo OSI ..... 21
2.8.- Comunicación Entre Capas ..... 26
2.9.- Protocolo de Control de Transmisión / Protocolo de Internet (Transmision Control Protocol / Internet Protocol) ..... 27
TCP ..... 27
IP ..... 27
2.10.-¿Qué es, cómo y dónde fue desarrollado TCP/IP? ..... 27
2.11.- Arquitectura de TCP/IP ..... 29
2.12.- La importancia de TCP/IP en la interconexión de redes ..... 30
2.13.- Ventajas de TCP ..... 31
2.14.- Pila TCP/IP en Sistema Operativo Windows ..... 31
2.15.- Optimización automática de la ventana de recepción ..... 31
2.16.- Ventanas de Recepción en la Pila Next Generation TCP/IP ..... 32
2.17.- Antecedentes de TCP: Números de Secuencia ..... 32
2.18.- Conversación entre hosts TCP ..... 33
3. Protocolo Internet IPv6 ..... 33
4. MTU Nombre ..... 34
5. Definición de SNMP ..... 35
5.1 Conceptos SNMP ..... 36
5.2 Operación de SNMP ..... 37
6. Términos Encontrados en Investigaciones Previas ..... 38
6.1. Remote Monitoring (RMON) ..... 38
6.2. SMI ..... 38
6.3. MIB ..... 38
6.4. Socket ..... 38
6.5. Net Flood ..... 39
6.6. TCP Syn Flood ..... 39
6.7. Tasa de transferencia (Throughput ) ..... 39
6.8. Tiempo de ida y vuelta (Roundtrip time) ..... 40
6.9. Ancho de Banda (Bandwidth ) ..... 40
6.10. Latencia (Latency) ..... 40
7. Comparación de los modelos de referencia OSI y TCP ..... 40
8. El sistema Operativo como Administrador de Recursos ..... 40
9. Sistema Operativo ..... 41
9.1. Red ATM (Modo de Transferencia Asíncrono) ..... 41
9.2. Net Flood ..... 42

9.3. Connection Flood ..... 42
III. Tecnologías Relacionadas al Estudio……..……44
1. Introducción al tema..... 44
2. Comparativa de rendimiento..... 46
2.1.- Comparativa de rendimiento (benchmark) general..... 46
2.2.- ¿Qué no es Benchmark?..... 46
2.3.- ¿Por qué Benchmark?..... 47
1.4.-Pasos para la Comparativa de Rendimiento..... 47
IV. Identificando el problema de investigación……..……49
1. Objetivos de investigación ..... 51
2. Pregunta de investigación..... 51
3. Hipótesis ..... 51
4. Metodología de investigación..... 54
4.1. Recolección de datos..... 55
4.2. Muestra..... 55
4.3. Configuración utilizada..... 58
4.4. Resultados Obtenidos..... 59
4.4.1. Técnica utilizada ..... 59
4.4.2.- Resultados de la Configuración LAN Respecto a la Variable Tiempo..... 59
4.4.3.- Resultados de la configuración LAN respecto a la variable paquetes enviados por Byte..... 61
4.4.4.- Resultados de la configuración LOOPBACK respecto a la variable tiempo..... 62
4.4.5.- Resultados de la configuración LOOPBACK respecto a las variable paquetes enviados por byte ..... 64
V. Conclusiones……..……67
5.1.- Conclusiones generales..... 67
5.2 Limitaciones..... 68
5.3 Trabajos futuros..... 68
VI. REFERENCIAS……70

## Lista de Figuras

Figura 1. Categorías de la importancia estratégica y su impacto (adaptado de Applegate, McFarlan, & McKenney, 1999)..... 4
Figura 2. Modelo de Porter (adapatado de Applegate et al., 1999) ..... 6
Figura 3. Estrategia de Latter (adaptado de Applegate et al., 1999) ..... 8
Figura 4. Cadena de valor (adaptado de Applegate et al., 1999) ..... 10
Figura 5. Estructura Linux (adaptada de Jaus, 2009) ..... 12
Figura 6. Throughput (Adaptado de QChec Console 2009)..... 15
Figura 7. Capas, protocolos e interfaces (adaptado de Comunicaciones, 1998)……17
Figura 8. Estructura en niveles de capas del modelo OSI (adaptado de Comunicaciones, 1998) ..... 21
Figura 9. Ordenamiento y funciones del modelo 051 (adaptado de GS Comunicaciones, 1998) ..... 24
Figura 10. Comunicación entre niveles del modelo OSI. ..... 26
Figura 11. Modelo TCP/IP (adaptado de http://www.textoscientificos.com/redes/tcp-ip/comparacion-modelo-osi) ..... 28
Figura 12. Niveles de arquitectura TCP/IP (adaptado de GS Comunicaciones, 1998) ..... 30
Figura 13. Ventanas TCP (adaptado de Cisco, 2007) ..... 31
Figura 14. Números de secuencia TCP (adaptado de Cisco, 2007)..... 33
Figura 15. Estructura de un datagrama (adaptado de Comer, 1996) ..... 34
Figura 16. Formato de encabezado base de cuarenta octetos del IPv6. Cada datagrama IPv6 comienza con un encabezado base (adaptado de Comer, 1996). ..... 34
Figura 17. Modelo de referencia TCP/IP (adaptado de Tanenbaum, 2003)……40
Figura 18. Configuración red LAN para Windows Server 2008 ..... 56
Figura 19. Configuración red loopback para Windows Server 2008..... 57
Figura 20. Configuración red LAN para FreeBSD 7.1 ..... 57
Figura 21. Configuración red loopback para FreeBSD 7.1 ..... 57
Figura 22. Configuración red LAN para CentOS 5.1 ..... 58
Figura 23. Configuración red loopback para CentOS 5.1 ..... 58

**Lista de Tablas**

Tabla 1. Niveles o capas del modelo OSI (adaptado de GS Comunicaciones, 1998) ……27
Tabla 2. Operaciones SNMP ..... 36
TABLA 3. Comparación entre CentOS y BSD (Adaptada de (A. Sandoval, 2008) …… ..... ……46
Tabla 4. Estadística descriptiva respecto a la variable tiempo ..... 60
Tabla 5. Tabla ANOVA (respecto variable tiempo)..... 60
Tabla 6. Comparaciones múltiples (respecto variable tiempo) ..... 61
Tabla 7. Desviación y error estándar (respecto variable n bytes)..... 62
Tabla 8. Tabla ANOVA (respecto a variable n bytes) ..... 62
Tabla 9. Comparaciones múltiples ..... 62
Tabla 10. Desviación y error estándar (respecto variable tiempo)..... 63
Tabla 11. Tabla ANOVA (respecto a variable tiempo)..... 63
Tabla 12. Comparaciones múltiples ..... 64
Tabla 13. Desviación y error estándar (respecto variable tiempo)..... 65
Tabla 14. Tabla ANOVA (respecto a variable n bytes) ..... 65
Tabla 15. Comparaciones múltiples ..... 66

# I. Introducción

La realización del estudio comparativo del desempeño de sistemas operativos bajo el protocolo TCP en IPv6 se lleva a cabo utilizando un diseño factorial, que según lo define Gutiérrez y de la Vara (2004), el objetivo es estudiar la relación entre los factores y la respuesta, con la finalidad de conocer mejor cómo es ésta relación y generar conocimiento que permita tomar decisiones que mejoren el desempeño del proceso, en el caso en particular de éste estudio lo que se busca es determinar los niveles de los factores de tipo cualitativo para poder interpretar los datos generados tanto para las variables dependientes que son los SO Windows Server 2008, CentOS 5.1 y FreeBSD 7.1 así como para las independientes que es la tecnología empleada, en éste caso la red LAN y Loopback configuradas. Una vez que se tenga el diseño factorial se procederá a correr aleatoriamente el experimento bajo todas las posibles combinaciones seleccionadas.

Dichas combinaciones se basarán en un experimento factorial o arreglo factorial, el cuál según define Gutiérrez y de la Vara (2004) es un conjunto de puntos experimentales o tratamientos que pueden formarse considerando todas las posibles combinaciones de los niveles de los factores. En éste estudio el número de factores serán las redes LAN y Loopback y los niveles de prueba corresponderá a los SOs Windows Server 2008, CentOS 5.1 y FreeBSD 7.1, formándose con ello un diseño factorial de 2x3 que servirá para definir las combinaciones o puntos experimentales.

La presente investigación se origina partir de un análisis previo de las características de cada elemento, que incide en los resultados esperados para poder realizar un estudio más eficiente. Para ello se realizarán pruebas en las que se comparará el desempeño en que se transmiten paquetes de información a través del round trip time, desde un equipo con Windows 2008 así como CentOS 5.1 y FreeBSD 7.1.

Los cuales se evaluarán para los siguientes escenarios:

1. Configuración LAN respecto a la Variable Tiempo
2. Configuración LAN respecto a la Variable Bytes
3. Configuración Loopback respecto a la Variable Tiempo
4. Configuración Loopback respecto a la Variable Bytes

La configuración LAN Respecto a la Variable Tiempo, Analizando la estadística descriptiva de la variable tiempo podemos observar que en promedio el SO Windows requirió más tiempo (.2408) comparado con los otros dos (FreeBSD .3600 y CentOS .4700). En consecuencia, Windows requiere aproximadamente 2 veces más tiempo que CentOS y 1.5 veces más que FreeBSD, y FreeBSD requiere 1.3 veces más tiempo respecto a CentOS.

Los resultados de la configuración LAN respecto a la variable paquetes

envíados por Byte, Analizando la estadística descriptiva de la variable bytes podemos observar que en promedio el SO Windows Server 2008 requirió más tiempo (93.70) comparado con los otros dos (FreeBSD 90.86 y CentOS 93.25). En consecuencia, Windows envía aproximadamente 0.99 bytes por uno enviado por CentOS y 0.96 bytes envíados, por uno de FreeBSD, y FreeBSD requiere 1.02 veces más bytes respecto a CentOS.

El Resultado observado para la configuración LOOPBACK respecto a la variable tiempo, Analizando la estadística descriptiva de la variable tiempo podemos observar que en promedio el SO Windows requirió más tiempo (1.862) comparado con los otros dos (FreeBSD 0.492 y CentOS 0.496). En consecuencia, Windows requiere aproximadamente 0.266 fracción de tiempo por un paquete enviado por CentOS y 0.264 fracción de tiempo por un paquete enviado por FreeBSD, y FreeBSD requiere 1.007 veces más tiempo respecto a CentOS.

Para el resultado de la configuración LOOPBACK respecto a las variable paquetes enviados por byte, Analizando la estadística descriptiva de la variable bytes podemos observar que en promedio el SO Windows Server 2008 requirió más tiempo (1.862) comparado con los otros dos (FreeBSD 0.492 y CentOS 0.496). En consecuencia, Windows envía aproximadamente 1.232 bytes por uno enviado por CentOS y 0.623 bytes envíados, por uno de FreeBSD, y FreeBSD requiere 1.976 Veces más bytes respecto a CentOS

# II. Contexto General de la Investigación

Las organizaciones, frecuentemente, deben decidir la adquisición de tecnología que les ayude a cumplir con sus metas. Para esto, es necesario seleccionar una entre diversas opciones. El desarrollo de nuevas tecnologías, así como la diversificación de las existentes, avanza a gran velocidad dando como resultado el tener que seleccionar la adecuada, lo cual requiere el tener que consultar estudios comparativos (benchmarking), para poder hacer la selección. A través de estos estudios las organizaciones pueden analizar los resultados para reducir la incertidumbre en la selección de alternativas. Por ejemplo, se puede observar el desempeño de un sistema operativo respecto a otro. Sin embargo, dado que los avances tecnológicos suceden a gran velocidad, existe la necesidad de realizar estudios que permitan evaluar dos o más tecnologías dentro de un mismo dominio de aplicación. Se pueden mencionar algunas de las tecnologías que se pueden evaluar, entre las más comunes están: sistemas de bases de datos, software de propósito específico, sistemas operativos, entre otros.

De acuerdo con Applegate et al. (1999), al comprar tecnología se pueden utilizar cinco preguntas clave que servirán de guía en la evaluación del impacto que se pueda observar. Dichas preguntas se explican a continuación:

a) ¿Puede la tecnología crear barreras de entrada? En éste sentido Applegate et al. afirman que entre más difícil sea el generar una tecnología como la ofertada por una organización, será más difícil que ésta tenga competencia, ya que para los competidores les tomaría mucho tiempo implementar dicha tecnología, debido a la complejidad y costos de ésta, lo cual generará una barrera de entrada respecto a sus competidores. Otro punto que también crea una barrera es cuando la tecnología aumenta la eficacia de las ventas, por lo que dependiendo de la complejidad y rentabilidad de la tecnología será más difícil tener competencia en el mercado.
b) ¿La tecnología puede impactar en los costos? De acuerdo con Applegate et al. cita que para poder reducir costos en productos y servicios es necesario fomentar la dependencia de ellos. Para esto se requiere que para el usuario sea muy sencillo el uso de tecnología, y que se cree una adicción a usarla. Por ejemplo, el empleo de compra de boletos de avión y acumular millas de viaje, lo cual reducirá los costos, ya que no se requerirá una oficina con empleados y papeleo. Además, hará que el cliente sea leal a la aerolínea al acumular millas de viaje.
c) ¿La tecnología puede cambiar la base de la competencia? Este punto lo define Applegate et al. como una oportunidad en la que para poder efectuar un cambio en la base de la competencia, primero se debe analizar la situación actual de la organización, lo cual incluye personal, y clientes. Una vez realizado dicho estudio, se procede a realizar el cambio de la base de la competencia. Por ejemplo, si se tiene una revista con artículos relacionados a equipos computacionales que se expende en puestos de revistas, entonces, eso significa que los subscriptores son individuos que cuentan o

tienen acceso de manera ordinaria a un equipo computacional. Por lo tanto, quizá sea más práctico para ellos leer la revista a través de internet, y por lo tanto, tampoco se requerirá tener un vocero que entregue en su domicilio dicha revista.

d) ¿Puede la tecnología cambiar el equilibrio de poder en relación con el proveedor? De acuerdo con Applegate et al., el uso de la tecnología permite a los corporativos tomar el control respecto a los productos y/o servicios que soliciten éstos a los fabricantes, en base a lo que se va a consumir en el momento actual o próximo, evitando así comprar productos que no son necesarios y tener así que pagar costos de almacenamiento, así como racionalizar el proceso de producción.

e) ¿La tecnología puede generar nuevos productos? Al respecto Applegate et al., definen que gracias al uso de la tecnología algunos corporativos han alterado la base de su competencia dentro de la industria, es decir, que los datos que les ha proporcionado el uso de la tecnología les ha permitido generar nuevos productos. Por ejemplo, una automotriz anteriormente, sólo producía motores y chasis. En la actualidad el uso de la tecnología, ha hecho que estas también fabriquen las computadoras para el control del sistema de frenado de los automóviles, automatizar la calefacción y/o el clima, etc.

# 1. La Tecnología en las Estrategias Competitivas de las Organizaciones

Las organizaciones en la actualidad utilizan aplicaciones informáticas. Más sin embargo, no todas le dan el mismo enfoque al uso de éstas. La Figura 1 explica el impacto que tiene para algunas organizaciones el uso de tecnología de información (TI).

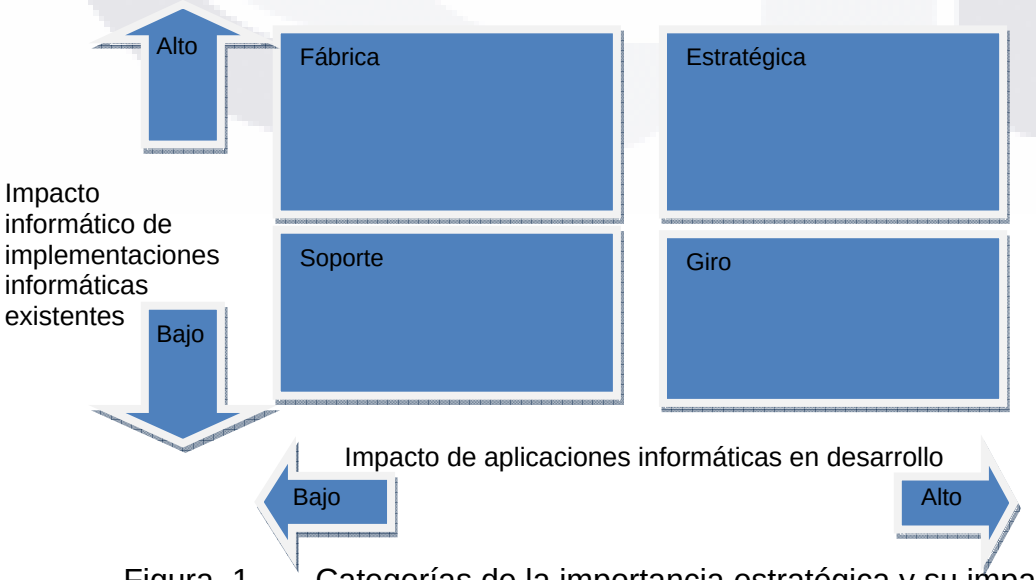


Figura 1. Categorías de la importancia estratégica y su impacto (adaptado de Applegate, McFarlan, & McKenney, 1999)

En la Figura 1, se pueden apreciar dos ejes, el primero de ellos explica el impacto informático que pueden tener las implementaciones TI utilizadas en las organizaciones, además, explica que las pequeñas interrupciones en el servicio o en la calidad pueden tener un profundo impacto para la permanencia en el mercado de las organizaciones, dependiendo del cuadrante en que se esté ubicada será el impacto que se observará.

Al comprender la posición de una organización respecto a cualquiera de los ejes permitirá el desarrollo de una adecuada estrategia de gestión de TI. A continuación se explican las categorías en las cuales se ubican los ejes:

1. ESTRATÉGICA.- En éste cuadrante las organizaciones planifican y se organizan administrativamente y toda la información está estrechamente ligada con las aplicaciones informáticas, de hecho, en algunas de estas empresas el jefe de TI, forma parte del consejo de administración de la organización. Ejemplos de organizaciones que se pueden ubicar en éste cuadrante están: los bancos de las compañías de seguros, y las principales cadenas comerciales.
2. GIRO.- En éste cuadrante las empresas reciben apoyo de las aplicaciones informáticas, pero no son absolutamente dependientes de ellas. Sin embargo, el rápido crecimiento de la organización requiere del desarrollo de aplicaciones computacionales para permitir al cliente centralizar la relación de producción y programación, mejorando así el servicio, reduciéndose los costos operativos y administrativos de la organización. En ésta categoría se ubican empresas de manufactura.

3. FÁBRICA.- En éste cuadrante se tienen aplicaciones informáticas desarrolladas, pero si por alguna razón existen una interrupción en el sistema, habrá consecuencias financieras. Mas sin embargo, no afectará la planificación y organización administrativa de la organización. Ejemplo negocios locales

4. SOPORTE.- En éste cuadrante las aplicaciones informáticas son utilizadas de acuerdo al perfil del profesional que está en cierta área de la empresa, es decir, el conocimiento y las habilidades se centran en el empleado, así que si existe un algún fallo en el sistema el impacto que tendrá será mínimo económicamente hablando. Ejemplo profesionista independiente

Con lo anterior se observa como el uso de TI en las organizaciones tiene un efecto sobre éstas. Sin embargo, existen más factores que intervienen directamente en el desempeño o ciclo de vida de una organización, ya que el utilizar TI por si solo al ejecutarse efectivamente produce beneficios respecto al número de clientes. Por tal motivo, las organizaciones deben buscar que el desarrollo de TI involucre a todos las áreas de la organización, lo cual implica una gran complejidad para involucrar los procesos de planeación y organización de la empresa. Por ello, es necesario que los directivos vean el uso de las TI desde una perspectiva estratégica y no solo táctica, el cual se

puede encontrar en el modelo de Porter (1980). Él menciona que la mejora está directamente relacionada con la disminución de costos, el incremento en la calidad y la mejora en el ciclo de producción. Sin embargo, no siempre que se desea mejorar el resultado es favorable, por lo cual agrega que es necesario conocer el área competitiva de la organización. Para llevar a cabo esto, se puede utilizar el modelo propuesto por él, en el cual describe la interacción de las fuerzas que involucran la mejora o fracaso de las organizaciones. Con este modelo se puede explicar la forma en que impacta e implica cada una de las fuerzas. Dicho modelo se presenta en la Figura 2.

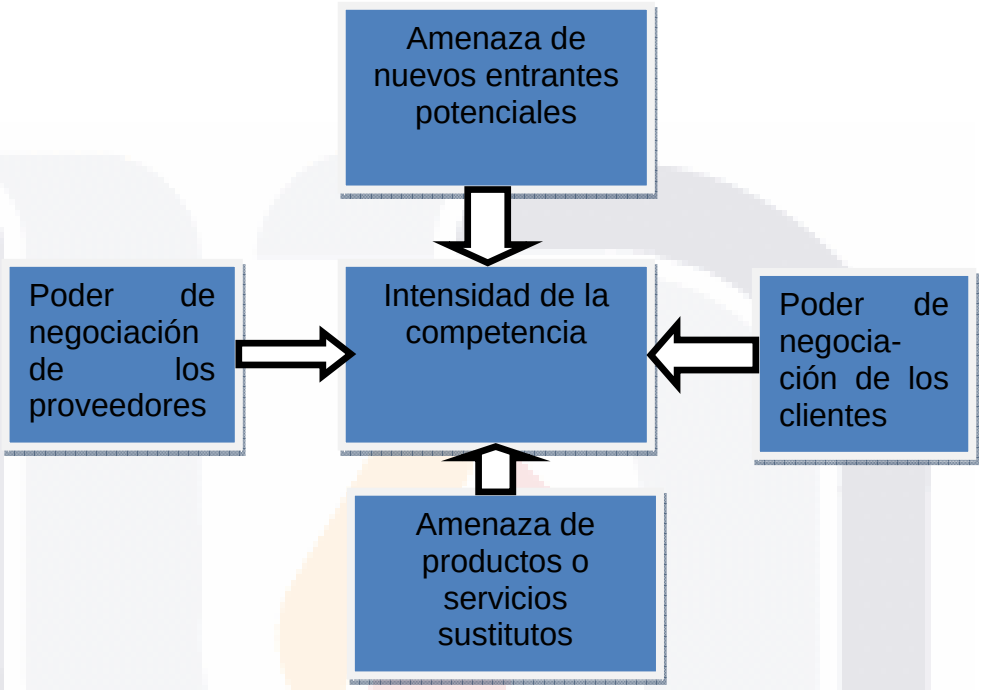


Figura 2. Modelo de Porter (adapatado de Applegate et al., 1999)

Estas fuerzas dan forma a la competencia, en la cual se involucra una organización dentro de su industria. A continuación se explica brevemente cada una de ellas.

- La intensidad de la competencia, es la fuerza en dónde se puede identificar la posición relativa de una organización. Además, las fuerzas económica y competitiva en un segmento de la industria son el resultado de las cuatro fuerzas básicas restantes. Una vez identificada dónde está posicionada la organización, es importante tener una visión para elegir una estrategia que conlleve al éxito o permanencia en el mercado, la cual se puede implementar con el uso de TICs. Este aspecto es muy relevante ya que no todos los ejecutivos de las organizaciones son capaces de realizar esto debido a que sus paradigmas actúan como filtros impidiéndoles identificar aspectos relevantes. Si una organización no conoce su posición relativa en la industria, le será prácticamente imposible definir una estrategia específica que le permita definir un rumbo exitoso. Barker (1989) explica que un paradigma es una regla o reglamento en donde se establece un límite, o bien una fórmula, para el éxito. En Wikipedia (2009a) se encuentra la siguiente definición de paradigma “es un

patrón o modelo, o bien un ejemplo". Algunos ejecutivos de las organizaciones presentan en ocasiones el efecto paradigma, es decir no proyectan soluciones creativas, o bien, no ven las oportunidades de negocio. Para evitar esto se debe evitar la que Baker (1989) llama "parálisis paradigmática", es decir, la incapacidad de identificar y adoptar nuevas ideas con valentía y confianza. Es por esto que para poder aplicar el modelo de Porter se debe identificar correctamente cada una de las fuerzas, y así aprovechar las oportunidades. Porter describe que para blindar a una organización ante la competencia, principalmente, se tiene que poner atención en los siguientes factores: a) liderazgo en costo, b) diferenciación, y c) enfoque.

- El poder de negociación de los proveedores describe el balance que existe en las transacciones realizadas entre la organización y sus proveedores. Este balance impacta directamente en la competitividad de la organización ya que entre más poder tenga el proveedor, más influencia tiene en la producción de bienes y servicios de la organización. Por ejemplo, si el proveedor es un monopolio, este definitivamente decide el precio y condiciones de entrega de sus bienes/servicios. En consecuencia, la organización tendrá que adaptarse a las reglas dictadas por el proveedor, ya sea en precio, condiciones de entrega, calidad, etc. Entonces, es muy importante el que existan más proveedores del mismo bien/servicio ya que de esta forma se genera la competencia entre ellos, beneficiando directamente a la organización puesto que se cambia el balance del poder de negociación hacia ella al tener la opción de decidir quién será su proveedor. A través del uso de TICs se puede cambiar el balance ya que entre menos dependiente se esté de un proveedor particular, mayor poder tendrá una organización.

- La fuerza de poder de negociación de los clientes implica que éstos serán determinantes para definir el precio del producto o servicio solicitado, así como el aumentar estándares de calidad y reforzar la competencia. Es decir, si el cliente es muy importante para una industria u organización, en consecuencia, obliga a obtener productos/servicios a condiciones favorables para él. Latter (Applegate et al., 1999) presenta un modelo derivado del de Porter, el cual tiene dos variantes: costo y diferenciación. El núcleo del nuevo modelo tiene dos principios. En el primero llamado mecanismo competitivo, se tiene la creencia de que las ventajas competitivas son la meta de la estrategia, es decir, una organización puede reducir sus costos o diferenciar sus productos o servicios. El segundo, llamado ámbito competitivo dice que la empresa debe definir el tipo de ventaja competitiva, en la cual se busca atender y el ámbito o la manera en que se atenderá.

La Figura 3, muestra las estrategias genéricas relacionadas con la ventaja competitiva y ámbito de aplicación. En consecuencia, podemos hacer la siguiente pregunta ¿se podría cambiar el equilibrio de poder en las relaciones con los proveedores? Para llevar a efecto dicho cambio, habría

que mejorar el flujo de información de las empresas, lo cual se lograría al reducir el inventario, y por consecuencia, el número de almacenes de productos.

| Ámbito competitivo | Ventaja Competitiva: Bajo costo | Ventaja Competitiva: Diferenciación |
|---|---|---|
| Objetivo amplio | Liderazgo en costo | Diferenciación |
| Objetivo reducido | Centrarse en el costo | Centrarse en la diferenciación |

Figura 3. Estrategia de Latter (adaptado de Applegate et al., 1999)

- La amenaza de nuevos entrantes afecta directamente a la unidad estratégica del negocio, lo cual implica un incremento en la competencia. Con esto, se puede afectar el nivel de competitividad en la industria. Entre menos competidores existan en una industria, una organización específica tiene mayor probabilidad de vender los bienes/servicios ofertados en condiciones favorables para ella. En consecuencia, es importante que a través del uso de TICs una organización cree barreras que impidan la entrada de más competidores dentro de la misma industria.

- Los productos substitutos representan una amenaza a la unidad estratégica de negocio ya que ofertan el efecto final similar al de una organización particular. Por ejemplo, si una organización produce refrescos, esta tiene la amenaza de aquellas que producen jugos, agua, etc. Esto podría implicar la limitación en las utilidades a los mercados potenciales y un tope a los precios. Generando con esto, un cambio en la relación precio/desempeño, así como la redefinición de productos y servicios. Ante la oferta de productos substitutos la organización deberá:

    - Hacer un cambio de la estructura de competitividad de la industria y ampliación de sus alcances
    - Generar nuevas posibilidades de competir con relación a los productos substitutos
    - Crear un aumento de la diferenciación de un producto en ciertos aspectos relacionados con la misma información y comunicación
    - Reducir costos con innovaciones en los procesos o en el análisis de identificación de los mismos. Crear barreras de entrada a nuevos entrantes, y también de salida Lograr mejorar los canales de distribución y el desempeño de la fuerza de ventas
    - Darle alta responsabilidad al negocio, permitiendo reaccionar eficazmente a los cambios del medio pero entendiendo las reacciones históricas de la empresa.

Para contrarrestar el efecto de las fuerzas la organización deberá crear barreras de entrada. Una forma de hacerlo es a través de la identificación de una estrategia para mantener o bajar el precio de su producto o servicio ofertado. Para esto, la organización deberá hacer hincapié en la diferencia de su producto o servicio respecto a otros, aprovechar la curva de aprendizaje que está comenzando la competencia ya que ésta no conoce el mercado, seleccionar el mercado de más enfoque y utilidad, así como el identificar la posición de la organización respecto a la competencia (es decir identificar fortalezas y debilidades, oportunidades y amenazas). También debe definir las prioridades para obtener la ventaja respecto a nuevos entrantes y/o competencia.

No todas las fuerzas impactan de igual forma a un segmento de la industria. Las oportunidades e intensidad de reglas de competencia varían de una industria a otra. Lo que para el líder es una estrategia de ventaja necesaria para otras no lo es, ya que cada negocio es diferente y no tienen la misma importancia. Debido a esto Porter detalló en su modelo el cómo y porqué existen diversas necesidades y su impacto o importancia en el producto ofertado. Porter definió una serie de estrategias que ayudan a una organización a sobresalir, las cuales son:

1. Mecanismo competitivo, es decir bajar sus costos o diferenciar sus productos o servicios
2. Objetivo competitivo. En éste se define la meta y estrategia de negocio y segundo definir el tipo de ventaja competitiva que se busca atender y la que se pretende atender.

Una barrera de entrada exitosa no solo oferta un nuevo producto o servicio que satisfaga las necesidades del cliente, sino que lo mantiene cautivo del servicio/producto, es decir entre mayor sea la necesidad de este, mayor será la barrera para la entrada de otros.

Para identificar el posicionamiento de las organizaciones, éstas deben:

- Conocer las fuerzas que impulsan la estrategia de la industria
- Identificar las oportunidades en las que se puede alcanzar o mantener las ventajas competitivas
- Identificar el posicionamiento deseado o final
- Identificar cómo y dónde se genera un valor agregado sobre las actividades de la cadena de valor de la empresa y del sistema de valor del sector en donde se encuentra
- Reforzar y crear nuevas competencias ya que éstas servirán de base para construir la estrategia de competitividad de la empresa y de su reposicionamiento

Lo anterior nos lleva a una serie de preguntas, en las cuales el común denominador sería encontrar una serie de oportunidades de negocio. Dichas oportunidades se encuentran en un análisis sistemático llamado cadena de valor, el cual es el conjunto de actividades interdependientes que brinda un producto o servicio. De acuerdo a la combinación de actividades se afectará

uno o más de dichos valores de la cadena, lo cual se reflejará en una mejor efectividad, dependiendo del cambio de actividad, y otras al alterar la relación entre ellas. Por lo tanto, las acciones que se lleven a cabo, afectarán significativamente la cadena de valor existente entre cliente y proveedor. Dicha cadena de valor condiciona todas las actividades relacionadas y permea de arriba hacia abajo, influyendo en todas las actividades para darle un valor agregado a cada una de ellas. El fin principal es mejorar el producto/servicio ofertado al cliente para que la empresa se posicione mejor dentro de su industria.

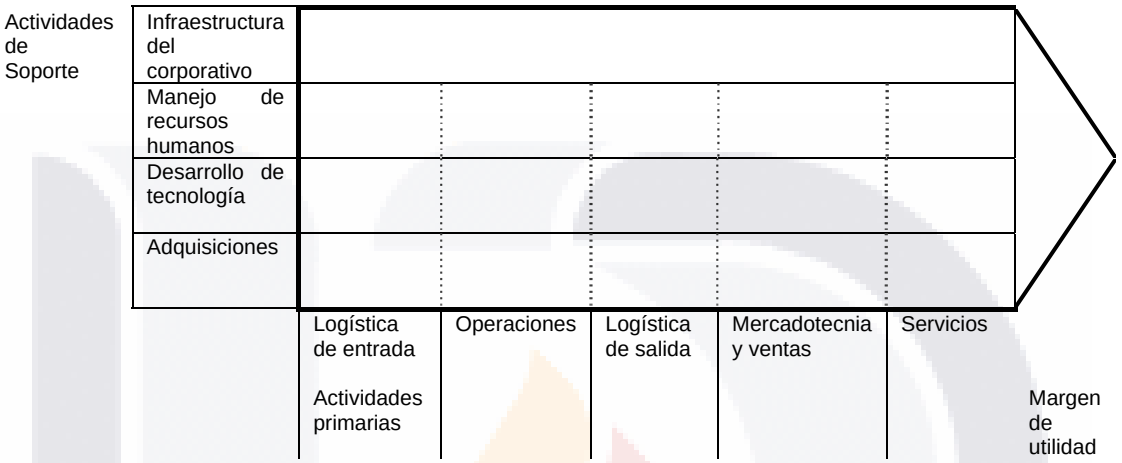


Figura 4. Cadena de valor (adaptado de Applegate et al., 1999)

En la cadena de valor intervienen:

- Logística de entrada. Se refiere a todos los materiales recibidos, almacenados y distribuidos en locales de fabricación
- Operaciones. Es cuando se transforma el material en producto terminado
- Logística de salida. Almacenamiento y distribución de productos
- Mercadotécnia y ventas. Promoción y fuerza de ventas
- Servicios. Mantenimiento o mejora del valor del producto
- Infraestructura organizacional. Apoyo de toda la cadena de valor, tales como la gestión general, planificación, finanzas, contabilidad, servicios jurídicos, el gobierno, los asuntos pendientes, y la gestión de la calidad
- Manejo de recursos humanos. Reclutamiento, contratación, formación y desarrollo
- Desarrollo de tecnología. Mejoramiento de productos y procesos de manufactura
- Adquisiciones. Compra de insumos

Dado todo lo anterior es claro identificar que una buena elección de tecnología contribuye al mejoramiento total de la organización. Por el contrario, una mala decisión pude provocarle serios problemas. En consecuencia, es vital el elegir la mejor tecnología. Por ejemplo dentro de las tecnologías que regularmente se deben elegir están, el invertir en software, en manejadores de bases de datos, adquisición de nuevo equipo, instalación

de infraestructura, entre otros. Se sabe que en el mercado existe una gran variedad de ellos y la decisión de compra debe hacerse para minimizar riesgos y maximizar el dinero invertido en ella. Un ejemplo pueden ser los sistemas operativos.

Existen distintos tipos de sistemas operativos (SO). Deitel (1999) los clasifica como: multiprogramación ó multiusuario, por lotes, de tiempo real, de tiempo compartido, distribuidos, de red, y paralelos. Y dentro de cada clase existen distintas características, es decir no todos son la solución ideal para cada tipo de problema u organización. Entonces, es necesario reducir el nivel de incertidumbre al decidir cual sistema operativo seleccionar. Un benchmarking puede ayudar a tomar la mejor decisión, con esto se estará minimizando la incertidumbre.

**Porque Comparar Sistemas Operativos**

De acuerdo a la clasificación de los sistemas operativos, es importante conocer las características de cada uno de ellos, esto con la finalidad de poder compararlos y poder elegir en base a las necesidades que se tengan. Por ejemplo, se tienen la siguiente clasificación de SO y la característica de cada uno de ellos:

- Multiprocesador.- soporta abrir un mismo programa en más de una CPU o varios procesos simultáneamente, cada uno en un CPU diferente
- Multitarea.- Permite que varios programas se ejecuten simultáneamente
- Multitramo.- Permite que diversas partes de un solo programa funcionen al mismo tiempo
- Multiusuario.- Dos o más usuarios ejecutan sus programas al mismo tiempo
- Tiempo Real.- Responde a las entradas inmediatamente.

Ahora bien, dentro de los SO quizá se haya escuchado hablar de los siguientes: Linux, Windows, Fedora, CentOS, Windows NT, FreeBsd, por mencionar algunos, pero la pregunta ahora es ¿Cuál es la diferencia entre cada uno de los sistemas operativos mencionados anteriormente?

Linux es el núcleo (kernel) de un SO, es decir, lo que interacciona directamente con el hardware. El núcleo es un medio de comunicación para el resto de los componentes del sistema operativo, es decir es un puente entre el hardware y el software de un equipo computacional, tal y como se puede apreciar en la siguiente figura:


Figura 5. Estructura Linux (adaptada de Jaus, 2009)

Al utilizar un núcleo, es posible montar una serie de programas sobre el SO para que este los administre. Algunos llaman a ése SO igual que al núcleo, es decir, simplemente Linux, pero la forma correcta de llamarle es GNU/Linux. Esto debido a que existen diversas maneras de montar un sistema operativo basándose en el núcleo Linux más otros componentes y cada una tiene su propio nombre. Por ejemplo, existen Mandriva, RedHat, Suse, FreeBSD, CentOS, por mencionar algunos, las cuales son llamadas correctamente como distribuciones y aunque tomen nombres distintos todas tienen algo en común: todas emplean el sistema operativo Gnu/Linux. Dentro de las características que da García (García, 2008) de Linux, está el ser multiusuario y multitarea así como tener sistema de archivos distribuidos.

A diferencia de las numerosas distribuciones de Linux, de acuerdo con Sandoval (2005), cada proyecto de las distribuciones mantiene su propio árbol de fuentes y su propio kernel. En la práctica, sin embargo, las diferencias en el entorno de usuario entre los distintos BSD son menores que las que hay en Linux. Los tres distribuciones BSD libres son:

- FreeBSD.- Tiene como meta ofrecer alto rendimiento y ser un SO amigable al usuario final además de ser uno de los favoritos entre proveedores de contenidos web. Este funciona en computadoras personales (PCs) y en procesadores Alpha de Compaq. El proyecto FreeBSD cuenta con un número de usuarios significativamente mayor que los otros proyectos. FreeBSD se clasifica como un SO multiusuario, capaz de efectuar multitarea con apropiación y multiproceso en plataformas compatibles con múltiples procesadores; el funcionamiento de FreeBSD está inspirado, en la variante 4.4 BSD-Lite de UNIX. Aunque FreeBSD no puede ser propiamente llamado UNIX, al no haber adquirido la debida licencia de The Open Group, este sí está hecho para ser compatible con la norma POSIX, al igual que varios otros sistemas clones de UNIX. El sistema FreeBSD incluye el núcleo, la estructura de archivos del sistema, bibliotecas de la API de C, y algunas utilidades básicas.

- NetBSD.- Que tiene como meta la Portabilidad: no en vano su lema es "of course it runs NetBSD" (que podría traducirse como "por supuesto que se ejecuta en NetBSD"). Funciona en máquinas que

abarcan desde dispositivos móviles como las PDAs hasta grandes servidores; e incluso ha sido usado por la NASA en misiones espaciales. Es una excelente elección para utilizar viejo hardware de Intel.

- OpenBSD.- Tiene como meta la seguridad y la integridad del código. Combina el concepto de código abierto y una revisión rigurosa del código que dan como fruto un sistema muy correcto, elegido por instituciones preocupadas por la seguridad tales como bancos, entidades de cambio y departamentos gubernamentales de los EEUU. Al igual que NetBSD, funciona en gran variedad de plataformas.

Existen dos distribuciones BSD más que no son de código abierto, éstas son: BSD/OS y el MacOS X de Apple. A continuación se describe brevemente cada uno de ellos

- BSD/OS.- Es un derivado más antiguo de 4.4BSD. No es código abierto pero es posible conseguir licencias de su código fuente a un precio relativamente bajo. Parecido a FreeBSD en muchos aspectos. El 31 de diciembre de 2004 se finalizó el desarrollo y venta de este sistema.
- Mac OS X.- Es la última versión del sistema operativo para la gama Macintosh de Apple Computer Inc. El núcleo BSD Unix de éste sistema operativo, Darwin, está libremente disponible como SO de fuente abierto totalmente funcional para arquitecturas x86 y PPC. El sistema gráfico Aqua/Quartz y la mayoría de las demás aspectos característicos de Mac OS X son código cerrado. Varios desarrolladores de Darwin son también parte del comité de FreeBSD y viceversa.

Ahora bien, existen diversas maneras de referirnos a Unix, o mejor dicho de entender el significado que se le puede dar. Estos se catalogan de la siguiente manera:

a) Familia Unix.- Dicho término se utiliza en la literatura cuando se esté hablando de un grupo genérico de SO con determinados criterios, y dentro de éstos se encuentran diferentes versiones tales como son BSD, AIX, Xenix, GNU. Dentro de la versión de BSD, están las distribuciones de: a) FreeBSD, b) OpenBSD, por citar algunos. Respecto a la versión GNU tenemos las distribuciones: a) Fedora, b) CentOS, c) Debian, y d) Red Hat.
b) SO Unix.- Se emplea dicho término en la literatura para referirse a las subfamilias de SOs descendientes de la primera versión creada por laboratorios Bell, siendo ésta la propietaria intelectual y de codificación, lo cual significa no ser código abierto.
c) Marca Unix.- Término empleado al referirse que se es propiedad de The Open Group, la cual es una organización de estandarización, independiente si el SO es de la familia Unix o un descendiente del original. Esto implica que no es código abierto.

Matzan (2006) menciona que para las versiones BSD y GNU se pueden

seleccionar las versiones como Fedora, FreeBSD, CentOS, Debian, etc., esto por mencionar las distribuciones más significativas. Además, él describe que las distribuciones más conocidas orientadas a servidores son: RedHat Enterprise Edition, Suse Linux Enterprise Edition, Mandriva Corporate Server. Dado esto, aunque sea el mismo SO, ya sea orientado o no servidor, puede presentar desempeños diferentes. Así pues, es necesario identificar y comprender cuales son las características de productividad de cada uno de ellos para tomar la mejor decisión.

Algunas distribuciones dicen cumplir con ciertos elementos claves, los cuales son críticos para el desempeño de las aplicaciones de las empresas. Esto sugiere un análisis que permita identificar las ventajas y desventajas de cada uno de ellos. Generalmente, no existen estudios empíricos que lleven a cabo esto, aunque existen una serie de benchmarks que pueden usarse como base.

Con el fin de contribuir a la solución de este tipo de problemática se llevó a cabo un estudio empírico efectuado por Mohamed et al. (2006) donde se evaluó el desempeño de los SO Windows 2003 Server, Red Hat 9, y FreeBSD 4.9. Estas pruebas de desempeño se hicieron mediante la transferencia de datos empleando un loop back. El centro de operaciones de videoconferencia de la UNAM (VNOC, 2009) define loop back como *un tipo de prueba que está disponible en las opciones de los equipos, el cual crea una conexión virtual consigo mismo*.

Mohamed obtuvo para el escenario de loop back lo siguiente: en Windows 2003 se observó un mayor aumento en el envío de paquetes de ida y vuelta o Round Trip Time (RTT) respecto a FreeBSD 4.9, y en último sitio se ubicó Red Hat 9 en donde el tamaño de paquetes enviados fue de 150 bytes en adelante. Un menor aumento en el RTT de 30000 ns en FreeBSD 4.9 y Windows 2003, mientras que Red Hat se mantuvo constante. Sin embargo, después de 1440 paquetes de octetos, Windows 2003 tiene un aumento muy significativo en comparación con el resto de los SOs. En consecuencia, Mohamed concluye que al realizar la prueba de loopback el pobre desempeño mostrado por Windows 2003 en comparación a los otros SOs, se debió a que en IPv6 existió una fragmentación de paquetes en las pruebas de 1440 bytes, mientras que para Red Hat 9, dicha fragmetación se presenta en los 16384 bytes y para FreeBSD 4.9 se da en los 14436 bytes enviados.

Otro escenario de este estudio, consistió en la simulación de una LAN y una WAN donde la unidad de medida fue en base al throughput. Wikipedia (Wikipedia, 2009c) define throughput como "*volumen de trabajo o de información que fluye a través de un sistema. Así también se le llama al volumen de información que fluye en las redes de datos utilizando una herramienta que trabaje bajo el protocolo SNMP (Simple Network Managmente Protocol).*" La Figura 2 muestra el funcionamiento del throughput.

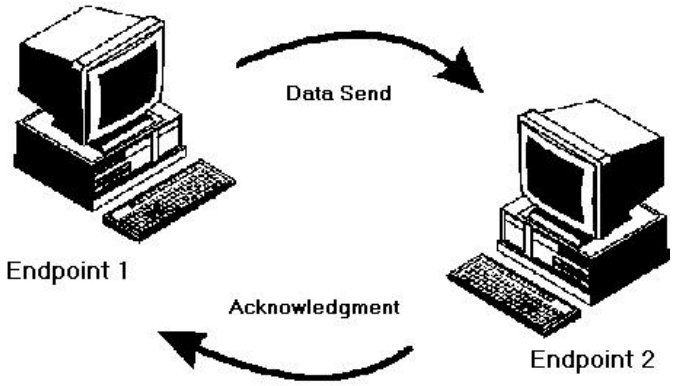


Figura 6. Throughput (Adaptado de QChec Console 2009)

Los resultados observados fueron los siguientes: para el rendimiento de TCP se midió en base al envío de paquetes de los 32 a los 1500 bytes observándose en todos los casos un mejor desempeño en Red Hat 9, seguido de FreeBSD4.9 y por último Windows 2003. Al medir el throughput los resultados fueron similares, salvo que en ésta ocasión Red Hat 9.0 mostró un desempeño muy por encima de FreeBSD y Windows 2003 en ése orden. En conclusión, éste escenario observó una disminución en el rendimiento y un aumento del RTT debido a la sobrecarga del ruteador, así como el que Red Hat 9 y FreeBSD muestran poco rendimiento en la carga útil, esto debido a que el buffer se encuentra saturado, ya que los paquetes se fragmentaron para luego volverse a re-ensamblar, en paquetes de información de gran tamaño

De acuerdo a otro estudio realizado por Zeadallyy Raicu (Zeadally, 2000), en el cual se evaluó una arquitectura de red compararando el rendimiento de IPv4 y la próxima generación de Internet (IPv6). Además, se compilaron las estadísticas de rendimiento de cada una de las redes empleando los protocolos Transmision Control Protocol (TCP) y User Datagram Protocol (UDP), en dónde se evaluó el rendimiento de dichos protocolos; la demora (tasa de pérdida de paquetes). Para este último, se tomó como referencia el viaje de ida y vuelta de la información y el tiempo utilizado en la transferencia.

Dicho estudio se implementó bajo los SO Windows 2000 y Solaris 8, ambos bajo la misma configuración de hardware. Adicionalmente, realizaron pruebas con diferentes configuraciones, incluyendo un par de ruteadores comerciales, los cuales apoyan la doble pila IPv4-IPv6 del protocolo. Para las pruebas se utilizaron seis parámetros de rendimiento. Ellos concluyeron que se obtiene una menor degradación en el rendimiento del TCP en una red a gran escala, mostrando el protocolo TCP un rendimiento ligeramente mayor que el observado en UDP. Este último mostró poca fluctuación en la frecuencia de demora, y una menor tasa de pérdida de paquetes. Así como poca diferencia en el viaje de ida y vuelta de éstos. Lo anterior haciendo referencia en la comparación de una red IPv6 respecto a otra con IPv4.

Zeadally y Raicu requirieron configurar dos estaciones de trabajo con las mismas características. El modelo bajo el cual se llevó a efecto el estudio se realizó en termino de las siguientes variables: a) Throughput, b) Round Trip Latency, c) CPU utilization y d) Socket-Creation Time y TCP-Connection Time.

1. El Throughput, se midió enviando paquetes de información de aplicación a aplicación empleando los SO Windows 2000 y Solaris 8,donde los paquetes con un tamaño menor a 256 bytes, el protocolo IPv4 triplicó su desempeño en comparación a IPv6. Mientras que para paquetes de 1024 bytes o superior, mostró un decremento en el envío de éstos. Los resultados fueron poco diferentes en Windows 2000 ya que el throughput es muy similar para IPv4 e IPv6 en mensajes pequeños de TCP. Para mensajes mayores a los 512 bytes, IPv4 fue 11% mayor. Al emplear el protocolo UDP se mostró en ambos SOs desempeños similares en throughput para mensajes menores a 256 bytes en IPv4 e IPv6; pero al incrementarse el tamaño del mensaje el throughput para IPv6 tuvo un rendimiento 25% menor respecto a IPv4. El estudio reveló que la diferencia en el desempeño de IPv4 e IPv6 es más pronunciado para paquetes pequeños con TCP respecto a UDP, ya que el protocolo TCP emplea el algoritmo de Nagle. Este algoritmo retarda el envío ya que requiere confirmar su envío, dicha optimización afecta el desempeño de los paquetes que emplean IPV6 más que IPv4, ya que la sobrecarga de encabezados se asocia con IPv6.
2. Para el escenario RTT se enviaron paquetes de información de hasta 1Kbyte, mostrando que para IPv6 se obtuvo una menor latencia del 30% respecto a IPv4 en Windows 2000. Con Solaris 8, se observó un incremento del 5% en latencia de IPv6 comparada con IPv4. En paquetes mayores a 1 Kbyte, se incrementó la latencia en ambos SO alrededor de 1 o 2% usando TCP en IPv6. Quizá se asocie lo anterior a la amortización de las cabeceras asociadas a los paquetes de mayor tamaño. Los resultados para UDP muestran 30% mayor latencia con IPv6 que IPv4 en Windows 2000 para mensajes mayores a 1 Kbyte, pero al incrementarse el tamaño de los mensajes, la latencia entre IPv4 e IPv6 decrece. Con Solaris, se obtuvo 5% mayor latencia en IPv6 respecto IPv4
3. Se midió el uso de CPU en el envío de paquetes empleando el administrador de tareas de Windows 2000 y se observó que TCP sobre IPv6 usó 20% más recursos respecto TCP sobre IPv4. Respecto a Solaris, ésta medición no fue llevada a cabo.
4. El tiempo que tomó la conexión de socket fue superior en Windows 2000 respecto a Solaris empleando el protocolo UDP y TCP, lo mismo sucedió respecto a las versiones IPv6 e IPv4

# 2. Aspectos relevantes de Tecnologías Relacionadas a la Investigación

Se conoce como protocolo de comunicaciones a un conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre sistemas (Wikipedia, 2009b), por tal motivo el estudio que pretendo realizar involucra la comunicación entre Sistemas Operativos y protocolo de comunicación, por lo que a continuación se explicarán dichos conceptos.

## 2.1.- Definición y Concepto de Protocolo

Tanenbaum ( 2003) **define protocolo como:** un conjunto de solicitudes y respuestas convenidas entre un transmisor y un receptor, con el fin de comunicarse a través de una red u otra interface.

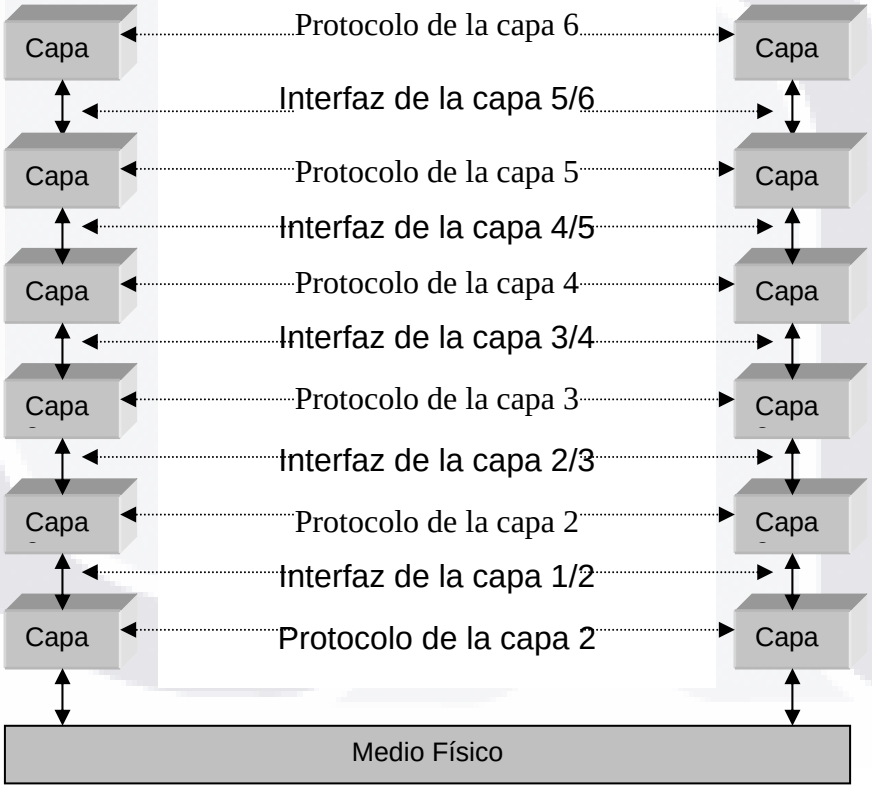


Figura 7. Capas, protocolos e interfaces (adaptado de Comunicaciones, 1998)

## 2.2.- Funciones de los Protocolos

Las descripciones para las funciones de los protocolos en la literatura (Comunicaciones, 1998) son como sigue:

- Segmentación y Re-ensamble: conforme se pasa la información hacia las

capas inferiores (transmisión) la información se rompe en unidades (segmentación). En recepción se hace el proceso contrario (re-ensamble).

- Encapsulado: proceso mediante el cual a las unidades de datos se les agregan encabezados (headers) y finalizadores (trailers). La información que llevan los ambos es de tipo dirección (fuente, destino, control del protocolo, prioridad, códigos de detección de errores, etc.).
- Control de la Conexión: Existen 2 tipos de control de la conexión:
    - Orientado a conexión: Transmisión con previo aviso.
    - Orientado a no conexión: Transmisión sin previo aviso.
- Entrega Ordenada de Tramas: las tramas pueden venir en desorden (típico en redes que usan datagramas). El protocolo debe proporcionar algún mecanismo para poder ordenar las fracciones desordenadas. A cada unidad se le debe asignar un número único.
- Control de Flujo: Propósito: Evitar la saturación del receptor. Mecanismo: uso de número de secuencia. Implentación: en varias capas (enlace, red, transporte).
- Control de errores: Métodos para recuperación de la línea en caso de tramas perdidas o dañadas. Usa algún método de secuencia de las tramas o confirmación continua.

## 2.3.- Organización y modelado de los protocolos

Literatura previa (Comunicaciones, 1998) explica que para organizar las tareas de los protocolos ha sido necesario clasificarlas y ubicarlas en un conjunto de capas de trabajo, las cuales funcionan para simplificar la meta final: la comunicación. Y define que: El concepto de capa en un protocolo es de alguna manera similar al de las subrutinas de un programa complicado. La manera de estructurarlos se reconoce como MODELO y a la forma de acomodarlos en capas se le reconoce como MODELOS POR CAPAS.

## 2.4.- Modelado de 7 Capas de OSI de la ISO

### Antecedentes

El modelo ISO (International Standards Organization) es un estándar que cubre todos los aspectos de las redes de comunicaciones y fue creado en 1947 (Forouzan, 2002) por la Organización Internacional de Estandarización (OSI), quién es un organismo multinacional dedicado a establecer acuerdos mundiales sobre estándares internacionales.

La literatura (Comunicaciones, 1998) describe que las técnicas de transmisión (protocolos) fueron desarrollados por os fabricantes como una respuesta a la necesidad de las comunicaciones en el área de la computación, con el fin de explotar las mayores velocidades disponibles de transmisión y para implementar los grados de control más sofisticados. Pero su gran inconveniente fue que cada fabricante trabajaba por separado, y no existía compatibilidad entre equipos de diferentes marcas. Si un cliente compraba equipo a un fabricante, quedaba comprometido en continuar con esa marca en crecimientos y expansiones futuras; su equipo instalado no podía crecer con sistemas diferentes. Además, sugirió otro problema: cuando distintos departamentos de una organización adquirieron tecnologías

de redes procedentes de diversos fabricantes y el intercambio de información fue necesario, esto no era posible. Un escenario similar se presentó cuando una empresa realizaba la compra de otra o establecía sucursales, ya que cada oficina manejaba diferentes tecnologías de red.

Los problemas de la heterogeneidad de las redes de cómputo de una organización y la dependencia hacia un solo fabricante influyeron en el desarrollo de los sistemas abiertos. Estos buscan de manera básica lograr la independencia del hardware y software, portabilidad de la aplicación y cumplimiento de los estándares.

ISO define un sistema abierto como todo el conjunto de interfaces, servicios y formatos de soporte, además de otros aspectos de usuario para la interoperabilidad o portabilidad de aplicaciones, datos o personas, según se especifica en los estándares y perfiles de tecnología informática.

## 2.5.- Importancia de OSI

El objetivo del modelo OSI es permitir la comunicación entre sistemas distintos sin que sea necesario cambiar la lógica del hardware o software subyacente (Forouzan, 2002). El modelo OSI no es un protocolo; es un modelo para comprender y diseñar una arquitectura de red flexible, robusta e interoperable. Este es una arquitectura por niveles para el diseño de sistemas de red que permite la comunicación entre todos los tipos de computadoras. Está compuesto por siete niveles separados, pero relacionados, cada uno de los cuales define un segmento del proceso necesario para mover la información a través de una red. Los niveles son: físico (nivel uno, conexión de equipos), enlace de datos (nivel dos, detección de errores), red (nivel tres, enrutamiento de mensajes), transporte (nivel cuatro, integridad de mensajes), sesión (nivel cinco, diálogos de control), presentación (nivel seis, interpretación de los datos) y aplicación (nivel siete, datos normalizados).

La literatura (Comunicaciones, 1998) también explica que el modelo OSI surgió frente a la necesidad imperante de interconectar sistemas de procedencia diversa en los que cada fabricante empleaba sus propios protocolos para el intercambio de señales. Este modelo fue creado como tal, es decir, que no necesariamente todos los fabricantes tenían que sujetarse a él. Pero al hacerse éste un estándar, todo aquel que no fuera compatible o hecho con base en OSI de alguna manera iba quedar relegado en el mercado ya que por ningún motivo el usuario deseaba seguir obligado a vivir con una sola marca, con todas las desventajas que esto representaba.

Existieron gigantes de las Telecomunicaciones que en un principio se opusieron al desarrollo de su tecnología con base en el modelo OSI, pero conforme vieron sus ventajas y desventajas, se sujetaron al nuevo estándar.

El modelo de referencia para la interconexión de sistemas abiertos OSI fue aprobado por el organismo internacional ISO en 1984, bajo la norma ISO 7498, después de varios años de arduo trabajo.

## 2.6.- Arquitectura de TCP/IP

El modelo de referencia OSI proporciona una arquitectura de 7 niveles, alrededor de los cuales se pueden diseñar protocolos específicos que permitan a diferentes usuarios comunicarse abiertamente. La elección de los 7 niveles se dividió básicamente en los 3 puntos siguientes:

1. La necesidad de tener suficientes niveles para que cada uno no sea tan complejo en términos el desarrollo de un protocolo detallado con especificaciones correctas y ejecutables.
2. El deseo de no tener tantos niveles y provocar que la integración y la descripción de éstos lleguen a ser demasiados difíciles.
3. El deseo de seleccionar fronteras naturales, con funciones relacionadas que se recolectan en un nivel y funciones muy separadas en diversos niveles.

También, se tomó en cuenta para el desarrollo del modelo OSI que cada nivel debe contar con ciertas premisas, las cuales son las siguientes:

- Cada nivel realiza tareas únicas y específicas y debe ser creado cuando se necesite un grado diferente de abstracción.
- Todo nivel debe tener conocimiento de los niveles inmediatamente adyacentes y sólo de éstos.
- Todo nivel debe servirse de los servicios del nivel anterior, a la vez que los debe de prestar al superior.
- Los servicios de un nivel determinado son independientes de su implantación práctica.
- Los limites de cada nivel se deben seleccionar, teniendo en cuenta que minimicen el flujo de información a través de las interfaces establecidas.

En la figura siguiente se muestra la estructura en niveles (capas) del modelo OSI.

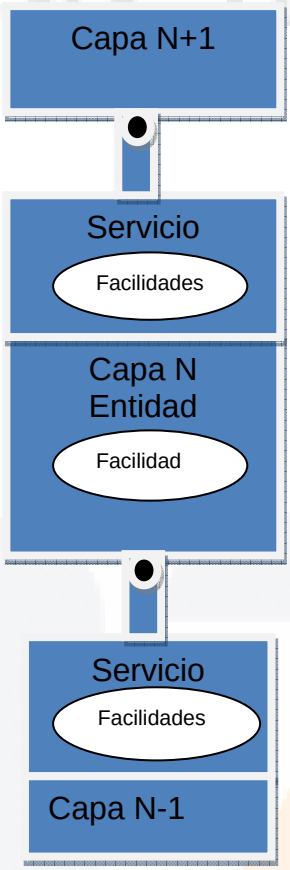


Figura 8. Estructura en niveles de capas del modelo OSI (adaptado de Comunicaciones, 1998)

## 2.7.- Capas del Modelo OSI

Como se explicó anteriormente, el modelo OSI está formado por 7 capas. Cada una con una función diferente. La literatura (Comunicaciones, 1998) explica cada capa de la siguiente manera:

### Capa Física

El nivel físico es el encargado, primordialmente, de la transmisión de los bits de datos (0s ó 1s) a través de los circuitos de comunicaciones. El propósito principal de este nivel es definir las reglas para garantizar que cuando la computadora emisora transmita el bit “1”, la computadora receptora verifique que un “1” fue recibido y no un “0”. Es el nivel de comunicación física de circuitos.

Adicionalmente, esta capa provee los medios mecánicos, eléctricos, funcionales y de procedimiento para establecer, mantener y liberar conexiones físicas entre el dispositivo terminal (DTE) y el punto de conexión a la red (DCE), o entre dos DTE.

- Mecánicos: define el tipo de conector, sus dimensiones físicas, la distribución de pines, etc.
- Eléctricos: concierne a las características eléctricas, como su voltaje, nivel, impedancia, etc.

- Funcionales: define el significado de los niveles de tensión en cada uno de los pines del conector.
- De procedimiento: define las reglas aplicables a ciertas funciones y la secuencia en que éstas deben incurrir.

Como ejemplo, algunas de las normas dentro de este nivel son: X2 1, V. 10,V.11, V.24, *V.35,* Y.430, Y.431 del CCITT, ISO 2,110 (EIA 232), 4,902(FIA-449) y 9,314 (FDDI).

**Capa de Enlace**

Es el nivel de datos en donde los bits tienen algún significado en la red, y este nivel puede verse como el departamento de recepción y envío de una compañía de manufactura, el cual debe tomar los paquetes que recibe de la Capa de Red y prepararlos de la forma correcta (tramas) para ser transmitidos por el nivel físico. De igual forma sucede cuando recibe paquetes (bits) del nivel físico y tiene que ponerlos en la forma correcta (tramas) para verificar si la información que está recibiendo no contiene errores, si los paquetes vienen en orden, si no faltan paquetes, etc., para entregarlos a nivel de red sin ningún tipo de error.

Dentro de sus funciones se incluyen la de notificar al emisor (la computadora remota) si algún paquete (trama) se recibe en mal estado (basura); si alguna de las tramas no se recibieron y se requieren que sean enviadas nuevamente (retransmisión), o si una trama esta duplicada, también cuando la trama llegó sin problemas. En resumen, es responsable de la integridad de la recepción y envío de la información, así como de saber dónde comienza la transmisión de la trama y dónde termina, y garantizar que tanto la computadora transmisora como la receptora estén sincronizadas en su reloj y que emplean el mismo sistema de codificación y decodificación.

En esta capa se determina el uso de una disciplina de comunicaciones conocida corno HDLC (*Hígh Level Data Link Control).* El HDLC es el protocolo de línea considerado como un estándar universal, que muchos toman como modelo. Los datos en HDLC se organizan en tramas. La trama es un encuadre que incluye bits de redundancia y control para corregir los errores de transmisión; además, regula el flujo de las tramas para sincronizar su transmisión y recepción, también enmascara a las capas superiores de las imperfecciones de los medios de transmisión utilizados. Dentro de esta capa se encuentra el protocolo HDLC (3,309), el procedimiento LAP B (7,706) y las normas IEEE 802.2-7 para LAN.

**Capa de Red**

El nivel de red es el responsable del direccionamiento de mensajes y de la conversión de las direcciones y nombres lógicos a físicos. También determina la ruta del mensaje desde la computadora emisora hasta la computadora receptora, dependiendo de las condiciones de la red.

Dentro de las funciones de ruteo de mensajes evalúa la mejor ruta que debe seguir el paquete, dependiendo del tráfico en la red, el nivel de servicios, etc. Los problemas de tráfico que controla tienen que ver con el ruteo *(routting),*

intercambio *(switching)* y congestionamiento de paquetes en la red.

Asimismo, maneja pequeños paquetes de datos juntos para la transmisión a través de la red, así como la reestructuración de tramas de datos grandes (números de bits) en paquetes pequeños. En la computadora receptora se re-ensamblan los paquetes en su estructura de datos original (trama).

A la información proveniente de la capa de transporte se le agregan componentes apropiados para su ruteo en la red y para mantener un cierto nivel en el control de errores. La información es presentada según el método de comunicaciones para acceder a la red de área local, la red de área extendida (como los enlaces El) y la conmutación de paquetes (como *X.25,* etc.).

El diseño de este nivel debe considerar que:

- Los servicios deben ser independientes de la tecnología empleada en la red de datos.
- El nivel de transporte debe ser indiferente al número, tipo y topologías de las redes utilizadas.
- La numeración de la red debe ser uniforme a través de LANs y WANs.

El servicio de red se define en la recomendación X.213 (ISO 8,348 y 8,880 para LANs). Como ejemplo de este nivel, tenemos las recomendaciones *X.25,*X.32, X.3, X.28, X.29 del CCITT para redes de conmutación de paquetes, la ISO 9,420 protocolo de enrutamiento para LAN y las 8348, 8208, 8473, 8648 para sistemas de proceso de información.

**Capa de Transporte**

El nivel de transporte es llamado ocasionalmente el nivel de *host-to-host* o el nivel de *end-to-end,* debido a que en él se establecen, mantienen y terminan las conexiones lógicas para la transferencia de información entre usuarios. En particular de la capa 4 hasta la 7 son conocidas como niveles *end-to-end* y los niveles 1 a 3 son conocidos como niveles de protocolo.

El nivel de transporte se relaciona más con los beneficios de *end-to-end,* como son las direcciones de la red, el establecimiento de circuitos virtuales y los procedimientos de entrada y salida a la red. Solamente al alcanzar el nivel superior de transporte (sesión) se abordarán los beneficios que son visibles al usuario final.

Este nivel puede incluir las especificaciones de los mensajes de difusión (*broadcast*), los tipos de datagramas, los servicios de correo electrónico, las prioridades de los mensajes, la recolección de la información y su administración, seguridad, tiempos de respuesta, estrategias de recuperación en casos de falla y segmentación de la información cuando el tamaño es mayor al máximo del paquete según el protocolo.

Al recibir información del nivel de red, el nivel de transporte verifica que la información esté en el orden adecuado y revisa si existe información duplicada o extraviada. Si la información recibida está en desorden, lo cual es posible en redes grandes cuando se rutean las tramas, el nivel de

transporte corrige el problema y transfiere la información al nivel de sesión en donde se le dará un proceso adicional.

Algunos de los principales parámetros de calidad de los que se hace mención son los siguientes:

- Retardo en el establecimiento de la conexión.
- Falla en el establecimiento de la conexión.
- Protección contra intrusos
- Niveles de prioridad.
- Interrupción por congestión.
- Retardo en la liberación de la conexión.
- Error en la liberación, etc.

En este nivel trabajan las recomendaciones X.214 (ISO 8,072) y X.224 (ISO 8,073). La siguiente figura muestra el ordenamiento y funciones de las capas de acuerdo a lo mencionado.

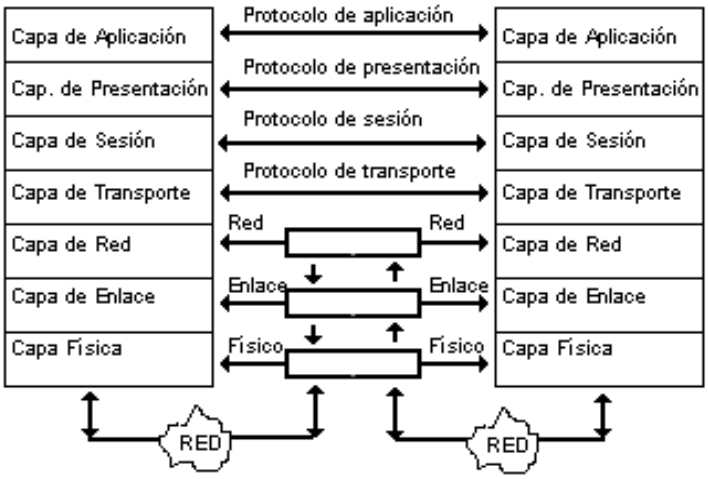


Figura 9. Ordenamiento y funciones del modelo 051 (adaptado de GS Comunicaciones, 1998)

**Capa de Sesión**

Este nivel es el que permite que 2 aplicaciones en diferentes computadoras establezcan, usen y terminen la conexión llamada sesión. El nivel de sesión maneja el diálogo que se requiere en la comunicación de 2 dispositivos. Establece reglas para iniciar y terminar la comunicación entre dispositivos y brinda el servicio de recuperación de errores; es decir, si la comunicación falla y ésta es detectada, el nivel de sesión puede retransmitir la información para completar el proceso en la comunicación. El nivel de sesión es el responsable de iniciar, mantener y terminar cada sesión lógica entre usuarios finales.

Para entender mejor este nivel, se puede pensar en el sistema telefónico. Cuando se levanta el teléfono, espera el tono y marca un número, en ese momento se está creando una conexión física que va desde el nivel uno

(físico) como un protocolo de persona a red. Al momento de hablar con la persona en el otro extremo de la línea, se encuentra en una sesión persona a persona. En otras palabras, la sesión es el diálogo de las dos personas que se transporta por el circuito de la conexión telefónica.

También en este nivel se ejecutan funciones de reconocimiento de nombres para el caso de seguridad relacionado a aplicaciones que requieren comunicarse a través de la red, las cuales se listan a continuación:

- Establecimiento de la conexión a petición del usuario.
- Liberación de la conexión cuando la transferencia termina.
- Intercambio de datos en ambos sentidos.
- Sincronización y mantenimiento de la sesión para proporcionar un intercambio ordenado de los datos entre las entidades de presentación.

En el nivel de sesión están las recomendaciones X.215 (150 8,326) y X.225 (ISO 8,327).

**Capa de Presentación**

El nivel de presentación define el formato en que la información será intercambiada entre aplicaciones, así como la sintaxis usada entre las mismas. Se traduce la información recibida en el formato del nivel de aplicación a otro intermedio reconocido. En la computadora receptora, la información es traducida del formato intermedio al usado en el nivel de aplicación de dicha computadora y es, a su vez, responsable de la obtención y liberación de la conexión de sesión cuando existan varias alternativas disponibles.

El nivel de Presentación maneja servicios como la administración de la seguridad de la red, como la encriptación y desencriptación, también brinda las reglas para la transferencia de información *(data transfer)* y comprime datos para reducir el número de bits que necesitan ser transmitidos.

En el nivel de presentación se encuadran por ejemplo, las normas para videotex, telefax y teletex y las normas *X.225* del CCITT.

**Capa de Aplicación**

Al ser el nivel más alto del modelo de referencia, el nivel de aplicación es el medio por el cual los procesos de aplicación acceden al entorno 051. Por ello, este nivel no interactúa con uno más alto.

Este proporciona los procedimientos precisos que permiten a los usuarios ejecutar los comandos relativos a sus propias aplicaciones. Estos procesos de aplicación son la fuente y el destino de los datos intercambiados. Se distinguen primordialmente 3 tipos de procesos de aplicación:

- Procesos propios del sistema.
- Procesos de gestión.
- Procesos de aplicación del usuario.

La transferencia de archivos *(file transfer)* y el acceso remoto a archivos, son probablemente sus aplicaciones más comunes. Las normas más conocidas de este nivel son: X.400 (Correo Electrónico) y X.500 (Directorio) del CCITT; otras son las FTMA (ISO 8,571), DS (9,594), MHS (10,021), ODA (8,613), VT (9,041), RDA (9,570), DTA (10,026) y CMIP.

## 2.8.- Comunicación Entre Capas

La literatura (Comunicaciones, 1998) define el Funcionamiento del modelo OSI como un conjunto completo de estándares funcionales que especifican interfaces, servicios y formatos de soporte para conseguir la interoperabilidad. El modelo OSI se compone de 7 capas, cada una de ellas con una función específica. La utilidad principal del modelo OSI radica en la separación de las distintas tareas que son necesarias para comunicar dos sistemas independientes.

Es importante indicar que no es una arquitectura de red en sí misma, sino que exclusivamente indica la funcionalidad de cada una de ellas. El modelo de referencia OSI se constituye como el marco de trabajo para el desarrollo de protocolos y estándares para la comunicación entre dos capas homónimas ubicadas en equipos separados.

Conforme se avanza en la explicación y funcionamiento de cada una de las capas, se identifica como muchos de los términos se duplican de capa en capa. Un nivel representativo ofrece un conjunto de servicios a la entidad de la capa superior
donde la capa superior se llama Usuario de Servicio y la capa inferior Proveedor de Servicio. La Figura 10 muestra la relación de comunicación entre capas y en la Tabla 1 la función principal de cada una de ellas.

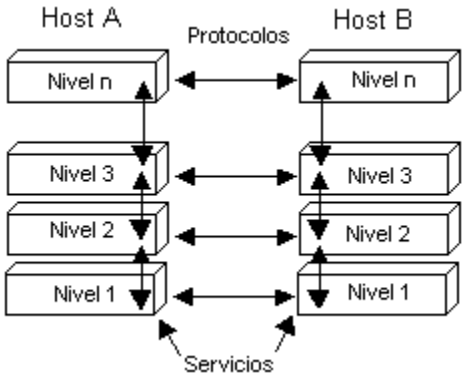


Figura 10. Comunicación entre niveles del modelo OSI.

| Nivel | Nombre | Función |
|---|---|---|
| 7 | Aplicación | Datos normalizados |
| 6 | Presentación | Interpretación de los datos |
| 5 | Sesión | Diálogos de control |
| 4 | Transporte | Integridad de los mensajes |
| 3 | Red | Enrutamiento de los mensajes |
| 2 | Enlace | Detección de errors |
| 1 | Físico | Conexión de equipos |

Tabla 1. Niveles o capas del modelo OSI (adaptado de GS Comunicaciones, 1998)

## 2.9.- Protocolo de Control de Transmisión / Protocolo de Internet (Transmision Control Protocol / Internet Protocol)

### TCP

TCP conocido como Protocolo de Control de la Transmisión (Tanenbaum, 2003), es un protocolo confiable orientado a la conexión que permite que una corriente de bytes originada en una máquina se entregue sin errores en cualquier otra máquina de la interred. Este protocolo fragmenta la corriente entrante de bytes en mensajes discretos y pasa cada uno a la capa de interred.

### IP

De acuerdo con Tanenbaum (2003), Protocolo Internet (IP, Internet Protocol) es un protocolo de datagramas en el que un transmisor inyecta un datagrama de hasta 64KB en la red y tiene la esperanza que llegue. No se ofrecen garantías. El datagrama puede fragmentarse en paquetes más pequeños al atravesar Internet. Estos paquetes viajan de forma independiente, quizá por rutas distintas. Cuando todos los fragmentos llegan al destino, se ensamblan en el orden correcto y se entregan.

Es decir, tal y como se cita (Silberschatz Abraham, 2005), cuando un protocolo genera un mensaje para enviarlo a su correspondiente protocolo en otra máquina, por ejemplo en una red IP, el protocolo TCP (un protocolo de nivel de transporte) actúa como un cliente de IP (un protocolo de nivel de red): los paquetes TCP se pasan al nivel IP para entregarlos al protocolo TCP que se encuentra en el otro extremo de la conexión TCP. IP encapsula el paquete TCP en un paquete IP, el cual se pasa de forma similar al nivel de enlace de datos inferior, para ser transmitido a través de la red a su correspondiente protocolo IP en la computadora de destino. Este protocolo IP entrega entonces el paquete TCP al protocolo TCP de dicha máquina.

## 2.10.-¿Qué es, cómo y dónde fue desarrollado TCP/IP?

Según Forouzan (2002), TCP/IP no es sólo un protocolo, sino que comprende todo un conjunto muy completo de diversos protocolos que prestan diversos servicios. TCP/IP es, probablemente, uno de los protocolos de comunicaciones más viejos en los estándares de redes internas. A inicios de 1969, éste fue desarrollado por el Departamento de Proyectos Avanzados de Investigación de la Defensa de Estados Unidos (DARPA: Defense's

Advanced Research Project Agency) con el propósito de resolver los problemas de la heterogeneidad de las tecnologías de redes de cómputo. El protocolo que se dio dentro de TCP/IP comenzó con el usado para construir el primer switcheo de paquetes en el mundo, ARPANET. Éste es el que conduce el desarrollo del Worldwide Internet, hoy una de las redes heterogéneas más grandes del mundo.

El protocolo TCP/IP se emplea en Internet y algunas veces en redes más pequeñas, especialmente en las que conectan sistemas de computación que trabajan bajo el sistema operativo UNIX (Forouzan, 2002). Él explica que es posible que el protocolo que desarrollado por el Organismo Internacional de Estándares (ISO) para el modelo OSI eventualmente desplazó al protocolo TCP/IP en varios ambientes. Las siglas TCP/IP son por el nombre de 2 protocolos que realizan todas las funciones de inicio del protocolo TCP/IP TCP conocido como Protocolo de Control de la Transmisión, es un protocolo confiable orientado a la conexión que permite que una corriente de bytes originada en una máquina se entregue sin errores en cualquier otra máquina de la interred (Tanenbaum, 2003). Este protocolo fragmenta la corriente entrante de bytes en mensajes discretos y pasa cada uno a la capa de interred.

IP es un protocolo conocido como Protocolo de Interred. El trabajo de la capa de interred es paquetes IP a dónde se supone deben ir. El protocolo TCP/IP será extensamente usado por varias organizaciones dentro de los siguientes 100 años. TCP/IP es ahora una forma extremadamente importante de tecnología para redes.

Forouzan (2002) describe que TCP/IP pertenece a una familia de protocolos que se utilizan en Internet. Además, éste protocolo se desarrolló antes que el modelo OSI, por lo tanto, los niveles del TCP/IP están compuestoa por cinco niveles llamados capas, las cuales son (ver Figura 11): físico, enlace de datos, red, transporte y aplicación.


Figura 11. Modelo TCP/IP (adaptado de http://www.textoscientificos.com/redes/tcp-ip/comparacion-modelo-osi)

Adicionalmente, Forouzan (2002) explica que TCP/IP define dos protocolos en el nivel de transporte: Protocolo de Control de Transmisión (TCP) y Protocolo de Datagramas de Usuario (UDP). En el nivel de red, el principal protocolo definido por TCP/IP es el **Protocolo entre Redes (IP)**

Tanenbaum ( 2003) explica que el modelo TCP/IP originalmente no distinguía en forma clara entre servicio, interfaz y protocolo, aunque se ha tratado de reajustarlo después a fin de hacerlo más parecido a OSI. Además, como consecuencia, en el modelo OSI se ocultan mejor los protocolos que en el modelo TCP/IP y se pueden reemplazar con relativa facilidad al cambiar la tecnología, La capacidad de efectuar tales cambios es uno de los principales propósitos de tener protocolos por capas. Adicionalmente, explica que el modelo de referencia OSI se desarrolló antes de que se inventaran los protocolos. Este orden significa que el modelo no se orientó hacia un conjunto específico de protocolos, lo cual lo convirtió en algo muy general. Por último, lo contrario sucedió con TCP/IP: primero llegaron los protocolos, y el modelo fue en realidad sólo una descripción de los protocolos existentes. No hubo el problema de ajustar los protocolos al modelo. El problema fue que el modelo no se ajustaba a ningún otro protocolo; en consecuencia, no fue de mucha utilidad para describir otras redes que no fueran del tipo TCP/IP.

## 2.11.- Arquitectura de TCP/IP

Parte del poder del protocolo TCP/IP se determina por la habilidad para permitir que diferentes tipos de dispositivos y de proveedores ínter-operen con cualquier otro, soportando una gran variedad de dispositivos; pero siempre se pueden presentar problemas substanciales por compatibilidad (Comunicaciones, 1998). El hardware y software de estos dispositivos necesitan ser compatibles dentro del orden, para lo cual las arquitecturas de redes han sido desarrolladas en la construcción de redes complejas, usando una gran variedad de equipo.

En redes de computadoras modernas las funciones de transmisión de datos se realizan por un complejo hardware y software en varios dispositivos conectados a la red. Las funciones del software empleadas en los dispositivos en red son divididos dentro del nivel independiente de funciones. La comitiva del protocolo TCP/IP realiza una arquitectura por niveles, teniendo los 4 niveles de software ilustrados en la Figura 12.

Los 4 niveles de *software* TCP/IP son construidos sobre el entendimiento del *hardware* de la red que opera en el nivel inferior al *software* TCP/IP. El *software* de comunicación TCP/IP es dividido dentro de niveles

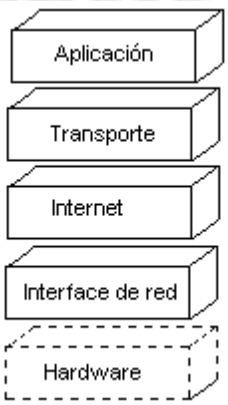


Figura 12. Niveles de arquitectura TCP/IP (adaptado de GS Comunicaciones, 1998)

TCP/IP hace posible desarrollar una aplicación en un ambiente dentro de *Internet* para facilitar la comunicación con una aplicación corriendo en otro ambiente como si ambos fueran conectados directamente. La comunicación parece simple hacía éstos. *Internet* puede ser un complejo integrado de muchas redes físicas y muchos ruteadores entre los dos ambientes realizando los programas de comunicación. Cada uno de los ambientes de comunicación maneja un *software* que implementa los 4 niveles de la arquitectura TCP/IP para tomar las funciones de comunicación.

El protocolo de comunicaciones es flexible y permite la transmisión de tramas sin errores entre diferentes sistemas. Debido a que es un protocolo de transferencia de información, puede enviar grandes volúmenes de información a través de redes no confiables, garantizando que ésta será recibida sin errores al momento de alcanzar su destino final.

Cuando se emplea TCP/IP, la información viaja en segmentos creados por TCP entre emisor y receptor para acceder a alguna aplicación. Los segmentos creados por TCP son encapsulados por IP, y esta encapsulación es llamada *datagramas* IP. El *datagrama* IP permite que los segmentos TCP que fueron hechos por alguna aplicación, sean transmitidos o ruteados en la Red de Área Local o en la Red de Área Extendida.

## 2.12.- La importancia de TCP/IP en la interconexión de redes

A medida que el desarrollo de plataformas de *hardware* (computadoras personales, estaciones de trabajo UNIX, computadoras centrales) y de sistemas operativos (DOS, UNIX, etc.) continúa creciendo en varias direcciones, la estructura TCP/IP proporciona las herramientas para vincular estos diferentes sistemas con servicios de transporte para las funciones de igual a igual, cliente-servidor de archivos, de transporte, y otras funciones de redes. TCP/IP aprovecha al máximo sus características y servicios para desarrollar redes corporativas (locales, globales y remotas) tanto en el entorno de propietarios como de proveedores múltiples.

## 2.13.- Ventajas de TCP

Uno de los servicios que provee TCP es el control de flujo, el cual regula la cantidad de datos enviada durante un período de transmisión dado. Este proceso de control de flujo se conoce como uso de ventanas. TCP usa las ventanas para determinar de forma dinámica el tamaño de la transmisión. Los dispositivos negocian el tamaño de la ventana a un número específico de bytes para transmitir antes del ACK. La Figura 13 explica la cantidad de datos enviados en un periodo de transmisión dado:

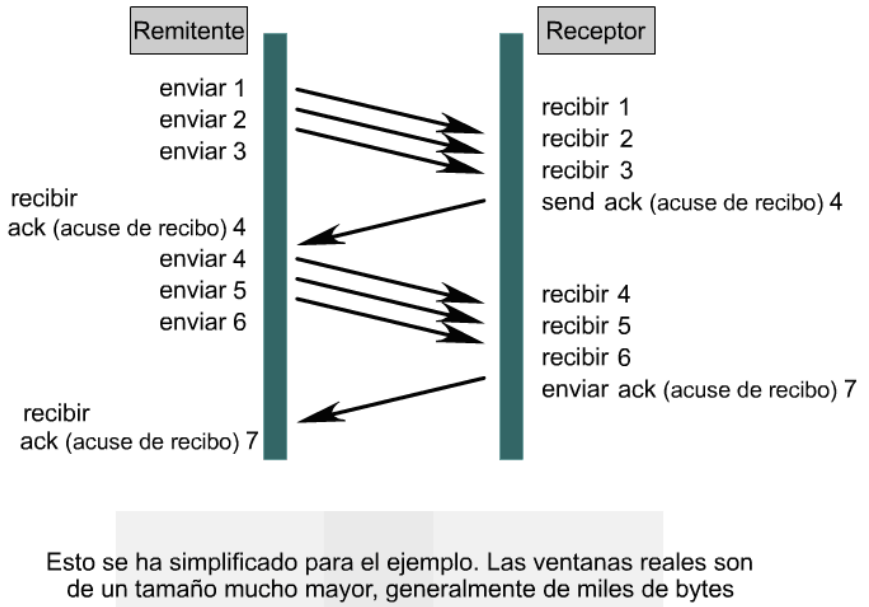


Figura 13. Ventanas TCP (adaptado de Cisco, 2007)

## 2.14.- Pila TCP/IP en Sistema Operativo Windows

La pila TCP/IP de Microsoft Windows 2000 ya presentó un conjunto de mejoras en el rendimiento como el ajuste de escala de la ventana de recepción de TCP, las confirmaciones selectivas y una estimación más precisa del tiempo de ida y vuelta (RTT, roundtrip time). Durante los años posteriores al lanzamiento de Windows 2000 se han producido numerosos cambios en el ancho de banda de red, por ejemplo, el uso más extendido del ancho de banda alto o de los vínculos inalámbricos. La pila Next Generation TCP/IP de Windows Vista y Windows Server "Longhorn" (actualmente como versiones beta en fase de prueba) contiene un nuevo conjunto de mejoras que incrementan el rendimiento en entornos de red de ancho de banda, latencia y pérdida altos (Microsoft, 2008a).

## 2.15.- Optimización automática de la ventana de recepción

El tamaño de la ventana de recepción de TCP corresponde a la cantidad de

bytes en un búfer de memoria de un host receptor que se utiliza para almacenar los datos entrantes en una conexión TCP. Una vez establecida la conexión, el tamaño de la ventana de recepción se anuncia en cada segmento de TCP. Anunciar el espacio restante en el búfer de memoria de recepción es un mecanismo de control de flujo del lado del receptor, el cual impide que el emisor envíe datos que el receptor no puede almacenar. Un host de envío sólo puede enviar cómo máximo la cantidad de datos anunciada por el receptor antes de esperar una confirmación y una actualización del tamaño de la ventana de recepción (Microsoft, 2008b). La optimización automática se puede configurar manualmente. Los valores del Registro como sigue:

*HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\TCPWindowSize*
*HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\Interface\TCPWindowSize*

Se pueden establecer en un máximo de 65.535 bytes (sin ajuste de escala de la ventana) o 1.073.741.823 (con ajuste de escala de la ventana). Sin un ajuste de escala de la ventana sólo se puede conseguir un rendimiento de aproximadamente 5 megabits por segundo (Mbps) en una ruta con un tiempo de ida y vuelta de 100 milésimas de segundo, independientemente del ancho de banda de la ruta (Microsoft, 2007).

## 2.16.- Ventanas de Recepción en la Pila Next Generation TCP/IP

Para solucionar el problema de determinar correctamente el valor del tamaño máximo de la ventana de recepción para una conexión según las condiciones actuales de la red, la pila Next Generation TCP/IP admite la optimización automática de la ventana de recepción. Esta establece continuamente su tamaño óptimo mediante la medición del producto del retraso de ancho de banda y la velocidad de recuperación de la aplicación, y lo ajusta conforme a las condiciones cambiantes de la red.

La optimización automática de la ventana de recepción permite el ajuste de escala de la ventana de TCP de forma predeterminada hasta un tamaño máximo de 16 megabytes. Mientras los datos fluyen en la conexión, la pila Next Generation TCP/IP supervisa la conexión, mide el producto del retraso de ancho de banda actual y la velocidad de recepción de la aplicación, y ajusta el tamaño de la ventana de recepción para optimizar el rendimiento. La pila Next Generation TCP/IP ya no utiliza los valores del Registro TCPWindowSize (Microsoft, 2008b).

Al conseguir un mejor rendimiento entre los interlocutores de TCP, el uso del ancho de banda de red se incrementa durante la transferencia de datos. Si todas las aplicaciones se optimizan para recibir datos TCP, el uso general de la red se puede incrementar notablemente, por lo que la utilización de Calidad de servicio (QoS) resulta más importante en las redes que funcionan casi al límite de su capacidad o a capacidad completa (Microsoft, 2008b).

## 2.17.- Antecedentes de TCP: Números de Secuencia

TCP divide los datos en segmentos. Los segmentos de datos viajan, entonces, desde el transmisor hacia el receptor después del proceso de

sincronización y la negociación del tamaño de ventana que dicta el número de bytes que es posible transmitir por vez. Los segmentos de datos que se transmiten deben re-ensamblarse una vez recibidos.

No hay garantía alguna de que los datos llegarán en el orden en que se transmitieron. TCP aplica los números de secuencia a los segmentos de datos que transmite de modo que el receptor pueda re-ensamblar adecuadamente los bytes en su orden original. La Figura 14 muestra lo anteriormente descrito.

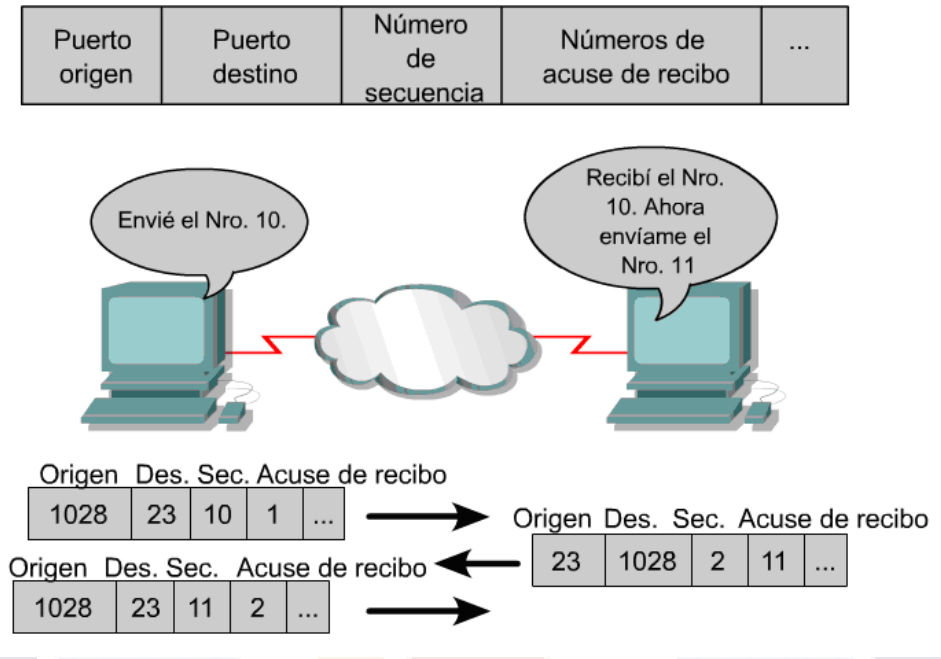


Figura 14. Números de secuencia TCP (adaptado de Cisco, 2007)

### 2.18.- Conversación entre hosts TCP

Los hosts que corren TCP/IP asocian los puertos de la capa de transporte con determinadas aplicaciones. Los números de puerto se usan para realizar el seguimiento de las distintas conversaciones que atraviesan la red al mismo tiempo. Los números de puerto son necesarios cuando un host se comunica con un servidor que provee múltiples servicios. Tanto TCP como UDP utilizan números de puerto o socket para enviar información a las capas superiores.

# 3. Protocolo Internet IPv6

Baker (1997) describe las características más importantes de IPv6 de la siguiente manera:

1. Soporta la entrega orientada a no conexión, es decir, permite que cada datagrama[1] sea ruteado independientemente.

[1] Datagrama: Paquete de longitud variable que consta de: cabecera (informa encaminamiento y entrega) y datos

2. Permite al emisor seleccionar el tamaño de un datagrama (ver Figura 15) y requiere que el emisor especifique el máximo número de saltos que un datagrama puede realizar antes de ser eliminado.

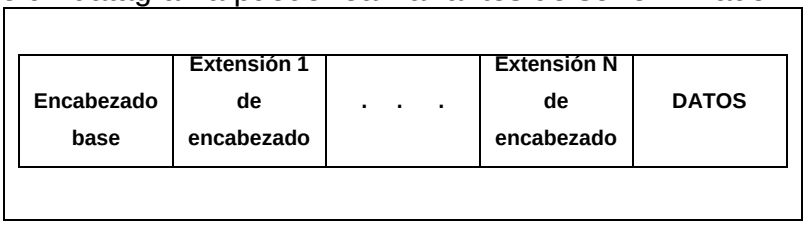


Figura 15. Estructura de un datagrama (adaptado de Comer, 1996)

3. Utiliza direcciones largas, pasó de treinta y dos bits (así era en IPv4), a 128 bits.
4. Datagrama con un encabezado base de tamaño fijo, seguido por ceros o más encabezados de extensión, seguidos a su vez por datos.
5. Cada datagrama IPv6 comienza con un encabezado base de cuarenta octetos, es decir un datagrama debe contener cuando menos cuarenta octetos además de los datos (ver Figura 16).

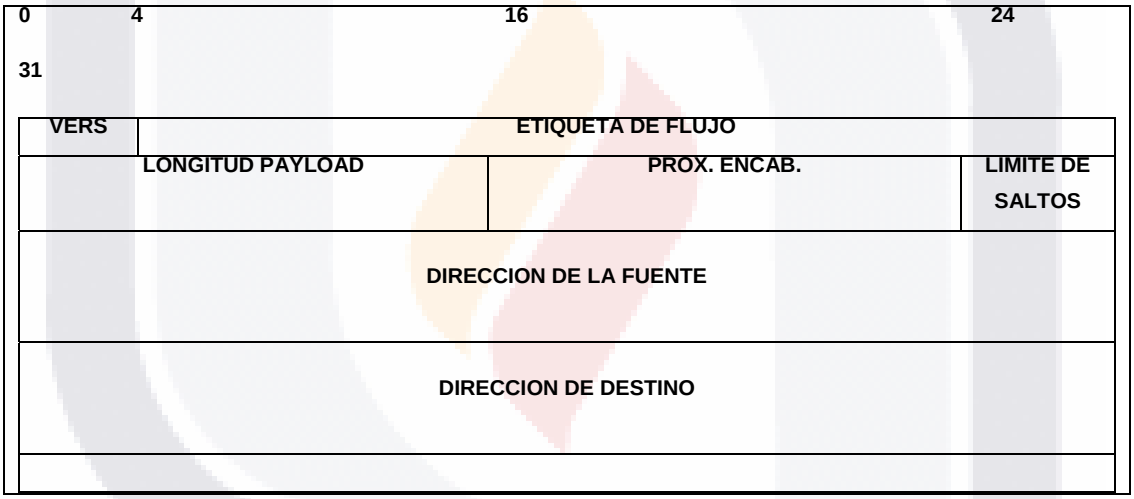


Figura 16. Formato de encabezado base de cuarenta octetos del IPv6. Cada datagrama IPv6 comienza con un encabezado base (adaptado de Comer, 1996).

Un protocolo de red de redes que utilice la fragmentación de extremo a extremo requiere que el emisor descubra el path MTU para cada destino y que fragmente cualquier datagrama que salga si es mayor que el path MTU. La fragmentación de extremo a extremo no se adapta al cambio de ruta (Comer, 2003).

# 4. MTU Nombre

Conocido también como Unidad de Transferencia Máxima en una red, el tamaño de la MTU puede ser muy pequeño; las limitaciones de los

datagramas para que se ajusten a la MTU más pequeña posible en una red de redes hace que la transferencia sea ineficiente cuando estos datagramas pasan a través de una red que puede transportar tramas de tamaño mayor (Comer, 2003).

En la Figura 15 podemos observar que el diagrama de IPv6 tiene un encabezado base de tamaño fijo, seguido por ceros o más encabezados de extensión, seguidos a su vez por datos. Además que sólo el encabezado base es indispensable, los encabezados de extensión son opcionales.

En la Figura 16 el campo inicial VERS de cuatro bits especifica la versión del protocolo. VERS siempre contiene el número seis en un datagrama IPv6. Como en el IPv4, los campos source address (dirección fuente) y destination address (dirección destino) especifican la dirección del emisor y del recipiente. En IPv6 sin embargo, cada dirección requiere diez y seis octetos.

# 5. Definición de SNMP

Estándar utilizado para administrar dispositivos IP (Douglas, 2001). El protocolo sencillo de gestión de red (SNMP, Simple Network Management Protocol) es un marco de trabajo para gestionar los dispositivos en una internet que utiliza el conjunto de protocolos TCP/IP. Ofrece un conjunto de operaciones fundamentales para monitorizar y mantener una Internet (Forouzan, 2002).

El Protocolo simple de administración de red (SNMP: Simple Network Management Protocol) es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. El SNMP permite que los administradores de red administren el rendimiento de la red, detecten y solucionen los problemas de red y planifiquen el crecimiento de la red. El SNMP usa UDP como su protocolo de capa de transporte.

Una red administrada con SNMP está compuesta por los tres componentes clave que se detallan a continuación:

- **Sistema de administración de la red (NMS: Network Management System):** El NMS ejecuta aplicaciones que monitorean y controlan los dispositivos administrados. La gran mayoría de los recursos de procesamiento y de memoria que se requieren para la administración de red se suministra a través del NMS. Deben existir uno o más NMS en cualquier red administrada.
- **Dispositivos administrados:** Los dispositivos administrados son nodos de red que contienen un agente SNMP y que residen en una red administrada. Los dispositivos administrados recopilan y guardan información de administración y ponen esta información a disposición de los NMS usando SNMP. Los dispositivos administrados, a veces denominados elementos de red, pueden ser routers, servidores de acceso, switches y puentes, hubs, hosts del computador o

impresoras.

- **Agentes:** Los agentes son módulos del software de administración de red que residen en los dispositivos administrados. Un agente tiene conocimiento local de la información de administración y convierte esa información a un formato compatible con SNMP.

## 5.1 Conceptos SNMP

SNMP emplea dos conceptos que son gestor y agente. El gestor es la estación que controla y monitorea un conjunto de agentes. SNMP trabaja al nivel de aplicación, permitiendo de esta manera que las estaciones gestoras monitoricen y controlen agentes. Una estación de gestión, denominado gestor, es una estación que ejecuta un cliente de SNMP; una estación gestionada, denominada agente, es un encaminador (o una estación) que ejecuta el servidor de SNMP (Forouzan, 2002).

El agente almacena información sobre presentaciones (tráfico, paquetes recibidos y/o enviados en la red) en una base de datos. El gestor tiene acceso a los valores de ésta base de datos. La siguiente tabla presenta las Operaciones SNMP.

| Operación SNMP | Dirección | Descripción |
|---|---|---|
| GetRequest | Gestor SNMP(cliente) Agente SNMP(servidor) | El mensaje GetRequest se envía desde el gestor(cliente) al agente (servidor) para recuperar el valor de una variable (Forouzan, 2002). |
| GetNextRequest | Gestor SNMP(cliente) Agente SNMP(servidor) | Este mensaje se envía desde el gestor al agente para recuperar el valor de una variable. El valor recuperado es el valor del objeto que sigue al objeto definido en el mensaje. Se utiliza fundamentalmente para recuperar valores de las entradas de una tabla. Si el gestor no conoce los índices de las entradas, no puede recuperar los valores. Sin embargo, puede utilizar GetNextRequest y definir el objeto (Forouzan, 2002). |
| GetResponse | Agente SNMP(servidor) Gestor SNMP (cliente) | Este mensaje es enviado desde un agente al gestor en respuesta a GetRequest y GetNextRequest. Contiene el valor de la(s) variable(s) solicitada por el gestor (Forouzan, 2002). |
| SetRequest | Gestor SNMP(cliente) – Agente SNMP(servidor) | Este mensaje es enviado desde el gestor al agente para fijar (almacenar) un valor en una variable (Forouzan, 2002). |
| Trap | Agente SNMP(servidor) – Gestor SNMP (cliente) | Este mensaje es enviado desde el agente al gestor para informar de un evento; por ejemplo, si el agente es reiniciado, informa al gestor e indica la hora de reinicio (Forouzan, 2002). |

Tabla 2. Operaciones SNMP

La forma en que se medirá el desempeño de la red será a través del Análisis de protocolo en una LAN switcheada, la cual se configura con los siguientes elementos:
1.- Espejeo de puerto.- guarda la información antes de medir desempeño
2.- Espejeo de switch.- guarda la información antes de medir desempeño
3.- Prueba interna con RMON.- analiza los resultados observados en la medición de desempeño, almacenada previamente en espejeo de switch y puerto.

## 5.2 Operación de SNMP

El SNMP es un protocolo de la capa de aplicación diseñado para facilitar el intercambio de la información de administración entre los dispositivos de red. Al usar el SNMP para tener acceso a los datos de información de administración, tales como los paquetes enviados a una interfaz cada segundo o el número de conexiones TCP abiertas, los encargados de las redes pueden administrar más fácilmente el rendimiento a fin de encontrar y resolver los problemas en las redes.

El SNMP es en la actualidad el protocolo más popular para la administración de redes corporativas, universitarias y de investigación.

La actividad de estandarización continúa aun cuando los proveedores desarrollan y lanzan al mercado aplicaciones administrativas novedosas basadas en el SNMP. El SNMP es un protocolo sencillo, aunque su conjunto de funciones es lo suficientemente poderoso como para manejar los difíciles problemas relacionados con la administración de redes heterogéneas. El SNMP permite que los administradores de red administren el rendimiento de la red, detecten y solucionen los problemas de red y planifiquen el crecimiento de la red. El SNMP usa UDP como su protocolo de capa de transporte.

Una red administrada con SNMP está constituida por los siguientes tres componentes:

- Sistema de administración de la red (NMS: Network Management System): El NMS ejecuta aplicaciones que monitorean y controlan los dispositivos administrados. La gran mayoría de los recursos de procesamiento y de memoria que se requieren para la administración de red se suministra a través del NMS. Deben existir uno o más NMS en cualquier red administrada.

- **Dispositivos administrados:** Los dispositivos administrados son nodos de red que contienen un agente SNMP y que residen en una red administrada. Los dispositivos administrados recopilan y guardan información de administración y ponen esta información a disposición de los NMS usando SNMP. Los dispositivos administrados, a veces denominados elementos de red, pueden ser routers, servidores de acceso, switches y puentes, hubs, hosts del computador o impresoras.

- **Agentes:** Los agentes son módulos del software de administración de red que residen en los dispositivos administrados. Un agente tiene conocimiento local de la información de administración y convierte esa información a un formato compatible con SNMP.

# 6. Términos Encontrados en Investigaciones Previas

## 6.1. Remote Monitoring (RMON)

Surgió de la necesidad de crear un protocolo que analizara y monitoreara múltiples LANs, analiza los resultados en una estación de trabajo centralizada.

Aparte del protocolo SNMP, se emplean los siguientes protocolos que trabajan de manera cojunta a SNMP, como lo son SMI (Structure of Management Information) y MIB(Management Information Base).

## 6.2. SMI

SMI es un componente utilizado en la gestión de red; sus funciones son nombrar objetos; definir el tipo de datos que se pueden almacenar en un objeto y mostrar cómo codificar los datos a transmitir por la red (Comer, 2003).

## 6.3. MIB

La base de Información de gestión (MIB) es el segundo componente utilizado en la gestión de red; cada agente tiene su propio MIB, que es una colección de todos los elementos que pueda manejar el gestor; los objetos en el MIB se clasifican en ocho grupos: sistema, interfaz, traducción de direcciones, ip, icmp, tcp, udp y egp; cada grupo tiene variables definidas y/o tablas Comer (2003).

Al trabajar sobre protocolo TCP en IPv6 sobre WinNT y Solaris (Zeadally, 2000), se encontraron las siguientes oportunidades de investigación: a) pruebas realizadas sobre configuración punto a punto; b) investigación de tamaño de direccionamiento de sockets asumen que el desempeño se debe a el S.O., validar que no sea debido a que IPv6 tiene direccionamiento de socket constante; c) Así mismo se observará como trabaja o se manipula la asignación de bloques. d) se realizará el monitoreo de los procesos en ejecución, empleando la medición por switches, para determinar el desempeño observado en los procesos. En base a dichas características, las condiciones actuales bajo las que se realiza el estudio, serán las siguientes: Se genera una hipótesis de lo que se observará, y se realiza un estudio de análisis de datos para aceptar o refutar lo analizado, en base a esto el modelo será cuantitativo.

## 6.4. Socket

Los sockets se emplean para enviar datos. Inicialmente un socket se crea en un estado desconectado, lo que significa que el socket no está asociado con ningún destino externo; la llamada del sistema en estado conectado enlaza un destino permanente a un socket; un programa de aplicación debe llamar al estado conectado para establecer una conexión antes de que pueda transferir datos a través de un socket de flujo confiable; los sockets utilizados con servicios de datagrama sin conexión necesitan no estar conectados antes de usarse, pero haciéndolo así, es posible transferir datos sin especificar el destino en cada ocasión (Comer, 2003).

## 6.5. Net Flood

El objetivo de este ataque es degradar la capacidad de conexión a la red de un sistema, saturando sus enlaces de comunicaciones. Por ejemplo, si el enlace de una organización dispone de un ancho de banda de 34 Mb. y un atacante dispone de un enlace de 155 Mb., prácticamente la totalidad del tráfico cursado por la organización pertenecerá al atacante, por lo que no podrá enviarse tráfico útil (Siles Peláez, 2002).

Para disponer de altos anchos de banda puede recurrirse a la obtención de múltiples sistemas desde los que efectuar el ataque (ver las vulnerabilidades DDoS) o apoderarse de sistemas mal administrados y protegidos que posean redes de gran capacidad, como por ejemplo, los existentes en las universidades (Siles Peláez, 2002).

Las dos técnicas aplicadas en este tipo de ataques se basan en los protocolos ICMP y UDP, al tratarse de protocolos no orientados a conexión y que permiten el envío de paquetes sin requisitos previos: *ICMP Flood y UDP Flood* (Siles Peláez, 2002).

## 6.6. TCP Syn Flood

Dentro de los ataques DoS, existe uno asociado directamente al protocolo TCP. Consiste en el envío masivo de paquetes de establecimiento de conexión (SYN) contra un sistema. La recepción de estas solicitudes provoca que el sistema destino, objetivo del ataque, reserve cierta cantidad de memoria (*buffers*) para almacenar las estructuras de datos asociadas a cada una de las nuevas conexiones en curso (Siles Peláez, 2002).

Asimismo, ciertos sistemas imponen un número máximo de conexiones en este estado, por lo que una vez alcanzado éste, no será posible establecer más conexiones. Tras un periodo de tiempo controlado por un temporizador (que suele ser de 2 minutos), las conexiones que continúan en este estado expiran, permitiendo la creación de nuevas conexiones. Esto solo será posible si el ataque *TCP SynFlood* ha cesado, ya que mientras se mantenga, serán sus nuevos inicios de conexión los que ocuparán el espacio de memoria liberado por las sesiones expiradas.

Suponiéndose un número máximo de conexiones igual a 30, y el temporizador igual a 2 minutos, se podría desarrollar un ataque de este tipo enviando un paquete SYN cada 4 segundos: tiempo necesario por cada conexión para poder enviar el máximo de 30 conexiones en los 120 segundos de expiración (Siles Peláez, 2002).

## 6.7. Tasa de transferencia (Throughput )

Se pude explicar como el promedio de ancho de banda entre dos aplicaciones conectadas para medir el tiempo en el que su ancho de banda envía información entre las aplicaciones en un largo periodo de tiempo (Zeadally, 2000).

### 6.8. Tiempo de ida y vuelta (Roundtrip time)

El tiempo de ida y vuelta se define como la prueba de conexión hacia un determinado host en máquina cliente, hacia la máquina servidor para medir el tiempo de ida de un mensaje del cliente hacia el servidor más el tiempo que tarda en regresar hacia la máquina que hizo la petición (Zeadally, 2000).

### 6.9. Ancho de Banda (Bandwidth )

El ancho de banda se define como el máximo número de paquetes transmitidos durante una unidad de tiempo (Sándor & Hassan, 2004).

### 6.10. Latencia (Latency)

La latencia se define como el tiempo total que un mensaje tarda en viajar del origen al destino (Sándor & Hassan, 2004).

# 7. Comparación de los modelos de referencia OSI y TCP

De acuerdo con Tanenbaum (2003), los modelos de referencia OSI y TCP/IP tienen mucho en común. Ambos se basan en el concepto de un gran número de protocolos independientes. También la funcionalidad de las capas es muy similar. Por ejemplo en ambos modelos las capas por encima de la de transporte, incluida ésta, están ahí para prestar un servicio de transporte de extremo a extremo, independiente de la red, a los procesos que deseen comunicarse. Estas capas forman el proveedor de transporte. También en ambos modelos, las capas encima de la de transporte son usuarios del servicio de transporte orientado a aplicaciones. La siguiente figura muestra el modelo TCP/IP comparándolo con el modelo de referencia de OSI.

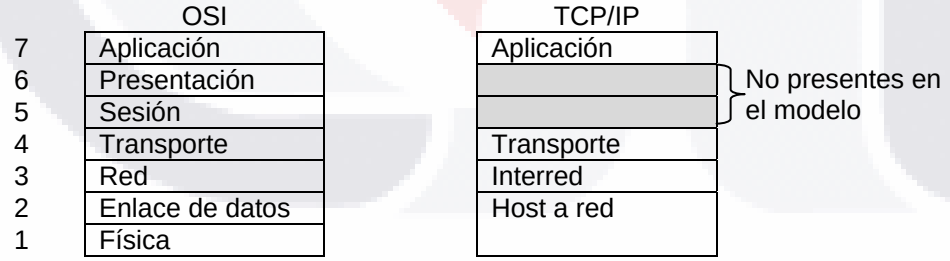


Figura 17. Modelo de referencia TCP/IP (adaptado de Tanenbaum, 2003)

# 8. El sistema Operativo como Administrador de Recursos

¿Qué es un Sistema Operativo? Según Tanenbaum (2003), es una capa de software cuya labor es administrar a un sistema de cómputo moderno que consta de uno o más procesadores, memoria principal, discos, y otros dispositivos de entrada/salida y por lo tanto el SO proporcionará a los programas de usuario una interface más sencilla para comunicarse con el hardware

# 9. Sistema Operativo

El sistema Operativo de acuerdo a lo definido por Tanenbaum (2003) tiene como misión administrar todos los elementos de un sistema complejo.

La administración de recursos incluye el multiplexaje (compartimiento) de recursos en dos formas: en el tiempo y en el espacio. Cuando un recurso se multiplexa en el tiempo, diferentes programas o usuarios se turnan para usarlo. El otro tipo de multiplexaje es en el espacio. En lugar de que los clientes se turnen, cada uno recibe una parte del recurso. Por ejemplo la memoria principal por lo normal se divide entre varios programas en ejecución, de modo que todos puedan estar residentes al mismo tiempo (por ejemplo para poder turnarse la CPU).

IP es el mecanismo de transmisión utilizado por los protocolos TCP/IP; es un protocolo basado en datagramas sin conexión y no fiable (Forouzan, 2002). De acuerdo a una evaluación realizada (Zeadally, 2000) sobre en IPv6, tanto en plataforma Windows como en Solaris se mostró una serie de resultados en base a una prueba de desempeño de equipos con características y configuraciones particulares. Principalmente, se evaluaron los protocolos IPv6 e IPv4 donde se midieron los tiempos de respuesta sobre los sistemas Windows 2000 y Solaris en el envío de mensajes menores a 256 bytes y mayores a 512 bytes, mostrando finalmente el desempeño de cada una de las pruebas con las características anteriores.

## 9.1. Red ATM (Modo de Transferencia Asíncrono)

Según cita Martínez (2008), los dispositivos de acceso integrados (IAD) ATM Link Access (LA) permiten efectuar la convergencia de servicios múltiples, tales como voz, LAN y datos sobre una línea de acceso DSL, utilizando la red ATM y DSLAM existentes. Los IAD basados en ATM garantizan la calidad del servicio (QoS) y la gestión de punta a punta hasta el establecimiento del cliente.

Zeadally (2000) explica que las redes ATM permiten el soporte en tiempo real de aplicaciones multimedia, (video conferencia, teletrabajo) y otras aplicaciones. El formato estándar en redes ATM es que integra redes de área local (LAN), así como redes de área amplia (WAN). Además, al medir el desempeño en una red ATM, la prueba en que se evaluó el tiempo de respuesta de un applet en la red ATM se evaluó en relación a un jitter en donde se comparó el rendimiento de las APIs basadas en Java respecto a la API de Winsock 2 en diferentes protocolos TCP/IP, UDP/IP y el modo de transferencia asíncrona (ATM), mostrando un mejor rendimiento en plataformas diferentes a Unix, debido a que en dicha plataforma se tiene que habilitar la opción de jitter, mientras que el SO Windows NT traen el modo de transferencia asíncrona (ATM) configurado de forma nativa.

Jitter la define la literatura (Aguilera, 2009) como la variación en el retardo, en términos simples la diferencia entre el tiempo en que llega un paquete y el tiempo que se cree que llegara el paquete.

## 9.2. Net Flood

Según Softonic, (2009) Net Flood es un concepto genérico que engloba diversos tipos de ataques de denegación de servicios, uno de ellos es el "smurf" o pitufo, el cual consiste en transmitir paquetes a través de la red de forma que el paquete se replica tantas veces como existan equipos que puedan transmitir el paquete. Un segundo tipo de ataque es llamado "IP spoofing" es similar al "smurf" la diferencia consiste en que el atacante utiliza la dirección IP de otro sistema, es decir el atacante utiliza una red intermedia para multiplicar sus recursos.

Otra definición citada por Siles Peláez (2002) define que el objetivo de éste ataque es degradar la capacidad de conexión a la red de un sistema, saturando sus enlaces de comunicaciones.

Siles Peláez (2002) dice que dentro del Net Flood, existe una técnica llamada "smurf", en donde se envía un paquete ICMP con la dirección IP de la máquina a atacar así como la dirección IP destino de la dirección broadcast.

Un estudio similar al anterior fue efectuado por Siles Peláez (2002) en donde fue posible controlar el número de paquetes por unidad de tiempo ICMP en los ruteadores.

En ésta segunda condición de prueba se configuró el ruteador para controlar el número de paquetes por unidad de tiempo y así evitar una medición errónea.

El objetivo del ataque fue degradar la capacidad de conexión a la red de un sistema, saturando sus enlaces de comunicaciones, es decir que el enlace del atacante sea mayor al del atacado provocando con esto que el tráfico del atacado pertenecerá completamente al del atacante, por lo que no se podrá enviar tráfico útil, donde las técnicas aplicadas en el ataque se basaron en los protocolos ICMP y UDP, que son protocolos no orientados a conexión y permiten el envío de paquetes sin requisitos previos.

## 9.3. Connection Flood

Esta prueba consiste en controlar el tiempo que un socket pueda permanecer en estado time_wait, evitando así un consumo de recursos excesivo (Siles Peláez, 2002).

De acuerdo con Siles Peláez (2002), al enviar un paquete orientado a conexión empleando el protocolo TCP, existe un límite máximo de conexiones soportadas, cualquier conexión posterior a la última será rechazada, éstas últimas después de cierto periodo de tiempo y no ser atendidas expiran y dejan de consumir recursos, pero al estar siendo atacadas constantemente cualquier conexión provocará que se mantengan abiertas todas las sesiones y harán que colapse el servidor, dicho ataque es posible prevenirlo mediante el uso de un cliente que establezca conexiones

contra un sistema, lo cual es posible debido a que connection flood se basa en servicios TCP y se conoce la identidad del atacante (dirección IP).

# III. Tecnologías Relacionadas al Estudio

## 1. Introducción al tema

**CentOS.** Sandoval (2005) menciona que CentOS (**C**ommunity **ENT**erprise **O**perating **S**ystem) es una distribución de Linux de clase empresarial derivada de los archivos fuentes provistos libremente al publico por Red Hat. CentOS cumple completamente la política de redistribución y apunta a ser 100% compatible a nivel binario (programas) con Red Hat Enterprise Linux. En CentOS los principales cambios con respecto a Red Hat Linux Enterprise (RHEL), es la eliminación de las ilustraciones y marcas de Red Hat en los paquetes. CentOS es gratuito, y está orientado a los usuarios que necesiten un sistema operativo de nivel empresarial, pero sin pagar los costos de certificación y soporte de Red Hat.

CentOS es desarrollado por un creciente grupo de programadores. Estos reciben ayuda de una activa comunidad de usuarios, los cuales incluye administradores de sistemas, administradores de red, empresas, administradores, contribuidores del núcleo Linux y entusiastas de Linux de todo el mundo (Sandoval 2005).

CentOS es una versión muy probada, lo cual significa que es estable, y además es sucesor de Red Hat, lo cual lo hace un SO más fiable gracias a su fiabilidad y soporte, además de ser código abierto (Sandoval 2005).

En la actualidad éste sistema operativo es muy empleado por grandes empresas, por lo cual considero conveniente emplearlo. Según se describe en (Wikipedia, 2009a) **CentOS** (**C**ommunity **ENT**erprise **O**perating **S**ystem) es un clon a nivel binario de la distribución Linux Red Hat Enterprise Linux **RHEL**, compilado por voluntarios a partir del código fuente liberado por Red Hat. No sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto. Algunas de las distribuciones basadas en RedHat Linux más importantes son: Mandriva Linux, Yellow Dog Linux (sólo para PowerPC), y CentOS (compilada a partir de las fuentes de Red Hat).

**FreeBSD** es el sistema operativo que conserva la esencia de la distribución original de UNIX de Laboratorios Bell así como el de la Universidad de Berkley (FreeBSD, 2008). BSD también tiene características avanzadas de conexión en red, rendimiento, seguridad y compatibilidad siendo las características anteriores una de las razones de su elección para el presente estudio debido a que dichas características es difícil encontrarlas en otros SOs por sí solos, incluso algunos de los más comerciales.

Los desarrolladores de FreeBSD se han enfrentado a algunos de los problemas más difíciles en el diseño de sistemas operativos para poder ofrecerte estas avanzadas características. Los problemas más comunes son

los siguientes:

- **Bounce buffering** trata sobre la limitación en la arquitectura ISA de los PC's que limita el acceso directo a memoria en los primeros 16 megabytes.
  *Resultado:* sistemas con más de 16 megabytes operan más eficientemente con periféricos DMA en el bus ISA.

- **Un buffer de caché conjunto de memoria virtual y sistema de ficheros** continuamente ajusta la cantidad de memoria usada por los programas y el cache de disco.
  *Resultado:* los programas reciben una excelente gestión de memoria y un alto rendimiento en los accesos a disco, liberando al administrador del sistema del trabajo de ajustar los tamaños de los cachés.

- **Módulos de compatibilidad** que permiten la ejecución de programas de otros sistemas operativos en FreeBSD, incluyendo programas para Linux, SCO, NetBSD y BSDI.
  *Resultado:* los usuarios no tendrán que recompilar programas ya compilados para algunos de los sistemas compatibles, teniendo acceso a programas como las extensiones para BSDI de Microsoft FrontPage Server o WordPerfect para SCO y Linux.

- **Módulos de kernel de carga dinámica** que permiten tener acceso a nuevos sistemas de ficheros, protocolos de red o emuladores de binarios en tiempo de ejecución sin necesidad de generar un nuevo kernel.
  *Resultado:* Se puede ganar mucho tiempo y desarrolladores de terceras partes pueden ofrecer subsistemas completos como módulos de kernel sin necesidad de distribuir el código fuente o complejos procedimientos de instalación.

- **Librerías compartidas** reducen el tamaño de los programas, ahorrando espacio de disco y memoria. FreeBSD usa un avanzado esquema de librerías compartidas que ofrecen muchas de las ventajas de ELF, ofreciendo la versión actual compatibilidad ELF con programas de Linux y nativos de FreeBSD.
  *Resultado:* Naturalmente, cómo FreeBSD es un esfuerzo en constante evolución, puedes esperar nuevas características y niveles más altos de estabilidad con cada release.

- Los argumentos pro-BSD de la seguridad son relativos. Como BSD tiene una única distribución, se hacen muchos esfuerzos en verificar el código de todo, mientras que Linux no se preocupa más allá del kernel, siendo responsabilidad de las distribuciones incluir versiones estables y verificadas. En particular OpenBSD se esfuerza en asegurar la máxima seguridad de su distribución, y para ello usualmente tiene versiones viejas de muchos paquetes comparados con lo que uno puede encontrar en Linux y otros BSD. La distribución FreeBSD es de las más estables de dicha familia y estable.

Ahora bien, comparando las distribuciones FreeBSD y CentOS se puede observar que, se tiene soporte para diversas arquitecturas, tal y como se observa en la tabla comparativa siguiente:

| Distribución Linux | FreeBSD | CentOS |
|---|---|---|
| **Tipo** | Instalable | Instalable |
| **Escritorio por defecto** | KDE | GNOME |
| **Arquitecturas con soporte** | alpha, amd64, i386, ia64, pc98, powerpc, sparc64 | i386, ia64, ppc, s390, s390x, x86_64 |
| **Sistemas de archivos journaled** | n/a | Ext3 |
| **Eficiencia en el arranque** | s/i | Cerca de 30 segundos, esto depende de la configuración de hardware. Los scripts de arranque estan escritos en estilo BSD. |
| **Eficiencia del sistema** | s/i | Altamente responsivo después de ser instalado. |
| **Recomendable para** | | Empresas |
| **Licencia** | BSD | GNUP, GPL |

Tabla 3. Comparación entre CentOS y BSD (Adaptada de A. Sandoval, 2008)

# 2. Comparativa de rendimiento

## 2.1.- Comparativa de rendimiento (benchmark) general

De acuerdo con (Corbett, 1998), el término benchmark en Japón se le conoce como “dantotsu” y se define como lucha para ser el mejor. De acuerdo con Sole y Bist (1995) el nivel de benchmark establece el grado de desafío para una ligera mejora en el proceso de desarrollo, a un cambio radical en el proceso. Benchmark puede dividirse en diferentes tipos dependiendo en cómo y con qué objetivo se hizo la comparación. Además, mencionan algunos tipos comunes de evaluación, los cuales incluyen:

- Benchmarking interno: comparación dentro de las mismas organizaciones,
- Benchmarking externa: comparación con organizaciones externas
- Benchmarking en la industria: comparación con los competidores,
- Benchmarking genérico: identificación de las mejores prácticas,
- Proceso de benchmarking: comparación de los procesos de trabajo discretos y sistemas
- La evaluación comparativa de rendimiento: comparación de atributos de rendimiento, por ejemplo, precio, tiempo de salida al mercado, y
- Benchmarking estratégico: abordar cuestiones estratégicas.

## 2.2.- ¿Qué no es Benchmark?

De acuerdo a Jones (2004), los siguientes puntos definen lo que no es un benchmark:

- Competidor de análisis - benchmarking es mejor cuando involucra colaboración.
- Comparación de las tablas de la liga - el objetivo es aprender acerca de las circunstancias y procesos que subyacen a un rendimiento

superior.

- Una solución rápida, hecha una vez para siempre – los proyectos de benchmarking pueden prolongarse durante varios meses o incluso años, y es vital que se repita periódicamente a fin de no quedarse atrás.
- Recuperación - rapidez en la evolución de las circunstancias, las buenas prácticas de convertirse rápidamente y ganar la competencia, o, tomar ventaja
- Copia - el hecho de que otros están haciendo las cosas de manera diferente no significa necesariamente que estén mejor
- Espionaje - la apertura y la honestidad son vitales para el éxito de la evaluación comparativa.

La comparativa de rendimiento (benchmarking) ha permitido identificar en las cuestiones laborales de las empresas un rendimiento superior en su producción adaptándolo a su negocio, lo cual ha facilitado el éxito de éstas. En consecuencia, benchmarking es un proceso operativo de aprendizaje continuo y adaptación que le permite mejorar una organización de la posición competitiva.

## 2.3.- ¿Por qué Benchmark?

Aunque muchas organizaciones inician proyectos en los cuales se requiere tomar una decisión respecto a que sistema operativo utilizar por ejemplo, es por lo que éstas deciden utilizar la comparativa de rendimiento.

El benchmarking es un indicador del desempeño comparativo. Dicho proceso de da resultados en dos tipos: medidas de desempeño superior, y facilitadores. Los facilitadores del proceso, se desarrollan para satisfacer una necesidad específica de negocio en el contexto de un entorno empresarial y la cultura de la empresa.

El problema más general en la comparativa de rendimiento, es que no se puede acceder al nivel óptimo en una comparativa de rendimiento, a menos que se tenga información privilegiada.

Según Wohlin et al. (2002), el proceso de benchmarking se efectúa cuando se realiza una comparación de procesos similares en diferentes contextos, e implica múltiples puntos de referencia. Por ejemplo, dos puntos de datos no es una evaluación comparativa, y requiere una muestra representativa en términos de, por ejemplo, las organizaciones y aplicaciones.

## 1.4.-Pasos para la Comparativa de Rendimiento

Según Jones ( 2004), existen cinco etapas clave en el benchmarking, los cuales se describen a continuación:

1. Proto-planificación
   a. Decida lo que desea comparar
   b. Decidir contra lo que usted necesita comparar

   c. Identificar los productos necesarios
   d. Determinar las metodologías de recopilación de datos
2. La recopilación de datos
   a. Secundaria / investigación de antecedentes
   b. Investigación primaria - desde el punto comparativo de referencia
3. Análisis
   a. De las lagunas
   b. De los factores que crean las diferencias ó lagunas (facilitadores)
4. Implementación
   a. La planificación de la ejecución
   b. Roll-out de los nuevos modus operandi (cambios)
5. Monitoreo
   a. Recolección de datos
   b. La evaluación de los progresos
   c. Cambio iterativo

Estos pasos fueron utilizados para conducir el presente estudio. Mismo que se detalla en los siguientes capítulos.

# IV. Identificando el problema de investigación

El presente estudio se basó en tres investigaciones previas de comparativas de rendimiento de sistemas operativos bajo diversas condiciones. El primero de ellos fue un estudio realizado por (Zeadally, 2000), permitió identificar las pruebas a efectuar a las tecnologías seleccionadas. Dichas pruebas son bajo las siguientes condiciones:

1.- Se midió el throughput. En este aspecto los resultados revelaron diferencias considerables a través de la medición del throughput en Protocolo de Datagramas de Usuario (UDP) mostrando considerables diferencias en el envío de mensajes menores a 256 bytes sobre TCP en IPv6 e IPv4 mostrando mejor desempeño IPv6.

2.- Al medir el tiempo de respuesta que tarda un paquete en ser enviado y recibido, observaron retrasos considerables en aplicaciones de audio y video, sobre Windows 2000 en IPv4 e IPv6. En donde Windows 2000 con IPv6 tuvo el mejor desempeño, esto debido a que aplicaciones que implican audio y video presentan retraso en el envío y recepción de la información, el desempeño en el estudio mostrado por Zedally para éste punto mostró que IPv6 tuvo un mejor desempeño respecto a IPv4 empleando Windows 2000 y Solaris con TCP.

3.- Se observó la usabilidad de los procesos al ser enviados al destino, la disponibilidad fue medida con la herramienta de monitoreo de desempeño de Windows. Donde TCP con IPv6 usó en promedio veinte por ciento más recursos de CPU que IPv4. Esto implicó el uso del administrador de tareas de Windows 2000 en su herramienta integrada que monitorea el desempeño del CPU de los recursos de la computadora empleada.

4.- También se registró el tiempo que tarda en interactuar el cliente con el servidor y viceversa para visualizar el desempeño de los ruteadores. Para esto se empleó una configuración punto a punto en sus pruebas de desempeño. Utilizaron dos ruteadores, un Ericsson AXI 462 y un IBM 2216, en donde el desempeño de éste último ruteador, mostró un pobre desempeño, particularmente en IPv6.

Estos resultados presentan oportunidades para investigación adicional a dicho tema. Entre dichas oportunidades se tienen las siguientes:

Una segunda investigación (Sulaiman, Asaad, & Chieng, 2005), se llevó a efecto bajo las mismas condiciones ó características del equipo utilizado para cada uno de los sistemas operativos, junto con sus respectivas implementaciones de IPv6. Dicho estudio realizó tres pruebas, las cuales se describen a continuación:

1. El primer escenario consistió en configurar una interfaz de loopback, se realizó con el fin de evitar que factores externos tales como el ancho de banda, el cable y el ruido de colisión, los nodos intermedios, etc., pudieran afectar al desempeño de una pila de protocolos de red

en tiempo real.

2. El segundo escenario consistió en representar la comunicación entre dos sistemas en la misma red (LAN) en tiempo real.

3. El tercer escenario fué similar a la comunicación entre dos sistemas diferentes en la red, entre un sistema en el interior de la LAN y un sistema en la Internet, por mencionar un ejemplo.

Una tercera investigación (Sulaiman, Asaad, & David, 2005) sobre Windows 2003, RedHat 9.0 y FreeBSD4.9 comparó el desempeño de equipos con características y configuraciones similares, donde se evaluó utilizando el protocolo IPv6 mediante la transferencia de datos empleando loopback. En dicho estudio, el resultado observado para el escenario de loop back fue el siguiente: en Windows 2003 se observó un mayor aumento en el envío de paquetes de ida y vuelta o Round Trip Time (RTT) respecto a FreeBSD 4.9 y en último sitio se ubicó Red Hat 9 en donde el tamaño de paquetes enviados fue de 150 bytes en adelante. Se pudo observar un menor aumento en el RTT de 30000 ns en FreeBSD 4.9 y Windows 2003, mientras que Red Hat se mantuvo constante. Sin embargo, después de 1440 paquetes de octetos, Windows 2003 tiene un aumento muy significativo en comparación con el resto de los SO. En consecuencia, este estudio (Mohamed et al., 2006), se concluyó que al realizar la prueba de loopback, se dijo que el pobre desempeño mostrado por Windows 2003 en comparación a los otros SO, se debió a que en IPv6 existió una fragmentación de paquetes en las pruebas de 1440 bytes, mientras que para Red Hat 9, dicha fragmetación se presenta en los 16384 bytes y para FreeBSD 4.9 se dá en los 14436 bytes enviados.

De acuerdo con las pruebas realizadas en los artículos citados anteriormente, se llevó a cabo el presente estudio, el cual podría ser de interés para los administradores de red de las organizaciones, ya que les permitirá comparar y analizar que el tráfico que fluye a través de la red bajo diferentes escenarios y diferentes sistemas operativos todos ellos bajo las mismas condiciones de red, ver el tráfico que fluye en la red, ya sea entrante y/o saliente, y éste al ser evaluado en unidades de tiempos de conexión, les será de ayuda para detectar problemas e identificar el tráfico no deseado es decir intrusiones en la red, y/o también en base a las evaluaciones realizadas en la presente comparativa, decidirse por el uso de un sistema operativo para alguna condición similar a la evaluada.

Las condiciones bajo las cuales se llevó a efecto el estudio se basada en estudios en los que para que una aplicación envíe información a través de la red, lo primero que hace es encapsular esa información en un protocolo de transporte (TCP,UDP), a partir de ese momento la información pasa a llamarse payload o carga útil, por lo cual, el protocolo a utilizar será TCP/IPv6, en base a que no existe un estudio en donde se haya evaluado dicho protocolo, también por que el protocolo de transporte sirve para especificar cómo queremos que viaje nuestro paquete.

# 1. Objetivos de investigación

En el presente estudio se realiza un análisis comparativo de desempeño de los sistemas operativos Windows Server 2008, CentOS 5,1, FreeBSD 7.1 para poder determinar una comparativa de los S.O. al trabajar junto con el protocolo IPv6 tanto en una red local, como en un loopback en donde se pretende realizar un diseño experimental en el cual se busca encontrar bajo diversos factores y variables cual SO presenta un mejor desempeño al combinar las variables con los factores. Una investigación previa (Sulaiman, Asaad, & Chieng, 2005) sobre Windows 2003, RedHat 9.0 y FreeBSD4.9 comparó el desempeño de equipos con características y configuraciones similares, donde principalmente, se evaluó utilizando el protocolo IPv6 en donde se visualizó la disponibilidad de una variedad de implementaciones de IPv6, con el propósito de evaluar su desempeño en diferentes sistemas operativos (Windows 2003, Redhat Linux 9.0 y FreeBSD 4.9)

# 2. Pregunta de investigación

La presente investigación tiene por objetivo responder el siguiente cuestionamiento:

¿Cuál es el desempeño que tiene el protocolo de Internet versión 6 sobre el sistema operativo CentOS 5.1, FreeBSD 7.1, comparado con el sistema operativo Windows server 2008?

# 3. Hipótesis

En el presente estudio se mide el desempeño observado por los SO Win Server 2008, CentOS 5.1, y FreeBSD 7.1, utilizando el protocolo TCP/IPv6 en donde se realizaron pruebas de desempeño para las siguientes configuraciones:

1. Una LAN donde se miden las variable dependientes tiempo y bytes por segundo entre los diferentes SOs, con la finalidad de comprobar que el desempeño observado por CentOS, FreeBSD es mayor respecto Windows 2008.
2. Un Loopback donde se miden las variable dependientes tiempo y bytes por segundo entre los diferentes SOs, con la finalidad de comprobar que el desempeño observado por CentOS, FreeBSD es mayor respecto Windows 2008.

Lo anterior corresponde a un diseño experimental, ya que con la manipulación deliberadamente de la variable independiente estaremos realizando este diseño. Las hipótesis de Investigación son las siguientes:

H1: El desempeño del sistema operativo CentOS 5.1 es mayor respecto al sistema operativo Windows 2008 empleando loopback

- H1a: El tiempo utilizado para el desempeño del sistema operativo

CentOS 5.1 es mejor con respecto al sistema operativo Windows 2008 empleando loopback

- H1b: Los bytes transmitidos por paquete enviado para medir el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo Windows 2008 empleando loopback

H2: El desempeño del sistema operativo CentOS 5.1 es mayor respecto al sistema operativo Windows 2008 empleando LAN

- H2a: El tiempo utilizado para el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo Windows 2008 empleando LAN
- H2b: Los bytes transmitidos por paquete enviado para medir el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo Windows 2008 empleando LAN

H3: El desempeño del sistema operativo CentOS 5.1 es mayor respecto al sistema operativo FreeBSD 7.1 empleando loopback

- H3a: El tiempo utilizado para el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo FreeBSD 7.1 empleando loopback
- H3b: Los bytes transmitidos por paquete enviado para medir el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo FreeBSD 7.1 empleando loopback

H4: El desempeño del sistema operativo CentOS 5.1 es mayor respecto al sistema operativo FreeBSD 7.1 empleando LAN

- H4a: El tiempo utilizado para el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo FreeBSD 7.1 empleando LAN
- H4b: Los bytes transmitidos por paquete enviado para medir el desempeño del sistema operativo CentOS 5.1 es mejor con respecto al sistema operativo FreeBSD 7.1 empleando LAN

Las variables bajo las cuales se realizarán las mediciones en las condiciones de prueba son las siguientes:

- Tasa de transferencia (troughput).- Según Sándor y Hassan (Sándor & Hassan, 2004) es el máximo número de datos que se transfieren durante una unidad de tiempo en un instante dado.

- Latencia.- Según Sándor y Hassan (Sándor & Hassan, 2004), es el retardo de tiempo debido a diferentes tamaños de paquetes que se transfieren, y que es medida desde el hardware. De acuerdo a Coulouris et al. (Coulouris, Dollimore, & Kindberg, 2001) la latencia incluye:
  1.- El tiempo que se toma en llegar a su destino el primero de una trama de bits transmitidos a través de una red.
  2.- El retardo en acceder a la red, que es grande cuando la red está muy cargada.
  3.- El tiempo empleado por los servicios de comunicación del sistema

operativo tanto en el proceso que envía como en el que recibe, y que varía según la carga actual de cada sistema operativo.

Por lo que Coulouris et al. (2001) definen que a partir de las anteriores definiciones, el tiempo requerido por una red para transferir un mensaje de n bits de longitud entre dos computadores es:

Tiempo de transmisión del mensaje = latencia + longitud / tasa de transferencia

Además, explican que la ecuación anterior es válida para mensajes cuya longitud no exceda un máximo que viene determinado por la tecnología de red subyacente. Los mensajes más largos deberán ser segmentados y el tiempo de transmisión será la suma del tiempo de transmisión de cada segmento.

- Ancho de banda (bandwith).- De acuerdo a Sándor y Hassan (2004) es la máxima capacidad de transferencia en un periodo largo de tiempo para un tamaño específico de mensajes. El ancho de banda total del sistema de una red lo cita Coulouris et al. (2001) lo definen como la medida de la productividad del volumen de tráfico que puede ser transferido a través de la red en un intervalo de tiempo dado. En tecnología de red de área local, como Ethernet, se utiliza toda la capacidad de transmisión de la red en cada transmisión y el ancho de banda es igual a la tasa de transferencia. En cambio, en la mayoría de las redes de área extensa los mensajes pueden ser transmitidos simultáneamente sobre varios canales diferentes, de modo que el ancho de banda no guarda una relación directa con la tasa de transferencia. Las prestaciones de las redes se deterioran con situaciones de sobrecarga; cuando existen demasiados mensajes en la red al mismo tiempo. El efecto preciso de la sobrecarga es la latencia, la tasa de transferencia y el ancho de banda de una red depende esencialmente de la tecnología de la red.

- Roundtrip.- Segun Sándor y Hassan (2004), es el tiempo en que un mensaje enviado se comunica a host destino y regresa a su origen de envío.

- La fluctuación (Jitter).- De acuerdo a Coulouris et al. (2001) es la variación en el tiempo invertido en completar el reparto de una serie de mensajes. La fluctuación es importante para los datos multimedia. Éste es el caso que si se producen muestras de audio consecutivas a un ritmo variable el sonido presentará graves distorsiones.

**Esquemas de conmutación**

De acuerdo a Coulouris et al. (2001), una red se compone de un conjunto de nodos conectados a través de circuitos. Para transmitir información entre dos nodos cualquiera se necesita un sistema de conmutación. A continuación se definen los cuatro tipos de conmutación que se utilizan en las redes de computadoras.

a) Difusión. La difusión (broadcast) es una técnica de transmisión que no involucra conmutación alguna. Toda información es transmitida a todos los nodos, y depende de los receptores decidir si el mensaje va dirigido a ellos o no. Algunas tecnologías de red LAN como Ethernet se basan en la difusión. Las redes inalámbricas están basadas necesariamente en este tipo.

b) Conmutación de circuitos.- Hace algún tiempo, las redes telefónicas eran las únicas redes de telecomunicaciones. Su modo de operar era: cuando el emisor marcaba un número, el par de hilos de cobre que iba desde su teléfono hasta la centralita era conectado automáticamente al par de hilos que lo llevaba al teléfono del receptor. En llamadas de larga distancia las conexiones se realizaban empleando conmutadores.

c) Conmutación de paquetes.- En cada nodo de conmutación se encuentra un computador (dónde se conectan varios circuitos). Los paquetes que llegan a un nodo se almacenan en la memoria del computador de ese nodo y luego son procesados por un programa por un programa que les envía hacia su destino eligiendo uno de los circuitos salientes que llevará al paquete a otro nodo que estará más cerca del destino del nodo anterior.

d) Frame relay.- La transmisión basada en el almacenamiento y el reenvío de paquetes no es instantánea; generalmente se toma desde unas pocas decenas de microsegundos hasta unos pocos milisegundos para encaminar los paquetes en cada nodo de la red, dependiendo del tamaño del paquete, las velocidades del hardware y la cantidad de tráfico. Los paquetes pueden ser encaminados hacia bastantes nodos antes de que alcancen su destino.

Se utilizaron dos equipos PC, los cuales tienen las mismas características de hardware. Para hacer la comparación de desempeño, en un equipo se instalaron los SOs CentOS, FreeBSD, uno a la vez; y en el otro el SO Windows 2008. En dichos equipos se estará midiendo el throughput, así como el tiempo de respuesta para los siguientes escenarios:

5. Configuración LAN respecto a la Variable Tiempo
6. Configuración LAN respecto a la Variable Bytes
7. Configuración Loopback respecto a la Variable Tiempo
8. Configuración Loopback respecto a la Variable Bytes

# 4. Metodología de investigación

En la presente investigación, se empleó un diseño factorial, en donde intervienen factores tales como: tiempo de conexión a TCP (TCP-Connection Time), el volumen de información que fluye en la red (Througput), y tiempo que tarda un paquete enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino (Round-Trip delay Time RTT); Además, se analizó el efecto conjunto de éstos sobre una respuesta en cada uno de los sistemas operativos bajo estudio.

Además, para la recolección de información se empleó el estándar de métricas para la evaluación de protocolos. Se midió el round trip time en bytes para TCP, el throughput en bytes por segundo (Bps). Los equipos PC utilizados tienen la siguiente configuración: procesador Intel Pentium 4 a 733 Mhz, con 512 megabytes en memoria principal y 60 gigabytes en disco duro.

## 4.1. Recolección de datos

Para la recolección de datos se utilizó el siguiente diseño:

| Sistema Op. / Arquitectura | WINDOWS SERVER 2008 | CentOS 5.1 | FreeBSD 7.1 |
|---|---|---|---|
| LAN | X<br>Y | X<br>Y | X<br>Y |
| LOOP BACK | X<br>Y | X<br>Y | X<br>Y |

X= tiempo
Y= paquetes/seg

### 4.2. Muestra

Para llevar a cabo la muestra se instaló cada sistema operativo en cada una de las particiones del disco duro de las dos PCs. Una vez instalados los SOs en los equipos computacionales, se procedió a configurar las direcciones IPv6 para cada uno. Dicho proceso se efectuó de la siguiente forma:

a) Configuración de IPv6 en Windows Server 2008. E dicho SO, IPv6 ya se encuentra instalado, lo único que se requiere es habilitarlo. Para ello, una vez que se accedió al SO se abrió la carpeta conexiones de red desde el panel de control. Se hizo clic en la interfaz, en la cual se deseaba instalar IPv6 y se seleccionó propiedades. En la ficha General, se dio clic en Instalar. Para posteriormente seleccionar el Protocolo y hacer clic en agregar. En el cuadro se seleccionó el protocolo de red Microsoft TCP / IP versión 6 y se dio clic en Aceptar. A continuación, haga clic en Cerrar para cerrar la interfaz de la hoja de propiedades.
b) Respecto a la versión CentOS 5.1 como FreeBSD 7.1, se tuvo que editar el script de la siguiente ruta /etc/sysconfig/network agregando la línea NETWORKING_IPV6=yes, posteriormente para posteriormente proceder a editar la interfaz sobre la que se trabajaría en éste caso eth0, abriéndolo desde la siguiente ruta: /etc/sysconfig/network-scripts/ifcfg-eth0, ya una vez dentro de la ruta anterior, se agregaron las siguiente líneas, IPV6INIT=yes,
IPV6ADDR=2607:f0d0:1002:0011:0000:0000:0000:0002,
IPV6_DEFAULTGW=2607:f0d0:1002:0011:0000:0000:0000:0001. Es importante mencionar que las direcciones IPv6 tanto para CentOS y FreeBSD, fueron diferentes para cada SO con

el fin de poder realizar las pruebas de conectividad. Una vez guardadas las configuraciones realizadas, se procedió a reiniciar el servicio, mediante el comando service network restart.

Una vez configurados los equipos bajo IPv6 en cada SO, se procede ahora a preparar los equipos para que puedan capturar los paquetes que se van a estar enviando, es decir a instalar el software wireshark. Dicho software es una distribución gratuita que permite la captura de paquetes a través de la red y el análisis de protocolos de red, brindando la capacidad de configurarlo en base a las necesidades tanto para la versión Windows como Unix.

La instalación de Wireshark para Windows se hiso a través de su propio instalador, el cual se encargó de agregar las librerías plugins, servicios. Mientras que para la versión Unix, la instalación en CentOS 5.1, se realizó mediante el comando sudo apt-get install wireshark, y para FreeBSD se empleó yum install wireshark wireshark-gnome

Una vez realizado lo anterior, ya es posible la captura de los paquetes para así para probar los distintos escenarios, el cual según define Gutiérrez et al. (2004) consiste en planear un conjunto de pruebas experimentales, de tal manera que los datos generados puedan analizarse estadísticamente para obtener conclusiones válidas y objetivas acerca del Sistema. El diseño experimental se configuró en base a los siguiente escenarios:

1.- Red LAN para Windows Server 2008
Se conectó una red local con dos equipos en los que se utilizó Windows Server 2008. La configuración requirió de un cable cruzado conectando ambos equipos, se procedió a hacer un ping de un equipo hacia otro para posteriormente ejecutar el programa wireshark para comenzar la captura de paquetes, donde se seleccionó la interfaz en la que se configuró IPv6 y se dejó que se capturasen los paquete en modo promiscuo, para éste caso el tamaño de la muestra fue de 119 paquetes (ver Figura 18).

Figura 18. Configuración red LAN para Windows Server 2008

2.- Red Loopback para Windows Server 2008
La configuración loopback se utiliza para verificar la operatividad del software de la base TCP/IP, se procedió a mandar un ping y posteriormente capturar los paquetes con ayuda de wireshark, donde se seleccionó la interfaz en la que se configuró IPv6 y se dejó que se capturasen los paquete en modo promiscuo utilizando un tamaño de muestra de 131 paquetes (ver Figura 19).

Figura 19. Configuración red loopback para Windows Server 2008

3.- Red LAN para FreeBSD 7.1
Se conectó una red local con dos equipos en los que se utilizó FreeBSD 5.1. La configuración requirió de un cable cruzado conectando ambos equipos, se procedió a hacer un ping de un equipo hacia otro para posteriormente ejecutar el programa wireshark para comenzar la captura de paquetes, donde se seleccionó la interfaz en la que se configuró IPv6 y se dejó que se capturasen los paquete en modo promiscuo, para éste caso el tamaño de la muestra fue de 108 paquetes(ver Figura 20).

Figura 20. Configuración red LAN para FreeBSD 7.1

4.- Red Loopback para FreeBSD 7.1:
La configuración loopback se utiliza para verificar la operatividad del software de la base TCP/IP, se procedió a mandar un ping y posteriormente capturar los paquetes con ayuda de wireshark, donde se seleccionó la interfaz en la que se configuró IPv6 y se dejó que se capturasen los paquete en modo promiscuo utilizando un tamaño de muestra de 129 paquetes (ver Figura 21).

Figura 21. Configuración red loopback para FreeBSD 7.1

5.- Red Lan para CentOS 5.1:
Se conectó una red local con dos equipos en los que se utilizó CentOS 7.1. La configuración requirió de un cable cruzado conectando ambos equipos, se procedió a hacer un ping de un equipo hacia otro para posteriormente ejecutar el programa wireshark para comenzar la captura de paquetes, donde se seleccionó la interfaz en la que se configuró IPv6 y se dejó que se capturasen los paquete en modo promiscuo, para éste caso el tamaño de la

muestra fue de 119 paquetes (ver Figura 22).

Figura 22. Configuración red LAN para CentOS 5.1

6.- Red Loopback para CentOS 7.1:
La configuración loopback se utiliza para verificar la operatividad del software de la base TCP/IP, se procedió a mandar un ping y posteriormente capturar los paquetes con ayuda de wireshark, donde se seleccionó la interfaz en la que se configuró IPv6 y se dejó que se capturasen los paquete en modo promiscuo utilizando un tamaño de muestra de 131 paquetes (ver Figura 22).

Figura 23. Configuración red loopback para CentOS 5.1

Enseguida se procedió a clasificar el diseño de experimento de acuerdo con la clasificación nombrada por Gutiérrez (2004), se encontró que lo que se pretendía era estudiar el efecto de varios factores sobre una o más variables de respuesta, en éste caso las variables se clasificaron en dos tipos, a) variable dependiente fueron los SO, y en b)variables independientes que fue la tecnología utilizada es decir la red LAN y Loopback configuradas

### 4.3. Configuración utilizada

El estudio se llevó a cabo utilizando una herramienta para la captura de paquetes a través de la red llamada Wireshark versión 1.0.6, la cual está basada en la librería Libpcap. Esta se ejecuta en la zona de usuario, pero la captura se realiza en la zona de Kernel. Si no existiese ningún mecanismo de filtrado, el Kernel no sabría cuales son los paquetes en los que está interesada la aplicación, por lo que tendría que traspasar la frontera Kernel - User space por cada paquete que transite por la red.

Para evitar esto, la aplicación establece un filtro en la zona kernel que sólo deja pasar los paquetes que se le indiquen (por ejemplo los que utilicen el protocolo TCP).

No existe un sistema único de filtrado, por el contrario, prácticamente cada SO reescribe su propia solución. Los siguientes son los nombres de filtrado utilizados de acuerdo al sistema operativo utilizado: NIT para SunOS, Ultrix

Packet Filter para DEC Ultrix, BPF para sistemas BSD (originalmente) y LSF (Linux Socket Filter ) la implementación para Linux.

El funcionamiento de Wireshark es el siguiente: se recopilan los paquetes desde el driver del dispositivo de red y se entregan al destinatario. Enseguida, debe decidirse si el paquete debe ser aceptado (debido a la coincidencia de direcciones Ethernet) y en caso afirmativo, cuanto de ese paquete debe ser entregado a la aplicación (no se entrega un paquete con sus cabeceras Ethernet).

## 4.4. Resultados Obtenidos

### 4.4.1. Técnica utilizada

Para la recolección de la muestra, una vez instalados los sistemas operativos, se procedió a capturar los paquetes que viajaron por la red local y por el Loopback, utilizando Wireshark 1.0.6, en cada uno de los sistemas operativos. Una vez obtenida la información se procedió a su análisis estadístico. Las variables empleadas en el estudio fueron el tiempo en que se transmitieron los paquetes tanto en una configuración LAN como en una de Loopback, así como el tamaño de los paquetes.

### 4.4.2.- Resultados de la Configuración LAN Respecto a la Variable Tiempo

Los resultados con respecto al tiempo utilizado por los Sistemas Operativos para el envío de paquetes, se muestran en la Tabla 4. Se sabe que la **desviación estándar** es una medida de dispersión usada en estadística que nos dice cuánto tienden a alejarse los valores puntuales del promedio en una distribución, siendo los valores 0.27, 0.23, 0.49 para Windows, FreeBSD y CentOS respectivamente. Respecto a la variable tiempo tanto Windows Server 2008, FreeBSD 7.1 y CentOS 5.1 son diferentes en sus comportamientos, esto debido a que la media de Windows fue de 0.24, FreeBSD 0.36 y CentOS 0.47. Analizando dichos datos, se puede decir que Windows Server 2008 tiene un desempeño menor respecto a la variable tiempo, frente a FreeBSD y Centos, siendo éste último el mejor evaluado, es decir, al comparar la media entre los SO se observó que:

Analizando la estadística descriptiva de la variable tiempo podemos observar que en promedio el SO Windows requirió más tiempo (.2408) comparado con los otros dos (FreeBSD .3600 y CentOS .4700). En consecuencia, Windows requiere aproximadamente 2 veces más tiempo que CentOS y 1.5 veces más que FreeBSD, y FreeBSD requiere 1.3 veces más tiempo respecto a CentOS. Por lo que el SO que mas rápido trasnmite paquetes con respecto a la variable tiempo fue CentOS 5.1, seguido por FreeBSD 7.1 y por último Windows Server 2008. Por lo que de forma inicial muestra indicios

de desempeño diferente.

**Descriptiva**

Tiempo

| | | N | Media | Desviaci.-on estandar | Error estandar | Intervalo de confianza para la media 95% | | Minimo | Maximo | Diferencia entre los componen-tes |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior | | | |
| Windows Server | | 108 | .2408751 | .2792295 | .0268688 | .1876107 | .2941395 | .000054 | .687151 | |
| FreeBSD | | 106 | .3600883 | .2315763 | .0224926 | .3154895 | .4046871 | .000614 | .881132 | |
| CentOS | | 117 | .4700662 | .4978663 | .0460277 | .3789024 | .5612299 | .000049 | 1.002415 | |
| Total | | 331 | .3600653 | .3720403 | .0204491 | .3198381 | .4002925 | .000049 | 1.002415 | |
| Modelo | Efectos fijos | | | .3609211 | .0198380 | .3210395 | .3990911 | | | |
| | Efectos variables | | | | .0668406 | .0724734 | .6476572 | | | .012199419 |

Tabla 4. Estadística descriptiva respecto a la variable tiempo

Para corroborar si existen diferencias en el desempeño se llevó a cabo una prueba ANOVA. Los resultados de esta se observan en la Tabla 5. Se puede observar que existen diferencias significativas (p<=0.001) en el desempeño de los SOs con respecto al tiempo utilizado. Por lo tanto, cada uno de los SOs bajo estudio tienen un desempeño diferente y existe la probabilidad de que alguno de ellos sea mejor comparado con el resto.

**ANOVA**

Tiempo

| | Suma de cuadrados | df | Media de cuadrados | F | Sig. |
|---|---|---|---|---|---|
| Entre Grupos | 2.950 | 2 | 1.475 | 11.323 | .000 |
| Dentro de Grupos | 42.727 | 328 | .130 | | |
| Total | 45.677 | 330 | | | |

Tabla 5. Tabla ANOVA (respecto variable tiempo)

Para identificar el mejor de ellos, es necesario hacer un análisis posterior (Post Hoc) así como una serie de contrastes donde se compare el desempeño de cada sistema operativo comparado con el resto. Par dicho propósito, se utilizó la Diferencia Mínima Significativa (LSD) y Dunnet. Los resultados obtenidos de estos análisis con los datos analizados se muestran en la Tabla 6. Por lo tanto se acepta la hipótesis H2a de que CentOS tiene un mejor desempeño con respecto a Windows server 2008 en la configuración LAN (p<=.001). Además, se acepta la hipótesis H4a de que CentOS tiene un mejor desempeño con respecto a FreeBSD en la configuración LAN (p<=.024).

| | (I) Sistema_Operativo | (J) Sistema_Operativo | Diferencia de las medias (I-J) | Error estándar | Sig. | Intervalo de confianza al 95% | |
|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior |
| LSD | Windows Server | FreeBSD | -1.192131822 | .04934628 | .016 | -.216288311 | -.022138054 |
| | | CentOS | -2.291910477 | .04816136 | .000 | -.323935186 | -.134446910 |
| | FreeBSD | Windows Server | .1192131822* | .04934628 | .016 | .022138054 | .216288311 |
| | | CentOS | -1.099778655 | .04839705 | .024 | -.205185652 | -.014770079 |
| | CentOS | Windows Server | .22919104* | .04816136 | .000 | .134446910 | .323935186 |
| | | FreeBSD | .109977865 | .04839705 | .024 | .014770079 | .205185652 |
| Dunnett T3 | Windows Server | FreeBSD | -1.192131822 | .03504079 | .002 | -.203542367 | -.034883997 |
| | | CentOS | -2.291910477 | .05329625 | .000 | -.357569003 | -.100813092 |
| | FreeBSD | Windows Server | .119213182 | .03504079 | .002 | .034883997 | .203542367 |
| | | CentOS | -.109977865 | .05122963 | .096 | -.233492826 | .013537095 |
| | CentOS | Windows Server | .229191047 | .05329625 | .000 | .100813092 | .357569003 |
| | | FreeBSD | .109977865 | .05122963 | .096 | -.013537095 | .233492826 |
| Dunnett t (2-sided)[a] | Windows Server | CentOS | -2.291910477 | .04816136 | .000 | -.336309359 | -.122072737 |
| | FreeBSD | CentOS | -1.099778655 | .04839705 | .044 | -.217620380 | -.002335351 |

*. La diferencia de las medias es significativo al nivel 0.05.

a. Prueba Dunnett t trata como grupo de control , y compara todos los demás grupos con ella.

Tabla 6. Comparaciones múltiples (respecto variable tiempo)

### 4.4.3.- Resultados de la configuración LAN respecto a la variable paquetes enviados por Byte

Los resultados con respecto a los bytes enviados utilizado por los Sistemas Operativos para el envío de paquetes, se muestran en la Tabla 7. Se sabe que la **desviación estándar** es una medida de dispersión usada en estadística que nos dice cuánto tienden a alejarse los valores puntuales del promedio en una distribución, siendo los valores 1.518, 21.964, 2.776 para Windows, FreeBSD y CentOS respectivamente. Respecto a la variable tiempo tanto Windows Server 2008, FreeBSD 7.1 y CentOS 5.1 son diferentes en sus comportamientos, esto debido a que la media de Windows fue de 93.70, FreeBSD 90.86 y CentOS 93.25. Analizando dichos datos, se puede decir que Windows Server 2008 tiene un desempeño menor respecto a la variable tiempo, frente a FreeBSD y Centos, siendo éste último el mejor evaluado, es decir, al comparar la media entre los SO se observó que:

Analizando la estadística descriptiva de la variable bytes podemos observar que en promedio el SO Windows Server 2008 requirió más tiempo (93.70) comparado con los otros dos (FreeBSD 90.86 y CentOS 93.25). En consecuencia, Windows envía aproximadamente 0.99 bytes por uno enviado por CentOS y 0.96 bytes envíados, por uno de FreeBSD, y FreeBSD requiere 1.02 veces más bytes respecto a CentOS. Por lo que el SO que mas rápido transmite paquetes con respecto a la variable bytes fue Windows Server 2008, seguido por CentOS 5.1 y por último FreeBSD 7.1. Por lo que de forma inicial muestra indicios de desempeño diferente.

N_Bytes

| | | N | Media | Deviacion estandar | Error estandar | Intervalo de confianza del 95% de media | | Minimos | Maximos | Diferencia entre los component-tes |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior | | | |
| Windows Server | | 108 | 93.70 | 1.518 | .146 | 93.41 | 93.99 | 86 | 94 | |
| FreeBSD | | 106 | 90.86 | 21.964 | 2.133 | 86.63 | 95.09 | 60 | 243 | |
| CentOS | | 117 | 93.25 | 2.776 | .257 | 92.74 | 93.76 | 78 | 94 | |
| Total | | 331 | 92.63 | 12.589 | .692 | 91.27 | 93.99 | 60 | 243 | |
| Modelo | Efectos fijos | | | 12.566 | .691 | 91.27 | 93.99 | | | |
| | Efectos aleatorios | | | | .871 | 88.88 | 96.38 | | | .844 |

Tabla 7. Desviación y error estándar (respecto variable n bytes)

Para corroborar si existen diferencias en el desempeño se llevó a cabo una prueba ANOVA. Los resultados de esta se observan en la Tabla 8. Se puede observar que existen diferencias significativas (p<=0.206) en el desempeño de los SOs con respecto al tiempo utilizado. Por lo tanto, cada uno de los SOs bajo estudio tienen un desempeño similar y no existe la probabilidad de que alguno de ellos sea mejor comparado con el resto.

**ANOVA**

N_Bytes

| | Suma de cuadrados | df | Cuadrado de las medias | F | Sig. |
|---|---|---|---|---|---|
| Entre Grupos | 501.825 | 2 | 250.913 | 1.589 | .206 |
| Dentro de Grupos | 51795.208 | 328 | 157.912 | | |
| Total | 52297.033 | 330 | | | |

Tabla 8. Tabla ANOVA (respecto a variable n bytes)

### 4.4.4.- Resultados de la configuración LOOPBACK respecto a la variable tiempo

Los resultados con respecto al tiempo utilizado por los Sistemas Operativos para el envío de paquetes, se muestran en la Tabla 10. Se sabe que la **desviación estándar** es una medida de dispersión usada en estadística que nos dice cuánto tienden a alejarse los valores puntuales del promedio en una distribución, siendo los valores 4.922, 0.501, 0.501 para Windows, FreeBSD y CentOS respectivamente. Respecto a la variable tiempo tanto Windows Server 2008, FreeBSD 7.1 y CentOS 5.1 son diferentes en sus comportamientos, esto debido a que la media de Windows fue de 1.862, FreeBSD 0.492 y CentOS 0.496. Analizando dichos datos, se puede decir que Windows Server 2008 tiene un desempeño menor respecto a la variable tiempo, frente a FreeBSD y Centos, siendo éste último el mejor evaluado, es decir, al comparar la media entre los SO se observó que:

Analizando la estadística descriptiva de la variable tiempo podemos observar que en promedio el SO Windows requirió más tiempo (1.862) comparado con los otros dos (FreeBSD 0.492 y CentOS 0.496). En consecuencia, Windows requiere aproximadamente 0.266 fracción de tiempo por un

paquete enviado por CentOS y 0.264 fracción de tiempo por un paquete enviado por FreeBSD, y FreeBSD requiere 1.007 veces más tiempo respecto a CentOS. Por lo que el SO que mas rápido transmite paquetes con respecto a la variable tiempo fue Windows, seguido por CentOS 5.1 y por último FreeBSD 7.1. Por lo que de forma inicial muestra indicios de desempeño diferente.

**Descriptivas**

Tiempo

| | | N | Media | Desviacion estandar | Error estandar | Intervalo de confianza para la media al 95% | | Minimo | Maximo | Diferencia entre los componentes |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior | | | |
| Windows Server | | 99 | 1.862425 | 4.922544 | .494734 | .880641 | 2.84421 | .000076 | 39.10025 | |
| FreeBSD | | 128 | .4921907 | .501880 | .044360 | .404409 | .57997 | .000022 | 1.00004 | |
| CentOS | | 129 | .4960642 | .501845 | .044185 | .408636 | .58349 | .000023 | 1.00073 | |
| Total | | 356 | .8746429 | 2.692040 | .142677 | .594042 | 1.15524 | .000022 | 39.10025 | |
| Modelo | Efectos fijos | | | 2.628513 | .139310 | .600659 | 1.14862 | | | |
| | Efectos aletorios | | | | .437526 | -1.007882 | 2.75716 | | | .5090688 |

Tabla 9. Estadística descriptiva con respecto variable tiempo

Para corroborar si existen diferencias en el desempeño se llevó a cabo una prueba ANOVA. Los resultados de esta se observan en la Tabla 11. Se puede observar que existen diferencias significativas (p<=.001) en el desempeño de los SOs con respecto al tiempo utilizado. Por lo tanto, cada uno de los SOs bajo estudio tienen un desempeño diferente y existe la probabilidad de que alguno de ellos sea mejor comparado con el resto.

**ANOVA**

Tiempo

| | Suma de cuadrados | df | Cuadrado de las medias | F | Sig. |
|---|---|---|---|---|---|
| Entre Grupos | 133.807 | 2 | 66.903 | 9.683 | .000 |
| Dentro de Grupos | 2438.907 | 353 | 6.909 | | |
| Total | 2572.714 | 355 | | | |

Tabla 10. Tabla ANOVA con respecto a variable tiempo

Para identificar el mejor de ellos, es necesario hacer un análisis posterior (Post Hoc) así como una serie de contrastes donde se compare el desempeño de cada sistema operativo comparado con el resto. Para dicho propósito, se utilizó la Diferencia Mínima Significativa (LSD) y Dunnet. Los resultados obtenidos de estos análisis con los datos analizados se muestran en la Tabla 11. Por lo tanto, se acepta la hipótesis H1a de que CentOS tiene un mejor desempeño con respecto a Windows server 2008 en la configuración Loopback (p<=.001). Además, se rechaza la hipótesis H3a de que CentOS tiene un mejor desempeño con respecto a FreeBSD en la configuración Loopback (p<=.991).

**Comparasiones Múltiples**

Variable dependendiente:Tiempo

| | (I) Sistema_ Operativo | (J) Sistema_ Operativo | Diferencia de medias (I-J) | Error estandar | Sig. | Intervalo de Confianza al 95% | |
|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior |
| LSD | Windows Server | FreeBSD | 1.370235 | .351803 | .000 | .678340 | 2.062130 |
| | | CentOS | 1.366361 | .351208 | .000 | .675637 | 2.057086 |
| | FreeBSD | Windows Server | -1.370235 | .351803 | .000 | -2.062130 | -.678340 |
| | | CentOS | -.003873 | .327926 | .991 | -.648809 | .641062 |
| | CentOS | Windows Server | -1.366361 | .351208 | .000 | -2.057086 | -.675637 |
| | | FreeBSD | .003873 | .327926 | .991 | -.641062 | .648809 |

Tabla 11. Comparaciones múltiples

### 4.4.5.- Resultados de la configuración LOOPBACK respecto a las variable paquetes enviados por byte

Los resultados con respecto a los bytes enviados utilizado por los Sistemas Operativos para el envío de paquetes, se muestran en la Tabla 12. Se sabe que la **desviación estándar** es una medida de dispersión usada en estadística que nos dice cuánto tienden a alejarse los valores puntuales del promedio en una distribución, siendo los valores 51.904, 3.420, .721 para Windows, FreeBSD y CentOS respectivamente. Respecto a la variable tiempo tanto Windows Server 2008, FreeBSD 7.1 y CentOS 5.1 son diferentes en sus comportamientos, esto debido a que la media de Windows fue de 51.904, FreeBSD 3.420 y CentOS 0.721. Analizando dichos datos, se puede decir que Windows Server 2008 tiene un desempeño menor respecto a la variable tiempo, frente a FreeBSD y Centos, siendo éste último el mejor evaluado, es decir, al comparar la media entre los SO se observó que:

Analizando la estadística descriptiva de la variable bytes podemos observar que en promedio el SO Windows Server 2008 requirió más tiempo (95.56) comparado con los otros dos (FreeBSD 59.58 y CentOS 117.74). En consecuencia, Windows envía aproximadamente 1.232 bytes por uno enviado por CentOS y 0.623 bytes envíados, por uno de FreeBSD, y FreeBSD requiere 1.976 veces más bytes respecto a CentOS. Por lo que el SO que mas rápido transmite paquetes con respecto a la variable bytes fue CentOS 5.1, seguido por Windows Server 2008 y por último FreeBSD 7.1. Por lo que de forma inicial muestra indicios de desempeño diferente.

**Descriptives**

N_Bytes

| | | N | Media | Desvia-cion estandar | Error estandar | Intervalo de confianza para la media al 95% | | Minimo | Máximo | Diferencia entre los componen-tes |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior | | | |
| Windows Server | | 99 | 95.56 | 51.904 | 5.217 | 85.20 | 105.91 | 64 | 342 | |
| FreeBSD | | 128 | 59.58 | 3.420 | .302 | 58.98 | 60.18 | 56 | 94 | |
| CentOS | | 129 | 117.74 | .721 | .063 | 117.62 | 117.87 | 115 | 118 | |
| Total | | 356 | 90.66 | 37.009 | 1.961 | 86.80 | 94.52 | 56 | 342 | |
| Modelo | Efectos fijos | | | 27.428 | 1.454 | 87.80 | 93.52 | | | |
| | Efectos aleatorios | | | | 17.785 | 14.14 | 167.18 | | | 929.796 |

Tabla 12. Estadística descriptiva con respecto variable bytes

Para corroborar si existen diferencias en el desempeño se llevó a cabo una prueba ANOVA. Los resultados de esta se observan en la Tabla 13. Se puede observar que existen diferencias significativas (p<=146.655) en el desempeño de los SOs con respecto al tiempo utilizado. Por lo tanto, cada uno de los SOs bajo estudio tienen un desempeño diferente y existe la probabilidad de que alguno de ellos sea mejor comparado con el resto.

N_Bytes

| | Suma de cuadrados | df | Cuadrado de las medias | F | Sig. |
|---|---|---|---|---|---|
| Entre Grupos | 220659.652 | 2 | 110329.826 | 146.655 | .000 |
| Dentro Grupos | 265564.221 | 353 | 752.307 | | |
| Total | 486223.874 | 355 | | | |

Tabla 13. Tabla ANOVA (respecto a variable n bytes)

Para identificar el mejor de ellos, es necesario hacer un análisis posterior (Post Hoc) así como una serie de contrastes donde se compare el desempeño de cada sistema operativo comparado con el resto. Par dicho propósito, se utilizó la Diferencia Mínima Significativa (LSD) y Dunnet. Los resultados obtenidos de estos análisis con los datos analizados se muestran en la Tabla 14. Por lo tanto se acepta la hipótesis H1b de que CentOS tiene un mejor desempeño con respecto a Windows server 2008 en la configuración Loopback (p<=.001). Además, se acepta la hipótesis H3b de que CentOS tiene un mejor desempeño con respecto a FreeBSD en la configuración Loopback (p<=.001).

N_Bytes

| | (I) Sistema_Operativo | (J) Sistema_Operativo | Diferencia de medias (I-J) | Error estándar | Sig. | Intervalo de Confianza al 95% | |
|---|---|---|---|---|---|---|---|
| | | | | | | Límite inferior | Límite superior |
| LSD | Windows Server | FreeBSD | 35.977* | 3.671 | .000 | 28.76 | 43.20 |
| | | CentOS | -22.189* | 3.665 | .000 | -29.40 | -14.98 |
| | FreeBSD | Windows Server | -35.977* | 3.671 | .000 | -43.20 | -28.76 |
| | | CentOS | -58.166* | 3.422 | .000 | -64.90 | -51.44 |
| | CentOS | Windows Server | 22.189* | 3.665 | .000 | 14.98 | 29.40 |
| | | FreeBSD | 58.166* | 3.422 | .000 | 51.44 | 64.90 |

Tabla 14. Comparaciones múltiples

# V. Conclusiones

## 5.1.- Conclusiones generales

El presente estudio tiene por objetivo el identificar cual de tres alternativas de sistemas operativos distribuidos tiene un mejor desempeño. Específicamente, conocer si CentOS 5.1 tiene un mejor desempeño en términos de tiempo y bytes transmitidos por paquete comparándolo con Windows Server 2008, y FreeBSD 7.1. Considero fundamental este estudio ya que las organizaciones se enfrentan diariamente a un constante cambio en la tecnología así como en el ámbito comercial. En consecuencia, las empresas deben eficientar el uso de sus recursos tal como la inversión en tecnología. Una finalidad es el hecho de que éste experimento sirva a organizaciones que estén buscando evaluar SOs para servidores, en donde la característica común es que se evaluaron SOs de última generación, con la finalidad de decidirse por uno de ellos e implantarlo en su organización, ya que no existe en el mercado un estudio con las características de evaluación descritas anteriormente que evalúen el envío de paquetes a través de la red Para realizar la comparación, se usaron dos configuraciones: LAN y Loopback. Una vez hecho el estudio obtuvimos los siguientes resultados.

1.-Para la configuración LAN respecto a la variable tiempo, se aceptaron tanto la hipótesis H2a y la H4a, ya que CentOS 5.1 tiene un mejor desempeño comparado con Windows server 2008 y FreeBSD 7.1., esto pudo haber sido debido a lo siguientes:

- Proporciona balanceo de carga a través de Linux Virtual Server (LVS) el cual se encarga del enrutamiento IP y balance de cargas.
- En caso de encontrarse un nodo inoperante, provee servicios de conmutación de error.
- Cuenta con herramientas de administración de cluster que se encarga de mantener siempre disponibles los servicios.

Por lo cual, CentOS 5.1 reduce los tiempos de respuesta a peticiones efectuadas en tiempo real en una red LAN, favoreciendo así los costos de producción en las organizaciones, debido a que todas las peticiones son atendidas en el mismo instante que se solicitan.

2.-Para la configuración LAN respecto a la variable bytes, se rechazaron tanto la hipótesis H2b y en la H4b. Esto se debió a que CentOS 5.1 no mostró mejor desempeño comparado con Windows Server 2008; ni comparado con FreeBSD 7.1. En consecuencia, podemos decir que si la decisión de una organización está basada en el desempeño en cuanto a la cantidad de bytes enviados a través de una red LAN, cualquiera de los tres sistemas operativos analizados en el presente estudio le proporciona las mismas ventajas.

3.- Para la configuración Loopback respecto a la variable tiempo, se aceptó la hipótesis H1a, pero se rechazó H3a, al comparar CentOS 5.1 se obtuvo evidencia de un mejor desempeño en el escenario configurado respecto a Windows Server 2008, pero no fue mejor respecto a FreeBSD. En

consecuencia, si la decisión es en Loopback y depende del tiempo necesesario se recomienda cualquiera de los SOs CentOS 5.1 y FreeBSD 7.1 ya que tienen un desempeño similar. Por el contrario, Windows Server 2008 no es recomendable para este escenario.

4.- Para la configuración Loopback respecto a la variable bytes, se encontró evidencia para aceptar tanto la hipótesis H1a como la H3a, ya que CentOS 5.1 tuvo un mejor desempeño comparado con Windows server 2008 y FreeBSD 7.1. Esto pudo ser debido a las siguientes características:

- Utiliza el gestor de volúmenes lógicos (LVM), el cual proporciona virtualización de almacenamiento, es decir no se limita a tamaños de disco físicos locales

Bajo la configuración de Loopback y si lo más importante es el desempeño con respecto al número de bytes transmitidos se recomienda CentOS 5.1para condiciones en que se desarrollen o ejecuten programas bajo ambiente web local con volúmenes altos de información.

## 5.2 Limitaciones

El presente estudio se realizó utilizando dos configuraciones comparando tres sistemas operativos con versiones particulares, razón por la cual se debe tener mucho cuidado al tratar de generar lo resultados a otros SOs u otras versiones ya que podrían ser diferentes.

Adicionalmente, la recolección de datos se hizo corriendo únicamente el software específico necesario para el presente estudio. En consecuencia, los resultados podrían ser diferentes si se corre el análisis y la configuración está funcionando en forma normal.

Debido a que el hardware utilizado es muy particular, los resultados no pueden generalizarse a otros equipos ni a otras arquitecturas. Para esos casos, los resultados podrían ser diferentes.

Pueden existir algunas otras variables no identificados en este estudio que podrían afectar los resultados.

## 5.3 Trabajos futuros

Se sugiere que el presente estudio se conduzca utilizando otras configuraciones diferentes a las aquí utilizadas. Por ejemplo, en una red WAN y comparar otros sistemas operativos y/o otras versiones. De esta forma, se podría tener mayor conocimiento con respecto a la problemática aquí estudiada.

Ya que la recolección de datos se hizo corriendo únicamente el software específico necesario para el presente estudio. Considero muy importante el correr el análisis con la configuración funcionando en forma normal. De esta forma, los resultados son más reales.

Debido a que el hardware utilizado es muy particular, se recomienda ejecutar el presente estudio en otros equipos y/o en otras arquitecturas. De esta forma, se podrían generalizar los resultados obtenidos y servir como guía más universal a las organizaciones.

# VI. REFERENCIAS

Aguilera, A. (2009). VoIP por Alejandro Torres Aguilera. from http://atorresa.wordpress.com/category/jitter/

Applegate, L. M., McFarlan, F. W., & McKenney, J. L. (1999). *Corporate Information Systems Management text and cases*: Irwin McGraw-Hill.

Applegate M., L., Mc Farlan Warren, F., & McKenney L., J. (1999). *Corporate Information Systems Management: Text and cases* (5th edition ed.): McGraw-Hill.

Baker, A. (1997). *The windows NT device driver book: A guide for programmers*: Prentice-Hall PTR-Upper Saddle River.

Barker, J. A. (1989). *Discovering the future: The business of paradigms*.

Comer, D. E. (2003). *Redes globales de información con Internet y TCP/IP principios básicos, protocolos y arquitectura*: Prentice-Hall.

Comunicaciones, G. (1998). *Telecomunicaciones: Redes de datos*: Mc Graw Hill.

Corbett, L. M. (1998). Benchmarking Manufacturing Performance in Australia and New Zeland. *Benchmarking for Quality Management & Technology*, 271-282.

Coulouris, G., Dollimore, J., & Kindberg, T. (2001). *Sistemas Distribuidos, conceptos y diseños* (Tercera edición ed.): Addison-Wesley.

Deitel, H. M. (1999). *Introducción a los Sistemas Operativos*: Addison-Wesley Iberoamericana.

Douglas. (2001). *Essential SNMP*: O`Reilly.

Forouzan, A. B. (2002). *Transmisión de Datos y redes de comunicaciones*: Mc Graw Hill.

FreeBSD. (2008). About FreeBSD. Retrieved 29/01/2009, 2009, from http://www.freebsd.org/about.html

García, M. M. (2008). Parallel/Shared/Distributed Filesystems. Retrieved 02/02/2009, 2009, from http://lists.centos.org/pipermail/centos/2008-November/067622.html

Gutiérrrez, H. P., & Vara, R. S. d. l. (2004). *Análisis y diseño de experimentos* (Primera edición ed.): McGraw-Hill.

Jaus, G. (2009). Retrieved 04/02/2009, from http://glx-jaus.blogspot.com/

Jones, N. (2004). Benchmarking Training Article.

Martínez, M. (2008). Modelo para el desarrollo de servicios ATM y FrameRelay. Retrieved 29/01/2009, 2009, from http://es.geocities.com/marbry69/e3/T_2.htm

Matzan, J. (2006). The differences between Linux distributions. from http://www.softwareinreview.com/linux_guides/the_differences_between_linux_distributions.html

Microsoft. (2008a). Introducción Técnica a Windows Sever 2008. Retrieved 29/01/2009, from http://www.microsoft.com/spain/windowsserver2008/evaluation/overview.mspx

Microsoft. (2008b). Windows Server 2008, una nueva plataforma productiva. Retrieved 29/01/2009, from http://www.microsoft.com/latam/technet/articulos/tn/2007/jul-01.mspx

Mohamed, S. S., Buhari, M. S., & Saleem, H. (2006). Performance comparison of packet transmission over IPv6 network on different platforms. *The Institution of Engineering and Technology*.

Porter, M. E. (1980). *Competitive Strategy: Techniques for Analyzing Industries and Competitors*: Boston: Harvard Business School.

Sándor, J., & Hassan, C. (2004). Exploiting Fast Ethernet Performance in Multiplataform Cluster Environment. *Budapest University of Technology and Economics*.

Sandoval, A. (2005). Sistemas BSD y Linux. Retrieved 29/01/2009, from http://www.microteknologias.cl/linux_bsdlinux.html

Sandoval, A. (2008). Comparando BSD, Linux, Solaris y Unix. from http://microteknologias.wordpress.com/2008/09/07/comparando-bsd-linux-solaris-y-unix/

Silberschatz Abraham, G. B. P., Gagne Greg. (2005). *Fundamentos de sistemas operativos*: McGraw Hill.

Siles Peláez, R. (2002). *Análisis de Seguridad de la familia de protocolos TCP-IP y sus servicios asociados*: O'Reilly & Associates, Inc.

softonic, f. d. p. (2009). ¿Cómo funciona un ataque de denegación de servicio? Retrieved 21 Mayo 2009, 2009, from http://foros.softonic.com/seguridad/funciona-ataque-denegacion-servicio-53527

Sole, T. D., & Bist, G. (1995). Benchmarking in Technical Information. *IEEE Transactions on Professional Communication, 38*(2), 77-82.

Sulaiman, S. M., Asaad, M. A., & Chieng, D. (2005). Evaluation of IPv6 and Comparative Study with Different Operating Systems. *Cyberjaya, Malaysia*.

Sulaiman, S. M., Asaad, M. A., & David, C. (2005). Evaluation of IPv6 and Comparative Study with Different Operating Systems. *Cyberjaya, Malaysia*.

Tanenbaum, A. (2003). *Redes de Computadoras*: Prentice Hall.

VNOC, C. d. O. d. V. U. (2009). VNOC,Centro de Operaciones de Videoconferencia UNAM,. Retrieved 29/01/2009, from http://vnoc.unam.mx/index.php?option=com_glossary&Itemid=65&catid=20&func=display&search=loopback&search_type=1

Wikipedia. (2009a). Paradigma. from http://es.wikipedia.org/wiki/Paradigma

Wikipedia. (2009b). Protocolo de comunicaciones. Retrieved 27/1/2009, 2009, from http://es.wikipedia.org/wiki/Protocolo_de_red

Wikipedia. (2009c). Throughput Retrieved 27/1/2009, 2009, from http://es.wikipedia.org/wiki/Throughput

Wohlin, C., Aurum, A., Petersson, H., Shull, F., & Ciolkowski, M. (2002). *Software Inspection Benchmarking - A Qualitative and Quantitative Comparative Opportunity.* Paper presented at the Eighth IEEE Symposium on Software Metrics (METRICS' 02), Ottawa, Canada.

Zeadally, S. (2000). Implementation and performance of QoS-aware Java Applications over ATM networks. *Wayne State University, Detroit*.