UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

IPv6 PARA LINUX

TESIS

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A :

MAURICIO HERNANDEZ GARCIA

DIRECTOR DE TESIS:

ING. GERMAN SANTOS JAIMES

CIUDAD UNIVERSITARIA, 2000

A:

mi mamá,
por toda la paciencia, esfuerzo
y por enseñarme tanto;

mi hermano,
por su apoyo;

mi hermana,
por ser como es;

Germán,
por apoyarme y dirigirme;

Daniel Sol,
por darme la libertad de investigar y experimentar;

Eduardo Resendiz,
por ser mi compañero y amigo;

Sandra, Paco, Héctor, Hugo, Mariza, Edith,
Adriana, Gerardo, Marcela, Lilia, Gris y Mike,
por ser mis y amigos;

La Dirección General de Servicios de Cómputo Académico,
por abrirme sus puertas;

y sobre todo,

a la Universidad Nacional Autónoma de México y
a la Facultad de Ingeniería,
por hacerme quien soy.

Índice Temático

INTRODUCCIÓN

Introducción

Las redes de computadoras se han vuelto indispensables en la vida cotidiana, gracias a ellas es que podemos realizar trabajos colaboratorios en donde pueden intervenir varias personas que se encuentran en lugares distantes, se puede utilizar dispositivos remotos y puede interaccionarse con otros equipos y personas sin la necesidad de transportarse físicamente a los lugares en donde se requiera el equipo.

La historia de las redes de computadoras muestra los cambios radicales que han sufrido las mismas, desde ser un par de computadoras conectadas hasta lo que ahora llamamos Internet. Durante esta evolución, se han tenido que hacer varias consideraciones tanto de software como de hardware para que pudiera realizarse.

En un principio, las redes de computadoras eran tan pequeñas y tan simples que se tenían pocas convenciones para poder establecer la comunicación que se necesitaba. Actualmente es necesario hacer consideraciones más complejas acerca de los equipos y dispositivos que se encuentran conectados a la red, ya que algunos de ellos pueden presentar problemas de incompatibilidad que deben resueltos y que no pueden evitarse, ya que puede ser necesario que sea el caso de un dispositivo único que sea indispensable para la realización de alguna tarea específica y que no pueda ser reemplazado o substituido por ninguno otro.

Para poder hacer todo esto, es que se definen los protocolos de comunicación, en el caso de Internet, le llamamos IP (*Internet Protocol*) a uno de los protocolos que se usan; en estos protocolos se define la manera en la que los equipos y dispositivos conectados a la red deben trabajar, es un conjunto de "reglas" específicas y bien definidas.

La IETF *(Internet Engineering Task Force)* ha dado la recomendación de reemplazar este protocolo por una versión con características nuevas para que pueda soportar las nuevas necesidades de la red y los requerimientos de los usuarios cuando ejecutan aplicaciones que demandan más recursos.

La nueva versión del protocolo es la 6, comúnmente conocida como IPv6 *(Internet Protocol version 6)* o IPng *(Internet Protocol next generation)*. Esta versión tiene varias ventajas sobre la anterior, las que sobresalen (por ser evidentes) son: manejo de direcciones de 128 bits y resuelve el desbordamiento de las tablas de ruteo.

Otra de las grandes ventajas que ofrece este protocolo sobre otros que fueron contemplados para la evolución, es que permite la coexistencia entre el mismo y su versión anterior, de esta manera es posible crear mecanismos de transición gradual de un protocolo a otro, lo que no implica un cambio drástico en la tecnología que requeriría cambios simultáneos de hardware y software, con las dificultades que esto conlleva.

Aprovechando todas las características que presenta esta nueva versión del protocolo se podrá tener una red más eficiente en donde puedan ejecutarse aplicaciones que demanden más recursos y de manera eficiente permitiendo que el tráfico sea adecuado en cada momento para cada tipo de usuario, de cada tipo de aplicación y que, también se consideren cuestiones geográficas para hacer más eficiente el uso de los recursos de la red.

Los cambios serán evidentes en todas las redes, de todos los tamaños y tipos, pero serán más evidentes en las redes que son usadas comúnmente por los usuarios tradicionales. De esta manera veremos que Internet tendrá un funcionamiento más óptimo y que encontraremos en esta red algunas aplicaciones que actualmente son imposibles de encontrar por la cantidad y calidad de recursos que demandan.

Internet evolucionará aprovechando las capacidades de IPv6, ya que con este protocolo tendrá acceso a otros recursos, así como a una optimización de los mismos. El servicio que se prestará se adaptará a las aplicaciones de los usuarios para obtener el máximo beneficio y máximo desempeño de las redes de comunicaciones en las que estarán incluidos una amplia variedad de dispositivos que no se reducen solamente al campo de los equipos de cómputo.

Para poder hacer uso de las redes de comunicaciones es necesario tener el medio lógico que lo permita, por ello es necesario considerar una plataforma sobre la cual se ejecuten las aplicaciones que se requieran. Linux es un sistema operativo con características sobresalientes en el área de redes.

Linux tiene otras ventajas que lo hacen ser considerado como una buena opción de sistema operativo para integrarlo con las redes de comunicaciones empleando IPv6. Actualmente, se cuenta con algunas aplicaciones creadas para ser ejecutadas sobre este sistema operativo y que tienen implementación para la nueva versión del protocolo de comunicación. De esta manera se pueden realizar algunas pruebas de funcionamiento del sistema una vez que se haya concluido la etapa de configuración del mismo.

La configuración del sistema es muy importante, ya que de esto depende que pueda encontrarse en un estado funcional y que permita la comunicación con otros dispositivos conectados a la red, tanto los que se encuentran usando IPv4 como los que utilizan IPv6.

Veremos la forma en la que el sistema tiene que ser configurado para adoptar las capacidades que ofrece IPv6 empleando varias actualizaciones a programas de aplicación básicos, así como para programas de comunicación y monitoreo de la red.

Con lo anterior, veremos de que manera se conjuntan IPv6 y Linux ofreciendo una base sólida para la evolución de los sistemas adaptándose a las necesidades de las nuevas redes de comunicación de alto desempeño. Usando la nueva tecnología de IPv6 combinada con un sistema operativo potente, funcional y adaptado a las redes veremos que la transición no tiene que ser costosa ni difícil.

REDES DE COMPUTADORAS

Introducción

Las redes de computadoras son una herramienta que se ha vuelto indispensable para muchas personas en la actualidad. Gracias a ellas podemos trabajar en lugares diferentes accesando a archivos de datos, a programas de aplicación que se encuentran residentes en una computadora que se encuentra físicamente en un lugar lejano, compartir dispositivos de hardware y otros recursos.

Con esto, puede ahorrarse dinero y espacio al no ser necesario adquirir equipo que tenga funciones que puedan ser aprovechadas por un conjunto de computadoras, como lo son las impresoras, los scanners, los plotters, dispositivos de almacenamiento, etc. Con estas redes, pueden compartirse recursos como éstos sin sacrificar dinero y espacio, que en ocasiones puede ser un factor importante para tomar decisiones.

Otra de las ventajas de las redes de computadoras es que permiten que varios usuarios utilicen algún programa de aplicación en común o que trabajen en conjunto con otros usuarios en un proyecto determinado sin tener que estar físicamente en el mismo lugar. En este caso pueden ser necesarios recursos de software o de hardware que sean indispensables para el desarrollo del proyecto y siendo compartidos pueden aprovecharse de una mejor manera.

Para que las redes de computadoras puedan trabajar eficientemente o realizar todas las labores para las que fueron diseñadas es necesario establecer algunos estándares con los cuales se definan ciertos parámetros bajo los cuales operarán y que gracias a éstos será posible no sólo la comunicación entre los equipos, sino el "entendimiento" entre los mismos.

Algunos de estos estándares son aceptados mundialmente, ya que fueron aprobados (o propuestos) por instituciones u organizaciones internacionales, éstos son aceptados ampliamente por un mayor número de personas y fabricantes en el mundo, pero también existen estándares más cerrados que solamente son aceptados por una compañía y que se aplican solamente a su software o hardware.

Uno de estos estándares (el más conocido referente a redes de computadoras) es el modelo de red OSI (*Open Systems Interconnection*) y fué establecido por la ISO que es la International Organization of Standarization. Ésta es una organización internacional establecida en 1947 que tiene injerencia en 130 países y propone estándares en materia científica, tecnológica, económica y de otros tipos.

Otra de las organizaciones que tienen importancia reconocida por la emisión de recomendaciones relativas a las redes de computadoras es el ITU-T (*International Telecommunication Union*).

El IETF (*Internet Engineering Task Force*) es el encargado actual de establecer los estándares de comunicaciones para internet. El IETF emite sus recomendaciones a través de RFC (*Requests for Comments*), que pueden ser consultadas por diferentes medios. En estos documentos se describen las características y fundamentos de algunos conceptos o

especificaciones que se usan en las redes de comunicación. Así, tenemos que el RFC del protocolo IP es el 791, para el ICMP es el 792, para el TCP es el 793 y para IPv6 es el 1752.

Muchas de las aplicaciones actuales demandan una mayor cantidad o calidad de los recursos a las redes que se encuentran instaladas actualmente. La capacidad de las redes actuales está siendo insuficiente para soportar aplicaciones en las que se requiere mayor ancho de banda, calidad de servicio (que es un concepto relativamente nuevo en donde se considera la capacidad de decisión para la optimización de los recursos que se usan para cada aplicación), velocidad de transmisión, almacenamiento y transferencia de grandes cantidades de datos, transmisiones de audio y video, videoconferencia, supercómputo y otros.

Como alternativa de solución a los problemas de insuficiencia que se tienen actualmente ha surgido lo que se conoce como Internet2 que es un modelo de red experimental en donde se puede tener acceso a recursos con características como las que se mencionan en el párrafo anterior.

Se ha dicho que la solución de estos problemas tiene que estar en la base de la estructura de la red misma. Es por eso que se ha considerado trabajar (y dar prioridad) al estudio y desarrollo del protocolo de comunicaciones con el que se llevan a cabo las operaciones de las redes.

Por todo esto es que se fundamentan los trabajos sobre IPv6 para establecer el estándar que solucione problemas existentes, que abra posibilidades para desarrollo y crecimiento futuro en las comunicaciones y sobre todo (siendo indispensable), que permita el trabajo simultáneo y la coexistencia con la versión anterior.

Conceptos básicos

Puede considerarse una red de computadoras cuando se tienen dos o más computadoras conectadas y que pueden compartir recursos de hardware o software.

Los recursos de hardware que pueden compartir son muy variados y de diferentes tipos y características. Un ejemplo muy común es el que se tiene con las impresoras compartidas, en este caso se puede tener un conjunto de computadoras conectadas en red permitiendo que compartan una impresora, de esta forma, cada una de ellas podrá emplearla.

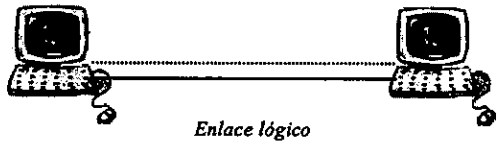
Para tener en funcionamiento una red se necesitan elementos de hardware pero de software también. Cuando se habla se hardware enfocado hacia las redes se puede oír de cable de red, que es uno de los medios físicos por el cual se transmite la información; de las tarjetas de red, que es el medio para conectar un equipo a la red; los dispositivos de interconexión de redes como concentradores, repetidores, puentes, ruteadores y gateways; el sistema operativo de red que permite que se establezca la comunicación lógica entre los dispositivos, así como la transferencia de información a través de la red y finalmente el software de aplicación que se tiene para el usuario final de la red. Todos estos elementos

determinan las características y capacidades de la red (individualmente o en conjunto) como son su velocidad, su confiabilidad, su capacidad de operación, etc.

Como se ha dicho anteriormente, para que los dispositivos que se encuentran conectados a la red puedan comunicarse es necesario respetar ciertos estándares que se han definido; así, los dispositivos de hardware y el software que se empleen deben cumplir con los mismos.

Uno de estos estándares que permiten la comunicación y que es de suma importancia es el protocolo de comunicación que se usa entre los dispositivos.

El protocolo de comunicación de una red es el conjunto de reglas que establecen la forma en la que se realizara el intercambio de información, definen la secuencia en la que se realizará la transmisión, el formato de los datos que serán enviados por la red, la forma en la que se revisará el contenido de los mismos, la forma de corregirán errores que ocurran durante la transferencia (si es que ocurren), el tamaño de los paquetes, etc. El protocolo es el medio "lógico" que permite que las computadoras puedan comunicarse y sobre todo "entenderse".



Enlace lógico

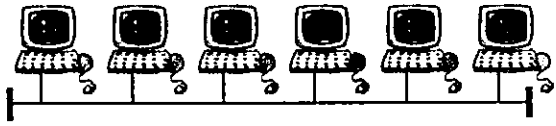
El protocolo de comunicación es el que define como se establece, mantiene y termina la comunicación entre los dispositivos de la red. Por ello, debe tener una forma de identificación para los dispositivos, esto es lo que se llama o se conoce como *dirección IP*. La dirección IP es un identificador único para cada dispositivo de la red.

La forma en la que se designan estas direcciones varía dependiendo de la naturaleza del dispositivo. Algunos de ellos tienen la capacidad de tomar una dirección IP asignada por el proveedor del servicio de acceso a la red y que por lo tanto puede variar dependiendo de la disponibilidad de las direcciones con las que cuente en el momento de la solicitud de conexión, este puede ser el caso de los equipos móviles; mientras que otros dispositivos tienen asignada una dirección IP que debe mantenerse fija.

Existen diferentes configuraciones de redes de computadoras que se definen y toman sus nombres y características de acuerdo a la forma en la que se encuentran dispuestas física y lógicamente. A estas configuraciones se les conoce como topologías.

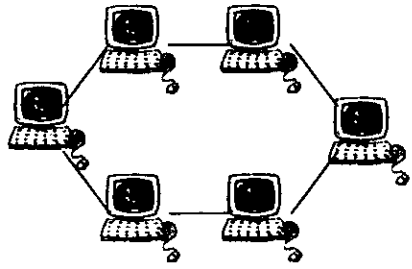
Una red de computadoras puede tener una topología física determinada y al mismo tiempo tener una topología lógica diferente. Las principales topologías de redes definidas son:

- Bus. La característica principal de esta topología es que las computadoras que componen esta red se encuentran conectadas por medio de un solo cable con dos elementos "terminadores" de línea en los extremos del mismo. En esta red, los paquetes de información se transmiten a través del cable en intervalos de tiempo determinados que se han especificado para evitar colisiones entre los mismos y así la pérdida de la información.



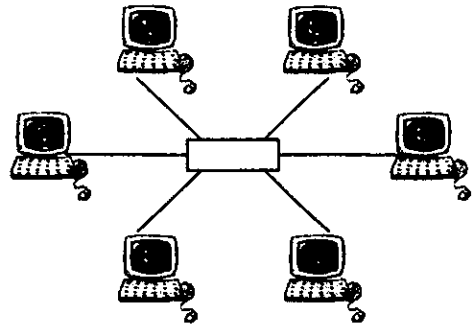
Topología de bus

- Anillo. Esta topología está formada por un conjunto de computadoras conectadas formando un anillo, es decir, la conexión se realiza entre pares de computadoras solamente. La comunicación entre ellas se realiza de forma que los paquetes que se envían a través de la red recorren el anillo hasta encontrar el dispositivo destino. Cuando se envían datos, se tiene que ver que la línea se encuentre "desocupada", para evitar colisiones de paquetes y pérdida de información por esta causa. Después, el paquete recorre la red y los equipos conectados verifican su procedencia y así descartan la posibilidad de tomarlo y de esta manera continúa hasta llegar al equipo para el cual fue enviado.



Topología de anillo

- Estrella. Esta topología requiere de un dispositivo específicamente para el direccionamiento de los paquetes a través de la red. Este dispositivo se encarga de la interconexión y permite la comunicación de las terminales que se encuentran en la red.



Topología de estrella

Cada topología tiene características específicas que la hace diferente a las demás. Estas características deben tomarse en cuenta cuando se diseña la red para poder elegir la topología que se implementará de acuerdo a las necesidades y a las limitaciones que se tengan.

Otra forma de identificar a las redes es por medio de asignaciones en las que se toma en cuenta la distancia que existe entre los dispositivos; así, pueden distinguirse las redes PAN *(Personal Area Network)* conocidas también como bluetooth, las LAN *(Local Area Network)* y las WAN *(Wide Area Network)*. Las primeras son las que se forman cuando la distancia entre el equipo es pequeña, aproximadamente 10 metros, emplean ondas de radio para sus comunicaciones a una frecuencia similar a las microondas; las LAN son redes que se forman con dispositivos ubicados en un área relativamente pequeña como universidades y empresas por ejemplo, cuentan con un gran número de equipos conectados como servidores, impresoras, computadoras, etc.; las WAN son redes de área amplia, por lo que conectan equipos que se encuentran separados por grandes distancias.

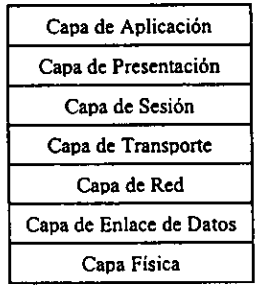
Aunque se tienen estas definiciones para identificar de una manera específica las redes de computadoras, debe decirse que todas las redes, se pueden interconectar entre si (y de hecho así es como se hace), de esta manera, podemos decir que las definiciones de redes que se acaban de dar pueden ser identificadas como topologías virtuales, ya que generalmente una red personal puede ser una subred de una red local y esta a su vez, una subred de una red de área amplia. Son conceptos que pueden ser útiles cuando se trata el caso de una manera más local para identificar problemas y satisfacer necesidades de los usuarios.

Lo importante de todo esto es que se mantengan y respeten los estándares para que entre todos los tipos de red pueda realizarse la comunicación; de otra manera, podrá existir un enlace físico entre varias redes de computadoras a través de los gateways y otros dispositivos, pero si no se toman en cuenta características lógicas como lo son los protocolos, la comunicación será imposible.

Para permitir que varias redes puedan comunicarse entre sí aunque tengan características diferentes se han establecido algunos estándares que definen el comportamiento de la red. Uno de ellos, el mas importante y el que define el modelo actual de red es el modelo de referencia OSI *(Open Systems Interconection)* propuesto por la ISO *(International Standard Organization)*.

La ISO, publicó en 1977 este modelo de referencia, con el cual se establecen las bases para la comunicación entre dispositivos, en él se consideran diferentes funciones que se han agrupado en secciones llamadas capas, en donde cada una de ellas está relacionada de alguna manera con las demás; así, el trabajo en conjunto eficiente es el que determina el éxito o fracaso de la comunicación.

El modelo de red OSI se compone de siete capas que tienen funciones específicas que en conjunto permiten que la comunicación efectiva entre equipos pueda realizarse. Algunas de ellas requieren del trabajo eficiente de otras, tal es el caso de las capas superiores que requieren que se realice eficientemente el trabajo de las capas inferiores para poder trabajar. El modelo se representa gráficamente de la siguiente manera:



Modelo de red OSI

En el esquema se puede ver la relación que existe entre capas, las superiores requieren de las inferiores, de tal manera que la capa superior (la capa de aplicación) depende de las otras 6.

Para que la comunicación entre dispositivos pueda realizarse es necesario que en ellos se respete el modelo y los parámetros que define.

El funcionamiento (abstracto) que se realiza por este modelo debe llevarse a cabo en el receptor y en el transmisor. Así podemos decir que cada capa se comunica con su similar en el extremo opuesto del sistema de comunicaciones sin olvidar que trabajan en conjunto.

El orden en el que se hace referencia a ellas es en sentido vertical de abajo hacia arriba; es por eso que se dice que la capa de aplicación requiere del trabajo de las otras seis. Las

funciones de cada una de ellas son especializadas por lo que solamente se comentarán algunas de ellas.

Capa Física. Es la que se relaciona directamente con los componentes físicos del sistema de comunicaciones, en ella se tratan las características de hardware básico con las que debe funcionar el equipo conectado a la red de computadoras. Define estándares como son los voltajes, la corriente, el tipo de cable, las tarjetas de red, concentradores, etc.

Capa de Enlace de Datos. En ella es en donde se trata la forma en la que se realizará el intercambio de información a través de la red. Define el modo en el que se enviarán los datos, la sincronización entre los equipos, así como la forma en la que se detectará la existencia de errores en la transmisión y por lo tanto, también de la corrección de estos errores.

Capa de Red. Especifica la interfaz con la cual los equipos estarán interconectados y la forma en la cual se realizará la conmutación entre los paquetes que se envíen a través de la red.

Capa de Transporte. Esta es una capa que relaciona la interfaz de comunicación de la red con las capas superiores del modelo. Es en donde se hace evidente la transición de hardware a software. Aquí es en donde se trata la optimización de rutas para la comunicación.

Capa de Sesión. Es la que se encarga de mantener y monitorear la comunicación entre los dispositivos conectados a la red. Es la que define la forma en la que la comunicación se lleva a cabo. Es aquí en donde se especifican los protocolos de comunicación que se emplean y establecen los parámetros con los cuales tiene que realizarse la comunicación entre los dispositivos.

Capa de Presentación. En esta capa es en donde se hace la revisión de la sintaxis de los datos que se han transmitido o recibido. Para poder realizar el trabajo del parser (analizador sintáctico) requiere de una serie de tablas con las que verifica y valida la información.

Capa de Aplicación. Es la capa en la que el usuario de la red ejecuta sus aplicaciones. Estas pueden ser aplicaciones específicas para una tarea. Se requiere (en la mayoría de los casos) que estas aplicaciones sean versiones de software para ejecución en red.

En cada una de estas capas se realizan funciones específicas con las que los datos se adecuan para ser transmitidos y recibidos. Las funciones que se realizan son básicamente de control para poder detectar errores en la transmisión, pérdida de datos durante la misma, corrección para estos errores, etc.

Durante el proceso se añaden datos que permiten este control de la transmisión. Este proceso puede llamarse encapsulamiento ya que los datos (que es lo que quiere enviarse específicamente) son cubiertos o encapsulados por datos de control que añaden cabeceras o segmentos de información que se usan como parámetros de control.

En cuanto a la recepción de los datos, se realiza un proceso que es inverso al anterior; es decir, aquí se lleva a cabo un desencapsulamiento en donde se aíslan los datos de las cabeceras que se han añadido y por lo tanto se "desencapsulan", mientras que son evaluados los datos de control para comprobar que la recepción se realizó de manera correcta y si no es el caso, ejecutar funciones de corrección de errores.

De esta manera es como se puede establecer la comunicación entre programas de aplicación (que se encuentran en el nivel superior del modelo de red).

Existe un programa en el que intervienen los gobiernos de Estados Unidos y del Reino Unido que se ha creado para promover el estándar OSI y su aplicación, se le conoce como GOSIP (*Government Open Systems Interconnection Profile)*. Su trabajo, además de promover el modelo de referencia OSI, es actualizarlo.

Protocolos de comunicación

Un protocolo de red es un conjunto de normas y reglas que definen la forma en la que se llevará a cabo la comunicación entre los dispositivos conectados a la red. Es entonces uno de los intentos por estandarizar y permitir la coexistencia y trabajo conjunto entre aplicaciones en la red.

Dos de los principales protocolos que se conocen y que se usan generalmente son el TCP y el IP. Es conveniente mencionar que estos no son los únicos dos protocolos que intervienen en la comunicación entre los dispositivos de la red, son los que se mencionan con más frecuencia, pero es necesario utilizar una serie de diferentes protocolos que complementan su trabajo.

En esta sección se mencionarán algunos de los protocolos que intervienen en el proceso de comunicación entre computadoras, enfatizando en el TCP y aún más en el IP que es el que nos sirve de base para la realización de esta tesis. En un capítulo posterior se tomará la nueva versión de este protocolo y será entonces en donde se muestren e indiquen las diferencias que fundamentan su transición.

El TCP es el Protocolo de Control de Transmisión (*Transfer Control Protocol*) y se encuentra específicamente en la capa de transporte del modelo de red OSI. Este protocolo se encuentra en un nivel superior al de IP en el modelo de red, por lo que realiza algunas funciones que el IP no, de esta manera se complementan para poder trabajar y permitir las transmisiones de información de una forma correcta. El TCP se encuentra en las computadoras de usuario, aunque es un protocolo de comunicación, no se encuentra en otros dispositivos como los puentes o gateways. La definición formal de este protocolo se encuentra en la recomendación 793 del IETF. Este es un protocolo orientado a conexión, por lo que se encarga de los procesos necesarios para establecer, mantener y terminar la conexión entre las computadoras que intervienen en la transmisión.

El UDP es el Protocolo de Datagramas de Usuario *(User Datagram Protocol)*. Puede utilizarse en lugar del TCP en algunas ocasiones, en aquellas en donde no sea indispensable

el control de errores en la transmisión ya que no cuenta con los servicios que éste último ofrece.

El IP es el Protocolo Internet (*Internet Protocol*) que trabaja en conjunto con el TCP para poder realizar la transferencia de información a través de la red; se encuentra en la capa de red y tiene datos que determinan la ruta que siguen los datos. Para poder establecer esta ruta, el IP realiza una consulta a las tablas de direccionamiento (uno de los problemas del IPv4 es el agotamiento de las direcciones definidas en estas tablas).

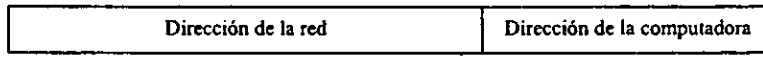
El RFC que corresponde al IP es el 791 y se publicó en septiembre de 1981 por el IETF, en este documento se tienen las especificaciones internacionales que se consideran como estándar en cuestión de este protocolo. Esta recomendación es emitida por el IETF aunque fue formulada por la DARPA (*Defense Advanced Research Projects Agency*). En esta recomendación se dice que las dos principales funciones que tiene el IP son las de direccionamiento y fragmentación, esto es lo que permite que los datagramas lleguen a su destino y que puedan ser fragmentados para ser enviados a través de la línea.

La versión actual (que se usa comúnmente) del protocolo internet es la 4, por lo que se le ha llamado y se le conoce como IPv4, de la misma manera, la versión más reciente se conoce como IPv6 *(Internet Protocol versión 6)*, o IPng *(Internet Protocol next generation)*. Las dos son versiones de un mismo protocolo, pero una es más reciente y por consecuencia tiene muchas ventajas sobre la anterior. En esta sección se tratarán conceptos sobre IPv4 para establecer las bases de análisis de la última versión en un capítulo posterior.

Es común escuchar de TCP/IP cuando se habla de protocolos de redes. Este término surge por la forma en la que se procesan los datos antes de ser enviados por la red. De acuerdo al modelo de red OSI, en cada una de las capas se realizan funciones específicas que permiten que los datos sean enviados y recibidos correctamente por los equipos de la red. Así, el TCP se encuentra en una capa superior al IP (cabe aclarar que la designación de TCP/IP se da por la secuencia en la que se realizan las funciones de cada capa del modelo de red en el transmisor), por lo que los datos a ser enviados pasan primero por las funciones del TCP para adecuarlos añadiéndoles una cabecera que servirá para el control de la transmisión y finalmente pasan por el IP que también se encarga de realizar algunas funciones sobre los datos que ha recibido (que ahora incluyen los datos que recibió el TCP y el encabezado que le fue añadido).

Finalmente, después de haber pasado por los dos se ha formado una trama que se define con el término TCP/IP. Es aquí en donde se puede ver claramente que los dos funcionan en conjunto.

El TCP/IP identifica a las computadoras y a las redes, para ello utiliza identificadores de 32 bits. Estos identificadores son los que se conocen como direcciones IP. La estructura de una dirección IP se compone de la concatenación de la dirección de red y la dirección de la computadora. El siguiente diagrama muestra gráficamente la estructura básica de la dirección IP.



Estructura de direcciones IP

Existen tres clases básicas de redes que se definen de acuerdo con el tamaño del campo de direcciones de la red. Por lo tanto, el tamaño de la red, se define con el número de bits que se tienen en cada uno de los campos. Aquí se muestran las características con las que se definen las tres principales clases de redes:

Red	Bits en el campo de dirección de red	Bits en el campo de dirección de la computadora	Número de redes	Número de computadoras
Clase A	7	24	2^7	2^{24}
Clase B	14	16	2^{14}	2^{16}
Clase C	21	8	2^{21}	2^8

Principales clases de redes

Una de las características principales del IP es que es un protocolo no orientado a conexión por lo que no pueden perderse datos si se interrumpe la transmisión. Para poder emplearlo, es necesario que se use en los dos dispositivos de la red un protocolo común de transporte que generalmente es el TCP.

La fragmentación es otra de las características con las que cuenta, gracias a ella es que los paquetes de información pueden enviarse a través de la red y llegar a las subredes aunque éstas tengan una naturaleza diferente evitando así el uso de recursos extra en los gateways quienes realizarían este trabajo en caso de que el IP no lo hiciera. Para ello, el IP proporciona reglas de fragmentación y reconstrucción de paquetes.

IP es un protocolo de datagrama por lo que no proporciona una manera de detección y corrección de errores, éstos los realiza el TCP. De esta manera, pueden recibirse datos en un orden diferente al que fueron enviados, duplicados de paquetes, o incluso perderse.

Para poder realizar estas funciones de monitoreo de la transmisión, el IP se acompaña con un módulo que se conoce como IMCP (*Internet Control Messages Protocol*). Éste es un protocolo de control y reporte de mensajes que ocurren durante la transmisión. Los datos de este protocolo se añaden en la parte de usuario, que es en donde se ponen los datos en el datagrama; consta de varios campos para la realización de sus funciones, entre los principales está el de código de error que es en donde se describe o se define el tipo de error que se encuentra durante la transmisión.

Los errores que pueden surgir cuando el datagrama se envía son la pérdida de información, lo que puede provocar que el mensaje no sea comprendido por el destino; que haya

excedido el tiempo de vida del datagrama; que se tenga una dirección inalcanzable como destino, esto puede suceder cuando el extremo no está disponible; en algunos casos, este protocolo puede provocar una redireccion del datagrama dependiendo de las condiciones que se tengan, para evaluar estas condiciones se envía una señal de eco, con la que se evalúa la señal que se recibe al regreso.

El datagrama es el resultado de encapsular la información que se transmite y añadir cabeceras (que son datos de control) en los diferentes niveles del modelo de red. Para tener un buen entendimiento del datagrama de IP se muestra a continuación de manera esquemática una representación en donde se incluyen los componentes del mismo seguidos de una descripción para cada uno de ellos.

<table>
<tr><td>Versión</td><td>IHL</td><td>Tipo de servicio</td><td colspan="2">Longitud total</td></tr>
<tr><td colspan="3">Identificador</td><td>Banderas</td><td>Desplazamiento de fragmentación</td></tr>
<tr><td colspan="2">Tiempo de vida</td><td>Protocolo</td><td colspan="2">Checksum de cabecera</td></tr>
<tr><td colspan="5">Dirección de la fuente</td></tr>
<tr><td colspan="5">Dirección del destino</td></tr>
<tr><td colspan="4">Opciones</td><td>Relleno</td></tr>
<tr><td colspan="5">Datos</td></tr>
</table>

Encabezado de IPv4

El datagrama de IP consta de 12 campos fijos y 2 campos opcionales en donde se tiene información que se utiliza para identificar el datagrama, poder relacionarlo con los que llegan antes y después en la transmisión de la información, poder determinar si existió algún error durante la misma y si es el caso corregirlo.

Como ya se ha dicho, los datos que se envían son cubiertos por otros datos que encapsulan a los primeros, de esta manera, cada campo del datagrama de IP es añadido para realizar una función de control específica. Debido a que se realiza un encapsulamiento, los datos que se mandan se encuentran modificados por las cabeceras de control; de esta manera, cuando se recibe el datagrama tienen que irse "retirando" las cabeceras una a una hasta que finalmente se tienen los datos que se necesitan.

El primer campo es en de *versión*, tiene 4 bits; en el se incluye la versión de IP que se utiliza durante la transmisión.

El campo IHL es el campo de *longitud de cabecera* es en donde se indica el tamaño de la cabecera del datagrama, tiene 4 bits.

El campo de *Tipo de Servicio* se usa para determinar el tipo de servicio que será necesario para los datos que se envían, consta de 8 bits.

Longitud Total tiene 16 bits; es el campo que determina la longitud total del datagrama (incluyendo la cabecera), para determinar la longitud de los datos, IP obtiene la diferencia entre este campo y el de longitud de cabecera. La longitud máxima de este campo es de 2^{16}.

El campo *Identificador* se utiliza para relacionar los fragmentos que provienen de un datagrama original. Gracias a esto puede saberse la secuencia en la que tienen que ir, y además relaciona la dirección de la fuente con los mismos consta de 16 bits.

El campo de *Banderas* es el que determina si el datagrama puede fragmentarse o no, en él se tienen varios bits que se utilizan como banderas y que de esta manera determinan la naturaleza y capacidad de fragmentación del datagrama en cuestión; consta de 3 bits.

Desplazamiento de Fragmentación es el campo en el que se incluye un valor que determina el desplazamiento que tiene el fragmento del datagrama en el datagrama mismo; es decir, indica cual es la ubicación del fragmento dentro del datagrama; tiene 13 bits.

Tiempo de Vida es el campo que determina el tiempo que el datagrama lleva en la red. En algunos casos, este campo se decrementa cada vez que el datagrama pasa por un gateway, en otros se decrementa en intervalos de tiempo determinados; consta de 8 bits.

Protocolo es el que identifica cual es el protocolo que le sigue al IP en la estructura de niveles que va a recibir el datagrama, frecuentemente se trata del TCP. Este campo es de 8 bits.

El campo de *Checksum de la cabecera* detecta la ocurrencia de errores en la cabecera durante la transmisión, no se ocupa de la integridad de los datos, esto se lo deja al protocolo de nivel superior y solamente verifica si esto ocurrió o no; consta de 16 bits.

Los campos de *Dirección de la Fuente* y *Dirección del Destino* son los que contienen la dirección IP de la computadora que transmite la información y de la que la recibe, por lo tanto con estos dos campos pueden identificarse los extremos de la transmisión. Cuando ocurre algún error durante la misma, se consultan estos campos para restablecer la conexión, reenviar el datagrama o corregir de alguna manera el error que haya surgido. Estos dos campos cuentan con 32 bits.

El campo de *Opciones* indica si existen opciones especiales para la identificación del datagrama, este campo no existe en todos los datagramas ya que solamente es incluido cuando se necesita información adicional. La longitud de este campo es variable.

El campo de *Relleno* verifica que la cabecera del datagrama este alineada con una división de intervalo de 32 bits. Esto es para que se respete la secuencia que se debe llevar en la transmisión de datagramas; así, los datagramas deben ser enviados en secuencias de 32 bits solamente. Su longitud es variable.

El último campo es el de *Datos* que es en donde se encuentran los datos del usuario, ahí es en donde van los datos que se quieren enviar o recibir.

El protocolo IP realiza diferentes funciones para que los datagramas lleguen a su destino, para ello realiza consultas a tablas de direcciones y con las mismas determina la ruta óptima hacia su destino.

Generalmente, la ruta más corta se considera como la más óptima entre los dos puntos extremos de la comunicación. Los gateways mantienen las tablas de direcciones de las computadoras o dispositivos conectados a la red, por ello pueden ser estáticas o dinámicas, aunque generalmente son dinámicas ya que la ruta que puedan tomar los datagramas depende de la disponibilidad de los equipos en el momento de la transmisión.

Estas tablas se forman con las direcciones de los dispositivos que se encuentran conectados a la red; para mantenerlas actualizadas, se usan algunos protocolos que se encargan de realizar algunas funciones como la selección de la ruta óptima que generalmente es aquella en donde los puntos extremos de la comunicación se enlazan por la menor cantidad de dispositivos.

En redes LAN se tiene por ejemplo el protocolo de gateway interno (IGP) que es el que lleva la relación de las direcciones de los dispositivos de su red correspondiente en particular; pero como hemos dicho antes, las redes están interconectadas y forman entre si una gran red que enlaza una gran cantidad de dispositivos en el mundo, como lo es Internet.

Para poder comunicar a las subredes entre si se necesitan de dispositivos especiales que realicen esta función; así es como se tiene a los gateways externos que se encargan de establecer la conexión entre subredes. En este caso se usa el protocolo de gateway externo (EGP) que se encarga de mantener las direcciones de los equipos y dispositivos que pertenecen a redes externas.

Un protocolo que trabaja en conjunto con el UDP es el RIP *(Route Information Protocol)* que es el protocolo que mantiene la información de ruteo entre los dispositivos. Esta información la obtiene de la consulta a las tablas de direccionamiento y consideración del número de saltos que deben realizarse entre los dispositivos, por lo que obtiene la mejor ruta entre los mismos considerando la distancia menor entre ellos.

Por último mencionaremos solamente algunos protocolos de nivel de aplicación. Éstos son los que se conocen por las funciones que realizan, en ocasiones se les identifica como aplicaciones de red y no como protocolos como lo son. *Telnet* es un protocolo para servicios entre terminales; *TFTP* es el protocolo trivial de transferencia de archivos; *FTP* que es el protocolo de transferencia de archivos y el *SMTP* que es el protocolo simple de transferencia de mensajes que se usa para el correo electrónico.

INTERNET

Introducción

Las redes de computadoras se han extendido por todo el mundo rápidamente gracias a la utilidad y a las facilidades que representan para las personas. Desde la primera red que comenzó siendo prácticamente de dos equipos ha evolucionado y crecido hasta envolver a prácticamente el mundo entero. Las pequeñas redes locales que se encuentran en instituciones educativas, corporaciones comerciales, etc., se unen para formar una gran red, que se le llama (por la forma en la que se constituye) la red de redes que conocemos como Internet.

Internet es una gran red que cubre casi a todo el mundo aprovechando las potencialidades de las redes que se encuentran dispersas; de esta manera, podemos ver que nunca fue concebida para ser lo que ahora es. Ha ido creciendo sin ninguna planeación y sin nadie que supervise su funcionamiento.

No existe ninguna empresa, ninguna organización, ninguna asociación ni grupo de personas que pueda decir que es "dueña" de Internet; no hay nadie que pueda acreditarse el funcionamiento de la misma, no hay nadie a quien se le pueda reclamar por el mal funcionamiento, por las fallas, por los retrasos o errores de la misma.

Cualquier persona puede estar conectado a Internet y para hacerlo no necesita cubrir una serie de requisitos especiales, solamente es necesario un equipo de cómputo (no muy demandante por cierto) y una línea telefónica. El primer paso es la contratación del servicio de acceso, no de Internet; este tiene que hacerse con uno de los proveedores del servicio a los que comúnmente se les conoce como ISP's *(Internet Service Provider)*. Como lo dice su nombre, estas compañías solamente proporcionan el servicio de acceso y pueden encontrarse en casi cualquier lugar en donde hay compañías telefónicas, algunas de ellas incluso lo hacen.

El contenido de Internet tampoco es supervisado por nadie, cualquier persona puede poner a disposición del mundo literalmente lo que quiera, sea o no sea cierto, sirva o no, tenga el carácter que tenga. Han habido varios intentos por evaluar y restringir el contenido de la información en Internet, pero todos éstos han fallado. Uno de los ejemplos más conocidos es el que surgió en Estados Unidos en los 90's. Pero de la misma manera ha habido campañas para mantener la libertad e independencia de la red.

Internet, es lo que es gracias a su contenido, gracias a la posibilidad de encontrar y publicar prácticamente lo que sea. Es una red libre en la que cualquier persona puede mantener la identidad que quiera (en la mayoría de los casos). Es un medio de comunicación que se ha vuelto indispensable para muchas personas de manera personal e institucional, Internet ha cambiado la forma de comunicación entre ellas hasta el punto de hacer obsoletos otros medios.

Internet es la red de redes que cubre al mundo y lo ha hecho cambiar radicalmente en muchos aspectos. En el presente capítulo hablaremos de algunas de sus características y de la forma en que ha llegado a ser lo que ahora es.

Características

Internet es una red inmensa, pero una red. Absorbe a una infinidad de redes de todos los tamaños. Cada una de estas redes tiene características específicas que la hacen diferente de las demás, pero para poder conectarse a Internet tiene que cumplir con ciertos requisitos indispensables para establecer la comunicación entre los equipos. Estos requisitos o normas son los que se han definido para Internet y que han sido adoptados por un gran número de personas y ya se han establecido como estándares.

Dada la naturaleza de Internet, es posible conectar una gran variedad de dispositivos a la misma, prácticamente cualquier dispositivo que se encuentra conectado a una computadora con acceso a Internet puede ser accesado desde la red. Todos estos dispositivos cuentan con características en común que los hacen tener la capacidad de comunicación a través de la red.

No todos los dispositivos cuentan con componentes de software que les permitan integrarse a la red, estos son los controladores o drivers, que en algunos casos, como cuando se usan sistemas operativos que no son comerciales y por lo tanto, no cuentan con el apoyo de grandes compañías de software, suele ser difícil encontrarlos o desarrollarlos.

Una parte importante para el acceso a Internet es el sistema operativo que se utiliza como plataforma para el uso de las aplicaciones de la red. Existen varios sistemas operativos que cuentan con funciones específicas para redes y por lo mismo, para Internet. La elección del sistema operativo que se use depende de algunas consideraciones como la capacidad del mismo, los requerimientos de hardware y software, el valor comercial y la compatibilidad de programas de aplicación que a fin de cuentas se usarán en la vida diaria, para el trabajo común y no solamente en la red.

Hay varias marcas comerciales y otras de software libre de sistemas operativos con los que se tiene acceso a Internet. De esto hay que tener alguna consideración, ya que cuando se trata lo de la ayuda en línea, se necesita algún tipo de apoyo por el fabricante, puede ser fácil obtenerlo o mucho muy difícil. Esto se puede ver si comparamos dos sistemas operativos que se encuentran actualmente con un gran número de usuarios, Microsoft Windows y Linux.

Por un lado, el primero de ellos está respaldado por una de las compañías de software más grandes del mundo, es un software comercial, hay que pagar por él una cantidad que depende de la versión que se desee; cuando se necesita apoyo técnico solamente hay que buscarlo y con relativa facilidad se encontrará y los problemas podrán resolverse en un tiempo determinado o por lo menos se sabrá que es algo que no puede resolverse. Frecuentemente se liberan nuevas versiones que incluyen correcciones a los errores de versiones anteriores y en donde se encuentran nuevas aplicaciones u optimizaciones a las que ya existían.

Por el otro lado, cuando se trata de software libre, como es el caso de Linux, no hay apoyo directo del fabricante, en la mayoría de los casos, ya que existen distribuciones que se han hecho comerciales y que ahora se distribuyen como tales por algunas empresas de software

que han añadido algunas cosas al sistema. Linux sigue siendo un sistema operativo que se mantiene principalmente por particulares, por hackers específicamente. Ha sido un esfuerzo que surgió a principios de los 90's como un proyecto personal y se ha extendido a una gran cantidad de usuarios y desarrolladores en todo el mundo. El apoyo que tiene se ha hecho cada vez más fuerte por el número de personas que lo manejan y desarrollan aplicaciones actualmente.

Otro sistema operativo que merece ser mencionado por tener una estrecha relación con Linux, es el UNIX. Puede decirse que Linux es una versión "lite" de UNIX, que fue su modelo y que Linux se desarrolló mucho tiempo tratando de imitarlo, pero que con el tiempo logró separar su trayectoria, y aún conserva gran parte de las características con las que nació.

Podemos decir que éstos son dos de los sistemas operativos más comunes con los que se trabaja y por lo tanto, con los que se accesa a Internet, pero veremos a lo largo de esta tesis un análisis mas exhaustivo de Linux, ya que es el enfoque que se tomó como base para la misma. En un capítulo posterior lo abordaremos de una manera más detallada.

El sistema operativo es entonces la base de software que permite a las computadoras la conexión con Internet si nos enfocamos al módulo de redes que tienen, porque no es esencialmente un sistema operativo de red el que lo hace en este caso. Entonces, este módulo del sistema tiene implementadas funciones que le permiten establecer la comunicación con otras computadoras, esto lo hace apegándose a los estándares que se han definido (y explicado en el capítulo anterior) por organizaciones internacionales y que son aceptadas mundialmente.

Cada sistema operativo tiene su forma particular de realizar la configuración del sistema y tiene opciones particulares, pero una de ellas que es común y que es de gran importancia es la de configuración del protocolo de comunicaciones.

La comunicación entre estas computadoras se realiza gracias al protocolo que utilizan. Éste debe encontrarse definido en las computadoras que deseen tener acceso a Internet; el que se usa generalmente es el TCP/IP, que como explicamos anteriormente, es un conjunto de otros protocolos que realizan determinadas funciones con lo que se llega a establecer, mantener, terminar y corregir la transmisión de datos entre los diferentes equipos.

Este protocolo puede seleccionarse entre otros y establecerse como predeterminado para el equipo por medio de la configuración del sistema. Esta configuración dependerá del sistema operativo que se use, así como de la versión del mismo, pero esencialmente contará con un campo de selección del protocolo.

Bueno, éstos son los requisitos básicos del sistema para la conexión del mismo a Internet, pero es indispensable obtener el acceso a través de un proveedor de servicio especializado, que son los ISP's. Estos proveedores ofrecen el servicio de acceso a Internet, a través de sus equipos; o sea que los proveedores tienen un enlace permanente hacia la red y a través de ellos se realiza la conexión de equipos a la misma.

Esta puede realizarse por diferentes medios, el más común es a través de la línea telefónica que se realiza con un módem convencional. La velocidad de la transmisión depende de algunos factores como son el medio por el que se realiza, el módem que se utiliza y la compresión que realiza el software.

Existen algunos términos para identificar la velocidad de transmisión de los equipos, éstos términos se muestran en la siguiente tabla.

Velocidad	Tipo
1200 bps	RTC o Punto a Punto
2400 bps	
4800 bps	
9600 bps	
14.4 Kbps	
19.2 Kbps	
56 Kbps	DS0
64 Kbps	E0
1.544 Mbps	T1
2 Mbps	E1
10 Mbps	Ethernet
45 Mbps	T3
100 Mbps	FDDI

Tipo y velocidad de conexión

Éstos son los tipos y velocidades con las que puede tenerse acceso a Internet y a todo lo que se encuentra en ella. En Internet, como hemos dicho puede encontrarse cualquier tipo de información, en eso no hay restricción, pero también proporciona ciertos servicios que la identifican y hace que la gente la utilice por los mismos aún sin darse cuenta de que está haciendo uso de ellos.

Los servicios básicos que proporciona Internet son:

- Correo electrónico
- Acceso remoto
- Transferencia de archivos
- World Wide Web

No son los únicos servicios que se tienen, pero muchos otros se derivan de éstos y no se encuentran aislados independientemente y por lo tanto no realizan funciones específicas que deban tratarse con más detalle.

Existen muchas aplicaciones en Internet, muchos programas y servicios que pueden encontrarse de diferentes maneras, pero los cuatro servicios básicos son los que se mencionaron anteriormente. Aquí mencionaremos las características de los mismos, así como su funcionamiento básico.

El correo electrónico es uno de los servicios más utilizados por Internet; con él, la gente puede comunicarse rápidamente desde cualquier parte del mundo. Casi cualquier persona que tiene acceso a una computadora conectada a Internet tiene este servicio. Existen muchas formas de obtenerlo, no es necesario pagar por él, ya que existen muchos servidores de correo electrónico que prestan sus servicios de manera gratuita. Solamente es necesario llenar una forma de registro con datos personales.

Cuando se han cubierto los requisitos que solicitan los prestadores del servicio, se asigna una dirección que es única y con la que se identifica al usuario en la red. Esta dirección esta compuesta de dos partes principalmente separadas por un carácter especial que se ha vuelto todo un símbolo representativo de esta generación, el tan conocido @. A continuación se muestra una dirección estándar de correo electrónico:

maurik@servidor.unam.mx

La primera parte, la del lado izquierdo, es la que representa e identifica al usuario de manera única en el servidor de correo al cual pertenece su cuenta, es el destinatario al que se dirigen los correos electrónicos o mail's. La parte derecha, la que se conoce como dominio, es la que identifica a la computadora en donde reside el servicio de correo, está compuesta por identificador de la computadora, que es el nombre que se le asignó y con el cual se le identifica de manera única entre todos los equipos y dispositivos conectados a la red.

La forma en la que se designan los nombres a los servidores de correo es relativamente arbitraria, uno puede poner el nombre que desee a su servidor, pero es conveniente comprobar que no existe otro servidor con el mismo nombre, por lo menos en la misma red. Existen identificadores especiales para organizaciones, instituciones y empresas comerciales; también lo hay para cada país. De esta manera, podemos ver que hay cierta relación entre las direcciones de los servidores, ésta se designa de acuerdo con las funciones que realizan.

En un principio, las redes eran pequeñas y se estableció una convención para las redes que se encontraban en Estados Unidos; así, los dominios que se asignaban describían el tipo de la institución que era dueña del servidor como se muestra en la tabla de dominios por organización.

Dominio	Organización
com	comercial
edu	educativa
gov	gobierno
int	internacional
mil	militar
net	gestión de redes
org	no lucrativa

Dominios por Organización

Posteriormente, cuando las redes siguieron creciendo y fueron uniendo cada vez más computadoras que se encontraban en otros países e incluso en otros continentes, fue necesario establecer una nueva convención de dominios que ahora consideran la ubicación geográfica del servidor, como se muestra en la tabla de dominios geográficos.

Dominio	Organización
at	Austria
ca	Canadá
es	España
fr	Francia
mx	México
jp	Japón
us	Estados Unidos

Dominios Geográficos

La asignación de dominios la hacía la NSI *(Network Solutions Incorporated)*, pero después de una serie de platicas con el Departamento de Comercio de Estados Unidos y el ICANN *(Internet Corporation for Assigned Names and Numbres)* se ha decidido otorgar a otras compañías la asignación de estas direcciones. Esta fue una iniciativa propuesta por la administración del gobierno estadounidense para evitar el monopolio en ese ramo. Desde 1993, la NSI tenía esta función la asignación de los dominios de primer nivel (org, net, com), pero el ICANN será el que se encargue ahora de la regulación de los mismos, actualmente se tienen consideradas 64 compañías para compartir estas funciones con la NSI.

Existen muchos programas de aplicación diseñados para manejar correo electrónico, algunos en ambiente gráfico, otros en línea de comandos, de diferentes compañías de software, para diferentes sistemas operativos, en muchas versiones. Muchos de ellos, la mayoría permite la transferencia de archivos adjuntos con el mensaje, aunque esto se relaciona directamente con otro de los servicios de Internet del que hablaremos mas tarde.

Cabe mencionar que estos programas de aplicación tienen como base un protocolo conocido por sus iniciales como SMTP *(Simple Mail Transport Protocol)*. Este protocolo define la forma en la que se lleva a cabo el intercambio de correos electrónicos, define el formato que deben tener los archivos que son enviados a través de las aplicaciones de correo electrónico específicas. Pertenece al conjunto de protocolos del TCP/IP.

Para ser posible la transferencia de correo electrónico por la red, es necesario que intervengan otros factores que son conocidos como agentes que se encargan de los procesos que se realizan y que permiten el intercambio de esta información.

El más importante de ellos es el agente de transporte, éste es el que se encarga de tomar el mensaje que se ha generado con las especificaciones del SMTP y enviarlo a través de la red. Es el encargado de hacer que llegue hasta su destino y que no encuentre problemas en

la transmisión. Este agente es un programa que se ejecuta en segundo plano, y que atiende a las peticiones de correo que surjan en cualquier instante.

Otro agente que es conocido (aunque transparente) para todos los usuarios, es el agente de usuario. Este es el que proporciona la interfaz entre el usuario y el agente de transporte, es la aplicación de correo electrónico como tal, es la que permite al usuario ejecutar las funciones que se conocen propiamente como correo electrónico. Es el que se encuentra incluido implícitamente en las aplicaciones de correo electrónico y pasa generalmente desapercibido por los usuarios. Como hemos dicho, existen muchos programas de este tipo que proporcionan estos servicios, cada uno tiene características que los hacen diferentes a los demás, pero tienen que respetar y acatar ciertas funciones y formatos que son indispensables para que puedan ser usados como lo que son, agentes de usuario de correo electrónico.

Otro protocolo que interviene en el servicio de correo electrónico (en algunas ocasiones, dependiendo de la configuración personal del sistema) es el POP *(Post Office Protocol)* es el que se encarga de monitorear la existencia de mensajes nuevos en el servidor de correo y cuando detecta su existencia, los baja a la computadora que tiene el usuario en particular, y de esta manera evita el acceso frecuente al mismo.

El formato de un correo electrónico incluye información que lo identifica además de incluir el cuerpo del mensaje que es en realidad, lo que se quiere enviar. Los principales datos que se muestran en un mail son:

- Datos del remitente (dirección electrónica y nombre)
- Datos de el (los) destinatario(s) (dirección electrónica y nombre)
- Fecha en la que se envió
- Encabezado (subject)
- Archivos adjuntos
- Mensaje

Éstos son solo los datos básicos que contiene un mail, en realidad puede tener otros como los nombres de algunos servidores por los que ha pasado antes de llegar a su destino, nombres de otras personas a las que se ha mandado el mismo mensaje, si se trata de un mail contestado (al que se ha hecho replay) o reenviado (usando forward), formato en el que se envió, y otros.

El formato de las direcciones electrónicas fue descrito anteriormente; la fecha no necesita descripción, incluye fecha y hora; en encabezado del mail es un identificador que tiene cierta importancia para reconocer y asignar prioridad o importancia al mail ya que en él se puede colocar información descriptiva del contenido del correo, este es uno de los datos que se muestra en un listado del contenido del buzón de mensajes recibidos; los archivos adjuntos son los que se integran y se mandan en el mail, pueden ser de cualquier tipo, pero la limitación que se tiene está en el tamaño del mismo. Es recomendable que el tamaño de estos archivos sea lo menor posible ya que la velocidad de la transmisión del mail

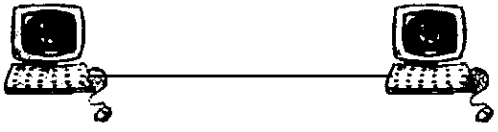
dependerá de esto. Estos archivos no son enviados en el cuerpo del mail; esto es, no se integra su contenido y se envía junto con el mensaje, sino que se envía encapsulado; así se recibe el mensaje (o cuerpo del mail) de forma independiente con una referencia que indica el contenido de un archivo adjunto.

El mensaje es el principal motivo de enviar un mail, es en él en donde se encuentra la información. Contiene generalmente texto o datos que se representan por caracteres alfanuméricos, aunque existen algunos agentes de usuario (o aplicaciones de mail) que permiten incluir imágenes, texto con formato e incluso, hasta audio y animaciones.

Existen normas de "convivencia" que tienen que ver directamente con el sistema de correo electrónico, no son reglas que tengan que cumplirse al pié de la letra pero son consideraciones que hacen que el uso del correo tenga una utilidad y que se use como lo que es, un servicio de Internet.

Dada la naturaleza de la red, existen millones de usuarios con intereses específicos que no siempre pueden ser intereses comunes, por tanto no siempre es bien recibido un correo que tiene información que no se desea; es decir, el uso principal de los mail's es para comunicación personal, con ellos pueden aprovecharse las ventajas de la red y eliminarse tiempos y algunas situaciones que trae consigo el envío físico de un documento. Por todo esto es bueno recordar y considerar que el servicio de correo tiene su razón de ser y tiene que usarse para lo que está hecho y por lo tanto debe evitarse el uso del mismo incluyendo información que no pueda ser solicitada o no deseada.

Ahora, en el caso del servicio de acceso remoto, podemos decir que siendo uno de los principales servicios de Internet, es de los que se usan mas frecuentemente. Cuando se hace un acceso remoto se establece la conexión entre dos computadoras. Se dice que una de ellas es el cliente y el otro el servidor. El cliente es el que solicita el acceso y por lo tanto los servicios que proporciona el servidor.



Cliente/Servidor

Para que esta conexión pueda realizarse, se ejecutan en realidad dos programas, uno en el servidor y el otro en el cliente. El programa del cliente es el que hace las peticiones mientras que el del servidor las satisface. El programa cliente solamente se ejecuta cuando es necesario establecer la conexión y lo ejecuta el usuario, pero el programa servidor es un programa que se encuentra en segundo plano y que siempre está en ejecución, se dice que los programas servidores se mantienen escuchando los puertos para verificar los intentos de conexión.

Hay que aclarar que el programa cliente es uno en particular y depende de la computadora que se este utilizando en ese momento como cliente, depende del sistema operativo que se use, de la versión del mismo, etc.; pero el programa servidor tiene que ser flexible, para poder atender las peticiones de conexión de cualquier cliente que pueda solicitarlo.

Un puerto es un identificador numérico para una aplicación determinada, no se trata de un dispositivo de hardware como podría pensarse. Mediante los puertos, entonces puede identificarse el tipo de servicio que se solicita. Cuando no se hace referencia a un puerto específico cuando se realiza la conexión, se está intentando conectar al puerto que es el que se ha dado por defalut. Un puerto puede atender varios usuarios simultáneamente.

El acceso remoto se realiza comúnmente con un programa llamado telnet. Para poder establecer la conexión es necesario indicar a telnet la dirección o el nombre de la computadora a la cual se desea conectar y si es necesario, el puerto específico que hace referencia al servicio deseado. En muchos casos se dice que se ha establecido una conexión virtual entre el cliente y el servidor.

Mediante el acceso remoto, se establece una conexión entre dos o más computadoras simultáneamente, de esta manera se puede decir que el cliente "entra" al servidor. Esto puede explicarse, porque una vez establecida la comunicación, puede trabajarse en el cliente como si se tratara del servidor.

En casos especiales, puede ser necesario configurar algunos parámetros de manera especial para satisfacer completamente las necesidades del cliente; por ejemplo, cuando se trata de aplicaciones gráficas, es necesario exportar el display del servidor hacia el cliente para que en el monitor de éste último pueda visualizarse de una manera correcta la aplicación.

También puede tenerse acceso total a los archivos y aplicaciones que se encuentran en el servidor (si se tiene una cuenta que lo permita). Puede trabajarse entonces, como si se estuviera físicamente en la consola del servidor estando en cualquier otro lugar.

La transferencia de archivos es otro de los servicios que caracterizan a Internet. Este es un servicio que se ha usado mas que un medio de comunicación como un medio de transporte para cualquier clase de documentos, textos, informes y datos. Finalmente, un archivo puede contener la información que sea, puede ser un texto o datos binarios que pueden ser un programa, una aplicación, una imagen, un vídeo, un archivo de audio o cualquier contenido que pueda ser almacenado electrónicamente.

De la misma forma que con el servicio de acceso remoto, cuando se realiza una transferencia de archivos común, es necesario contar con dos programas, uno que reside en el cliente y otro que se encuentra en el servidor. En este caso, el cliente hace peticiones sobre un archivo específico al servidor, este acepta (o niega) primero la conexión y después, inicia la transferencia.

Para tener acceso a archivos en un servidor determinado puede ser necesario contar con una cuenta en el mismo que permita establecer el acceso de manera parcial o total. Existe la posibilidad de que el servidor que posee el archivo que nos interesa tenga una cuenta de

acceso limitado que se conoce como anónima *(anonymous)*. Con esta cuenta pueden obtenerse archivos que son de dominio público y que no tienen restricción de acceso para esta consulta, sin embargo, no permiten la recepción de archivos; esto es, no reciben archivos.

La transferencia de archivos se realiza principalmente utilizando un protocolo especial que se ha creado para establecer las bases y los parámetros que tienen que cumplirse. Éste es el protocolo de transferencia de archivos, conocido como FTP *(File Transfer Protocol)*. Este es el que permite que puedan ser transferidos archivos por la red, que lleguen a su destino y sobre todo, que sean recibidos en el lugar en el que se necesitan o fueron solicitados.

Esta es la forma más común de referirse a la transferencia de archivos (con FTP) pero no es la única. Actualmente existen programas de aplicación que permiten la inclusión de archivos (binarios y de texto) dentro de los correos electrónicos o como aplicaciones individuales. La idea de los servidores FTP sigue siendo una de las que caracteriza a Internet, sobretodo cuando se habla de FTP anónimo, pero no es la única forma de transferir archivos.

Es necesario aclarar que la forma en que se realizan las trasferencias de archivos tiene que mantener y seguir las convenciones dadas por el FTP, y aunque no se utilice este servicio propiamente, los programas de aplicaciones que se utilicen para ello tienen que usarlo.

Una forma de transferencia de archivos es la que se realiza por medio de lo que se conoce comúnmente como Web y que es el famoso World Wide Web. Basa su funcionamiento (como casi todo lo que hemos visto) en un protocolo, se trata del HTTP (*HyperText Transfer Protocol*) que es el protocolo de transferencia de hipertexto, es el que se encarga de establecer las normas y reglas de formato que deben tener los archivos que son transferidos a través de la red y que pueden ser visualizados por un explorador convencional. El explorador, es un programa de aplicación que permite visualizar documentos que han sido transferidos por la red y que tienen un formato especial. Estos documentos pueden contener cualquier tipo de archivos en ellos, pueden ser audio, vídeo, texto (con formato o sin él), imágenes, animaciones, etc.

Esta es la manera en que mucha gente identifica y conoce Internet. Internet no solamente es un conjunto de documentos en la red que solamente pueden consultarse y que para poder tener una utilidad es necesario encontrar la información deseada empleando alguna de las herramientas que se han creado para esto como son los motores de búsqueda. Internet no es sólo lo que se ve a través del explorador.

El explorador es un programa de aplicación creado para correr sobre un sistema operativo determinado, reside en la computadora que se conecta a la red y que para poder establecer la comunicación con los demás equipos y dispositivos conectados debe estar configurado correctamente tomando en cuenta las normas y recomendaciones que se han establecido para el uso de Internet. Estos programas también son conocidos como navegadores.

En el World Wide Web puede encontrarse fácilmente (y en un ambiente gráfico muy amigable) la información que sea, pueden encontrarse documentos que han sido escritos en

cualquier parte del mundo por cualquier persona; es por eso que tiene que ser considerada la veracidad de lo que se puede encontrar en la red.

El lenguaje que se usa más comúnmente cuando se escriben documentos para publicarse en Internet es el HTML *(HyperText Markup Language)* que es el lenguaje de hipertexto. Es un lenguaje que es interpretado por el explorador que tiene grandes facilidades para la inclusión de datos en formatos específicos. Permite la inclusión de imágenes, de audio y vídeo en sus documentos, así como la capacidad de interacción entre programas de aplicación residentes en la computadora cliente de Internet. Recientemente han aparecido otros lenguajes para la creación de documentos de hipertexto pero con ventajas sobre el html, tal es el caso del XML, XHTML, SHTML y el DHTML. Es necesario recordar que éstos son lenguajes interpretados que muestran un documento con formato en los exploradores, no se trata de lenguajes de programación con los que puedan programarse objetos, eventos o funciones.

Todos éstos permiten la interacción entre el cliente y el servidor. A través de formas en las que se incluyen cuadros de texto, listas con opciones, casillas de verificación y otros objetos es como se puede hacer que el usuario accese a funciones determinadas o secciones en las que se manejen características de privacidad o seguridad especiales.

Pero también podemos encontrar los lenguajes de programación para Internet. Cada uno de ellos tiene características específicas. Pueden clasificarse en dos grandes categorías:

- Los que se ejecutan en el servidor
- Los que se ejecutan en el cliente

Los primeros, son aquellos que corren o se ejecutan en el servidor al cual se está conectado a la red. Estos permiten el uso de archivos, librerías y datos que se encuentran en el servidor mismo que pueden ser indispensables para la ejecución del programa y que probablemente no se tengan en el cliente que está haciendo la solicitud del servicio.

Los segundos, son los que se ejecutan en el cliente permiten cierta flexibilidad porque se ejecutan "fuera de la red", lo que permite mantener menos saturada la línea de comunicación. Estos son programas que no demandan recursos especiales y que pueden encontrarse casi en cualquier computadora con las características suficientes para encontrarse conectada a Internet.

Los lenguajes de programación con los que se hacen estos programas pueden tener grandes diferencias entre sí, pero mantienen en común la característica de correr en la red. Los principales lenguajes que podemos encontrar para aplicaciones de Internet que se usan actualmente son: Java, Java Script, Perl y PHP.

Cada uno tiene sus características y dependiendo de la aplicación que se necesite puede elegirse entre ellos. El primero de ellos, Java, es un lenguaje de programación creado por Sun Microsystems orientado a objetos que se ha vuelto en un estándar de programación para aplicaciones en Internet y ha tomado fama rápidamente además de mantenerse sobre

todos los demás. Su concepción general está en la ejecución de sus programas y aplicaciones sobre una máquina virtual que permite la ejecución en diferentes plataformas sin accesar a los dispositivos, con una administración de la memoria independiente. Es un lenguaje que tiene que ser compilado e interpretado por lo que la ejecución de sus programas no tiene la misma velocidad que programas creados con otros lenguajes. Su máquina virtual permite absoluta seguridad en el equipo del cliente.

Java Script es un lenguaje que comparte muchas características de Java, pero su implementación es por medio de script's que se incluyen en los documentos que se encuentran en Internet, en las páginas precisamente y que responden a las acciones que el usuario realiza al recorrerlas. No tiene la potencia del Java puro ya que es una implementación reducida. Se trata de un lenguaje interpretado y que no requiere de recursos especiales del equipo mas que un explorador con capacidad de ejecución de programas de Java Script. Este lenguaje fue creado por Netscape.

Perl *(Practical Extraction and Reporting Language)* es otro de estos lenguajes, se usa comúnmente cuando se requiere interacción con el usuario a través de CGI's *(Common Gateway Interface)*. Los programas escritos en este lenguaje se ejecutan en el servidor y procesan y/o trabajan con los datos que le son enviados desde el cliente generalmente por medio de una forma. Perl es un lenguaje (como su nombre lo dice) creado pensando en ser la base para extracción de información y creación de reportes, pero poderoso como la mayoría de los lenguajes de programación, dejando el límite al programador solamente.

PHP es el Hipertext PreProcessor, se trata de un lenguaje interpretado que puede incluir código en el formato HTML y que se ejecuta en el servidor.

Estos son algunos de los lenguajes de programación que han servido para el desarrollo de aplicaciones (sólo algunas) que podemos encontrar en Internet. No son los únicos, pero son los más conocidos y de los que se escucha hablar comúnmente.

Existen aplicaciones de muchos tipos, aplicaciones que se han creado para el procesamiento y manejo de información en diferentes formatos. Estas aplicaciones demandan recursos del equipo que interviene en la comunicación, no solamente de una sola computadora, sino de todo el conjunto que interviene, tanto del cliente, del servidor, como de la línea por la que es transmitida la información, como de los dispositivos que se encuentran intermedios entre los extremos de la comunicación; o sea, gateway's, switch's, etc.

Conforme la tecnología ha ido avanzando, hemos visto que las aplicaciones se han hecho más demandantes de recursos. Esta es una de las razones por las cuales se ha pensado (y se trabaja) en una evolución de Internet. Este es un paso lógico que tiene que darse y que conforme va avanzando el tiempo se ve indispensable, entre otras cosas por la cantidad de personas que se encuentran conectados, por los recursos que se tienen que administrar y que tienen que mantener su identidad particular, así como por el avance en las aplicaciones.

Vemos que las aplicaciones actuales en donde se manejan audio y vídeo en tiempo real (por poner solamente un ejemplo) absorben los recursos actuales de la red y del equipo y en muchas ocasiones esto se hace insuficiente y por lo tanto es necesario hacer algunas

consideraciones para que se pueda proporcionar los recursos necesarios ya que algunas de estas aplicaciones pueden ser (o volverse) indispensables y no es posible, por lo tanto prescindir de ellas.

Estas son algunas de las características que considero que son más importantes de Internet, creo que con ellas es con lo que se ha formado, con lo que se ha desarrollado y se ha hecho lo que ahora es. Todos estos elementos trabajan en conjunto o individualmente y nos han formado una representación muy personal de lo que es Internet. Hemos visto las características generales, las funciones generales que podemos encontrar en la red. Es cierto que estas no son las únicas y que no tendrán siempre la importancia que tienen ahora.

En la siguiente sección hablaremos sobre la nueva generación de Internet, el famoso Internet2. Ahí veremos sus principales características y la forma en la que se ha ido dando a la transición.

Nueva Generación

La nueva generación de Internet es la que se conoce como Internet2 comúnmente. Internet2 surge como una necesidad de evolución de la red actual para satisfacer necesidades de recursos que demandan los usuarios por las aplicaciones actuales que se ejecutan en la red y una falta de planeación en la red actual.

Cuando Internet surgió, era una red muy pequeña y no se hicieron previsiones sobre el alcance y crecimiento que tendría en el futuro. Uno de los principales ejemplos (y el más mencionado) es el agotamiento de las direcciones que se ofrece. Actualmente, el protocolo principal de Internet, el IP tiene una estructura de 32 bits para la representación de las direcciones, de esta manera, vemos que existen 2^{32} direcciones. Esta es una cantidad mucho muy respetable, pero para las necesidades actuales, en donde se piensa en la posibilidad de manejar cualquier dispositivo usando la red, son insuficientes. Este es un problema que será (mejor dicho, es) resuelto por la versión 6 de este protocolo, el IPv6; esto se relaciona con la evolución de Internet, pero de ninguna manera se trata de lo mismo. En un capítulo posterior se tomará a fondo el tema de IPv6.

En la Internet actual existen infinidad de aplicaciones cada una de ellas tiene una demanda específica de los recursos de la red; así, es insuficiente para algunas de ellas el ancho de banda de la red actual, lo que puede tener como consecuencias una velocidad de transmisión baja y fallas en su ejecución.

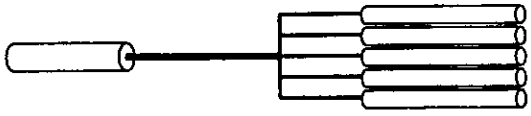
Vemos que existen dos factores principales que han sobresalido en el tema de la evolución de Internet y que por tanto se consideran como deficiencias de la red actual y son: el ancho de banda y la calidad de servicio.

El ancho de banda es la capacidad de un medio de transmitir señales a través del mismo; así, un ancho de banda muy grande permite la transmisión de un gran número de señales por el mismo canal simultáneamente. Este dato se refiere al número de frecuencias

diferentes que pueden transmitirse, representa un rango en el cual se incluyen una serie de frecuencias de señales y que, por tanto, entre mayor sea, permitirá la transmisión de mas señales por el mismo medio físico.

La información que se quiere transmitir tiene que ser codificada de forma discreta, esto se realiza mediante un proceso de modulación de la señal que contiene la información, de esta manera, los datos se modifican empleando una frecuencia determinada y así es como se mandan. Cada frecuencia que se encuentra dentro del ancho de banda del canal puede transmitir una señal de información.

El ancho de banda puede ser relacionado con la velocidad con la que se realizan las transmisiones, esto se debe a que si se usan varias frecuencias en la transmisión, pueden enviarse datos por diferentes canales virtuales concentrándolos en el extremo receptor. Puede verse como si la señal se dividiera y viajara por un canal particular para cada una y al llegar al receptor se concentraran en un solo punto.



Canales virtuales

Aunque en realidad, el ancho de banda representa la capacidad del medio para transmitir señales a diferentes frecuencias, se representa como una cantidad de datos por unidad de tiempo; así, podemos decir que el ancho de banda podrá incrementarse de 1.55 Mbps hasta 655 Mbps.

Calidad de servicio (QoS) es el concepto que define los recursos que son necesarios para una aplicación específica. Este concepto agrupa parámetros como la velocidad, el ancho de banda, el tiempo de respuesta y el tiempo de latencia. Mediante la especificación de estos parámetros, pueden establecerse los medios para proporcionar solamente los recursos que son indispensables para la aplicación y mantener de esta manera una optimización de los mismos.

Internet2 formará una red inteligente que asigne solamente los recursos solicitados por la calidad de servicio de las aplicaciones, ni mas ni menos, de esta manera se podrá tener un mejor control y administración de los recursos de la red sin descuidar a los usuarios.

Internet2 es un esfuerzo colaborativo en el que intervienen universidades y algunas instituciones particulares para proporcionar servicios de calidad y tecnología avanzada, así como difundir e impulsar su desarrollo en el mundo.

La base del desarrollo de estas tecnologías se encuentra (como suele ser) en las universidades. En México, el proyecto de Internet2 se formalizó el 8 de abril de 1999 en un

evento que se realizó en la Residencia Oficial de los Pinos, en él se establecieron los estatutos formales para la constitución del CUDI *(Corporación Universitaria para el Desarrollo de Internet)*. El CUDI, es una asociación civil, "es el organismo que representa jurídicamente los intereses de las Universidades e Instituciones que conforman el proyecto de Internet2 en México"[1].

Está formado por 29 miembros que se agrupan en Asociados Académicos, Asociados Institucionales y Afiliados. Cada tipo de membresía es diferente y por lo tanto realizan funciones determinadas dentro del CUDI.

Los miembros del CUDI agrupados por tipo de membresía son:

- Asociados Académicos
  - Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)
  - Instituto Politécnico Nacional (IPN)
  - Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM)
  - Laboratorio Nacional de Informática Avanzada (LANIA)
  - Universidad Autónoma de Nuevo León (UANL)
  - Universidad Autónoma de Tamaulipas (UAT)
  - Universidad Autónoma Metropolitana (UAM)
  - Universidad de Guadalajara (U. de G.)
  - Universidad de Las Américas-Puebla (UDLA-P)
  - Universidad Nacional Autónoma de México (UNAM)
  - Universidad La Salle (ULSA)
  - Universidad Veracruzana (UV)
  - Universidad Autónoma de Ciudad Juárez (UACJ)

- Asociados institucionales
  - Consejo Nacional de Ciencia y Tecnología (CONACYT)
  - Teléfonos de México, S. A. de C. V. (TELMEX)
  - Cabletron Systems S.A. de C.V.
  - Marconi Communications de México S.A. de C.V. (Fore System)
  - Nortel Networks de México S.A. de C.V.

- Afiliados
  - Universidad Anáhuac del Sur (UAS)
  - Universidad de Colima (UCol)
  - Universidad Iberoamericana (UIA)
  - Instituto Tecnológico Autónomo de México (ITAM)
  - Universidad Autónoma de Coahuila (UAC)
  - Universidad Autónoma de Chihuahua (UACH)
  - Universidad Tecnológica de México (UNITEC)

[1] http://www.Internet2.edu.mx

- Universidad del Valle de México (UVM)
- Universidad Autónoma de La Laguna (UAL)
- Instituto Latinoamericano de Comunicación Educativa (ILCE)
- Instituto Mexicano del Petróleo (IMP)

Internet2 ha tenido un fuerte impulso en la UNAM. En ella se lleva a cabo desarrollo tecnológico tanto de investigación, difusión y prospección de tecnologías de este tipo.

Las aplicaciones que se han determinado como indispensables hasta ahora para el uso de Internet2 y sobre las cuales el desarrollo de la misma se está orientando se agrupan en áreas en las que pueden relacionarse temas en común, así como características que definen los recursos que las definen. Estas aplicaciones son:

- Bibliotecas digitales
- Telemedicina
- Educación a Distancia
- Bases de Datos
- Colaboratorios
- Control
- Middleware
- Realidad Virtual
- Video sobre Demanda
- Videoconferencia H.323
- Multicast
- Supercómputo

Éstas no son las únicas aplicaciones que se tienen pensadas para su desarrollo con tecnologías de Internet2, pero son las principales y las demás podrán basarse en estas mismas. Estas aplicaciones son las que definen los recursos específicos para su funcionamiento.

Internet2 no podrá evadir las aplicaciones tradicionales de Internet; así, con el avance y desarrollo, seguirán funcionando las aplicaciones principales que vimos en el capítulo de Redes de Computadoras como correo electrónico, acceso remoto y transferencia de archivos. Deberán adecuarse las implementaciones de las aplicaciones para cada caso y adquirir características nuevas que permitan aprovechar los recursos que Internet2 proporcionará.

Internet2 y sus aplicaciones específicas no requieren de un análisis especial para el desarrollo de esta tesis, por lo que recomiendo consultar los URL's que se dan en la bibliografía y que se dan como referencia para obtener más información y profundizar en el tema.

Ahora, haciendo referencia a la infraestructura técnica, hay que decir que TELMEX será el que la proporcione en el caso de México; él será el encargado de implementar el backbone de Internet2 en México, así como la red nacional de fibra óptica. Los enlaces principales se encontrarán en Tijuana, Guadalajara, Ciudad de México y Monterrey.

IPv6

Introducción

IPv6 es la nueva versión del protocolo IP (IPv4), ha surgido como una solución a los problemas que tiene su predecesor y no solo eso, también tiene contemplado el crecimiento a futuro de las redes de comunicaciones (ya no sólo de computadoras), así como la demanda de recursos de las aplicaciones que se tendrán.

Surgió como una propuesta en 1994 como opción ante la previsión de los problemas que se encontrarían en un futuro no tan lejano. IP ha sido la base de los protocolos TCP/IP aunque deja parte del trabajo del control de transmisión al TCP, pero es el que permite que se lleven a cabo las comunicaciones entre los equipos que pertenecen a las redes.

Los requerimientos de recursos de las aplicaciones que han surgido últimamente, así como el aumento en el número de usuarios de Internet hacen necesaria la transición hacia IPv6 ya que este cuenta con características que lo hacen superior al anterior y sobre todo, que hace posible contar con aplicaciones que dependen de él para su funcionamiento.

Existen diferencias notables entre los protocolos IPv4 e IPv6, en este capítulo veremos algunas de ellas con detalle y justificaremos de esta manera la migración hacia este último.

Esto suena muy atractivo, la posibilidad de contar con aplicaciones que antes eran imposibles debido a las características de transferencia dadas por el protocolo y demás ventajas nos hacen tomar la decisión de adoptar este nuevo protocolo como estándar para nuestras redes.

Podemos pensar que ha pasado un tiempo razonable desde que se dictó la recomendación y desde que se ha aceptado como estándar el protocolo; en realidad son ya algunos años. Vemos que se ha dificultado en cierta manera la transición de un protocolo a otro.

La transición es un proceso relativamente sencillo pero involucra dispositivos de hardware así como configuración de software. Veremos que el proceso puede realizarse en un tiempo relativamente corto y puede elegirse entre algunas opciones de configuración para ello.

Veremos que IPv6 parece ser un protocolo que "salvará" literalmente a las redes de comunicaciones. Estudiaremos a fondo las características que lo hacen ser lo que es para entender el funcionamiento del mismo y así darnos cuenta de la forma en la que estos cambios en el protocolo afectan las redes y sobre todo las aplicaciones que algún día necesitaremos.

Características

IPv6 es un protocolo de comunicación que ha aparecido con el propósito de solucionar los problemas de los que padece su predecesor, cuenta también con un diseño que se ha considerado a futuro, ya que tiene contemplada la posibilidad de soporte de extensiones y actualizaciones que se requieran.

Desde 1992 comenzó el proceso de selección de un protocolo para adecuarse a las nuevas necesidades de las redes de computadoras. Fue en ese año cuando se propusieron cuatro de ellos, pero finalmente se optó por el IPng (más conocido como IPv6). Entonces comenzaron los trabajos formales para definirlo.

El 25 de julio de 1994 los directores del área de redes del IETF (*Internet Engineering Task Force*) hicieron la recomendación formal de este protocolo en el RFC 1752 "The Recommendation for IP Next Generation". Esta recomendación fue aprobada finalmente el 17 de noviembre de ese mismo año.

IPv6 tiene características que lo definen y que lo hacen superior a la versión anterior. Muchas de estas características se han sido implementadas debido a las necesidades de los usuarios y a las aplicaciones de los mismos.

Las aplicaciones actuales requieren de ciertos recursos de red (como alta velocidad de transferencia, mayor ancho de banda, transferencia de datos en tiempo real, seguridad en las transferencias, etc.) que no están disponibles o que tienen un desempeño ineficiente. IPv6 fue creado para permitir el uso de estas aplicaciones.

En este capítulo veremos a fondo las características de este protocolo, las diferencias que existen entre las dos últimas versiones y la estructura de la última versión.

Las características principales del nuevo protocolo son:

- cuenta con un mayor espacio de direcciones
- permite el uso de aplicaciones unicast, anycast y multicast
- permite que se realice una transición gradual entre versiones
- permite la coexistencia con IPv4
- permite la autoconfiguración de equipos
- facilita la computación móvil
- proporciona calidad de servicio (QoS)
- cuenta con seguridad e integridad de los datos
- soporta el tráfico multimedia en tiempo real

Las redes de computadoras han ido creciendo rápidamente debido a la cantidad de usuarios que requieren de el uso de los equipos conectados a las mismas. La red que mas importante y mas conocida es Internet, ya que a ella se encuentran conectados millones de computadoras que la hacen ser una red global e inmensamente grande.

Con la evolución en el campo tecnológico, han aparecido gran cantidad de dispositivos electrónicos que pueden ser conectados a la red como las ya clásicas computadoras portátiles, los teléfonos celulares, los radiolocalizadores, los PDA's (*Personal Digital Assistant*), etc. Además de todos estos equipos nuevos, también es necesario incluir en la lista a todos los dispositivos que se encuentran conectados actualmente.

Para que cada uno de estos dispositivos sea identificado en la red de manera única se le asigna un número independiente que se relaciona com la red a la que esté conectado. Este número identificador es el que se conoce como dirección IP del dispositivo; la versión 4 de este protocolo utiliza la notación decimal puntuada para representar estas direcciones de la siguiente manera:

123.456.789.123

Cada grupo de números representa una cantidad de 8 bits dando en total (los cuatro segmentos) 32 bits.

Este es uno de los principales problemas que se encuentran en la configuración del IPv4. Este protocolo utiliza un esquema de direcciones de 32 bits, lo que da un espacio de 2^{32} (4,294,967,296) direcciones que parece bastante grande pero que comienza a ser insuficiente y se hace evidente al compararlo con el número de habitantes de la tierra (6,000,000,000 aproximadamente).

IPv6, a diferencia de IPv4 tiene un esquema de direcciones de 128 bits, de esta manera se tiene un número mucho mayor de direcciones disponibles (2^{128}). Es un número realmente grande, podemos decir (para dar una idea de la magnitud de esto) que se tendrán disponibles 667,088,217,668,500,000 direcciones por mm^2 de la tierra. Ahora parece una cantidad excesiva, pero con el tiempo veremos que tanto lo es.

Las direcciones en IPv6 se representan en notación hexadecimal con : como separadores; así, por ejemplo tenemos que una dirección es:

3ffe:1abc:0000:0001:0000:0000:0000:0002

Es un número difícil de recordar, para facilitar esto se ha tomado una convención que se llama compresión de ceros, en donde se agrupan segmentos de ceros contiguos (tantos como sean) y se representan con una notación de ::. Así, la dirección anterior, utilizando compresión de ceros se representa como:

3ffe:1abc:0000:0001::0002

Se ha simplificado ya, pero sigue resultando difícil de recordar, y para hacer aún mas fácil de recordar esta dirección, se utiliza la supresión de ceros que se encuentran a la izquierda de los números representativos de cada segmento de la dirección. Finalmente, tenemos que esta dirección puede representarse como:

3ffe:1abc:0:1::2

Ya hemos visto la forma en la que se representan las direcciones para cada uno de estos protocolos de manera independiente, pero en un periodo de tiempo (relativamente largo) los dos protocolos tendrán que coexistir (mas adelante hablaremos de esta transición) así es que se vuelve necesario un formato que haga que cada protocolo entienda las direcciones del otro.

Para que IPv4 pueda "entender" direcciones de IPv6, éstas tienen que mapearse. Se dice que una dirección IPv6 "mapeada" a IPv4 es una dirección IPv6 representada en un formato compatible con IPv4, como ejemplo, tenemos:

::ffff:123.456.789.123

El proceso inverso es el que se tiene cuando una dirección IPv4 tiene que representarse con formato IPv6 para ser compatible con este último protocolo, a este proceso se le llama completar; así, tenemos que una dirección IPv4 completada con IPv6 es:

::123.456.789.123

Existen tres tipos de direcciones IPv6 diferentes, cada tipo determina de una manera diferente la forma en la que la información se envía a través de la red. Estos tipos son:

- unicast
- anycast
- multicast

Las direcciones *unicast* son las que representan de manera única una interfaz; de esta manera, los paquetes enviados a una dirección unicast son enviados directamente a la interfaz especificada por esa dirección. Estas direcciones se forman por la combinación de la dirección MAC (*Media Access Control*) y un prefijo relacionado con datos de la red.

Por ejemplo, supongamos que la dirección MAC de un dispositivo de red es: 08-00-02-12-34-56; si la dirección del enlace local tiene un formato de 48 bits, entonces su dirección IPv6 unicast será:

FE80:800:212:3456

y si el identificador del enlace local es de 64 bit, se tiene esta otra dirección:

FE80:A00:2FF:FE12:3456

Las direcciones *anycast* representan un conjunto de direcciones de interfaces que pertenecen a una red o segmento de red específico. Cuando se envían paquetes a una dirección anycast, éstos son entregados a la interfaz mas cercana, esto se determina mediante métricas de distancia. Ya que no representa únicamente una dirección específica, tiene ventajas para el gateway por default así como para los equipos móviles.

Las direcciones *multicast* igual que las anteriores, representan un conjunto de interfaces. La diferencia se encuentra en que los paquetes son entregados a todas las interfaces que componen ese conjunto simultáneamente.

Como hemos visto, se hace necesario un periodo de tiempo en el cual se tenga la existencia de los dos protocolos, durante este periodo podrán realizarse las pruebas correspondientes tanto de software como de hardware para adecuar los sistemas y los equipos a IPv6. Otra de las características que influyó en la decisión sobre adoptar IPv6 como protocolo de comunicación es que permite realizar un proceso de transición relativamente fácil y rápido, aunque la idea fundamental y la que determinó esta decisión es que permite hacerlo. Se dice que en el IETF se hizo la observación de desechar cualquier idea para mejorar el protocolo si no incluía esta característica por maravillosa que fuera. La capacidad de transición es necesaria ya que la cantidad de sistemas y equipos que utilizan IPv4 lo demanda.

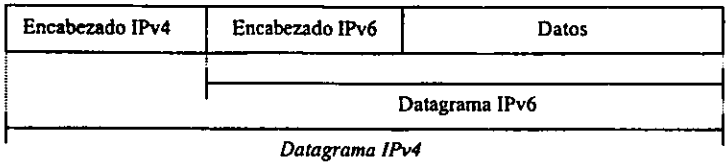
La transición se realizará de una manera relativamente sencilla y casi de forma transparente al usuario. Esta se realizará como se hace una actualización de software común, con algunas herramientas que permitan hacerlo fácilmente a través de los proveedores de servicio de acceso a internet y otros. Se puede decir que el trabajo difícil se deja a los administradores de las redes solamente.

En el periodo en el que se realiza la transición será necesaria la coexistencia de los dos protocolos ya que es necesario mantener activa la red para la mayoría de los usuarios (que permanecen usando IPv4) y para la minoría (que están experimentando con IPv6) así como para aquellos que se encuentran en las pruebas relacionadas con los dos protocolos.

Para la realización de las pruebas con IPv6 en las redes actuales se pueden usar varios mecanismos para establecer el contacto que se llaman "Mecanismos de transición para nodos y ruteadores" y que se definen en el RFC1933. Debido a que las redes actuales trabajan con IPv4, estos mecanismos se enfocan al uso de IPv6 usando estas redes.

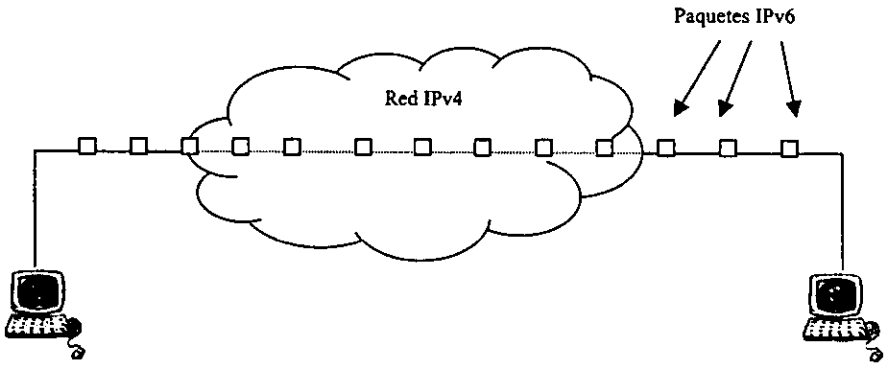
Existen dos principales formas de trabajar con IPv6 sobre redes IPv4. En esta sección se analizará la forma en la que se trabaja con cada una de ellas, así como las características de las mismas.

La primera de ellas es el encapsulamiento de paquetes IPv6 con encabezados IPv4, este mecanismo es el que se conoce como "túnel". El encapsulado se hace encabezando el datagrama completo (encabezado y datos) de IPv6 con IPv4. La siguiente gráfica muestra el formato del paquete.



Datagrama IPv4

Se dice que un datagrama consta fundamentalmente de dos partes básicas, los datos y el encabezado del mismo. Los datos es la información o los datos que se envían a través de la red y el encabezado tiene datos de control que permiten llevar a cabo la transferencia de manera correcta. Así, cuando se encapsula un datagrama IPv6 con IPv4 se agrega un encabezado o cabecera correspondiente a los datos que se forman por el encabezado IPv6 y los datos originales.



Túnel IPv6 sobre IPv4

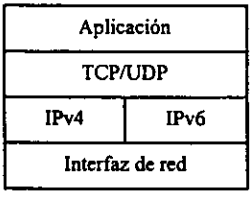
Vemos entonces que los datagramas de IPv6 viajan a través de la red encapsulados usando IPv4; o sea que, viajan a través de un túnel de IPv6 que se ha creado en la red IPv4. El encapsulamiento se lleva a cabo en la fuente y el desencapsulamiento en el destino de la transferencia. Esta técnica se utiliza principalmente para evitar el problema que causa no tener ruteadores configurados para utilizar IPv6.

Existen dos tipos diferentes de túneles:

- los túneles configurados. Son aquellos en donde la dirección del nodo destino se proporciona manualmente y desde el nodo que se encarga de encapsular; esto se usa solamente en conexiones punto a punto.
- los túneles automáticos. En los túneles automáticos, la dirección de destino es incluida en el paquete y esta dirección es compatible con IPv4.

El otro mecanismo para trabajar simultáneamente con los dos protocolos es la utilización de una capa IP dual (lo que se conoce como stack doble) en el host y los ruteadores. El stack doble permite a los dispositivos de red utilizar el protocolo que sea necesario dependiendo de las necesidades inmediatas de las aplicaciones en ejecución.

Esto puede representarse gráficamente de la siguiente manera:



Stack doble

Este mecanismo es el más sencillo para realizar la transición de un protocolo a otro.

En este punto es necesario mencionar que para que se realice la transición de una manera gradual y progresiva como se pretende, se necesita actualizar el DNS (Domain Name Service) ya que como hemos visto, la representación de las direcciones con los protocolos ha cambiado y se ha vuelto más complicada. La labor principal del DNS es permitir la relación entre las direcciones numéricas con las alfanuméricas que son mas fáciles de recordar y mas representativas. Uno de los cambios principales que tiene que llevar a cabo el DNS es con el nuevo registro (AAAA) que tiene que ser implementado para permitir la interpretación de direcciones IPv6 ya que, en el caso de IPv4 se utiliza un registro A.

El agotamiento de las direcciones que se dio a causa de las limitaciones de IPv4 trajo como consecuencia la saturación de las tablas de ruteo. En ellas se encuentran las direcciones de los dispositivos y equipos que tienen asignados una dirección IP y que son alcanzables desde algún punto de la red. Debe decirse que no existe una única tabla de ruteo y que su contenido depende de las características de la red a la que pertenecen. Estas tablas se mantienen actualizadas con información de tablas de ruteo de redes cercanas por medio de un intercambio de datos.

Debido a que la cantidad de dispositivos conectados a la red aumenta (y seguirá aumentando) considerablemente, las tablas de ruteo llegarán a un punto en el que se saturarán y esto traerá consigo una baja en la velocidad de resolución de rutas para transferencia de información a través de la red.

Otra de las ventajas de IPv6 es la capacidad de autoconfiguración de los equipos. Esta autoconfiguración se refiere a la asignación de la dirección IP para el dispositivo al momento de establecer la conexión con la red.

Existen dos formas de autoconfiguración en IPv6:

- Stateless. Es la que realiza el host directamente para obtener una dirección IPv6 válida para establecer su conexión a la red.
- Stateful. Utiliza un servidor DHCP (*Dynamic Host Configuration Protocol*) para IPv6 para asignar direcciones disponibles a los dispositivos cuando éstos lo solicitan.

Esta, como se ve es una ventaja que deriva en otra, la computación móvil. Gracias a esto, se podrá contar con direcciones disponibles en el momento y en el lugar en que se necesiten. El único requisito será la elección de un punto de acceso a la red, que podrá estar definido por las características del dispositivo en particular.

Otra de las ventajas de este nuevo protocolo es que el encabezado que utiliza se ha simplificado considerablemente en relación al que se tiene para IPv4. Con esto se ha logrado la simplificación en la transmisión de los paquetes, así como el aumento en las velocidades de transmisión. Para poder entender los fundamentos de este protocolo, se hace necesario profundizar en el análisis del encabezado del mismo.

La siguiente gráfica muestra el encabezado de los paquetes IPv6, en los siguientes párrafos se describen las características que presentan cada uno de los campos que lo componen.

Versión	Prioridad	Etiqueta de flujo	
Longitud de carga útil		Encabezado siguiente	Límite de saltos
Dirección de la fuente			
Dirección del destino			

Encabezado IPv6

Podemos ver que el encabezado se ha simplificado enormemente[1]. Ahora se tienen menos campos y las funciones de los mismos se ha definido para adaptarse a las necesidades y características del protocolo.

El campo de *versión* es de 4 bits de longitud, representa la versión del protocolo (IPv6).

El campo de *prioridad* también tiene una longitud de 4 bits y se usa para especificar la prioridad que tiene el paquete que se transfiere. Este campo puede tomar uno de 16 valores que corresponden al nivel de la prioridad. Estos valores van del 0 al 15 y se asignan de la siguiente manera:

[1] En el capítulo "Redes de Computadoras" se muestra el encabezado de IPv4.

- 0 al 7. Son los valores de menor prioridad, los paquetes que usan este identificador pueden reducir la velocidad con la que son enviados en caso de un congestionamiento de la red.
- 8 al 15. Son valores de máxima prioridad, son asignados a los paquetes que requieren una transferencia en tiempo real. La tasa de transferencia se mantiene constante aunque se pierdan algunos paquetes del flujo.

La *etiqueta de flujo* tiene 24 bits de longitud. Se usa para identificar los paquetes que pertenecen al mismo flujo de datos. Los paquetes que son enviados tienen características comunes que son la dirección de destino, la dirección de la fuente, etc., pero la etiqueta de flujo (en conjunto con estas últimas) forman el identificador único para el flujo y permite identificarlo. Este campo debe tener un valor aleatorio entre 1 y FFFFFF.

El campo de *longitud de carga útil* tiene 16 bits, en él se indica la longitud del bloque de datos que es encabezado en el paquete. La máxima longitud de carga útil que puede tenerse es de 64 Kbytes. Cuando se necesita que esta longitud sea mayor, se incluye un valor de cero en este campo y con ello se agrega una extensión del encabezado con el campo Jumbo Playload que permite realizar transferencia de datos con longitudes mayores en los paquetes.

El campo de *encabezado siguiente* identifica al tipo de encabezado del siguiente paquete en el flujo, tiene una longitud de 8 bits y se representa mediante valores decimales de la siguiente manera:

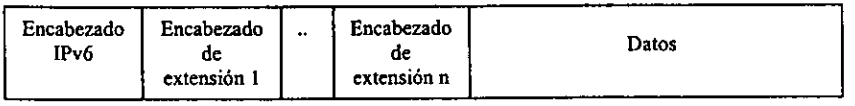
Valor decimal	Siglas	Descripción
0		Reservado (IPv4)
1	HBH	Opción Salto por Salto (IPv6)
2	ICMP	Protocolo Internet de Mensajes de Control (IPv4)
3	GGP	Protocolo Puerta de Enlace a Puerta de Enlace
4	IP	IP en IP (encapsulación IPv4)
5	ST	Trama
6	TCP	Protocolo de Control de Transmisión
8	EGP	Protocolo de Puerta de Enlace a Exterior
17	UDP	Protocolo Datagrama de Usuario
43	RH	Encabezado de Ruteo (IPv6)
44	FH	Encabezado de Fragmentación (IPv6)
46	RSVP	Protocolo de Reservación
51	AH	Encabezado de Autenticación
58	CMP	Protocolo Internet de Mensajes de Control (IPv6)
59	Null	Sin encabezado siguiente (IPv6)
60	DOH	Encabezado de Opciones Destino (IPv6)
83	VINES	VINES
88	IGRP	IGRP
89	OSPF	OSPF

Valores para el encabezado siguiente

El campo de *límite de saltos* es de 8 bits de longitud, tiene un valor numérico que puede ser como máximo 255. Este valor se decrementa en cada equipo por el que pasa el paquete, de esta forma se pueden evitar problemas de loops cuando se tiene mal la dirección del destino. Cuando el contador llega a cero, se desecha el paquete.

Los dos últimos campos, los de *dirección de la fuente* y *dirección del destino* son de 128 bits. En ellos es en donde se deben incluirse las direcciones de los puntos extremos para la transferencia de los paquetes.

Los paquetes IPv6 pueden tener extensiones de encabezados para poder incluir algunas características extras. Estrictamente, y de manera gráfica, un datagrama IPv6 puede representarse de la siguiente manera:



Datagrama IPv6

Existen 9 tipos de encabezado de extensión para paquetes IPv6 y una implementación completa del mismo debe soportar estas extensiones. Como se muestra en la gráfica, pueden incluirse varios encabezados, esto puede hacerse siguiendo el RFC 1883 en donde se establece la secuencia en la que se incluyen. Esta secuencia es la siguiente:

1-. Encabezado IPv6.
2-. Encabezado de Opción Salto por Salto.
3-. Encabezado de Opciones Destino.
4-. Encabezado de Ruteo.
5-. Encabezado de Fragmentación.
6-. Encabezado de Autenticación.
7-. Encabezado de Encapsulación con Seguridad de Carga Util.
8-. Encabezado de Opciones Destino.
9-. Encabezado de Capa Superior.

Otra de las ventajas de este protocolo es que permite reconocer la Calidad de Servicio necesaria para cada aplicación, de esta manera es como se pueden asignar solamente los recursos necesarios para satisfacer esos requerimientos sin desperdiciarlos ni ocuparlos cuando no es necesario. El concepto de Calidad de Servicio QoS *(Quality of Service)* se define por el conjunto de recursos como ancho de banda, velocidad de transferencia, tiempo de latencia, tiempo de retardo y prioridad en la transferencia.

Gracias a esto puede ser optimizada la transferencia de paquetes a través de la red administrando mejor los recursos de la misma.

En cuanto a la seguridad e integridad de los datos, IPv6 tiene características que permiten verificar que los paquetes que se envían (y reciben) provengan del lugar del que se supone que provienen. Así se eliminan conceptos como el de suplantación de hosts eliminando también los ataques que se realizan de esta manera (ataques por réplica), hace que los sniffers y analizadores de tráfico no puedan usarse ya que protege los paquetes hasta su destino y finalmente reduce el uso de los firewalls.

IPsec *(Internet Protocol Security)* es el protocolo que se encarga de las funciones de seguridad que se tratan con IPv6 y con otros protocolos superiores como TCP y UDP, ICMP. Se encuentra descrito en el RFC 2401. Se aplica en los hosts, en los ruteadores y firewalls.

Es un protocolo muy flexible, ya que permite al sistema elegir entre los protocolos de seguridad que utilizará, así como los algoritmos con los que se realizará el cifrado y las llaves necesarias para ello.

El IPsec actúa sobre otros dos definidos por los RFC 2402 y 2406, "Authentication Header" y "Encapsulating Security Payload". El primero realiza funciones de autenticación de los datos de la cabecera de los paquetes, proporciona integridad en las transmisiones que no están orientadas a conexión, mientras que el segundo proporciona confidencialidad por medio de cifrado. Los dos protocolos pueden usarse de manera independiente o en conjunto utilizando túneles o sin ellos.

Con todas las características que hemos visto, pueden tenerse aplicaciones avanzadas para redes. Muchas de estas aplicaciones han empezado a surgir con el concepto de Internet2. Es cierto que IPv6 e Internet2 son conceptos muy diferentes, no puede decirse siquiera que se parezcan.

El primero es un protocolo de comunicación y el segundo es un modelo de red avanzada. Es cierto también, que Internet2 podrá ser lo que promete aprovechando las ventajas de IPv6, ya que sin ellas sería imposible constituir las redes de tecnología avanzada que se espera.

LINUX

Introducción

Linux es un sistema operativo que ha revolucionado los sistemas del mundo. Comenzó siendo un proyecto personal de un estudiante de informática de Helsinki y ha ido evolucionando hasta convertirse en unos de los más optimizados y utilizados del mundo. Cuenta con características que lo hacen uno de los mejores, más confiables y seguros que se usan actualmente.

Linux cuenta con las características de un sistema operativo que cuenta con el respaldo de una compañía de software, pero sin tenerlo de una manera formal (por lo menos de manera general). Ha sido diseñado y desarrollado por mucha gente que intercambia información y hace pruebas aprovechándose de las ventajas de Internet.

Se dice que ha sido desarrollado especialmente por hackers, y es por eso que tiene gran potencia y se ha convertido en un sistema realmente sólido, tanto que ha empezado a opacar algunos otros que si son comerciales.

Linux se distribuye bajo el concepto de software libre; es decir, se distribuye el sistema como tal y además el código fuente que lo integra, de esta manera se tiene un acceso total y libre al mismo.

Una de las principales ventajas con las que Linux nació (dado su origen) es que está diseñado para trabajar en red, lo que lo hace muy potente en cuanto a la ejecución de aplicaciones de redes y por lo tanto en Internet. Tiene implementación de funciones y aplicaciones que lo hacen sobresalir entre los demás sistemas operativos en este rubro.

Dada esta flexibilidad y fortaleza, es un sistema ideal para pruebas de desarrollo, implementación y prospección de aplicaciones para redes de tecnología avanzada como lo es Internet2, aprovechando los recursos y características con las que cuenta.

Uno de los pasos indispensables para la evolución de este sistema hacia las aplicaciones que demandan recursos especiales es la implementación del protocolo IPv6 en el mismo. Esto permitirá aprovechar las capacidades de Linux como sistema operativo de red al mismo tiempo de desarrollar y probar las nuevas tecnologías de redes.

En este capítulo veremos algunas de las características de Linux que han hecho a tanta gente hablar tanto del mismo. Veremos las ventajas y las desventajas que presenta y trataremos de ir acercándonos a la integración del sistema con el nuevo protocolo de comunicación, así como con las aplicaciones para el mismo.

Conceptos básicos

Linux es un sistema operativo que tiene relativamente poco tiempo y que se ha desarrollado de una manera muy rápida y no solo eso, lo ha hecho de manera satisfactoria. Comenzó siendo un proyecto personal de Linus Torvalds, un estudiante de informática de la

universidad de Helsinki. Fue un intento por hacer más accesible el sistema MINIX para los estudiantes y compañeros de Linus.

La primera versión de lo que conocemos ahora como Linux apareció el 5 de octubre de 1991[1], esta era la versión 0.02. En ella solamente se podía utilizar el bash (que es un shell del sistema que se conserva aún) y gcc (un compilador para c).

Como se dijo, Linux nació siendo un acercamiento al MINIX, que era a su vez, un acercamiento al UNIX, por lo tanto, podemos decir que Linux, desde su nacimiento ha sido un intento de ofrecer las capacidades del UNIX pero en otras condiciones.

Se ha dicho que su nombre "Linux" es un juego de palabras en el que se pone lo anterior de manifiesto, se dice que es la versión "lite" de UNIX. Esto es porque comenzó siendo una versión limitada del mismo.

La primera característica que sobresale de Linux es que es un sistema operativo libre, se ofrece y distribuye bajo GNU Public License. Esto quiere decir que es un software que se distribuye libremente tanto el sistema como tal, así como el código fuente del mismo. Esto permite que pueda ser adquirido por cualquier persona que lo desee y no sólo eso, se da la libertad de adaptarlo a las necesidades particulares de cada quien modificando el código del mismo.

Esta licencia fomenta el uso, distribución y desarrollo del software libre, pero mantiene los derechos de autor y reconoce el trabajo de las personas que han trabajado en el mismo. Tal es el caso del kernel de Linux, los derechos legales son propiedad de Linus Torvalds.

El software libre tiene sus desventajas. Como no se tiene el apoyo de ninguna compañía de software establecida, puede ser difícil encontrar ayuda o documentación para la solución de problemas o para el desarrollo del software del que se trate. Linux ha sido desarrollado por muchos programadores en el mundo y no se tenía un apoyo privado.

Anteriormente se decía que esa era una de sus mayores desventajas, pero actualmente, Linux cuenta con el apoyo de muchas compañías tanto de software como de hardware. Ahora se tiene atención en línea permanente, existen foros, grupos de noticias y discusión, listas de correo, bibliografía en libros y revistas, información en CD's, etc. Algunas de las compañías que han declarado su aceptación, implementación, uso y en algunos casos desarrollo de aplicaciones para Linux son[2]:

- Apple
- Compaq
- Computer Associates
- Corel
- Dell
- Hewlett Packard

[1] http://www.linux-es.com
[2] Network Computing, Agosto 1999.

- IBM
- Novell
- Oracle
- SAP
- Sybase

Algunas de estas compañías desarrollan software de aplicación para Linux, otras han adaptado sus sistemas para la compatibilidad con el mismo, y otras mas lo distribuyen como una de las opciones de software precargado de sistema operativo para los compradores de su equipo.

Ahora, podemos ver que gracias a la licencia pública de este sistema operativo existen diferentes versiones del mismo, aunque en realidad no se les conoce como versiones sino como distribuciones. Estas distribuciones son compilaciones de programas y archivos que se encuentran listas para instalarse de una manera funcional y cada una tiene características particulares.

Cada distribución tiene su grupo de desarrollo encargado que se mantiene realizando actualizaciones y optimizaciones. Algunos de estos grupos se han constituido formalmente como empresas de software que han adquirido prestigio gracias a su trabajo de participación y colaboración en Linux. Todas las distribuciones tienen características que mantienen como estándar y esto es lo que las mantiene siendo distribuciones de Linux como tales, el ejemplo claro es el kernel del mismo. El trabajo de actualización y optimización del kernel lo realiza Linus Torvalds y un pequeño grupo de programadores.

Las distribuciones de Linux que se pueden encontrar actualmente son:

- Caldera
- Conectiva Linux
- Corel Linux
- Debian
- ESware
- Eurielec
- Hispafuentes Linux
- Mandrake
- MkLinux
- Red Hat
- S.u.S.E.
- Slakware
- PPP

Algunas de estas distribuciones son más conocidas, desarrolladas y difundidas que otras; puede cambiar la presentación que tienen, las aplicaciones que manejan, la interfaz con la

que cuentan, la forma de actualizar el sistema y otras cosas, pero en esencia todas son Linux.

Linux puede obtenerse de diversas fuentes, como ya hemos dicho se trata de un software libre, por lo tanto, gratis. Si se cuenta con acceso a Internet, puede adquirirse de alguno de los tantos sitios públicos en los que se distribuye, puede encontrarse en CD's que se distribuyen con copias de libros y revistas, puede adquirirse por una cantidad (normalmente simbólica) en las asociaciones o grupos de Linux o puede adquirirse como opción preinstalada de sistema operativo en algunas marcas de equipos de cómputo. Es necesario aclarar que algunas de estas distribuciones son comerciales (las menos) y por lo tanto hay que pagar por ellas el precio establecido que por cierto, no es simbólico.

El kernel no es la única característica en particular con la que cuentan las diferentes distribuciones de Linux. Como lo hemos venido diciendo se trata de un sistema operativo completo y funcional que tiene un mejor desempeño que otros. Ahora veremos algunas de las características que lo hacen ser lo que es.

Fue creado para su ejecución en procesadores intel 386 ó superiores, pero su desarrollo le ha permitido su actual característica multi-plataforma; esto es, no se limita a su ejecución en un tipo determinado de procesadores. Actualmente puede correrse en procesadores 386, 486, Pentium, Pentium II, Pentium III, Pentium Pro, Amiga, Atari, Alpha, ARM, MIPS, PowerPC, SPARC y AMD, así como en los clones de intel.

Como se ha dicho, Linux tiene una relación estrecha con el UNIX pero debido a que este último es una marca comercial registrada se dice estrictamente que los estándares a los que se apega Linux que lo relacionan con el UNIX están dados por el POSIX.

POSIX (*Portable Operating System Interface*) es un conjunto de estándares emitidos por el IEEE (*Institute for Electrical and Electronical Engineers*) para promover la portabilidad de las aplicaciones UNIX. Es gracias a esto s estándares que se encuentran compatibilidades entre los dos sistemas operativos de los que hemos estado hablando, aunque es necesario mencionar que otros sistemas han adoptado estos estándares y empiezan a hacerse cada vez mas parecidos.

Los estándares POSIX se identifican de la siguiente manera: POSIX.*x*, donde *x* es un número que va desde 1 hasta 22. Linux se apega a estos estándares, pero son dos los que lo hacen evidente, el POSIX.1 y el POSIX.2. El primero de ellos se refiere a las funciones escritas en C que usan las aplicaciones para accesar al sistema operativo y el segundo se refiere al intérprete de comandos y programas básicos que se tienen en el mismo; este es el estándar que por el cual Linux es tan parecido al UNIX.

El nacimiento de Linux estuvo dirigido hacia el UNIX, por lo tanto se formó con algunas características de este último y las conserva actualmente. Mantiene hasta ahora estrechos lazos con este sistema por lo que son compatibles considerablemente. Con el paso del tiempo, vemos que Linux tiene su línea definida y que se mantiene independiente del UNIX actual, comparten muchas características, es cierto, pero es independiente y cuenta con características que lo definen.

Podemos ver la capacidad multitarea de Linux como una herencia del UNIX. Con ella se permite la ejecución de diversos programas o procesos simultáneamente. Maneja una administración multitarea prioritaria, que a diferencia de la multitarea compartida permite la ejecución libre de un proceso hasta su terminación de manera simultanea con los demás administrando y asignando los recursos necesarios para que puedan realizarse las tareas necesarias.

La capacidad de multitarea de un sistema operativo lo hace ideal para otra de las capacidades que lo identifican, la capacidad de ser usado por varias personas al mismo tiempo, lo que se llama soporte multiusuario. Con esta característica, el sistema puede atender las peticiones de asignación de recursos para programas y aplicaciones para diferentes usuarios simultáneamente. Esto le permite a los usuarios del sistema ejecutar programas de aplicación al mismo tiempo, haciendo uso de terminales virtuales para ello. Hay que decir que esta es una ventaja que le permite al sistema convertir el equipo en un servidor como tal.

Para tener acceso al sistema en modo multiusuario, es necesario contar con las llamadas terminales virtuales, que son las que permiten emular al sistema en otro equipo; desde ahí, pueden ejecutarse programas y accesarse archivos de datos desde un equipo remoto. De esta manera, vemos claramente que el sistema puede convertir al equipo en el que se ejecuta en un servidor que por cierto cuenta con aplicaciones específicas para este trabajo, sin olvidar que cuenta también con la capacidad de conectarse a otros equipos, por lo que puede convertirse en un cliente también.

Otra de las características que distinguen a Linux es su capacidad de programación del shell. El shell es un intérprete de comandos, es el que se encarga de establecer y mantener el trabajo entre el usuario y el kernel. Con esta característica pueden simplificarse muchas tareas al hacer que el sistema las realice en secuencia de la misma manera que lo haría un usuario.

Ofrece compatibilidad binaria entre aplicaciones que se han desarrollado para diferentes sistemas UNIX, esto es lo que se conoce como la iBCS2 (*Intel Binary Compatibility Standard*); con esto pueden usarse en Linux aplicaciones que se desarrollaron para otros sistemas UNIX como editores de texto, hojas de cálculo, etc. usando un procesador x86.

Soporta varios sistemas de archivos diferentes; así, cuando se está trabajando bajo Linux, puede accesarse a archivos que se encuentren en otros sistemas de archivos como los de DOS, vfat, fat32 y otros.

Tiene incluido en el kernel soporte para diferentes protocolos de red, entre ellos el IPv6 que es el que forma parte del tema de esta tesis. La configuración del kernel, su compilación e instalación son tareas que tienen que realzarse cuando se necesita que el sistema tenga ciertas características con las que no cuenta, es un proceso relativamente fácil y rápido y gracias a él se puede tener un sistema Linux actualizado con la última versión del kernel y por lo tanto un kernel funcional, optimizado, probado, con nuevas funciones y sobre todo estable.

El formato con el que se identifican las versiones del kernel es el siguiente:

linux-xx.yy.zz

en donde xx es un número que identifica la versión base del kernel; yy es un número que identifica la característica de estabilidad del kernel, si es un número par, es estable; y zz es un número que identifica las últimas correcciones a las diferentes versiones. La versión del kernel, con el que se realizaron las pruebas a las que se hace referencia en esta tesis es 2.2.17. Debido a la naturaleza de software libre, podemos encontrar el kernel de esta forma en muchos lugares y de esta manera podemos tener actualizado el sistema.

La administración de los dispositivos en Linux se realiza de una manera muy peculiar y diferente a otros sistemas operativos como Windows por ejemplo. Linux reconoce y maneja los dispositivos de hardware como archivos independientes; de esta forma, aunque cada uno de ellos requiere de una configuración especial y particular son tratados de la misma manera. Cuando se necesita mandar algún tipo de información hacia los mismos, lo que se hace es redireccionar el flujo de los datos hacia el archivo que lo representa y no al hardware mismo.

Todos los programas o aplicaciones que se ejecutan sobre Linux son representados por uno o más procesos, estos procesos permiten al sistema administrar los recursos del mismo para poder asignar los que sean necesarios para la ejecución del proceso específico que se trate. Estrictamente, los procesos se identifican por medio de un número mediante el cual se le puede reconocer de manera abstracta entre los demás.

Una de las mayores ventajas de Linux es la administración de procesos, que como hemos dicho se realiza de forma multitarea prioritaria. Debido a esta característica, es posible alterar la ejecución de un proceso en particular; así, puede interrumpirse o eliminarse uno de ellos sin afectar a los demás.

La interacción con el kernel de Linux se realiza principalmente de dos formas: desde la línea de comandos en la que se tiene una representación textual para las funciones que se realizan, así como algunos programas de aplicación y herramientas de sistema y otra que es la interfaz gráfica de usuario. Esta última representa una ventaja al usuario ya que le presenta de un modo más amigable las funciones que maneja el sistema operativo.

Tiene muchas otras ventajas, ya que se han escrito muchos programas de aplicación y se han adaptado las funciones del sistema para representarse gráficamente, pero existe una desventaja. Se trata de las diferencias que existen entre las interfaces gráficas de cada distribución; estas suelen diferir un poco pero en esencia realizan las mismas funciones.

Configuración básica del sistema

Una de las ventajas de Linux es que puede ser configurado fácilmente. Existen diferentes tipos de configuraciones que dependen de las necesidades de los usuarios del sistema.

Algunas de ellas pueden ser complejas y realizar funciones específicas por lo que es necesario prestar una atención especial a su configuración.

En el caso de la configuración básica del sistema podemos decir que esta puede realizarse de manera prácticamente automática si es seleccionada durante la instalación del sistema.

El programa de instalación de Linux proporciona diferentes opciones de configuración predeterminada que son:

- Estación de trabajo
- Servidor
- Personalizada

La primera de ellas, cuenta con soporte para trabajo directamente en la computadora en la que se instala. Tiene seleccionadas las opciones de los programas de aplicación en su modo de cliente. Con todo eso, es posible conectarse a una red y poder trabajar de forma remota en otros equipos.

La instalación de servidor es la que permite proporcionar diferentes servicios a otras computadoras que se conectan y solicitan un servicio determinado. A este servidor pueden conectarse los usuarios remotamente y usar los recursos del sistema para realizar su trabajo. Esta instalación cuenta con los programas servidores encargados de establecer, mantener y terminar los servicios que solicitan los usuarios que son principalmente aquellos servicios que se proporcionan en internet como mail, web, ftp, acceso remoto, etc. Existen algunos otros programas servidores que realizan funciones especiales y que pueden ser instalados por el administrador del sistema si es necesario.

Cabe mencionar que esta última instalación incluye las opciones y archivos que se instalan en la instalación de estación de trabajo, por lo que el sistema puede trabajar como estación de trabajo y servidor simultáneamente.

La instalación personalizada es la que cuenta con mayor flexibilidad, ya que permite elegir los archivos y paquetes que serán instalados en el sistema. De esta manera puede elegirse una configuración especial que atienda a ciertas características particulares que definan las necesidades de los usuarios por el administrador del sistema.

Cuando se realiza esta instalación debe tenerse cierta precaución ya que pueden ser excluidos componentes que pueden ser necesarios para el correcto funcionamiento del sistema. Una de las ventajas con las que cuenta el programa de instalación de Linux es que realiza una verificación entre lo que se instala y puede mostrar un aviso de los componentes que no han sido seleccionados durante la instalación y que resultan necesarios para el sistema.

Para poder configurar el sistema para que use IPv6 no es necesaria una configuración especial. Estrictamente, podemos decir que puede elegirse la opción que se satisfaga las

necesidades del sistema, por lo que puede elegirse cualquiera que sea funcional, pero para cuestiones de pruebas de programas servidores y programas clientes es más conveniente elegir por la opción de configuración del sistema como servidor.

Así, podremos instalar y probar las aplicaciones que se encuentren en los extremos de los puntos de comunicación y nuestro sistema podrá trabajar como servidor atendiendo los servicios que sean solicitados y usando los programas cliente para realizar las peticiones de estos servicios a otros servidores.

Entonces, una configuración del sistema como servidor es la ideal para nuestros propósitos y como dijimos anteriormente, puede realizarse usando el programa de instalación del sistema operativo. Si no se tiene esta configuración, puede realizarse una actualización del sistema utilizando el disco de instalación de Linux, este proceso se realiza de una forma muy sencilla.

Si no se cuenta con los recursos para completar la configuración del sistema como servidor desde la instalación o se quiere configurar el sistema de manera independiente, para que podamos trabajar usando IPv6, es necesario que se configure el sistema con las opciones necesarias para comunicaciones por red.

Con esto, se deberá contar con soporte para los protocolos que se utilizaran, enfatizando en el IPv6, herramientas de configuración, mantenimiento y monitoreo de las funciones de red, así como los programas de aplicación (clientes y servidores) que se ejecutarán en el mismo.

Hay que recalcar que los tipos de instalación nos dan la oportunidad de tener configurado del sistema con opciones predeterminadas, pero no son las únicas opciones con las que se cuenta ni tampoco son configuraciones que deban mantenerse de forma estricta. Cada administrador deberá establecer la configuración particular para las necesidades de su sistema.

Aplicaciones para redes

Las aplicaciones para redes que están implementadas en Linux son básicamente las mismas que se encuentran para los demás sistemas operativos que tienen acceso a redes de computadoras. En esta sección daremos un tratamiento un tanto específico a ciertas aplicaciones básicas de red en Linux así como a la configuración de las mismas.

Antes de profundizar en las aplicaciones, veremos algunos fundamentos que hacen posible la comunicación entre el sistema y estas aplicaciones. Hablaremos un poco de algunas herramientas que sirven para la configuración y monitoreo del sistema y de algunas aplicaciones específicas para la red.

La mayoría de los servicios y las aplicaciones de red que se utilizan por este sistema operativo son "lanzados" o inicializados en un momento específico. La secuencia en la que el sistema arranca puede determinar la forma en la que se establece la disponibilidad de los servicios de red.

Cuando se levanta el sistema, empieza el proceso de carga de los demonios del sistema. Cada uno de ellos tiene una función específica, tiene una aplicación determinada que prestar y servicios que atender. Los demonios son programas que se ejecutan continuamente en el sistema y cuando encuentran una solicitud de servicio autorizada, la atienden proporcionando el servicio solicitado.

Los demonios para los servicios de red se levantan cuando el sistema arranca (generalmente). De esta manera, se mantienen atentos a la solicitud de los servicios. El programa que permite que sean cargados los servicios de red en Linux, es el inetd. inetd se conoce como el super-servidor, se inicia en tiempo de arranque del sistema (lo llama el /etc/rc.d/rc), escucha los puertos de conexión y cuando se detecta, determina el tipo de servicio que se requiere y lo inicializa. Es un demonio que permite levantar otros demonios. Lo que se logra con esto es evitar la carga del sistema manteniendo en ejecución solamente los programas servidores que se requieren en el momento en que son solicitados. Este demonio basa su funcionamiento en un archivo de configuración que es /etc/inetd.conf en el que se indica el tipo de servicio que está disponible, así como algunas características de conexión y accesibilidad.

El archivo /etc/inetd.conf está organizado básicamente en columnas en las que se tiene la información separada. Los datos que se muestran son los siguientes: nombre del servicio, tipo de socket, protocolo, estado de actividad, usuario, ruta del servidor y argumentos. En las siguientes líneas se muestra parte de este archivo del sistema.

```
# inetd.conf This file describes the services that will be available
#            through the INETD TCP/IP super server.  To re-configure
#            the running INETD process, edit this file, then send the
#            INETD process a SIGHUP signal.
#
# Version:   @(#)/etc/inetd.conf 3.10   05/27/93
# Authors:   Original taken from BSD UNIX 4.3/TAHOE.
#            Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#
# Modified for Debian Linux by Ian A. Murdock <imurdock@shell.portal.com>
# Modified for RHS Linux by Marc Ewing <marc@redhat.com>
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Echo, discard, daytime, and chargen are used primarily for testing.
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
#echo        stream tcp    nowait root   internal
#echo        dgram  udp    wait   root   internal
#discard     stream tcp    nowait root   internal
#discard     dgram  udp    wait   root   internal
#daytime     stream tcp    nowait root   internal
#daytime     dgram  udp    wait   root   internal
#chargen     stream tcp    nowait root   internal
#chargen     dgram  udp    wait   root   internal
#
# These are standard services.
ftp         stream tcp    nowait root   /usr/sbin/tcpd      in.ftpd -l -a
telnet      stream tcp    nowait root   /usr/sbin/tcpd      in.telnetd
gopher      stream tcp    nowait root   /usr/sbin/tcpd      gn
```

El valor de la primera columna (el nombre del servicio) debe ser un dato que se encuentre en el archivo /etc/services. En la segunda columna se especifica el tipo de socket que se utiliza durante la conexión; para los servicios que utilizan tcp debe usarse el tipo stream, mientras que para los que usan udp se usa dgram, los demás tipos se utilizan en casos especiales. La tercera columna especifica el tipo de protocolo que debe usarse, este debe estar definido en el archivo /etc/protocols. En la cuarta columna se indica la forma en la que se mantiene activo al servicio, se tienen solamente dos opciones que son nowait y wait, la primera de ellas se utiliza cuando se desea que el servicio se mantenga disponible y atento a varias conexiones simultaneas; así, cuando se encuentra una petición de atención por el servicio es atendida. Si mientras se atiende esta petición surge una nueva petición al mismo servicio, entonces la atiende, lo que no sucede con wait. Debido a que los servicios que usan tcp como protocolo están orientados a conexión (por la naturaleza del protocolo) es necesario establecer como norma este campo con el valor nowait. El siguiente campo es el que especifica el usuario al que es asignada la ejecución de este servicio, dependiendo de su uso (y por razones de seguridad) puede asignarse a un usuario específico, como es el caso de root. La sexta columna indica la ruta del programa servidor. Existe una forma de proteger el sistema de ciertos ataques cuando ésta se reemplaza por la ruta en la que se encuentra el tcpd (mas adelante hablaremos un poco de la función del tcpd). Finalmente, la última columna, es la que especifica el programa servidor (el servidor real) con los argumentos requeridos cuando se ejecuta.

El tcpd es un programa que substituye la ruta del programa servidor en el archivo de configuración del programa inetd (/etc/inetd.conf). Lo que hace principalmente, es proteger y monitorear los servicios de red informando de la petición del servicio al demonio correspondiente y revisa si el nodo remoto está autorizado para tal servicio; una vez que ha verificado esto, ejecuta el servidor real. Una observación importante es que este programa solamente se puede usar cuando se trata de servicios basados en tcp.

Como dijimos anteriormente, en el archivo /etc/services se indican los servicios que se encuentran disponibles en el sistema y algunos detalles específicos para cada uno de ellos como el puerto por el cual se atiende y el protocolo que utiliza. Cada programa de red debe consultar este archivo para obtener el número de puerto y protocolo que tiene definido para un servicio específico. Adelante se muestra una parte de este archivo.

```
# services    This file describes the various services that are
#             available from the TCP/IP subsystem.  It should be
#             consulted instead of using the numbers in the ARPA
#             include files, or, worse, just guessing them.
#
# Version:    @(#)/etc/services   2.00   04/30/93
#
# Author:     Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#

tcpmux        1/tcp                                 # rfc-1078
echo          7/tcp
echo          7/udp
discard       9/tcp         sink null
discard       9/udp         sink null
systat        11/tcp        users
```

```
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource            # resource location
name            42/udp          nameserver
whois           43/tcp          nicname             # usually to sri-nic
domain          53/tcp
...
```

En la primera columna se especifica el nombre del servicio. En la segunda (compuesta por dos campos) el número de puerto por el cual se atiende, así como el protocolo que debe usarse. La tercera columna indica el alias disponible (si lo tiene) y opcionalmente, pueden agregarse comentarios en la misma línea precedidos por un #.

Como se ve, las aplicaciones y los servicios de red en Linux tienen archivos de configuración de los que obtienen información necesaria para su ejecución como rutas de programas, argumentos, etc. Hay varios archivos de configuración que tienen que mantenerse sincronizados para funcionar correctamente.

La configuración de la red puede ser un proceso que se dificulta un poco cuando se hace "a mano" y para hacerlo más sencillo existen algunas herramientas que sirven para configurar el sistema de red y para monitorear su funcionamiento. Generalmente, cada distribución de Linux se acompaña de sus programas correspondientes con los que se puede hacer esto de manera sencilla.

Aquí hablaremos de tres de ellas que por su importancia sobresalen de las demás. La primera se llama ifconfig, la segunda route y la tercera netstat. Las dos primeras son independientes y tienen funciones específicas, pero se complementan una a otra para mantener en sincronía el sistema de red, mientras que la tercera se usa fundamentalmente para obtener datos cuantitativos del funcionamiento de la red. Aquí veremos cuál es su funcionamiento básico, así como la forma en la que se usa cada uno de ellos.

ifconfig permite al núcleo accesar a las interfaces que se encuentran conectadas al sistema. Es un programa que permite configurar las interfaces del sistema por medio de ciertos parámetros, así como proporcionar la información relativa a las mismas. Se utiliza de la siguiente manera:

```
$ ifconfig interfaz [-net | -host] dirección [parámetros]
```

Cuando se utiliza sin argumentos proporciona información de la interfaces de red del sistema que se encuentran activas, cuando se utiliza el parámetro –a se muestra la información relativa a todas las interfaces de red (activas y no activas), cuando se utiliza como parámetro el nombre de una interfaz del sistema, se muestra la información correspondiente a la misma.

ifconfig recibe como parámetro la familia de direcciones que se utiliza en la configuración de las interfaces. La familia de direcciones identifica el formato que se empleará para ubicar los equipos correspondientes. Actualmente, ifconfig identifica las siguientes familias de direcciones:

inet	default para TCP/IP
inet6	IPv6
ax25	paquetes de radio AMPR
ddp	Appletalk Phase 2
ipx	Novel IPX
netrom	paquetes de radio AMPR

El tipo de familia de direcciones tiene que especificarse antes que el nombre de la interfaz para configuración de las interfaces para poder ser asignada a la misma.

El nombre de la interfaz lo especifica el sistema. Una interfaz que merece especial atención es la de loopback que se identifica como *lo*. Esta interfaz permite hacer pruebas del sistema y con ella se pueden realizar conexiones al sistema emulando la transferencia de datos a través de una red común.

Otras interfaces que se especifican por el sistema son las de tipo Ethernet y que se identifican con eth*x*, en donde *x* es un dígito para identificar cada una de ellas. La primera (en el caso en el que existan varias) se identifica como eth0 y las posteriores con números ascendentes.

Con ifconfig se puede activar o desactivar interfaces de red utilizando simplemente parámetros como up y down. Así, podemos poner en funcionamiento (o no) una de ellas sin tener que desmontarla físicamente del sistema.

Es también con ifconfig con el que se establecen (o eliminan) las direcciones IP para las interfaces. Estas direcciones pueden encontrarse en formato de notación decimal puntuada (IPv4), notación hexadecimal (IPv6) y otras dependiendo de la selección de la familia a la que pertenezcan. Sucede lo mismo con la máscara de red correspondiente.

Una de las ventajas que sobresalen de las nuevas implementaciones de esta herramienta de red es la de configuración de túneles de IPv6 sobre IPv4. Con esto se permite enviar paquetes IPv6 a través de redes IPv4.

route es un programa que permite añadir y eliminar rutas a la tabla de ruteo del kernel. Puede recibir varios parámetros, pero básicamente se requieren dos, su uso se muestra a continuación:

```
$ route [add | del] destino
```

Como puede verse, la primera opción permite agregar una dirección a la tabla de ruteo, mientras que la segunda permite eliminarla. El destino es el que especifica la dirección a la cual se está haciendo referencia cuando se usa este programa de configuración.

Opcionalmente pueden indicarse otros datos como los que hacen referencia a los gateways que se encuentran entre las redes y otras opciones, pero estos requieren un análisis más específico.

Las interfaces que se han configurado mediante el programa ifconfig podrán utilizarse para la comunicación del sistema en la red cuando se tenga una entrada correspondiente a la tabla de ruteo del kernel. Se ve entonces claramente que estos dos programas de configuración de la red tienen que trabajar sincronizadamente para mantener al sistema en funcionamiento.

netstat es una herramienta que permite verificar la configuración de la red en el sistema así como la actividad de la misma. Muestra datos referentes a las conexiones de la red, la tabla de ruteo del sistema y estadísticas de las interfaces. Para ello recibe varios parámetros que permiten especificar el tipo de información que se requiere.

Puede mostrar el estado de los puertos asignados a servicios específicos, la tasa de transferencia de paquetes por cada interfaz, etc.

netstat es una herramienta muy útil ya que permite detectar fallas en los servicios de red de manera independiente, de esta manera puede trabajarse en problemas específicos y se reduce el campo de búsqueda para la detección de errores en la configuración.

Podemos decir que con la ayuda de las herramientas descritas anteriormente puede mantenerse en buen estado el funcionamiento de la red y por lo tanto, de los servicios y aplicaciones para la misma. Las aplicaciones específicas de la red basan su funcionamiento en la misma; por lo tanto, el buen estado de la red es determinante para las mismas.

Existen varias aplicaciones para redes implementadas con Linux, pero las más importantes por su uso frecuente y por ser de carácter básico en la comunicación entre redes de computadoras son telnet y ftp. Estas aplicaciones pueden servir de base o ejemplo de configuración para las demás. En el capítulo de Internet se vieron algunas de sus características fundamentales, en este capítulo nos enfocaremos a su uso bajo el sistema operativo Linux.

Como hemos visto, estas dos aplicaciones tienen dos programas cada uno, el cliente y el servidor. Los primeros son programas estándares que no requieren demasiada profundidad

en su explicación ya que para funcionar de manera correcta solamente requieren del buen funcionamiento de la red así como de una configuración eficiente de los programas servidores a los cuales se realizará la conexión.

Los programas servidores, a diferencia de los clientes, requieren de una configuración especial para poder aceptar las conexiones al sistema y por supuesto, atenderlas de manera correcta.

Telnet y Ftp son dos servicios de red que se encuentran incluidos en el archivo de configuración /etc/services del sistema. De este archivo podemos ver que:

servicio	puerto	Protocolo
Ftp	21	tcp
Telnet	23	tcp

Como hemos visto, estos servicios son levantados por el "super-servidor" inetd que en su archivo de configuración (inetd.conf) nos muestra los siguientes datos:

servicio	tipo de socket	protocolo	estado	usuario	Servidor	Servidor parámetros
ftp	stream	tcp	nowait	root	/usr/sbin/tcpd	in.ftpd –l
telnet	stream	tcp	nowait	root	/usr/sbin/tcpd [3]	in.telnetd

La ubicación de los programas cliente y servidor de estos programas se encuentra en:

servicio	cliente	Servidor
ftp	/usr/bin/ftp	/usr/sbin/in.ftpd
telnet	/usr/bin/telnet	/usr/sbin/in.telnetd

El ftpd es el programa servidor para la transferencia de archivos a través de la red. Este programa puede ser invocado por el inetd o ejecutarse de manera independiente y se mantiene escuchando el puerto 21 para detectar conexiones. Cuando se ejecuta, pueden asignarse ciertos parámetros al mismo para especificar funciones durante el proceso de ejecución del mismo.

El tipo de socket que tiene asignado este servicio es el de stream ya que se trata de un servicio que utiliza un protocolo orientado a conexión, el tcp. Como se trata de un socket de tipo stream, el estado de espera del servicio se especifica como nowait, de esta manera se

[3] En una sección anterior de este capítulo de explicó la razón de cambiar el programa servidor por tcpd.

mantendrá pendiente el servidor para atender varias conexiones simultáneamente. Debido a las características que este servicio presenta, tiene que ser asignado como usuario determinado root. Se utiliza el tcpd en lugar del servidor real para poder realizar el proceso de autorización del servicio. Finalmente, existen ciertos parámetros que son opcionales; una breve descripción de los mismos se tiene a continuación:

-d (debug) la información se escribe en un archivo de registro syslog.
-l cada conexión aceptada o rechazada es registrada en el syslog.
-T determina el tiempo máximo de conexión, 2 horas por default.
-t determina el periodo de inactividad, 15 minutos por default
-S solicita información de seguridad para las conexiones

Para deshabilitar el acceso por ftp se usa el archivo /etc/nologin, cuando este archivo existe, muestra el mensaje y sale del programa servidor automáticamente. Una vez que se ha establecido la comunicación y autorizado el uso del servicio, se muestra un mensaje de bienvenida al usuario (si está escrito en el archivo /etc/ftpwelcome).

El programa servidor de ftp responde a varias peticiones sobre el servicio provenientes del programa cliente, la siguiente lista muestra cada una de ellas [4]:

```
Request      Description
ABOR       abort previous command
ACCT       specify account (ignored)
ALLO       allocate storage (vacuously)
APPE       append to a file
CDUP       change to parent of current working directory
CWD        change working directory
DELE       delete a file
HELP       give help information
LIST       give list files in a directory (``ls -lgA'')
MKD        make a directory
MDTM       show last modification time of file
MODE       specify data transfer  mode
NLST       give name list of files in directory
NOOP       do nothing
PASS       specify password
PASV       prepare for server-to-server transfer
PORT       specify data connection port
PWD        print the current working directory

QUIT       terminate session
REST       restart incomplete transfer
RETR       retrieve a file
RMD        remove a directory
RNFR       specify rename-from file name
RNTO       specify rename-to file name
SITE       non-standard commands
SIZE       return size of file
STAT       return status of server
STOR       store a file
STOU       store a file with a unique name
```

[4] man ftpd

```
STRU        specify data transfer structure
SYST        show operating system type of server system
TYPE        specify data transfer  type
USER        specify user name
XCUP        change to parent of current working directory(deprecated)
XCWD        change working directory (deprecated)
XMKD        make a directory (deprecated)
XPWD        print the current working directory (deprecated)
XRMD        remove a directory (deprecated)
```

Las solicitudes de este servicio son autorizadas por el programa servidor de las siguientes formas:

1. verificando que el nombre de usuario se encuentre en el registro de usuarios del sistema (/etc/passwd) y que tenga una clave definida.
2. verifica que el usuario no se encuentre registrado en el archivo /etc/ftpusers
3. verifica que el usuario use un shell estándar
4. si el nombre del usuario es "ftp" o "anonymous", se le asigna como directorio raíz el que se encuentra en /home/ftp, se le restringe el acceso a los demás directorios y se solicita un password que por convención es el e-mail del usuario.

Los directorios principales que se encuentran bajo ftp son: /home/ftp/etc, /home/ftp/bin y /home/ftp/pub. Los dos primeros no tienen permisos de escritura y pertenecen a root, mientras que el segundo es el que se tiene definido para realizar las transferencias de archivos que se tienen a disposición de los usuarios externos.

telnetd es el programa servidor que se mantiene escuchando el puerto 23 para atender solicitudes de acceso remoto al sistema. También puede ser inicializado por el inetd o de forma independiente; además, puede o no recibir parámetros especiales. Las características de este servicio son básicamente las mismas que las del servidor de ftp.

El tipo de socket que tiene asignado este servicio es el de stream ya que se trata de un servicio que utiliza un protocolo orientado a conexión, el tcp. Debido a que se trata de un socket de tipo stream, el estado de espera del servicio se especifica como nowait, de esta manera se mantendrá pendiente el servidor para atender varias conexiones simultáneamente. Por las características que este servicio presenta, tiene que ser asignado como usuario determinado root. Se utiliza el tcpd en lugar del servidor real para poder realizar el proceso de autorización del servicio. Este servidor puede aceptar los siguientes parámetros:

-a se usa para especificar el modo de autenticación que se realizará. Puede tomar cualquiera de las siguientes opciones:
debug habilita el modo debug para la autenticación
user permite la conexión cuando el usuario remoto proporciona información válida para identificarse, se autoriza sin solicitud de password.

valid permite la conexión cuando el usuario remoto proporciona información válida para identificarse.

other permite la conexión si se proporciona cierta información de autenticación.

none es la configuración por default. No requiere información de autenticación, el programa login realiza la verificación de los datos

off deshabilita el código de autenticación. Cualquier verificación de usuario pasa a través del programa login.

-d modo debug

Permite ver que es lo que está haciendo el programa servidor. Puede usarse con algunas opciones que son:

options muestra la información sobre las opciones de negociación de la conexión

report complementa la información que se obtiene con options y agrega información adicional sobre los procesos que se realizarán.

netdata muestra el flujo de datos recibidor por telnetd

-edebug cuando el servidor de telnetd ha sido compilado con la opción de encriptación, esta opción permite la encriptación del código de debug

-h deshabilita la opción que muestra información específica del host cuando se realiza la conexión

-l se usa para especificar un programa de acceso diferente a /bin/login

-n inhabilita la opción para mantener "viva" la conexión con máquinas que se mantengan inactivas en un periodo de tiempo determinado.

-s puede habilitarse solamente si telnetd fue compilado con la opción para manejar SecurID cards. Permite que se le pase el parámetro –s al login. Es funcional cuando se realiza la conexión con equipos que se encuentran fuera del firewall.

-S establece la IP que identifica al tipo de servicio

-X esta opción solamente puede usarse si telnetd se compiló con la opción para autenticación, inhabilita el uso de la autenticación por tipo.

Si existe el archivo /etc/issue.net, telnetd muestra su contenido antes de proporcionar el prompt de sistema al usuario que accesa al mismo usando este servicio.

Lo que hace telnetd el desplegar una terminal virtual por cada cliente y después lanzar un proceso de acceso (login) para verificar los datos del usuario. Todos los mensajes de entrada, salida y de error son direccionados hacia la terminal virtual que se creó para el cliente específico, esto permite establecer varias conexiones de acceso remoto al mismo sistema y mantenerlas independientes.

Cuando se establece la conexión entre los programas telnet (cliente y servidor) hay un intercambio de opciones en forma de mensajes entre ellos, la siguiente lista muestra los mensajes del programa servidor al cliente [5]:

```
DO AUTHENTICATION
WILL ENCRYPT
DO TERMINAL TYPE
DO TSPEED
DO XDISPLOC
DO NEW-ENVIRON
DO ENVIRON
WILL SUPPRESS GO AHEAD
DO ECHO
DO LINEMODE
DO NAWS
WILL STATUS
DO LFLOW
DO TIMING-MARK
```

La primera instrucción indica al cliente que tiene que realizar la autenticación del usuario, después se activa el modo encriptado para la información que se procese en ese momento. Posteriormente pide que le sea enviada la información relativa al tipo de terminal, la velocidad de la linea serial, el nombre del sistema x window, datos de las variables de ambiente; todos los datos anteriores son los del cliente, de esta manera, el servidor sabrá como tratar la información para que se transmita de manera eficiente.

Las opciones anteriores no son las únicas ni se encuentran en una secuencia en la que deban tenerse en todas las conexiones, son aquellas que determinan el funcionamiento básico del servidor telnetd y por lo tanto se eligieron para complementar la explicación.

Como puede verse, estas aplicaciones tienen mucho en común. La forma en la que trabajan sus servidores es muy similar e incluso los archivos de configuración y aquellos en los que se muestra información de los mismos. Algunas otras aplicaciones para redes en Linux tienen características similares ya que su funcionamiento es un tanto general.

[5] man telnetd

IPv6 PARA LINUX

Introducción

Todas las capacidades que se tienen con IPv6 en comunicaciones en red pueden ser aprovechadas por un sistema Linux. Linux es un sistema operativo que permite la implementación de este protocolo para realizar sus trabajos en la red.

IPv6 puede ser implementado solamente en algunas distribuciones de este sistema operativo. Algunas versiones de estas distribuciones tienen la opción para configuración durante la instalación del sistema, algunas otras aceptan la configuración posterior a la instalación de dos formas diferentes: mediante paquetes o actualización y configuración de archivos individualmente por parte del usuario (o administrador) del sistema.

La implementación más sencilla es la que se hace desde la instalación; solamente hay que responder algunas preguntas al programa de instalación acerca de las opciones que se presentan. Cuando se instalan los paquetes de actualización sucede algo parecido, la diferencia se encuentra en que tienen que encontrarse los paquetes necesarios para mantener estable el sistema. Finalmente, la configuración por actualización de archivos (de programa y de configuración) es la que requiere un mayor esfuerzo, ya que requiere de un análisis detallado de algunos archivos y de conocimientos previos sobre configuración de los servicios de red, así como de las herramientas de configuración de la misma.

En este capítulo veremos la forma en la que puede configurarse Linux Red Hat 6.0 actualizando los recursos del sistema; es decir, posterior a la instalación del mismo, daremos datos generales para su configuración con las otras distribuciones y veremos los elementos necesarios para esta configuración.

En la configuración del sistema se requieren de ciertos elementos que se hacen necesarios y en ocasiones indispensables. Estos elementos son básicamente de software, mediante los cuales se realiza la selección de las opciones de configuración del sistema y actualización de las herramientas del mismo.

Requerimientos de software

Antes de comenzar con la actualización (o migración) del sistema hacia IPv6, es necesario verificar que el sistema cuenta con los elementos indispensables para esto. Podemos decir que hay tres requisitos de software que son necesarios como "base" para formar el sistema que utilice IPv6. Estos elementos son:

- Una distribución que soporte IPv6
- Un kernel con el que pueda configurarse el sistema con IPv6
- Librerías actualizadas para configuración del sistema

Como vimos anteriormente, son pocas las distribuciones que ofrecen la capacidad de configuración con IPv6. La distribución con la que trabajaremos será Red Hat. Lo que se

necesita primero es tener instalado el sistema como servidor; es decir, es necesario seleccionar la instalación por default que se nos presenta como opción al momento de la instalación, este modo presenta una gran ventaja y es que aquí no se hace necesario seleccionar individualmente los paquetes que se quieren instalar en el sistema (con la posibilidad de realizar una instalación incompleta) sino que se tiene este modo de instalación prácticamente automático.

Cualquier distribución de Linux se presenta disponible con la última versión estable del kernel que se encuentra en el momento en el que se liberó. Hay que tener en mente que no todas las versiones del kernel soportan la implementación para IPv6. Tenemos que recordar que este tipo de configuración es muy reciente por lo que solamente algunos de ellos tienen esta capacidad.

Las librerías son archivos necesarios en el momento de la compilación de algunos programas (evidentemente, antes de su ejecución). Estas librerías se vuelven indispensables ya que de éstas depende que los programas puedan ser compilados. Para la implementación de IPv6 en este sistema es necesario contar con el grupo de librerías contenidas en las *Glibc-2.1*.

Como podemos ver, este grupo de librerías contiene todo lo necesario para compilar los programas de instalación (o actualización) que se requieren. Es un conjunto de librerías recientes que solamente se encuentra en las versiones posteriores a la 6.0 en la distribución Red Hat. Estas librerías son indispensables para esta configuración del sistema.

Finalmente, se requieren de las aplicaciones que se encuentran diseñadas para trabajar usando IPv6. Estas aplicaciones igual que las demás opciones han surgido recientemente, por lo que se hace necesario conseguirlas. Existe la posibilidad de conservar las aplicaciones de IPv4 independientes de las de IPv6, pero también existe una forma de tenerlas coexistiendo y dejando al sistema reconocer cual de las dos es la que tiene que usar empleando mecanismos de direccionamiento para tomar esta decisión.

Distribución

Son pocas las distribuciones que ofrecen la posibilidad de configuración del sistema usando IPv6, específicamente son cuatro:

- Debian
- Polish Linux
- Red Hat
- S.u.S.E.

Estas distribuciones tienen características similares en cuanto al manejo de las herramientas y los servicios de red, pero se mantienen independientes unas de otras. Existe la posibilidad

de configuración del sistema desde el momento de la instalación del mismo o realizando una actualización como hemos dicho.

En esta tesis nos enfocaremos a la distribución Red Hat en su versión 6.0, más adelante veremos como puede configurarse el sistema, así como los detalles necesarios para ello.

Kernel

El kernel es la parte medular del sistema operativo Linux, es literalmente el núcleo del mismo. Es el que se encarga de establecer la comunicación entre los dispositivos y el sistema en si. Para poder hacerlo tiene que contar con los recursos necesarios de software para tener acceso al hardware del sistema.

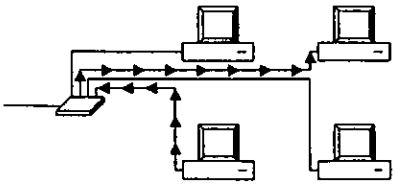
Se mantiene actualizado por un pequeño grupo de programadores entre los que se encuentra el mismo Linus Torvalds. Frecuentemente pueden obtenerse versiones recientes en las que se han agregado nuevas características y corregido algunas fallas de las versiones anteriores, estas pueden consistir en simples optimizaciones o nuevos desarrollos.

Para la implementación de IPv6 en un sistema Linux, tenemos que contar con una versión reciente del kernel. Específicamente, podemos decir que la versión inicial (estable) con la que se puede tener acceso a las características de IPv6 es la versión 2.2.0. Con esto, podemos afirmar que las versiones posteriores también cuentan con estas características.

Es importante señalar que hasta la última versión disponible (2.2.14) las opciones de configuración que se encuentran en el kernel (aún siendo una versión estable del mismo) indican que lo relacionado a IPv6 se encuentra en una fase experimental. Esto es porque aún se requieren de ciertos ajustes en la configuración del kernel para poder tener una actualización transparente.

Estas opciones de configuración son particulares para la comunicación del sistema utilizando IPv6, en estas opciones se integran módulos para establecer la forma en la que la comunicación debe realizarse, además de algunos datos necesarios para configuración de hardware, así como medios para la transición entre protocolos.

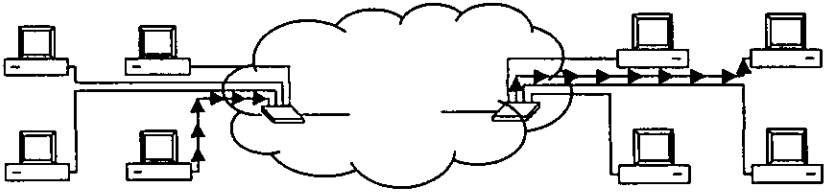
Podemos decir que la transición en un sistema Linux se realiza utilizando túneles para la comunicación con otros sistemas que se encuentran fuera de la red. Cuando se tienen equipos trabajando en la misma red local y utilizando IPv6, pueden comunicarse mutuamente sin ningún problema. Cuando los paquetes viajan a través de la red y son recibidos por un equipo que no reconoce paquetes IPv6, éste los deja continuar su camino por la red, no los toma. De esta manera los únicos equipos que pueden accesar a estos paquetes son aquellos que han sido configurados para esto.



Transmisión de paquetes IPv6 en una red local

El problema surge cuando los paquetes tienen que pasar a través de un ruteador para poder llegar a su destino, es entonces cuando se hace necesaria la implementación de un túnel para la comunicación entre los sistemas. Esto se debe a que las redes se comunican con IPv4 puro y a que los paquetes que se envían pertenecen a IPv6. Si el ruteador no reconoce los paquetes IPv6 (que es lo que generalmente ocurre en la actualidad ya que aún no se cuenta con ruteadores con estas características en las redes comunes) decide no transmitirlos a otras redes, por lo que los paquetes pueden seguir moviéndose dentro de la misma red local, pero no salir de ella.

Para poder establecer la comunicación con otras redes que manejen IPv6, los paquetes tienen que salir usando un túnel de IPv6 sobre IPv4.



Transmisión de paquetes IPv6 fuera de la red local

Posteriormente se tratará con detalle la configuración de túneles IPv6 sobre IPv4.

Paquetes

Los paquetes son conjuntos de archivos de programa, de configuración y de datos que hacen posible la implementación de ciertos componentes de software en el sistema. Algunas distribuciones tienen la posibilidad de realizar actualizaciones a su sistema mediante paquetes. Esto representa una gran ventaja para el administrador del sistema (o para el usuario en general cuando se trata de programas personales) ya que evita la

modificación de archivos de configuración (y por supuesto el conocimiento de los mismos) realizando las los cambios necesarios de manera automática.

La implementación de IPv6 en el sistema Linux requiere de ciertos paquetes instalados en el mismo, si no se cuenta con ellos el sistema no trabajará correctamente o incluso, dejará de funcionar. En esta sección veremos cuales son los paquetes indispensables que debe tener el sistema para poder realizar su actualización hacia IPv6.

modutils es un paquete opcional para la configuración del sistema con IPv6. Contiene los elementos necesarios para la ejecución del kerneld. Este es un demonio que se ejecuta cuando el sistema arranca permitiendo al kernel cargar los módulos con los que ha sido configurado. Frecuentemente pueden configurarse ciertos componentes del sistema para ser cargados como módulos por el kernel, esto permite una optimización en la forma en la que se administran los recursos del sistema. Cabe mencionar que este paquete es opcional ya que el kernel puede ser configurado monolíticamente; es decir, sin módulos externos, de esta manera se hace innecesario el uso del mismo. La última versión que se tiene es la modutils-2.1.121.

Las versiones de Linux que se distribuyen con el kernel anterior al 2.1.*x* cuentan con este paquete en versiones que no soportan la migración hacia IPv6, por esto se hace necesario verificar la versión del mismo.

Los paquetes *libtermcap-devel, ncurses-devel, XFree86-devel, libgr-devel, libpng-devel* y *zlib-devel* son paquetes que contienen encabezados y bibliotecas en C para que programas hechos en este lenguaje los utilicen en el momento de la compilación. Estos paquetes son indispensables para la actualización del protocolo ya que se requieren en el momento de la compilación de las funciones que serán agregadas.

También es necesario tener instalado el paquete *mc*. *mc* es el midnight commander que es un programa manejador de archivos en modo texto que ofrece ventajas para el manejo de los mismos como la compresión, descompresión, agrupación en archivos .tar, etc., también permite el uso del mouse para estas funciones.

Librerías

Las librerías son archivos que tienen encabezados y definiciones de funciones que utilizan los programas cuando son compilados para integrar estas funciones como parte del programa mismo. Los programas que las usan, dependen de estas librerías en tiempo de compilación para poder ser compilados correctamente con todos los elementos necesarios para poder efectuar las funciones y tareas para las que están hechos.

En el proceso de migración hacia IPv6, algunas librerías son de especial importancia, ya que hay programas que serán compilados nuevamente para poder agregarse nuevas funciones o características, y cuando esto sea hecho, se requiere de las mismas.

Aquí veremos cuales son las librerías indispensables con las que tiene que contar el sistema para poder realizar la actualización hacia IPv6.

libtermcap-devel contiene las librerías necesarias para accesar a la base de datos del sistema llamada termcap, se usa para tener acceso a los datos referentes a la presentación de caracteres por el sistema en las terminales cuando se compila o editan programas.

ncurses-devel es un paquete que contiene encabezados y librerías necesarios para la compilación de programas relacionados con el uso y manejo de caracteres en terminales independientes del sistema de manera optimizada.

XFree86-devel es un paquete que contiene todo lo necesario para el desarrollo de programas referentes a los clientes X (ambiente gráfico) del sistema. Tiene las librerías y encabezados necesarios para la compilación de estos programas.

libgr-devel contiene todo lo necesario para el desarrollo de programas que manejan diferentes formatos gráficos del sistema.

libpng-devel tiene las librerías y encabezados necesarios para el desarrollo de programas que utilicen ambiente gráfico y que estén relacionados con las herramientas de red PNG (Portable Network Graphics).

zlib-devel es la librería que contiene los archivos relacionados con la compresión y descompresión de archivos del sistema.

Las versiones de estas librerías con las que puede tenerse el sistema estable son:

- libgr-devel-2.0.13-17
- libpng-devel-1.0.3-2
- libtermcap-devel-2.0.8-13
- ncurses-devel-4.2-18
- Xfree86-3.3.3.1-49
- zlib-devel-1.1.3-5

Aplicaciones

Como hemos visto, las aplicaciones son los programas que se utilizan con fines específicos. Las aplicaciones de red para Linux que utilizan IPv6 son pocas aún, prácticamente son las aplicaciones de red que vienen implementadas en el sistema operativo desde su distribución.

Lo que es necesario, es realizar una actualización de las mismas, ya que no están diseñadas para trabajar con los paquetes de IPv6. La actualización o migración es un proceso relativamente sencillo que puede dividirse en dos partes.

La primera de ellas consiste en la búsqueda de las versiones actualizadas de estos programas de aplicación. Puede ser un tanto difícil conseguir algunas de ellas, pero la mayoría ya se encuentran disponibles en sitios FTP comunes a Linux, uno en particular es: ftp://ftp.bieringer.de/pub/Linux/IPv6.

La segunda consiste en la reinstalación de las aplicaciones modificando el código de las mismas mediante un patch (parche) antes de la compilación de las mismas; con estose actualiza el código del programa con elementos más recientes y que son necesarios para adaptarlo a IPv6.

Los principales programas de aplicaciones que tienen que ser actualizados, así como una breve descripción de los mismos se encuentra en los siguientes párrafos; posteriormente, en la siguiente sección veremos la metodología para la configuración del sistema incluyendo la instalación de estas aplicaciones.

Son muchas las aplicaciones que pueden ser utilizadas con IPv6 (cuando están diseñadas para hacerlo); hay aplicaciones de tipo general que son las que se utilizan en funciones generales y no están orientadas hacia funciones específicas que son las que se realizan con las aplicaciones particulares.

Algunas aplicaciones que pueden considerarse como específicas son las que se utilizan para el manejo y monitoreo de la red con IPv6. Hay un grupo de herramientas conocido como net-tools con las cuales se tiene acceso a estas aplicaciones. La última versión es la 1.54. Este grupo tiene las siguientes herramientas:

- hostname
- netstat
- arp
- rarp
- ifconfig
- route

hostname es una herramienta para mostrar o establecer el nombre que tiene un host en particular. El nombre del host es con el cual se identifica al sistema en la red, puede consultarse o modificarse con este programa. Tiene varios parámetros con los cuales puede obtenerse información relativa y específica acerca del host como son la dirección ip, el servidor de nombres que está configurado, el dominio del mismo y el alias con el que se conoce.

netstat es un programa con el que se puede monitorear el estado de la red. Muestra las conexiones activas del sistema en la red, el tipo de servicio que se esta prestando o al cual se está accediendo, los puertos por los cuales se esta estableciendo la comunicación, el estado de las conexiones, etc.[1]

[1] En el capítulo de Linux se trata con más detalle.

arp es un programa que permite la administración del cache ARP usando el protocolo ARP (Address Resolution Protocol). Con él puede vaciarse el contenido de la tabla de direcciones del cache o agregarse nuevas direcciones mapeadas. Trabaja específicamente sobre direcciones ethernet de los equipos que se encuentran en la red, aunque también incluye los valores de las direcciones IP correspondientes, muestra también información relativa a la interfaz de red de la que se trata.

rarp manipula la tabla rarp del sistema que utiliza el RARP (Reverse Address Resolution Protocol). Esta tabla tiene información que permite la resolución inversa de direcciones.

ifconfig es un programa que se usa para la configuración de las interfaces de red en el sistema. Con él pueden consultarse o establecerse datos como direcciones IP (IPv4 e IPv6) para las mismas, direcciones ethernet (solamente consulta), datos de la máscara de red, y algunos otros datos. [1]

route es una herramienta con la que se manejan las tablas de ruteo del sistema, con él pueden agregarse y eliminarse direcciones de manera que se mantenga actualizada la tabla y que el kernel del sistema pueda reconocerlas.[1]

Existen también aplicaciones de herramientas de depuración. Estas se orientan hacia otras aplicaciones de red en las que se implementa IPv6. Básicamente son tres:

- libpcap
- TCPdump
- traceroute

libpcap (Packet Capture Library) es una interface independiente del sistema que se usa para capturar paquetes a nivel de usuario. Proporciona la base para el trabajo de monitoreo de la red. Varias herramientas orientadas hacia aplicaciones de red como monitoreo, estadísticas, etc. requieren de una interfaz común que no proporcionan los diferentes proveedores de equipo, por esta razón es que se usa esta librería. La última versión que se encuentra disponible es la libpcap-0.4a6+ipv6-1.

TCPdump es un conjunto de programas que se utilizan como herramienta para el monitoreo de red y la adquisición de datos. Para poder realizar su trabajo requiere de una interfaz independiente del sistema, por lo que requiere que estén instalados los componentes de libcap. Con TCPdump se tiene acceso a los paquetes de la transmisión de datos dentro de la red, así es como puede analizarse y monitorearse el funcionamiento de la misma. La última versión disponible es la tcpdump-3.4a6+ipv6-1.

traceroute es una herramienta de monitoreo de red que se usa para rastrear los paquetes ip desde que son enviados hasta que llegan a su destino. Debido a que utiliza sockets ip directamente tiene que ser ejecutado forzosamente como root o tener la asignación de root durante la ejecución (con setuid). La última versión que se tiene es la traceroute-1.4a5+ipv6-1.

Las aplicaciones generales se agrupan (para su migración hacia IPv6) en un conjunto llamado inet6-apps Actualmente se encuentra disponible en su versión 0.36 y se reconoce como inet6-apps-0.36.

Las aplicaciones incluidas en las inet6-apps son:

- finger
- fingerd
- inetd
- ping
- tftp
- libinet6

finger es un programa cliente para mostrar información de los usuarios del sistema. Es un servicio que no siempre se encuentra activado en el sistema por razones de seguridad. Con él puede obtenerse información personal de un usuario en particular o un grupo de ellos.

fingerd es el programa servidor que recibe las conexiones al puerto que este definido en el sistema (generalmente se trata del puerto 79). Este es el encargado de proporcionar la información solicitada sobre un usuario del sistema.

inetd es conocido como el super-servidor, es el encargado de monitorear los puertos del sistema, determinar cual es el servicio solicitado y ejecutarlo para así atenderlo.

ping es un porgrama que se utiliza para determinar la disponibilidad de un dispositivo en la red. Lo que hace es enviar paquetes a través de la red hacia el dispositivo determinado, estos paquetes son reenviados hacia el dispositivo fuente. De esta manera, se reciben los paquetes como una señal de eco en la red proporcionando el tiempo realizado en este proceso; así puede decirse que si la señal se pierde, el dispositivo destino es inalcanzable.

tfpt es el programa de transferencia trivial de archivos. Es un programa similar al FTP, pero no tiene las mismas capacidades y debido a que no implementa autenticación de los usuarios, normalmente es un servicio que se tiene deshabilitado en el sistema por razones de seguridad. Hay una versión más reciente para actualizar este programa que se encuentra de manera independiente de las inet6-apps, esta es la versión tftpd-1.2a1.

libinet6 es una librería necesaria para la compilación de los algunos programas de aplicación sobre IPv6.

Otros programas de aplicación que pueden ser opcionales dependiendo de la configuración que se tenga o que se busque en el sistema, como estación de trabajo o como servidor pueden ser instalados. En los párrafos siguientes veremos algunas de estas aplicaciones que pueden ser utilizadas para realizar algunas funciones específicas con lo que podremos aprovechar las ventajas del sistema operativo y su manejo de paquetes a través de la red.

Telnet y *telnetd* son los programas para acceso remoto. Se encuentran agrupados en un archivo y la última versión estable y funcional en la que las podemos encontrar es telnet.95.10.23.NE+ipv6-3. En este paquete se encuentran los dos programas, el cliente y el servidor, que funcionan correctamente.

El demonio de *POP* es el que permite atender las solicitudes externas de los usuarios del sistema para enviar datos de correo electrónico usando el Post Office Protocol [2]. Este demonio se actualiza usando el qpopper2.2+ipv6-1.

Otro programa de aplicación que tiene que ser actualizado para poder trabajar usando paquetes IPv6 es el *Sendmail*. Es un agente de transporte de correo electrónico. Envía los mensajes a uno o varios destinos a través de la red. Es el encargado de establecer las conexiones entre las redes para que un correo electrónico pueda ser enviado y recibido a (y desde) cualquier punto de la red. La versión del programa de actualización es la sendmail-8.8.8+ipv6-1.

En demonio de HTTP (el programa servidor) no puede atender las solicitudes de los clientes que solicitan información usando http, este demonio tiene que ser actualizado para poder entender los paquetes de IPv6 de las solicitudes y enviar los datos usando ese mismo formato. La versión que se encuentra disponible actualmente es la apache_1.3a1+ipv6-2.

El programa cliente para HTTP instalado en el sistema no soporta la transferencia de paquetes IPv6 ya que no los reconoce, para que lo haga, tiene que actualizarse con el nuevo programa cliente que es el chimera-2.0a14+ipv6-1.

Un demonio que se necesita cuando se quiere probar la autoconfiguración del sistema cuando se usan sistemas linux como ruteadores es el Router ADVertisement, conocido como RADVD. Con este demonio es con el que se mantienen actualizadas las tablas de ruteo ya que aprovecha la detección de los sistemas que están conectados a la red en un momento determinado, representa una ventaja para las tablas de ruteo dinámicas. La versión que debe tenerse para que trabaje con IPv6 es la radvd-0.5.0.

Otro programa que tiene que ser actualizado ya que no maneja los paquetes de IPv6 es el TTCP, que se usa para monitorear los paquetes que viajan a través de la red TCP. La versión que se encuentra disponible para actualización es la ttcp+ipv6-1.

Sucede lo mismo con el PTCP y su versión de actualización es la ptcp+ipv6-1.

Los puertos del sistema se revisan continuamente para determinar si hay una solicitud de servicio específico que atender. TCP wrappers es un paquete que permite monitorear solicitudes de peticiones para diferentes servicios de red como: SYSTAT, FINGER, FTP, TELNET, RLOGIN,RSH, EXEC, TFTP, TALK y algunos otros y asigna la petición al programa servidor correspondiente para atender al servicio después de revisar la autorización. La versión actualizada del mismo es la tcp_wrapper_7.6+ipv6-1.[3]

[2]En el capítulo de Internet se habla de este protocolo.

[3] En el capítulo de Linux, se explicó con más detalle el funcionamiento del tcpd como wrapper

Secure Shell es un conjunto de programas que utiliza el sistema para realizar labores de autenticación, cifrado y encriptamiento cuando se realizan las transmisiones de datos a través de la red. Con estos programas puede tenerse cierta seguridad para la inclusión del sistema en una red. Los programas principales (cliente y servidor respectivamente) son ssh y sshd. En éstos programas se incluyen las variables de ambiente, que son las variables con las que el sistema maneja la terminal, el display y otras características. La última versión para actualización es la ssh-1.2.27-IPv6-1.5.

Otros programas cliente y servidor que tienen que actualizarse son los de SNTP (sntpcd y sntpsd respectivamente). Estos programas son los que hacen que el sistema sea capaz de enviar o recibir multicast usando IPv4 o IPv6. La versión (actualizada y necesaria) de estos programas se encuentran en sntp+ipv6-0.91. Este conjunto de programas incluye también una aplicación para establecer el reloj del servidor NTP.

Otro conjunto de programas que tienen que actualizarse, es el que se conoce como RIPE whois tools. Están escritos en C y trabajan con una base de datos llamada RIPE. Son básicamente tres programas con los que se mantiene actualizada una base de datos que contiene información de las computadoras en la red. La última versión de estos programas se encuentra en el ripe-whois-tools-3.0.4+6bone-extensions+ipv6-1.

Configuración del sistema con IPv6

La configuración del sistema, es como hemos dicho, prácticamente una actualización, por lo que resulta ser una tarea fácil cuando se cuenta con lo necesario para emprenderla. Para poder comenzar con este proceso es necesario conocer la versión de los programas que utilizan IP para verificar su posibilidad de actualización. En la sección anterior, hablamos de las últimas versiones que se encuentran disponibles y que son necesarias para realizar estas actualizaciones.

Una vez que se cuenta con el software necesario, puede comenzarse el proceso de actualización. Estos programas podemos encontrarlos en cualquiera de los servidores FTP que se encuentran las referencias que se dan en la bibliografía. En esta sección describiremos cuál es la forma y la secuencia en la que este proceso tiene que realizarse.

Hay que aclarar que esta configuración debe ser realizada por el administrador del sistema (root) ya que son necesarios los permisos para acceder a los archivos de configuración del sistema.

Hay que tener en cuenta que para comenzar la configuración del sistema con IPv6, debemos contar con una instalación de una distribución de Linux que la soporte. En el caso de Linux Red Hat 6.0, el sistema tiene que instalarse con la opción de servidor (en el caso de que necesitemos que proporcione servicios a diferentes clientes). Puede realizarse la instalación como estación de trabajo, pero ésta se encontrará limitada para trabajar como cliente en la red y sus funciones estarán limitadas, aunque sus capacidades estarán dentro del manejo y configuración del sistema como IPv6.

Si la configuración del sistema requiere que el kernel sea compilado (en alguna de sus opciones) como módulo debemos instalar las modutils:

```
$cd /usr/src
$tar vxzf ruta/modutils-2.1.121.tar.gz -C .
$ln -sf  modutils-2.1.121 modutils
$./configure
$make clean
$make
$make install
$reboot
```

El kernel se compila de la forma usual una vez que se han definido las opciones de configuración correspondientes a las capacidades de IPv6. Sin estas opciones de configuración, el kernel no podría emplear las funciones de IPv6 aunque se encuentren instaladas. Hay que recalcar que dentro de las opciones de configuración para el kernel, también deben considerarse las características propias del equipo en el que se tiene el sistema. Es importante no olvidar la configuración de éstas cuando se incluyan las opciones para IPv6.

Lo primero que hay que hacer en cuanto al kernel es conseguir una versión reciente (igual o superior a la 2.2.0 para que soporte IPv6). Normalmente se encuentra en versiones empaquetadas y comprimidas, en un solo archivo con extensión tar.gz; este archivo tiene que ser descomprimido en el directorio /usr/src para poder ser configurado, compilado e instalado.

Antes de comenzar a trabajar en la configuración del kernel debemos explicar algunas cosas. Los archivos con los que se puede configurar e instalar el kernel se encuentran en el directorio /usr/src/linux. Durante el proceso de configuración, compilación e instalación del kernel, el sistema buscará los archivos necesarios en el directorio mencionado.

Una buena práctica de administración del sistema es ubicar los archivos correspondientes para una versión en particular del kernel en un solo directorio para poder hacer referencia al mismo cuando sea necesario. Así, antes de descomprimir y desempaquetar un nuevo kernel, es recomendable cambiar el nombre del directorio (si existe) /usr/src/linux por /usr/src/linux-xx.yy.zz, en donde linux-x.yy.zz es la versión correspondiente del kernel que puede encontrarse tecleando

```
$uname -a
```

Ahora hay que crear la liga hacia el directorio que acabamos de renombrar con

```
$ln -sf /usr/src/linux-xx.yy.zz /usr/src/linux
```

con esto mantenemos estable el sistema y podemos entender mejor como se ha ido formando la estructura de archivos en este directorio.

Ahora, si contamos ya con el nuevo kernel al que queremos actualizar en algún directorio del sistema, podemos comenzar. Es necesario que nos encontremos en el directorio en donde se encuentran los códigos fuente para configurar el sistema, borrar la liga hacia el directorio del kernel anterior, descomprimir, desempaquetar, cambiar el nombre del directorio, crear la liga a éste último de la siguiente manera:

```
$cd /usr/src
$rm linux
$tar vxzf ruta/linux-xx.yy.zz.tar.gz -C .
$mv linux linux-xx.yy.zz
$ln -sf linux-xx.yy.zz linux
```

La configuración del kernel puede realizarse usando diferentes formatos, el de modo texto, el de cuadros de diálogo o el modo gráfico. Todos estos formatos representan las mismas opciones de configuración pero difieren en la forma en la que interactúan con el usuario. Para poder accesar al formato deseado (en el orden descrito) es necesario teclear sólo una de las siguientes opciones:

```
$make config
$make menuconfig
$make xconfig
```

Hay que mencionar que es conveniente guardar la configuración actual del kernel en un archivo por si es necesario emplearla por alguna falla del sistema. Esto puede hacerse desde el ambiente para configuración del kernel que se haya escogido o desde línea de comandos con:

```
$cp /usr/src/linux/.config /usr/src/configxx
```

En donde *xx* puede ser un valor cualquiera para la identificar la configuración actual del kernel.

En la siguiente sección veremos cuales son las opciones de configuración del kernel (2.2.14) indispensables para el funcionamiento correcto del sistema.

Networking options	Packet socket	yes o modulo
	Unix domain sockets	yes
	TCP/IP networking	yes
	IP: tunneling	yes
	The IPv6 protocol	yes o modulo
	IPv6: enable EUI-64 token format	yes
	IPv6: disable provider based address	yes
Code maturity label options	Prompt for development and/or incomplete code/drivers	yes
Loadable module support	Enable loadable module support	no
Console Drivers	Video mode selection support	yes
File systems	/proc filesystem support	yes
Kernel hacking	Magic sysrq key	yes

En los siguientes párrafos se da una breve descripción de las opciones de configuración relacionadas con IPv6 y se justifica la razón por la que la opción es seleccionada.

Packet socket (networking options). Es utilizado por ciertas aplicaciones que no utilizan un protocolo intermedio para comunicarse cuando realizan transferencias de paquetes a través de la red. Puede decirse que prácticamente, los dispositivos de red se comunican entre ellos.

Unix domain sockets (network options). Permite que el sistema maneje paquetes estándar de Unix, tiene que estar habilitado porque la mayoría de las aplicaciones que se usan comúnmente trabajan con paquetes de este tipo.

TCP/IP networking (network options). Se refiere al conjunto de protocolos que permiten que el sistema pueda comunicarse con los demás en la red, por lo tanto, es indispensable para que el sistema se encuentre comunicado con otros por medio de la red.
IP: Tunneling (network options). Es la opción que permite la configuración de túneles. Esta opción tiene que seleccionarse afirmativamente ya que es necesario poder encapsular los paquetes de IPv6 con IPv4 para que se transmitan en la red existente y que los dispositivos puedan comunicarse usando IPv6.

The IPv6 protocol (network options). Es la opción que permite que el kernel del sistema pueda trabajar con IPv6 para las comunicaciones con otros sistemas en la red. Actualmente se encuentra en etapa experimental (aunque se encuentra funcionando para aplicaciones comunes).

IPv6: enable EUI-64 token format (network options). Esta opción tiene que ser seleccionada para que el sistema pueda adoptar el formato de direcciones a la asignación de las mismas que maneja el 6bone.

IPv6: disable provider based address (network options). Es necesario ya que el formato de direcciones que se tienen por el proveedor no es compatible con el formato EUI-64, por lo que el sistema no podrá trabajar de manera correcta si coexisten los dos formatos de direcciones.

Prompt for development and/or incomplete code/drivers (Code maturity level options). Con esta opción seleccionada en la configuración del kernel, se tiene la posibilidad de utilizar elementos del sistema como drivers, sistemas de archivos, protocolos de red, etc. que se encuentran en fase de prueba y que se conocen como versiones alfa.

Enable loadable module support (Loadable module support). Esta opción es la que permite cargar módulos por el kernel. Los módulos son partes de código que se encuentran compiladas y que pueden ser cargadas o descargadas en cualquier momento por el kernel, de esta manera puede hacerse solamente cuando son necesarios para alguna función en el sistema. Esta opción tiene que seleccionarse como "no" ya que pueden encontrarse algunos problemas en el sistema.

Video mode selection support (Console drivers). Esta opción permite tener un mayor control sobre el modo en el que se presentan los datos cuando el sistema se encuentra en modo texto, con esto puede aprovecharse la resolución del monitor para presentar de una forma u otra el texto en el mismo.

/proc filesystem support (File systems). Es un sistema virtual de archivos en el que se encuentra información del sistema que se modifica frecuentemente y que se requiere para algunas aplicaciones. Esta opción tiene que ser seleccionada.

Magic sysrq key (Kernel hacking). Con esta opción se tiene cierto control sobre el sistema cuando se bloquea, permite el uso de teclas de control para vaciar el buffer de cache o reiniciar el sistema.

Una vez que se ha configurado el kernel, debe compilar de la siguiente manera:

```
$make dep
$make clean
$make zImage [4]
```

Si alguna de las opciones de configuración del kernel se ha seleccionado como módulo, es necesario teclear:

```
$make modules
$make modules_install
```

Aquí es en donde hay que instalar el kernel que hemos compilado, esto se hace de la siguiente manera:

```
$make install
```

[4] si el kernel es muy grande, usar bzImage

Ahora hay que copiar el kernel y ponerle un nombre para identificarlo y que el sistema pueda utilizarlo para arrancar:

```
$cp /usr/src/linux/arch/i386/boot/zImage ‘  /boot/vmlinuz-xx.yy.zz
```

Para que el sistema pueda arrancar con el kernel que acabamos de instalar, puede usarse una herramienta llamada liloconfig o de otra manera, es necesario agregar algunas líneas en el archivo lilo.conf. La estructura general que tiene este archivo (cuando hay un solo kernel instalado) es la siguiente:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
vga=extended
timeout=50
image=/boot/vmlinuz-aa.bb.cc
      label=Linux
      root=/dev/hda8
      read-only
```

Este archivo tendrá la capacidad de mostrar al usuario la opción para arrancar con el nuevo kernel para que el sistema pueda manejar IPv6. La forma en la que debe quedar es:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
vga=extended
timeout=50
image=/boot/vmlinuz-xx.yy.zz
      label=Linux-IPv6
      root=/dev/hda8
      read-only
      image=/boot/vmlinuz-aa.bb.cc
      label=Linux
      root=/dev/hda8
      read-only
```

Ya que actualizamos el archivo de configuración de lilo, tenemos que hacer que reconozca los cambios que acabamos de hacer, esto se hace tecleando:

```
$/sbin/lilo
```

Finalmente, para que el sistema pueda utilizar el kernel que acabamos de instalar, se tiene que reiniciar usando:

```
$reboot
```

Así, cuando el sistema arranque, y aparezca el prompt de lilo, tenemos la oportunidad de seleccionar (en el tiempo establecido) la versión del kernel con la que queremos trabajar. La opción que se tendrá ahora por default, es la que incluye el kernel que soporta IPv6.

Hay que tener en mente que la primera vez que se arranque el sistema y que se use el kernel que se acaba de compilar e instalar, tenemos que establecer las dependencias entre los módulos si es que se seleccionaron opciones de configuración como módulo en el kernel. Esto se hace de la siguiente manera:

```
$depmod -a
```

En este momento el sistema tendrá las capacidades que hemos seleccionado mediante la configuración del kernel; es decir, ahora tiene todo lo necesario para que los programas de aplicación utilicen los recursos del sistema para comunicarse usando IPv6.

Básicamente, las aplicaciones se instalan (o actualizan) de la misma manera. Se consiguen los archivos necesarios, se desempaquetan, se descomprimen, se hacen las modificaciones necesarias (con patches), se instalan y se configuran.

Para configurar el sistema de manera que pueda aprovechar las ventajas de IPv6 que se ha tomado el kernel, necesitamos de ciertas herramientas de red, éstas se encuentran agrupadas en las net-tools que se instalan de la siguiente manera:

```
$cd /usr/src
$tar vxzf ruta/net-tools-1.54.tar.gz -C .
$mv net-tools net-tools-1.54
$ln -sf net-tools-1.54 net-tools
$cd net-tools
$make clean
$make config
```

En este punto, debemos configurar las herramientas que se incluyen en este conjunto. Las opciones de configuración que el sistema requiere para trabajar con IPv6 son:

GNU gettext		Yes
Protocol Families	UNIX protocol family	Yes
	INET (TCP/IP) protocol family	Yes
	INET6 (IPv6) protocol family	Yes
Device hardware types	SIT (IPv6-inIPv4)	Yes

Ya que hemos seleccionado estas opciones, ahora procedemos a compilar e instalar estas herramientas de la siguiente manera:

```
$make
$make install
```

Ahora, utilizando estas herramientas podemos configurar las interfaces del sistema, agregar direcciones a la tabla de ruteo y monitorear la transmisión y recepción de paquetes IPv6.

Se requiere también de algunas aplicaciones de propósito general que proporcionan cierto tipo de servicios a otras aplicaciones de propósito específico. En esta sección se explica la forma en que se instalan.

Primero la librería para captura de paquetes (libpcap):

```
$cd /usr/src
$tar vxzf ruta/libpcap-0.4a6+ipv6-1.tar.gz -C .
$ln -sf libpcap-0.4a6+ipv6-1 libpcap
$cd libpcap
$./configure -prefix=/usr/inet6
$make clean
$make
```

Si se instala por primera vez esta librería con las características de IPv6, hay que crear un nuevo directorio (si no existe) para el almacenamiento de ciertos archivos de configuración.

```
$mkdir /usr/inet6/include/net
```

ahora se instala

```
$make install
```

y, si se instala por primera vez con las opciones de IPv6, se hace un respaldo de la librería actual, de la sección del manual que le corresponde y se actualizan con las nuevas versiones.

```
$mv /usr/lib/libpcap.a /usr/lib/libpcap.a.old
$ln -sf /usr/inet6/lib/libpcap.a /usr/lib/libpcap.a
$mv /usr/man/man3/pcap.3 /usr/man/man3/pcap.3.old
$ln -sf /usr/inet6/man/man3/pcap.3 /usr/man/man3/pcap.3
```

El TCPdump resulta muy útil cuando se trata del manejo de paquetes IPv6, su actualización requiere de la aplicación de un patch antes de la instalación. El procedimiento general de actualización es:

```
$cd /usr/src
$tar vxzf ruta/ tcpdump-3.4a6+ipv6-1.tar.gz -C .
$ln -sf tcpdump-3.4a6+ipv6-1 tcpdump
$cd tcpdump
```

Suponiendo que el patch se encuentra en el mismo directorio en donde se desempaquetaron los archivos de configuración del tcpdump (/usr/src/tcpdump), tenemos que hacer lo siguiente:

```
$cat tcpdump-3.4a6+ipv6-1-glibc21-rh60-patch.txt | patch -p1
```

Ahora se configura, compila e instala

```
$./configure --prefix=/usr/inet6
$make clean
$make
$mkdir /usr/inet6/sbin
$make install
$cp tcpdump.1 /usr/inet6/man/man1/
```

Cuando se realiza por primera vez esta actualización, hay que hacer un respaldo de los archivos anteriores

```
$mv /usr/sbin/tcpdump /usr/sbin/tcpdump.old
$ln -sf /usr/inet6/sbin/tcpdump /usr/sbin/tcpdump
$mv /usr/man/man1/tcpdump.1 /usr/man/man1/tcpdump.1.old
$ln -sf /usr/inet6/man/man1/tcpdump.1 /usr/man/man1/tcpdump.1
```

TraceRoute es otro paquete que requiere de actualización para trabajar con paquetes de IPv6, de igual forma que con el TCPdump, hay que aplicar un patch como se explica en esta sección:

```
$cd /usr/src
$tar vxzf ruta/ traceroute-1.4a5+ipv6-1.tar.gz -C .
$ln -sf traceroute-1.4a5+ipv6-1 traceroute
$cd traceroute
```

Aquí es en donde se aplica el patch, se configura, se compila e instala (debe encontrarse en el directorio /usr/src/traceroute).

```
$cat traceroute-1.4a5+ipv6-1-glibc21-rh60-patch.txt | patch -p1
$./configure --prefix=/usr/inet6
$make clean
$make
$make install
```

Si se realiza esta actualización por primera vez, puede guardarse un respaldo de los archivos de configuración que se tienen en el sistema

```
$mv /usr/sbin/traceroute /usr/sbin/traceroute.old
$ln -sf /usr/inet6/sbin/traceroute /usr/sbin/traceroute
$mv /usr/man/man#/traceroute /usr/man/man#/traceroute.old
$ln -sf /usr/inet6/man/man#8traceroute /usr/man/man8/traceroute
```

Las inet6-apps son las aplicaciones básicas de red para el sistema, son las funciones que permiten comunicarse con otros sistemas y realizar transferencias de archivos, como vimos anteriormente. Cuando este conjunto de aplicaciones surgió cubría ciertas características que se hacen insuficientes ahora, por lo que para poder usarlo en la configuración del sistema se requiere de un patch.

El procedimiento de configuración e instalación de las inet6-apps es el siguiente:

```
$cd /usr/src
$tar vxzf ruta/inet6-apps-0.36 -C .
$ln -sf /usr/src/inet6-apps-0.36 inet6-apps
$cd inet6-apps
```

Aquí, tiene que editarse el archivo GNUmakefile.conf si se quiere que en el sistema existan dos demonios independientes inetd, uno para cada versión del protocolo (IPv4 e IPv6). Cerca de la línea 87 de este archivo puede elegirse un desplazamiento en los puertos que atienden a los servicios específicos para permitir esta coexistencia. Ahí debe ponerse como comentario o no la línea que se refiere a este desplazamiento.

Ahora se aplica el patch para que modifique lo necesario en los archivos de configuración de este conjunto de aplicaciones, suponiendo que el patch se encuentra en /usr/src/inet6-apps se tiene que hacer lo siguiente:

```
$cat inet6-apps-0.36-glibc21-rh60-patch.txt | patch -p1
$make clean
$make
$make install
```

```
$chmod u+s /usr/inet6/bin/ping
$cp -r include/bsd /usr/inet6/include
```

Cuando se instala este conjunto de programas por primera vez, tiene que hacerse lo que se indica en los siguientes párrafos para preparar el sistema para actualizaciones posteriores; si no es la primera vez que se instalan, entonces puede saltarse esta sección y comenzar con la instalación de las siguientes aplicaciones.

Los datos de los archivos sobre los que se trabajará son los siguientes:

Aplicación	Ruta
finger	/usr/bin/finger
fingerd	/usr/sbin/in.fingerd
ftp	/usr/bin/ftp
ftpd	/usr/sbin/in.ftpd
inetd	/usr/sbin/inetd
ping	/bin/ping
tftp	/usr/bin/tftp
libinet6.a	/usr/lib/libinet6.a

Lo primero que hay que hacer ahora es un respaldo de las aplicaciones por si la actualización falla:

```
$mv /usr/bin/finger /usr/bin/finger.old
$mv /usr/sbin/in.fingerd /usr/sbin/in.fingerd.old
$mv /usr/sbin/inetd /usr/sbin/inetd.old
$mv /bin/ping /bin/ping.old
$mv /usr/bin/tftp /usr/bin/tftp.old
$mv /usr/lib/libinet6.a /usr/lib/libinet6.a.old
```

Ahora hay que crear una liga hacia las muevas aplicaciones instaladas y para la nueva librería:

```
$ln -sf /usr/inet6/bin/finger /usr/bin/finger
$ln -sf /usr/inet6/bin/in.fingerd /usr/sbin/in.fingerd
$ln -sf /usr/inet6/bin/inetd /usr/sbin/inetd
$ln -sf /usr/inet6/bin/ping /bin/ping
$ln -sf /usr/inet6/bin/tftp /usr/bin/tftp
$ln -sf /usr/inet6/lib/libinet6.a /use/lib/libinet6.a
```

También, hay que hacer un respaldo de las páginas del manual (man) para cada aplicación actualizada y después crear una liga a las mismas para que puedan seguir usándose de la forma usual:

```
$mv /usr/man/man1/finger.1 /usr/man/man1/finger.1.old
$mv /usr/man/man8/fingerd.8 /usr/man/man8/fingerd.8.old
$mv /usr/man/man1/ftp.1 /usr/man/man1/ftp.1.old
$mv /usr/man/man8/ftpd.8 /usr/man/man8/ftpd.8.old
$mv /usr/man/man8/inetd.8 /usr/man/man8/inetd..8old
$mv /usr/man/man8/ping.8 /usr/man/man8/ping.8.old
$mv /usr/man/man1/tftp.1 /usr/man/man1/tftp.1.old
```

Telnet es una aplicación que se usa regularmente, y la configuración de la misma se realiza de una manera muy similar a las aplicaciones anteriores, en las siguientes líneas se muestra el procedimiento:

```
$cd /usr/src
$tar vxzf ruta/ telnet.95.10.23.NE+ipv6-3.tar.gz -C .
$ln -sf telnet.95.10.23.NE+ipv6-3 telnet
$cd telnet
$make clean
$make
$make install
```

Si es la primera vez que se hace la instalación de esta aplicación tomando en cuenta la configuración de IPv6, se tienen que hacer las siguientes cosas para mantener un respaldo de la aplicación que se encuentra instalada y además finalizar la instalación.

```
$mv /usr/bin/telnet /usr/bin/telnetd.old
$mv /usr/sbin/in.telnetd /usr/sbin/in.telnetd.old
$ln -sf /usr/inet6/bin/telnet /usr/bin/telnet
$ln -sf /usr/inet6/bin/telnetd /usr/sbin/in.telnetd
```

Hay que hacer una copia de los manuales existentes en el sistema y agregar los nuevos:

```
$mv /usr/man/man1/telnet.1 /usr/man/man1/telnet.1.old
$mv /usr/man/man8/telnetd.8 /usr/man/man8/telnetd.8.old
$ln -sf /usr/inet6/man/man1/telnet.1 /usr/man/man1/telnet.1
$ln -sf /usr/inet6/man/man8/telnetd.8 /usr/man/man8/telnetd.8
```

En este punto, podemos decir que el sistema se encuentra estable y es funcional para el trabajo con paquetes IPv6 en las transmisiones a través de la red. Cuenta ya con las aplicaciones básicas que permiten esta comunicación en redes locales.

Debemos decir que la versión más reciente de esta distribución de Linux (Red Hat 6.2) cuenta con características específicamente desarrolladas para satisfacer los requerimientos de IPv6. Aunque el kernel cuenta con las mismas opciones de configuración (relacionadas con IPv6), el sistema puede configurarse mucho más fácil que la versión anterior (Red Hat

6.0). De hecho, existen algunas aplicaciones para IPv6 que se encuentran configuradas por default y que pueden elegirse desde la instalación del mismo.

Debemos aclarar también que esto es relativo al sistema solamente, por lo que la instalación y configuración de las aplicaciones se trata de manera independiente por lo que tienen que ser realizadas por el usuario o el administrador del sistema dependiendo de las funciones que realicen y los permisos que se requieran.

Recientemente se liberó una nueva versión de la distribución Red Hat de linux, se trata de la versión 7.0 que cuenta con el kernel en su versión 2.4.0. De acuerdo a la convención que se tiene en cuanto a la numeración para las versiones de software, esperamos que el nuevo kernel y la nueva distribución tengan características muy superiores a las actuales.

En la siguiente sección se tratan varias aplicaciones que pueden considerarse como adicionales para el sistema. Éstas pueden ser instaladas o no tomando en consideración las necesidades particulares del sistema específico con el que tratemos.

Aplicaciones específicas

Las aplicaciones específicas son las que se han desarrollado pensando en funciones particulares que son requeridas por los usuarios, algunas pueden ser de carácter comercial aunque la mayoría siguen la filosofía de Linux de mantenerse siendo software libre.

En esta sección veremos algunas de estas aplicaciones, las más recientes y más útiles que se han desarrollado pensando en el uso de IPv6. Daremos una breve descripción para conocer sus características y la referencia para consulta de actualizaciones y en un capítulo posterior, trataremos algunas de ellas con más detalle.

Existen diferentes versiones o presentaciones por varios desarrolladores de algunas aplicaciones, pero cada uno de ellos incluye en la propia características que la hacen diferente a las demás y que frecuentemente las hacen verse superiores o inferiores a las demás. Como es normal en estos casos, para tomar una decisión acerca del programa que se elige, es recomendable realizar una evaluación y comparación de las características con las de los competidores para poder realizar una elección que nos permita satisfacer nuestras necesidades empleando los recursos con los que contamos.

Estas aplicaciones pueden clasificarse de acuerdo a la función que desempeñan o a la manera en la que realizan su trabajo en diferentes grupos. Podemos decir que los grupos principales en los que podemos agrupar las aplicaciones son:

- ftp
- http
- irc
- mail
- ruteo

- monitoreo
- desarrollo
- noticias
- seguridad
- sistema
- otros

Los programas de aplicación se presentan de varias formas, pueden ser programas cliente, servidores o el par de ellos. No todos los desarrolladores presentan las dos versiones de las aplicaciones ya que no siempre son necesarias.

Las aplicaciones de ftp se usan para realizar transferencias de archivos; las de http para presentar información o accesar a ella en web; las de irc son las que permiten establecer una comunicación directa con otras personas en tiempo real a través de la red; las de correo electrónico nos permiten recuperar mail's, así como proporcionar servicios relacionados con el mismo; las de ruteo nos permiten enviar paquetes a través de la red en una forma específica para darnos un mejor control de los mismos; con las aplicaciones de monitoreo podemos observar el desempeño de los recursos con los que contamos así como de la información que enviamos; las aplicaciones de desarrollo tienen un papel importante ya que son las que nos permiten desarrollar aplicaciones que cuenten con características especiales; las aplicaciones de noticias son las que nos permiten realizar las conexiones a servidores de noticias para obtener información o nos permiten publicarla; las aplicaciones de seguridad también tienen importancia relevante ya que nos permiten mantener la integridad y la confiabilidad de nuestra información, así como de los servicios que proporcionamos o solicitamos; las aplicaciones de sistema son las que nos permiten configurarlo de tal manera que cuente con algunas características especiales y que pueda realizar tareas determinadas y por último, las aplicaciones que son de carácter más general y que se desarrollan para satisfacer necesidades específicas.

Ahora daremos algunos ejemplos de estas aplicaciones, así como una breve descripción de las mismas y una referencia para consulta de información acerca de las mismas.

apache 1.3.12
Es un servidor de HTTP muy popular, que trabaja eficientemente y que tiene características que lo hacen superior a otros. más información en http://www.apache.org.

bird 1.0.4
Es un demonio de ruteo dinámico para sistemas Unix. Soporta todos los protocolos de ruteo usados en Internet como BGP, OSPF, RIP y sus variantes respectivas para IPv6. Cuenta también con características que le permiten flexibilidad en mecanismos de configuración y un filtro de lenguajes de ruteo. Puede encontrarse más información en http://bird.network.cz.

bitchx 1.0c17
Es la versión actualizada del IRC (Internet Relay Chat) que ha sido modificada para adaptarse a las nuevas necesidades y características de la red. Se encuentra disponible en varias plataformas diferentes. más información en http://www.bitchx.com.

bsd-ftpd 0.3.2
Es un servidor de FTP GNU/Linux de OpenBSD. Es un servidor seguro que puede ser usado en lugar del wu-ftpd. Hay más información en:
http://www.elves.ens.fr:8080/home/madore/programs/#prog_ftpd-BSD.

cold
Es un sniffer y analizador de protocolos que ha sido desarrollado para propósitos educativos y comerciales, así como para localizar problemas. Trabaja en diferentes interfaces del sistema y soporta varios protocolos. Hay más información en http://www.panservice.it/cold.

cvs 1.10.8
Es un sistema de control de versiones que permite mantener el control de los cambios que se realizan en el sistema guardando los datos de cuando, quien y que cambios ocurren. A diferencia de otros programas similares, cvs puede trabajar en más de un archivo o directorio a la vez. Es una herramienta muy útil para archivos que son accesados simultáneamente por varios usuarios. Se encuentra disponible para Windows, Linux y Unix. Hay más información en http://www.cyclic.com.

exim
Es un agente de transferencia de mensajes (MTA) para usarse en Unix. Tiene características especiales para evasión de mensajes no autorizados de equipos, redes o usuarios. Puede encontrarse más información en: http://www.exim.org.

fetchmail 5.5.2
Es un programa que permite realizar conexiones a servidores de correo para "bajarlos", trabaja sobre TCP/IP y soporta cualquier protocolo remoto de correo (POP2, POP3, RPOP, APOP, KPOP, IMAP, ESMTP ETRN. Cuenta con más seguridad que otros clientes de correo y permite la encriptación mediante túneles. Puede encontrarse más información en http://www.tuxedo.org/~esr/fetchmail/.

gated 3.6
Es un ruteador que soporta la mayoría de los protocolos Unicast incluyendo RIP, OSF Router Discovery, RIPng y BGP-4+. Puede encontrar más información en: http://www.gated.org.

inn 2.3.0
El InternetNetNews es un sistema completo de noticias Usenet. Maneja la lectura de noticias en un servidor independiente. Con él se puede conectar a servidores de noticias de tal manera que puede obtenerse información específica de los mismos de una forma segura utilizando IPv6. Hay más información en: http://www.isc.org/inn.html.

ircii 4.4X
Es un programa cliente para IRC. más información en http://www.eterna.com.au/ircii.

inframail
Es una suite de protocolos de servidor como mail, web, ftp y listas de correo. Ofrece servicios de servidor POP, capacidades de IMAP, acceso por WebMail, Mail gratuito, listas de correo, listas de discución, servidor de noticias y cuenta con capacidades de administración por niveles. Existe una versión gratuita para evaluación, puede encontrarse más información en: http://www.infradig.com/infradig/inframail/index.shtml.

jipsy 0.2.1
Permite reemplazar algunas partes de las clases de java.net que permiten que las aplicaciones desarrolladas en este lenguaje puedan comunicarse usando IPv4 o IPv6 indistintamente. Esas aplicaciones no requieren de una re-compilación por lo que se encuentran listas para usarse en IPv6. Puede encontrarse más información de jipsy en: http://www.progsoc.uts.edu.au/~mpf/jipsy.

leafnode 1.9.17
Es un programa cliente lector de noticias que se ha desarrollado para sitios con limitaciones de ancho de banda y con pocos usuarios. Verifica cuales son los grupos de los que se han leído noticias y baja de solamente de éstos la información actual. No requiere de mantenimiento y su instalación es sencilla, hay más información en http://www.leafnode.org.

lftp 2.2.6
Es un cliente de FTP que tiene funciones "extendidas" por lo que es un programa de aplicación que resulta novedoso y funcional. Tiene la capacidad de realizar diferentes transferencias simultáneamente, ejecutar comandos en background, restablecer la conexión cuando se ha perdido y continuar la transferencia en el punto en el que se interrumpió; además, cuando el sistema se cierra mientras se realiza una transferencia, esta se manda a background y sigue realizándose hasta concluir. Esta aplicación trabaja en modo de línea de comandos y puede bajarse junto con la documentación de:
ftp://ftp.yars.free.net/pub/software/unix/net/ftp/client/lftp.

lukemftp 1.4
Es un programa para transferencia de archivos que tiene nuevas características como transferencias desde direcciones HTTP y FTP, completa direcciones mientras se escribe, muestra una barra de progreso de la transferencia, y otras mas. más información en ftp://ftp.netbsd.org/pub/NetBSD/misc/lukemftp.

lynx 2.8.3
Es un cliente de WWW con características especiales. Muestra la información en formato HTML como texto en terminales que no tienen la capacidad de ser configuradas gráficamente. Se encuentra disponible para su uso en diferentes plataformas. más información en http://lynx.browser.org.

mMosaic 3.6.5
Es un browser gratuito que soporta IPv6 para las transferencias HTTP y FTP; puede usarse simultáneamente en varias ventanas y se encuentra más actualizado respecto a sus versiones anteriores por lo que soporta más elementos de HTML que las versiones anteriores. más información en http://sig.enst.fr/~dauphin/mMosaic.

mozilla M18
Es un navegador de código libre Es una herramienta similar al Netscape Communicator que cuenta con el soporte de Netscape. Se encuentra disponible para distintas plataformas. Hay más información en: http://www.mozilla.org.

mrt 2.2.2a
Es un conjunto de herramientas de ruteo que incluye demonios de ruteo, librerías de programación y estadísticas de desempeño de ruteo en Internet. más información en: http://www.mrtd.net.

ncftp 3.0.1
Es un programa de aplicación que implementa los servicios para transferencia de archivos, se trata de una versión de FTP que cuenta con características mejoradas como trabajo en background y recursividad en la transferencia de directorios. Actualmente se tienen disponibles las versiones correspondientes para el cliente y el servidor. Hay más información en http://www.ncftp.com.

openssh 2.2.0p4
Es un paquete de herramientas libre que permite encriptar toda la información que se transmite a través de la red (datos y passwords) haciéndola más segura. Incluye capacidades para transmisiones en túneles por lo que puede ser aplicado durante la transición de IPv4 hacia IPv6. Puede encontrarse más información en http://www.openssh.com.

pftp 1.1.6
Es un programa que permite realizar transferencias de archivos, es un FTP con nuevas capacidades. Permite la transferencia de archivos y directorios recursivamente, enviar y recibir entrada y salida estándar del sistema, establecer el tamaño del buffer de la red, establecer el ancho de banda con el cual se realice la transferencia, mandar paquetes unicast, anycast y multicast en transmisiones de audio y Video. más información en: http://www.pftp.de.

postfix 1991261 pl02
Es un programa que intenta remplazar al Sendmail, trata de ser rápido, fácil de administrar, seguro y además, compatible con Sendmail para contar con una migración sencilla. más detalles en http://www.postfix.org.

postie
Es un servicio que permite enviar correos electrónicos en línea de comando. Soporta protocolos SMTP, ESMTP, MIME, BASE64, POP3, IMAP4, HTTP y puede usarse para Windows y Unix. Hay más información en:
http://www.infradig.com/infradig/postie/index.shtml

proftpd 1.2.0rc2
Es un servidor de FTP e alto desempeño enfocado hacia la simplicidad, seguridad y facilidad de configuración. su configuración es muy similar a la del servidor Apache y tiene muchas ventajas sobre otros servidores de FTP por como: servidores virtuales de FTP y visibilidad de directorios basada en la asignación de permisos. La información oficial se encuentra en http://www.proftpd.net.

rat 4.2.9
Robust Audio Tool (RAT), es una herramienta de aplicación de código abierto que permite establecer conferencias y realizar transmisiones de audio en internet. Con ella se puede establecer comunicación de audio entre dos o más participantes aprovechando las ventajas que se ofrecen con multicast. No demanda grandes requerimientos, es suficiente con una tarjeta de sonido y conexión a internet. Se ejecuta en ambiente gráfico en varias plataformas. Puede encontrarse más información en:
http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/index.html.

quake
Es un juego multiusuario que ha sido portado a IPv6 por Kame. Puede encontrar más información en: http://www.viagenie.qc.ca/en/quake.shtml.

scrollz 1.8j
Es un cliente IRC que permite la comunicación con otros usuarios de la red en tiempo real, esta basado en el código ircII pero no toma todas sus características, por lo que tiene un mejor desempeño. Esta desarrollado en C, lo que le permite manejar más eficientemente los recursos del sistema y dar un mejor servicio, puede encontrarse más información en http://www.scrollz.com.

solidpop3D 0.14
Es el servidor de correo POP en una versión actualizada que permite la configuración del mismo de manera sencilla y que cuenta con nuevas características. más información en http://solidpop3d.pld.org.pl.

taptunnel 0.31
Es un programa que puede usarse como cliente y como servidor para crear túneles Ethernet sobre redes TCP/IP. Puede usarse para conectar redes privadas usando redes públicas empleando una gran variedad de protocolos como IPX, ASP, IPv4, IPv6, DHCP y otros. Esta desarrollado para sistemas Linux con kernel 2.2 ó superior. más información en http://voodooz.extinct.org/~poettering/projects/taptunnel.

tcpwrappers 7.6+ipv6
Es un paquete que permite monitorear y filtrar peticiones externas de algún servicio en el servidor como systat, finger, ftp, telnet, rlogin, rsh, exe, tftp, talk y demás. Cuenta con un pequeño demonio que puede ser instalado sin cambios en el software o archivos de configuración. Los wrappers reportan el nombre del cliente y del servicio que solicitó. Se encuentra más información en ftp://ftp.porcupine.org/pub/security/index.html.

thttpd 2.19
Es un servidor de HTTP simple, pequeño, rápido portable y seguro. Maneja básicamente lo necesario para soportar HTTPv1, requiere de pocos recursos del sistema para ejecutarse, se compila en varios sistemas lo que hace que aproveche las características de cada uno y la implementación de IPv6 esta integrada por lo que no requiere de actualización. más información en http://www.acme.com/software/thttpd.

tin 1.4.4
Es un poderoso lector de noticias que se encuentra disponible para varias plataformas. Tiene opciones que permiten personalizar el formato con el que se presenta. Hay más información en: http://www.tin.org.

toolnet6
Software para intercambio de protocolos para windows 95/98/NT con el que pueden usarse aplicaciones para red en IPv4 usando IPv4/6. Hay más información en:
http://www.hitachi.co.jp/Prod/comp/network/pexv6-e.htm.

totd
Es un proxy DNS que traduce entre registros IPv4 e IPv6. Se usa para comunicar redes y hosts que usan solamente IPv6 con dispositivos IPv4 usando mecanismos de traducción de protocolos. Hay más información en:
http://vermicelli.pasta.cs.uit.no/ipv6/software.html.

vpnstarter 0.2.0
Es una aplicación para la implementación de VPN (Virtual Private Networks) empleando herramientas públicas. más información en http://detached.net/vpnstarter.

v6tun
Es un programa que permite aislar nodos IPv6 para permitir el uso de dispositivos unix como túneles para paquetes IPv6 sobre IPv4. El proceso sobre el túnel puede ser mediante ssh. El software pertenece al proyecto WIDE (http://www.wide.ad.jp) y se encuentra disponible en ftp://ftp.kyoto.wide.ad.jp/IPv6/v6tun.

wu-ftpd 2.6.0
Es otra versión del programa para transferencia de archivo, adaptado con nuevas características. Se trata particularmente del programa servidor. más información en http://www.wu-ftpd.org.

wwwoffle 2.5d
Es un conjunto de programas que facilitan el trabajo que realizan los browsers cuando el sistema usa una conexión dial-up a internet. Es un servidor proxy simple con características especiales que le permiten leer páginas aún estando desconectado. más información en http://www.gedanken.demon.co.uk/wwwoffle.

w3m
Es un cliente de web para IPv6, es muy similar al Lynx, pero cuenta con características que lo hacen superior. Hay más información en:

http://ei5nazha.yz.yamagata-u.ac.jp/~aito/w3m/eng/index.html.

xinetd 2.1.8.8p2
Es el super servidor que reemplaza al inetd instalado en los sistemas. Tiene nuevas características adaptadas para los nuevos protocolos. más información en:
http://www.xinetd.org.

zebra 0.88
Es una herramienta de software libre que maneja los protocolos de ruteo basados en IPv4 e IPv6, se distribuye bajo licencia publica GNU. Soporta varios protocolos, entre ellos BGP-4, RIPv1, RIPv2 y OSPFv2 por lo que hace al sistema funcional en cuando al manejo de protocolos recientes. Esta pensado para emplearse como servidor de ruteo debido a las características con las que se ha desarrollado y que lo hacen ser más que una herramienta de ruteo. Hay más información en: http://www.zebra.org.

zmailer
Es un paquete que sirve para la transferencia de mensajes. Ha sido desarrollado para gateways, servidores de mail y otros ambientes y trata de resolver los problemas del Sendmail. Es un intento para desarrollar un servidor de correo multiprotocolo. Hay más información en: ftp://ftp.funet.fi/pub/unix/mail/zmailer.

6tunnel 0.04
Es un programa que permite establecer la comunicación entre sistemas que se encuentran en diferentes redes IPv6 que están separadas y que tienen que establecer un contacto mediante un túnel. más información en ftp://amba.bydg.pdi.net/pub/wojtekka.

CASOS ESPECÍFICOS

Casos Específicos

En este capítulo veremos algunos casos específicos de aplicaciones que se encuentran desarrolladas actualmente para IPv6. Veremos la forma en la que el nuevo protocolo se distingue de su versión anterior aprovechando sus nuevas características para da un mejor rendimiento y funcionalidad a las aplicaciones que se han creado específicamente para IPv6.

Estos casos específicos estarán enfocados a las aplicaciones para Linux.

Veremos primero una aplicación que se ha desarrollado para hacer más eficiente el trabajo del sistema en cuanto a la atención de servicios por el mismo. Se trata del xinetd (eXtended interNET superDaemon) que es el reemplazo del inetd del que habíamos hablado anteriormente.

xinetd

xinetd hace todo lo que hace el inetd pero de una manera más segura. Igual que su predecesor, lee un archivo de configuración en donde se encuentra la lista de los servicios de IP que proporciona y a los que se mantiene escuchando el puerto correspondiente.

Hay que decir que no son compatibles, por lo que el inetd no puede realizar las funciones del xinetd, esto es precisamente porque el xinetd es la "nueva versión" del inetd y cuenta con características nuevas como soporte para IPv6 y un archivo de configuración con formato diferente (aunque muy parecido) al inetd.

Las características con las que cuenta son:

- Control de acceso
  Tiene implementadas funciones de control de acceso con las que se puede restringir el acceso a los servicios del sistema considerando el usuario que lo solicita o el host desde donde se hace la petición del mismo. Se puede configurar el xinetd con la opción para soporte de libwrap que trabaja de una manera similar al tcpd pero es más eficiente. Esto es porque los tcpwrappers atienden una sola conexión, en cambio, con la nueva versión pueden atenderse más de una conexión por lo que pueden limitarse los servicios a hosts específicos, a un número de conexiones, por servicio, etc. Puede también limitar el acceso a los servicios tomando en cuenta restricciones horarias. Permite realizar una selección de servicios que pueden ser proporcionados a ciertos hosts dependiendo de las direcciones IP que se proporcionen, con lo que pueden asignarse servicios específicos a ciertos hosts.

- Negación de servicios
  Debido a que el xinetd cuenta con capacidades de control de acceso, puede negar un servicio cuando las peticiones del mismo son repetitivas, por lo que se previenen los ataques a puertos específicos. Se puede limitar también el número de conexiones de

un host. También cuenta con la capacidad de limitar el tamaño de los archivos de registro del sistema para evitar que se hagan de un tamaño muy grande y ocupen un espacio considerable. Con todo lo anterior, se protege el sistema de ataques que puedan ocurrir aprovechando un servicio que se encuentra en el mismo.

- Registros de acceso extendidos
  Se puede establecer el nivel de registro para cada servicio independientemente , los archivos de registro que son los que tienen información relativa a los servicios y en ella se encuentra información como el nombre o la dirección del host desde el que se solicita el servicio, el usuario que lo pide, etc. Puede registrarse la hora en la que se inicia y en la que se finaliza un servicio y obtenerse el tiempo que los clientes lo usan. El registro también incluye los intentos de conexión para los servicios, de manera que se registra cada uno de ellos aunque no sea proporcionado.

- Soporte para IPv6
  Las versiones iguales o superiores a la 2.1.8.8pre11 cuentan con soporte para IPv6 lo que permite que el sistema tenga instalados servicios que usen este nuevo protocolo para realizar sus operaciones.

- Interacción con el usuario
  Pueden definirse varios mensajes para los usuarios que se desplegaran en los diferentes casos que se relacionen con los servicios y las conexiones; por ejemplo, cuando la conexión no pueda realizarse de manera correcta, cuando el servicio sea negado al usuario o al host, cuando existan cambios, etc.

El xinetd puede ejecutarse en varias plataformas como Solaris 2.6 (sparc y x86), Linux, BSDi e IRIX 5.3 y 6.2.

inetd y xinetd pueden coexistir en el mismo sistema sin problemas pero para ello, se tiene que especificar en el archivo de configuración respectivo cual será el servicio que atenderá cada uno.

Puede usarse el archivo de configuración del inetd como base o plantilla para crear el archivo de configuración del xinetd; esto se hace con ayuda del itox o con el xconv.pl que se distribuyen junto con el super servidor.

xinetd soporta los tcpwrappers si se incluye la opción --with-libwrap en la compilación el xinetd. Con esto se logra que el sistema verifique en los archivos host.allow y host.deny antes de establecer la conexión y conceder el servicio.

xinetd puede ser compilado con la opción --with-inet6 en el script de configuración para que soporte IPv6. Pueden usarse direcciones IPv4 mapeadas o direcciones IPv4 en notación puntuada normal y xinetd las mapeará a IPv6.

Cuando se compila con la opción de IPv6, se toman todos los sockets como sockets IPv6. El kernel del sistema tiene que estar compilado para tomar estos sockets y trabajar con ellos de manera correcta ya que si no es así, los servicios no se iniciarán cuando sean necesarios. Se concluye entonces, que solamente debe compilarse el xinetd si el kernel soporta IPv6.

El proceso para instalación y configuración de el xinetd es:

```
$tar vxzf ruta/xinetd-2.1.8.8p3.tar.gz -C /usr/src
$cd /usr/src/xinetd-2.1.8.8p3
$./configure --with-inet6 --with-libwrap --prefix=/usr/inet6
$make clean
$make
$make install
$xinetd/xconv.pl < /etc/inetd.conf > /etc/xinetd.conf
```

una vez que se ha obtenido el archivo de configuración del xinetd como se muestra en la última línea, puede ser necesario editarlo para agregar nuevos servicios o modificar su configuración.

El xinetd tiene varias opciones de configuración que se aplican en la compilación, que son:

--with-libwrap. Con esto pueden usarse tcpwrappers. Hace que se trabaje sobre el /etc/hosts.{allow|deny}.

--with-loadavg. Con esta opción, se soporta la configuración man_load que permite tener algunos servicios deshabilitados en niveles específicos.

--with-inet6. Es la que le da la opción para soporte de IPv6 convirtiendo todos los sockets en sockets IPv6 de tal manera que aceptan conexiones IPv4 e IPv6, pero las IPv4 son mapeadas a IPv6. Por ejemplo 127.0.0.1 ---> ::ffff:127.0.0.1

Existe otra forma de instalación que resulta mucho más sencilla, ésta es la que se realiza usando el archivo rpm correspondiente para el xinetd en la versión que se requiera. Suele ser diferente a la que se encuentra disponible en código, pero es funcional y prácticamente cuenta con las mismas características y nos evita cierto trabajo de configuración en línea de comandos.

Una vez que se tiene instalado en el sistema, puede usarse de diferentes formas dependiendo de las necesidades específicas que tengamos, para ello podemos aprovechar que pueden usarse algunos parámetros.

xinetd recibe una serie de parámetros en la línea de comandos, con los que se realizan funciones específicas y que permiten configurar al sistema de una forma específica. En esta sección se muestran los parámetros que puede aceptar, así como una breve descripción de los mismos.

-d
para entrar en el modo debug, produce una salida con información de lo que va ocurriendo durante la ejecución.

-syslog syslog_facility
esta opción se usa para mostrar mensajes en un formato específico

-filelog logfile
guarda los mensajes que se producen en un archivo especificado, si el archivo no existe, lo crea, no sirve esta opción en el modo de debug.

-f config_file
especifica el archivo de configuración que usará, por default es /etc/xinetd.conf

-pid
muestra el identificador del proceso asignado en la terminal

-loop rate
establece la frecuencia del lazo en la que el servicio se considera en error y se desactiva, se especifica en términos de número de servidores por segundo que pueden ser forzados por un proceso. La velocidad de la máquina determina el valor correcto de esta opción, por default, se tiene 10.

-reuse
si esta opción se usa, xinetd especifica la opción del socket.

-limit proc_limit
pone un límite al número de procesos ejecutándose concurrentemente que pueden ser iniciados por xinetd, es para que no se desborde la tabla de procesos.

-logprocs limit
esta opción pone un límite al número de servidores simultáneos con especificaciones de userid.

-shutdownprocs limit
esta opción pone un límite en el número de servidores concurrentes para shutdown

-cc interval
establece el tiempo con el que se revisa la consistencia del desempeño del xinetd, va en segundos

Ahora mostramos el archivo de configuración para el xinetd (/etc/xinetd.conf) para que pueda verse la estructura del mismo, se nota que se encuentra organizado por bloques en los que se trata a cada servicio independientemente de los demás. Resulta ser un poco más claro que el que corresponde al inetd.

```
# This file generated by xconv.pl, included with the xinetd
# package.  xconv.pl was written by Rob Braun (bbraun@synack.net)
#
# The file is merely a translation of your inetd.conf file into
# the equivalent in xinetd.conf syntax.  xinetd has many
# features that may not be taken advantage of with this
# translation.
# Please refer to the xinetd.conf man page for more information
# on how to properly configure xinetd.


# The defaults section sets some information for all services
defaults
{
      #The maximum number of requests a particular service may
      # handle at once.
      instances    = 25

      # The type of logging.  This logs to a file that is
      # specified.
      # Another option is: SYSLOG syslog_facility [syslog_level]
      log_type     = FILE /var/log/servicelog

      # What to log when the connection succeeds.
      # PID logs the pid of the server processing the request.
      # HOST logs the remote host's ip address.
      # USERID logs the remote user (using RFC 1413)
      # EXIT logs the exit status of the server.
      # DURATION logs the duration of the session.
      log_on_success = HOST PID

      # What to log when the connection fails.  Same options as
      # above
      log_on_failure = HOST RECORD

      # The maximum number of connections a specific IP address
      # can have to a specific service.
      per_source   = 5
}

service ftp
{
      flags        = REUSE NAMEINARGS
      socket_type  = stream
      protocol     = tcp
      wait         = no
      user         = root
      server       = /usr/sbin/tcpd
      server_args  = in.ftpd -l -a
}

service telnet
{
      flags        = REUSE NAMEINARGS
      socket_type  = stream
      protocol     = tcp
      wait         = no
```

```
        user            = root
        server          = /usr/sbin/tcpd
        server_args     = in.telnetd
}

service shell
{
        flags           = REUSE NAMEINARGS
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/tcpd
        server_args     = in.rshd
}

service login
{
        flags           = REUSE NAMEINARGS
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/tcpd
        server_args     = in.rlogind
}

service talk
{
        flags           = REUSE NAMEINARGS
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = nobody
        group           = tty
        server          = /usr/sbin/tcpd
        server_args     = in.talkd
}

service ntalk
{
        flags           = REUSE NAMEINARGS
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = nobody
        group           = tty
        server          = /usr/sbin/tcpd
        server_args     = in.ntalkd
}

service finger
{
        flags           = REUSE NAMEINARGS
        socket_type     = stream
        protocol        = tcp
        wait            = no
```

```
        user            = nobody
        server          = /usr/sbin/tcpd
        server_args     = in.fingerd
}

service linuxconf
{
        flags           = REUSE NAMEINARGS
        socket_type     = stream
        protocol        = tcp
        wait            = yes
        user            = root
        server          = /bin/linuxconf
        server_args     = linuxconf --http
}

service netbios-ssn
{
        flags           = REUSE NAMEINARGS
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/smbd
        server_args     = smbd
}

service netbios-ns
{
        flags           = REUSE NAMEINARGS
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        server          = /usr/sbin/nmbd
        server_args     = nmbd
}
```

Zebra

Ahora veremos un software de ruteo que tiene una gran potencia y soporte para diferentes protocolos, gracias a este software, puede tenerse un servidor de ruteo completo y funcional. Su nombre es Zebra y hablaremos de él en esta sección.

GNU Zebra es un software de ruteo que maneja diferentes protocolos. Comenzó en 1996 siendo un proyecto que trataba de proporcionar servicios de ruteo bajo la licencia GNU.

Zebra no es un conjunto de herramientas de ruteo solamente, es un software que permite la configuración de un equipo dedicado específicamente para ruteo. De esta manera, el

sistema actúa como un ruteador dedicado intercambiando información sobre las tablas de ruteo con otros ruteadores para mantenerse actualizado.

Aún no se encuentra disponible en una versión oficialmente estable, la última versión que se ha liberado (en octubre del año 2000) es (como todas las anteriores) una versión beta, zebra-0.89.

Se están desarrollando módulos con los cuales se puedan manejar en el futuro protocolos de ruteo multicast, así como soporte para MPS, se contempla que esto se pondrá tener en la versión 2.0 de zebra.

Esta diseñado modularmente, de tal forma que cada unos de los módulos que corresponden a los protocolos que usa pueden ser reemplazados, actualizados o se les puede dar mantenimiento individual e independientemente sin afectar a los demás. Además puede utilizarse en sistemas que usen un kernel multihilos, ya que cada protocolo es manejado por procesos independientes.

Soporta diferentes protocolos basados en TCP/IP como:

- BGP4
- BGP4+
- RIPv1
- RIPv2
- OSPFv2
- OSPFv3
- RIPng

Cuenta con tres características principales que son:

- Confiable
  Es confiable porque no interrumpe su trabajo aún cuando se realiza el mantenimiento o actualización de alguno de los módulos que lo componen. Además pueden hacerse modificaciones sin necesidad de que el servidor de ruteo se encuentre fuera de línea.
- Velocidad
  Zebra puede rutear los paquetes más rápidamente que el software común dedicado a esto, además permite que se envíen grandes cantidades de paquetes, lo que repercute en la velocidad de la transferencia de los paquetes.
- Modularidad
  Ya hemos hablado de que el diseño se encuentra en forma modular, lo que permite trabajar con un protocolo independientemente sin afectar el trabajo de los demás.

El diseño de Zebra se basa en muchos RFC's para mantenerse apegado a los estándares actuales y permitir el desarrollo y actualizaciones futuras. La mayor parte de estos RFC's se refieren a protocolos de ruteo específicamente, pero algunos se refieren a protocolos de transporte y capas de red.

Tiene capacidades para configurar las interfaces de red de manera que pueden utilizarse tablas de ruteo estáticas o dinámicas, dependiendo de la extensión de la red y de las necesidades particulares de la misma. Otra de las ventajas que tiene es que maneja dos modos de operación, uno de ellos permite realizar solamente consultas al sistema y a las tablas de ruteo, mientras que el otro modo permite la modificación de la configuración de las interfaces, con lo que puede asignarse un grado de libertad a los usuarios dependiendo de las funciones que requieran para su trabajo.

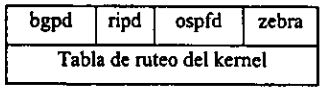
Una vez que se ha instalado zebra en el sistema, se tiene que editar el archivo en el que se encuentran definidos los servicios que presta el mismo, añadiendo el puerto asignado a cada uno de estos servicios, así como el protocolo que usaran. De esta manera, será necesario añadir lo siguiente en el archivo /etc/services:

```
zebrasrv    2600/tcp    # zebra service
zebra       2601/tcp    # zebra vty
ripd        2602/tcp    # RIPd vty
ripng       2603/tcp    # RIPngd vty
ospfd       2604/tcp    # OSPFd vty
bgpd        2605/tcp    # BGPd vty
ospf6d      2606/tcp    # OSPF6d vty
```

VTY (Virtual TeletYpe) es la interface individual y específica para los demonios correspondientes de los demonios para los protocolos de ruteo del sistema. Se trata de interfaces de línea de comandos que se usan para consultar o modificar la configuración del sistema relacionada con los protocolos de ruteo.

Para hacer uso de estas terminales, es necesario establecer una conexión al sistema usando telnet especificando el puerto en el que se tiene el servicio. Posteriormente se introduce el password correspondiente al nivel de usuario para proporcionar los servicios permitidos.

La relación entre los diferentes protocolos de ruteo puede esquematizarse de la siguiente manera para representar la arquitectura del software de ruteo.



Arquitectura del sistema

zebra es un demonio administrador que se ha definido para poder establecer las relaciones y las conversiones necesarias entre protocolos, además de ayudar a la interpretación de los mismos en la tabla de ruteo del kernel. Los demás demonios son los que se encargan de los protocolos para los que fueron creados individualmente.

Podemos ver que los diversos protocolos de ruteo deben trabajar sobre la misma tabla, que se encuentra manejada por el kernel, es por eso que se hace necesario contar con un demonio que pueda interpretar a cada uno de ellos y que todos puedan modificar la tabla de ruteo cuando sea necesario.

Es importante decir que para que zebra pueda trabajar correctamente en el sistema requiere que se encuentren instalados en el mismo los paquetes correspondientes a las aplicaciones y herramientas básicas para red que son inet6-apps y las net-tools.

Zebra se instala en la forma usual para los programas o herramientas que se instalan en el sistema. Hay que configurarlo, compilarlo e instalarlo. El proceso se realiza usando los siguientes comandos:

```
$configure
$make
$make install
```

el primero de ellos tiene varias opciones que puede tomar dependiendo de los argumentos que se le asignen en la línea de comandos cuando se ejecuta. El script de instalación contiene opciones por defecto que son las más comunes y que idealmente deben funcionar en la mayoría de los sistemas. Algunas de las opciones que pueden darse como parámetros son [1] :

--enable-guile
--disable-ipv6
--disable-zebra
--disable-ripd
--disable-ripngd
--disable-ospfd
--disable-ospf6d
--disable-bgpd
--disable-bgp-announce
--enable-netlink
--enable-snmp

Zebra se encuentra configurado por default para que cuente con soporte para IPv6, esta opción puede deshabilitarse usando los parámetros correspondientes durante la

[1]En la referencia correspondiente puede encontrar los detalles de configuración particulares.

configuración. Para que zebra pueda manejar paquetes IPv6 para rutearlos, es necesario que el sistema cuente con la configuración de las aplicaciones de red y configuración que permitan mantenerlo establemente usando este protocolo.

La instalación de los programas relativos a este software se realiza en /usr/local/sbin mientras que los archivos de configuración correspondientes a los mismos se almacenan en /usr/local/etc. Como en la mayoría de los programas (o scripts) de configuración, estas rutas pueden ser modificadas para atender a las necesidades específicas de los usuarios o administradores del sistema.

Una vez que se tiene instalado, puede invocarse de diferentes formas usando algunos parámetros con los que se define la forma en la que se mantendrá en ejecución el software de ruteo.

Evidentemente, zebra es un software de ruteo muy potente, que puede aprovecharse para configurar un ruteador dedicado en una máquina sin necesidad de grandes recursos ni de hardware ni de software. Maneja una amplia variedad de protocolos unicast y su diseño a futuro contempla en manejo de protocolos de ruteo multicast así como MPS.

Todas las ventajas de las herramientas de configuración, así como de manejo de paquetes pueden encontrarse en este software, por lo que representa una buena base sobre la cual apoyar un sistema que pueda trabajar como ruteador prestando servicios dedicados de forma estable y funcional.

6tunnel

6tunnel es una herramienta que permite crear y configurar túneles IPv6 sobre redes IPv4 de tal forma que las aplicaciones que usen el primer protocolo puedan establecer el contacto con equipos que se encuentren trabajando en otras redes que usen IPv6.

Se instala fácilmente, puede encontrarse como archivos fuente para configurarse, compilarse e instalarse o puede encontrarse también el paquete rpm. La opción que elegí, debido a la facilidad de instalación fue la segunda, de tal manera que se instala con las opciones por default.

La documentación que se encuentra acerca de este programa resulta ser poca pero directa. De alguna manera podemos decir que es suficiente para poder darnos una clara idea de los alcances que se tienen con este programa y la forma en la que tiene que usarse.

Una de las ventajas con las que cuenta es que se instala en el sistema la página del manual correspondiente por lo que se tiene acceso a la información "oficial" cuando se necesita directamente en nuestro sistema.

Como la mayoría de los programas y herramientas de configuración del sistema, 6tunnel debe recibir una serie de parámetros con los cuales se indica el modo en el que se ejecutará y las funciones que se encontrarán disponibles en el momento de su ejecución.

La mayoría de los parámetros que recibe son opcionales, pero existen dos de ellos que son necesarios (el puerto local y el host remoto). Ahora veremos cuales son los parámetros que puede recibir y hablaremos un poco de la forma en la que deben usarse.

6tunnel tiene la siguiente sintaxis:

> 6tunnel [-146dhfv] [-u username] [-i password] [-I password] [-l local_host] [-s source_host] local_port remote_host [remote_port]

Con las opciones que se muestran, es posible establecer el tipo de direcciones que se manejaran en los extremos (IPv4 o IPv6); obligar al sistema al establecimiento del túnel aunque no se encuentre disponible el extremo del mismo; establecer las direcciones de los extremos del túnel o proporcionar los puertos por los que se realizarán las conexiones.

Es otro de los programas que nos dan una apariencia muy simple de configuración, instalación y uso pero que como algunos de los mencionados, nos muestra sus inconvenientes y sus límites hasta que lo probamos e intentamos utilizarlo siguiendo las instrucciones proporcionadas.

taptunnel

taptunnel es una herramienta que sirve para crear y establecer túneles para unir redes TCP/IP. Para ello, utiliza una interfaz que se encuentra en el kernel de linux desde la versión 2.2 del mismo.

Para que pueda usarse es necesario que se configure, compile e instale el kernel tomando algunas opciones especiales que son las que se mencionan a continuación. Estas opciones tienen que seleccionarse como módulos.

Networking options	netlink device emulation
Network Device Support	ethertap network tap

Las opciones del kernel que tienen que seleccionarse, hacen que esté preparado para poder utilizar la interfaz con la que cuenta para establecer el túnel hacia el extremo que se indique.

Este programa puede trabajar como cliente o como servidor casi con la misma orden en la línea de comandos.

Se instala fácilmente, solamente hay que usar make para crear todos los archivos necesarios.

Este programa parece ser una buena opción para la configuración de los túneles a través de las diferentes redes, pero tiene el inconveniente de que en el momento de la instalación marca un error relativo a un archivo que no encuentra durante este proceso, por lo que la instalación no puede realizarse exitosamente.

En las últimas dos versiones disponibles de este programa, vemos que existe este mismo error y que al parecer no puede ser corregido fácilmente; así, podemos decir que el atractivo por la sencillez aparente que presenta se pierde al no poder instalarse.

Otra de las desventajas que se presentan cuando se considera este programa para la creación de túneles es la falta de información sobre el mismo, debemos decir que la documentación que se encuentra disponible es realmente pobre y no alcanza a cubrir las necesidades de abundar en datos relativos a la instalación o uso del mismo.

Es necesario mencionar, que hasta ahora podemos encontrar este programa en versión beta, por lo que no debe extrañarnos que existan algunos "huecos" en el mismo y que no se encuentre funcional al 100%. Todo lo anterior deberá ser corregido en la primera versión estable que se libere del mismo.

Mozilla M18

Es una aplicación similar al Netscape Communicator 5.0, pero con la diferencia de que se encuentra distribuida bajo el concepto de Software Libre. Se ha creado con la aprobación e incluso la colaboración de Netscape.

Se trata de una versión que reúne características avanzadas en las funciones relacionadas con las transferencia de datos a través de la red. Se le conoce como Mozilla Milestone 18 y se distribuye gratuitamente en su versión beta (la última disponible).

En el sitio en donde puede conseguirse este programa, así como información del mismo resalta un párrafo en el que se enfatiza en el hecho de que se trata de una versión de prueba, en la que existen errores conocidos (y algunos no conocidos); se aclara que el sistema puede sufrir daños en la configuración del mismo y por lo tanto puede dañar los archivos que se encuentran en él.

Por lo anterior, aunque resulta atractiva la posibilidad de instalarlo y tratando de mantener la configuración que se ha alcanzado hasta ahora, decidí postergar la instalación del Mozilla y solamente hacer referencia a las características del mismo en esta sección.

Como hemos dicho anteriormente, Mozilla es una aplicación muy similar al Netscape Communicator que cuenta con las características que pueden ofrecerse bajo licencia de software gratuito. Los requerimientos que precisa para su instalación son convencionales completamente. Actualmente se encuentra disponible en versiones para Windowsx (95, 98, 2000, NT), MAC OS (8.5, 8.6, 9) y Linux (Red Hat, SuSe). La arquitectura de la computadora debe apegarse a los requerimientos de los sistemas operativos.

Entre las características que sobresalen se encuentran:

- estabilidad
- soporte para java2
- inclusión de "temas"
- soporte para proxy
- descarga avanzada de archivos
- "completa" direcciones
- barra de herramientas personalizable
- soporte para IPv6

La instalación (en Linux) se realiza de una forma sencilla, solamente se descomprime y desempaqueta el archivo necesario y se ejecuta el script llamado mozila que se crea en el directorio principal durante la descompresión.

La instalación en otras plataformas requiere de algunos pasos mas que dependen de cada una de ellas.

La aplicación en general cuenta con una serie de errores que no la afectan completamente por lo que puede trabajarse con ella de una manera simple y limitada, pero el hecho de que se encuentren frecuentemente errores en la ejecución de la misma puede resultar molesto, sobre todo cuando no se tiene el ánimo o la intensión de participar en la localización y eliminación de los mismos.

Parece que la instalación de esta aplicación causará más incomodidades (o daño) al sistema de los beneficios que podrán obtenerse. Es atractiva la oferta, es cierto, pero cuando se piensa en dañar al sistema una vez que se ha configurado resulta ser una oferta con valor relativo que solamente puede tomarse de manera individual.

freenet6

freenet6 es un servicio proporcionado por frenet6.net desarrollado y apoyado por Viagénie Inc.

freenet6 ofrece lo necesario para poder establecer una conexión mediante un túnel desde cualquier máquina que se encuentre conectada a Internet hacia 6Bone.

Con esto, pueden realizarse pruebas de configuraciones particulares en redes que se encuentran aisladas y que usen IPv6.

El túnel se asigna temporalmente a las personas que lo solicitan llenando un formulario que se encuentra en su sitio web. En él, se solicitan algunos datos simples que son:

- dirección IPv4 local
- nickname
- país de origen

La única restricción que se tiene es que la dirección local que se proporcione tiene que pertenecer a una red pública.

freenet6 envía un script de configuración escrito en perl para el sistema que depende de la plataforma que se haya elegido para la configuración del mismo, así como de los datos ingresados en el formulario de solicitud del túnel. Las plataformas que son soportadas actualmente para proporcionar el script de configuración son:

- FreeBSD/INRIA
- FreeBSD/KAME
- Windows NT
- Cisco client
- Linux Debian GNU
- Solaris 8
- NetBSD/KAME
- OpenBSD/KAME

Si se cuenta con un equipo que trabaje bajo cualquiera de estas plataformas, el script de configuración que se recibe de freenet6 hace todo el trabajo, de otra manera puede ser necesario realizar algún tipo de modificación en el mismo. Frecuentemente, estas modificaciones suelen ser relativamente sencillas.

Particularmente, en el caso de Linux existe solamente disponible el script para la distribución Debian que se muestra a continuación.

```
#!/usr/bin/perl

# Perl script for autotunnel IPv6 with Debian GNU/Linux with Kernel 2.2.5

  print "Your system is using Debian GNU/Linux with IPv6 stack\n";

# Somes informations about tunnels values
  print "This script will create a tunnel between this computer\n";
  print "and the Freenet6 server (tunnels server)\n";
  print "Your IPv6 address (your tunnel end point) is 3ffe:b00:c18:1fff:0:0:0:307 \n";
  print "We establish a tunnel to the Freenet6 server at 3ffe:b00:c18:1fff:0:0:0:306 \n";
  print "Your IPv4 address is : 132.248.71.85 \n";
  print "The IPv4 address of the Freenet6 server is : 206.123.31.102 \n";

     # Setup the tunnel with values from Freenet6
       system(`ifconfig sit0 up`);
       system(`ifconfig eth0 add 3ffe:b00:c18:1fff:0:0:0:307`);
       system(`ifconfig sit0 tunnel ::206.123.31.102`);
       system(`ifconfig sit1 up`);
       system(`route -A inet6 add ::0/ gw fe80::206.123.31.102 dev sit1`);
```

```
print "End of the script for IPv6 with Debian GNU/LINUX \n";
```

Del script anterior, podemos ver algunos datos como:

- Dirección IPv6 asignada (local) 3ffe:b00:c18:1fff:0:0:0:307
- Dirección IPv6 de freenet6 3ffe:b00:c18:1fff:0:0:0:306
- Dirección IPv4 local 132.248.71.85
- Dirección IPv4 de freenet6 206.123.31.102

Tomando en cuenta lo anterior, es necesario realizar algunas modificaciones al script para que sea soportado por la distribución con la que trabajamos; entonces, para la distribución Red Hat, el script quedaría de la siguiente forma:

```
#!/usr/bin/perl5.00503
# Perl script for autotunnel IPv6 with RedHat/Linux with Kernel 2.2.5
  print "Your system is using Red Hat GNU/Linux with IPv6 stack\n";
# Somes informations about tunnels values
  print "This script will create a tunnel between this computer\n";
  print "and the Freenet6 server (tunnels server)\n";
  print "Your IPv6 address (your tunnel end point) is";
  print "3ffe:b00:c18:1fff:0:0:0:307 \n";
  print "We establish a tunnel to the Freenet6 server at";
  print "3ffe:b00:c18:1fff:0:0:0:306 \n";
  print "Your IPv4 address is : 132.248.71.85 \n";
  print "The IPv4 address of the Freenet6 server is : 206.123.31.102 \n";

     # Setup the tunnel with values from Freenet6
       system(`/sbin/ifconfig sit0 up`);
       system(`/sbin/ifconfig eth0 inet6 add 3ffe:b00:c18:1fff:0:0:0:307`);
       system(`/sbin/ifconfig sit0 tunnel ::206.123.31.102`);
       system(`/sbin/ifconfig sit1 up`);
       system(`/sbin/route -A inet6 inet6 add ::0/ gw fe80::206.123.31.102 dev
sit1`);

  print "End of the script for IPv6 with Red Hat/LINUX \n";
```

Si se tiene el sistema configurado correctamente para trabajar con IPv6 y se ejecuta este script, podrá tenerse acceso a la red IPv6 usando el túnel con freenet6.

Es recomendable que se revise este script para actualizar o reemplazar las rutas de los archivos que se usan en el mismo, así como para confirmar que se cuenta con las versiones que soporten este tipo de configuración.

Este túnel se encontrará disponible desde el momento en el que se llena el formulario y se mantendrá mientras la computadora se encuentre conectada a la red y cuente con su dirección IPv4 asignada.

Túnel hacia el Laboratorio de Interoperabilidad de la DTD en DGSCA, UNAM.

Buscando una alternativa para establecer una conexión entre equipos que trabajen con IPv6 y que se encuentren físicamente en redes distintas, usando la red convencional (que trabaja prácticamente con IPv4) contacté con los encargados de la asignación de direcciones y túneles del Laboratorio de Interoperabilidad de la Dirección de Telecomunicaciones Digitales que se encuentra en la Dirección General de Servicios de Cómputo Académico de la UNAM.

En este laboratorio se proporcionan direcciones IPv6, túneles e información para configuración de sistemas para que puedan usar este protocolo. Es un proceso sencillo que no requiere de ningún requisito extraordinario y que es totalmente gratuito. Con todo esto, se busca la difusión y la colaboración en la prospección y pruebas con el nuevo protocolo.

El contacto se realizó con el Fis. César Olvera Morales y con el Ing. Azael Fernández Alcantara quienes son los encargados del proyecto de IPv6 en la UNAM. Me proporcionaron una dirección que se usó como extremo (local) en la configuración del túnel, esta dirección es: 3FFE:8070:1:9::2.

Además de esta dirección, también fue necesaria la dirección (IPv4) de su equipo para establecer la conexión. El equipo con el que se estableció la configuración del túnel es un ruteador y sus datos básicos son:

Bay Networks BLN
bay-ipv6.redes.unam.mx
132.248.108.254
3ffe:8070::28

Con los datos anteriores, se configura el túnel usando las herramientas de configuración del sistema de las que hemos hablado con anterioridad.

Siguiendo la instrucciones que me proporcionó el Ing. Fernández, habilité la interfaz con la que se crea el túnel con

```
$/sbin/ifconfig sit0 up
```

posteriormente, asigné la dirección en el extremo del túnel usando la dirección del ruteador

```
$/sbin/ifconfig sit0 tunnel ::132.248.108.254
```

finalmente, asigné la dirección que se me fue asignada para el extremo (local) del túnel

```
$/sbin/ifconfig sit0 inet6 add 3FFE:8070:1:9::2/64
```

Con lo anterior, podemos decir que la conexión entre los dos equipos se ha establecido y que pueden realizarse transferencias de datos de uno de los extremos al otro usando IPv6 como protocolo.

La idea de contar con un túnel hacia un equipo que tenga "salida" hacia la red IPv6 es tratar de configurar el sistema como ruteador para que las máquinas que se encuentran en la misma red local puedan usarla para "salir" a la red a través del mismo.

Configurando el sistema como lo he descrito y usando el software de red adecuado, podemos tener una red en la que se tengan equipos de diferentes características, como sistemas operativos y versiones trabajando juntos y teniendo conexión hacia la red IPv6 usando un ruteador.

CONCLUSIONES

Conclusiones

Las redes de computadoras representan actualmente una herramienta indispensable en muchas áreas, tanto las que se relacionan con cuestiones profesionales, comerciales o personales; de hecho, las que tienen que ver con la comunicación personal son las que han impulsado el desarrollo de aplicaciones especiales en las que se permite establecer este tipo de comunicación fácilmente en diferentes ambientes, plataformas y arquitecturas.

Debemos decir que con el avance de la tecnología, este tipo de comunicación ha requerido ciertas características con las que no contaba con anterioridad, por lo que las redes han tenido que adaptarse a las necesidades de las aplicaciones y por lo tanto de los usuarios de la red.

Hay que pensar que la comunicación tiene que efectuarse sin importar las cuestiones de hardware o software de los equipos que intervienen en la misma. Generalmente, cuando se realiza una transferencia de información a través de la red se desconoce el equipo del que proviene o al que se envían los datos, pero lo que nos interesa, es que este proceso de lleve a cabo de manera exitosa.

Lo mismo sucede con las redes y subredes que se encuentran involucradas en este proceso. Es un hecho que no conocemos la estructura con la que cuentan ni la forma en la que se encuentran conectados la mayoría de los equipos.

Debido a todas estas diferencias, se hace necesario el establecimiento de protocolos de comunicación que permitan llevar a cabo de manera exitosa el intercambio de información entre los equipos. Uno de estos protocolos, el que representa la base de la arquitectura de la red es el protocolo IP. Como sabemos, este protocolo se encuentra actualmente en uso en la versión 4 pero debido a las características y necesidades de las aplicaciones que han surgido en estos tiempos, así como el incremento en el numero de usuarios de la red se ha propuesto una migración hacia una nueva versión de este protocolo.

IPv6 es la última versión del protocolo IP que ha sido diseñado y desarrollado para cubrir los puntos en los que falla su versión anterior. Además de que se ha creado pensando en el desarrollo a futuro de las redes, así como de las aplicaciones.

Cuenta con ciertas características que le permitirán adaptarse a nuevas necesidades y sobre todo, permite la coexistencia con su versión anterior. Esto resulta de gran importancia, ya que debe establecerse un periodo de transición entre las versiones de este protocolo ya que resulta ser imposible un reemplazo puntual por la magnitud de la red, así como las dificultades tecnológicas que esto conlleva a la mayoría de las personas que se encuentran conectadas a la red.

IPv6 resuelve desde su concepción el problema del espacio de direcciones que estaba cerca de agotarse para la versión anterior del IP. Uno de los principales "atractivos" de esta nueva versión es el encabezado con el que se diseño, podemos ver que tiene un menor número de campos y que aún así, trabaja de una manera más eficiente.

Para que pueda llevarse a cabo esta transición, es necesario que se cuente con herramientas que lo permitan; así como plataformas y aplicaciones básicas con las que pueda comenzarse la migración, así como continuar con el desarrollo de las nuevas versiones capaces de soportar el nuevo protocolo.

En cuanto a las plataformas disponibles, podemos decir que existen varias de ellas que cuentan con soporte para IPv6. Una de estas plataformas es Linux. Linux es un poderoso sistema operativo capaz de trabajar en diferentes arquitecturas de computadoras y requiere de dispositivos y hardware que pueden conseguirse fácilmente y que además no requieren de una gran inversión económica.

Linux ha nacido bajo el Unix y por lo tanto ha nacido en la red. Debido a esto, cuenta con características que lo hacen muy superior a otros sistemas operativos en cuanto al trabajo en la red. Esto se debe a la capacidad de configuración, así como a la posibilidad de modificación de su estructura desde la base del mismo.

Actualmente, Linux representa una base sólida en cuanto a las plataformas de desarrollo, a la investigación de las nuevas tecnologías, además de que la naturaleza del mismo, lo hace mantenerse actualizado y de acuerdo a las necesidades cambiantes de las aplicaciones y los usuarios.

Gracias a que ha sido desarrollado por personas que tienen una estrecha relación con la programación a niveles científicos y lúdicos, pueden aprovecharse los puntos de vista y las opiniones de desarrolladores que se encuentran esparcidos literalmente por todo el mundo. Cada uno de ellos refleja su interés especifico, pero a fin de cuentas, se mantiene un desarrollo formal.

De esta manera se han desarrollado aplicaciones para trabajar con IPv6. Actualmente se tienen las herramientas de configuración y algunas aplicaciones básicas para comunicación, configuración y monitoreo de redes con este protocolo. Las aplicaciones que atienden a los servicios básicos de Internet se encuentran en versiones estables y funcionales en este momento.

Aún no se cuenta con muchas aplicaciones desarrolladas para usuarios finales. Aunque se ha intentado, se ha quedado en eso, solo en el intento por lo que se tienen muchas versiones beta y todavía muchas mas versiones alfa de algunas aplicaciones.

Linux ha sabido adaptarse al nuevo protocolo y este ha podido aprovechar las ventajas con las que cuenta este sistema operativo. Podemos decir que gran parte del desarrollo actual y la investigación que se tiene sobre este protocolo tiene relación con este sistema operativo lo que se demuestra con el numero de aplicaciones que se desarrollan para el mismo.

Sabemos ahora, que un sistema Linux puede ser configurado para usar IPv6 como protocolo para sus comunicaciones con otros dispositivos de la red eficientemente. Vimos que la configuración del mismo para trabajo en redes locales no representa un problema mayor, pero que en cuanto a la comunicación con equipos que se encuentran en otras redes IPv6 aisladas por la red IPv4 surgen problemas.

Estos problemas se encuentran relacionados con el ruteo de los paquetes, algo que puede ser solucionado con la documentación adecuada.

Puedo decir que Linux e IPv6 en conjunto pueden trabajar como una herramienta para comunicaciones de alto nivel con un gran desempeño, pero que aún tendrán que esperar un poco de tiempo por las aplicaciones.

Aunque si lo vemos desde el punto de vista de la filosofía con la que se ha desarrollado Linux, podemos decir que no necesariamente tendremos que quedarnos esperando a que lo que necesitemos se encuentre instalado. Tenemos la ventaja de poder entrarle al código e intentar contribuir al desarrollo de las redes de comunicaciones con IPv6.

El desarrollo de las aplicaciones que trabajen con IPv6 y que corran en Linux, depende de mucha gente que no tiene intereses comerciales por lo que resulta un tanto incierta, aunque como sabemos, los desarrolladores para Linux, son gente que hace su trabajo por el solo gusto de hacerlo, sin ninguna obligación ni interés por lo que frecuentemente, resulta ser de una mejor calidad.

Con todo lo anterior, puedo decir que se tendrá al alcance de casi cualquier persona un sistema con características potentes con el que puedan realizarse comunicaciones en la red de manera eficiente y que se encuentre adaptado a las necesidades demandantes de las aplicaciones y los usuarios haciendo un uso más eficiente y optimo de los recursos de las redes de comunicaciones actuales y futuras.

ANEXO

Documentos generales

En esta sección se incluyen algunas referencias que fueron útiles para la realización de esta tesis. Pertenecen a la documentación traducida (lo mejor posible) para servir de consulta y como una referencia rápida de ciertos temas. Puede encontrarse una mejor traducción en los sitios del proyecto LUCAS.

```
Mauricio Hernández García
maurikk@servidor.unam.mx
septiembre 28, 2000.

En este documento esta la información que obtuve de los manuales que hay en el
sistema y que son referentes a los comandos que intervienen en la configuración
del sistema para ver si ahora si puedo hacer que funcione
correctamente con IPv6 y que salga bien.


ifconfig

NOMBRE
       ifconfig - sirve para configurar las interfaces de red

SYNOPSIS
       ifconfig [interface]
       ifconfig interface [aftype] options | address ...

DESCRIPCION
       Ifconfig se usa para configurar las tarjetas de red que se
       encuentran residentes en el kernel. Se usa cuando el sistema
       arranca para configurar las interfaces como va siendo
       necesario. Después de eso, se usa solamente cuando se
       requiere de una corrección o actualización del sistema.

       Si no se dan argumentos, ifconfig muestra el estado
       de las interfaces que se encuentran activas. Si se da solamente
       como argumento una interfaz, se muestra el estado de esa interfaz
       solamente; si se da como argumento -a, se muestra el estado
       de todas las interfaces (activas o inactivas). Con mas parámetros
       se configura la interface.

Familia de Direcciones
       Si el primer argumento después del nombre de la interface
       se reconoce como uno de los nombres soportados de las
       familias de direcciones, esa familia de direcciones se usa para
       establecer y mostrar todas las direcciones del protocolo.
       Las familias de direcciones que son soportadas actualmente son
       inet (TCP/IP), inet6 (IPv6), ax25 (AMPR Packet Radio), ddp
       Appletalk Phase 2), ipx (Novell IPX) y netrom (AMPR Packet
       Radio).

OPCIONES
       interface
              El nombre de la interface. Generalmente se trata de un
              nombre de controlador seguido por un número.

       up     Este parámetro activa la interface. Esta especificado
              implícitamente si la dirección es asignada a la
              interface.
```

down Este parámetro desactiva una interface.

[-]arp Activa o desactiva el uso del protocolo ARP en esa interface.

[-]promisc
Activa o desactiva el modo promiscuo de la interface. Si es seleccionado, todos los paquetes en la red serán recibidos por esta interface.

[-]allmulti
Activa o desactiva el modo all-multicast. Si es seleccionado, todos los paquetes multicast en la red serán recibidos por la interface.

metric N
Este parámetro establece la métrica para la interface.

mtu N Este parámetro establece la Unidad Máxima de Transferencia (MTU) de la interface.

dstaddr addr
Establece la dirección IP remota de una conexión punto-a-punto (como PPP). Se encuentra actualmente obsoleto.

netmask addr
Establece la máscara IP de la red para esta interface. Se encuentra puesto por default dependiendo de la clase de la red pero puede establecerse cualquier valor.

add addr/prefixlen
Agrega una dirección IPv6 a una interface.

del addr/prefixlen
Elimina una dirección IPv6 de una interface.

tunnel aa.bb.cc.dd
Crea una interface (IPv6-en-IPv4), para establecer el túnel hacia la dirección proporcionada.

irq addr
Establece la interrupción de línea que se usa para el dispositivo. No todos los dispositivos pueden cambiar dinámicamente su IRQ.

io_addr addr
Establece la dirección de inicio en el espacio de entrada/salida para este dispositivo.

mem_start addr
Establece la dirección de inicio para la memoria compartida usada por el dispositivo. Solamente pocos dispositivos la necesitan.

media type
Establece el puerto físico o tipo de medio para ser usado por el dispositivo. No todos los dispositivos pueden cambiar este parámetro y aquellos pueden variar en los valores que soportan. Los valores típicos son: 10base2 (thin ethernet), 10baseT (ethernet 10Mbps par trenzado), AUI (external transiver) y otros. Hay un tipo especial llamado auto que puede usarse para obtener el tipo de

dispositivo. No todos los dispositivos lo soportan.

[-]broadcast [addr]
Si se da una dirección como argumento, se establece la dirección del protocolo broadcast para la interface. Si no, establece o elimina el parámetro IFF_BROADCAST para la interface.

[-]pointtopoint [addr]
Este parámetro activa el modo punto-a-punto para la interface; o sea, por lo que el enlace directo entre dos maquinas. Si el parámetro de la dirección se proporciona se establece el protocolo de direcciones en el extremo del enlace.

hw class address
Establece la dirección de hardware de la interface, si el dispositivo soporta esta operación.

multicast
Establece el parámetro de multicast de la interface. Esto puede no ser necesario en dispositivos que se configuran automáticamente.

address
La dirección IP que se asigna a la interface.

txqueuelen length
Establece la longitud de la cola de transferencia para el dispositivo. Es útil para establecer valores pequeños para dispositivos lentos con baja latencia (enlaces por modem, ISDN) para prevenir transferencias en masa que causen problemas de tráfico.

NOTAS
Desde el kernel 2.2 no hay estadísticas explícitas para interfaces por alias. Si se requieren estadísticas por dirección, se tiene que usar el comando ipchains.

ARCHIVOS
/proc/net/socket
/proc/net/dev
/proc/net/if_inet6

RELACIONADOS
route, netstat, arp, rarp, ipchains

route

NOMBRE

route - muestra / maneja la tabla de ruteo IP

SYNOPSIS

route [-CFvnee]

route [-v] [-A family] add [-net|-host] target [netmask Nm] [gw Gw] [metric N] [mss M] [window W] [irtt I] [reject] [mod] [dyn] [reinstate] [[dev] If]

route [-v] [-A family] del [-net|-host] target [gw Gw] [netmask Nm] [metric N] [[dev] If]

route [-V] [--version] [-h] [--help]

DESCRIPCION

Route maneja las tablas de ruteo del kernel. Principalmente se usa para establecer rutas estáticas a hosts específicos o redes a través de una interface después de que ha sido configurada con ifconfig.

Cuando se usan las opciones add o del, route modifica las tablas de ruteo. Sin esas opciones, route muestra el contenido actual de las tablas de ruteo.

OPCIONES

-A family
se usa para especificar la familia de direcciones

-F opera en la tabla de ruteo del kernel llamada FIB (Forwarding Information Base). Se encuentra definida por default.

-C opera en el cache de ruteo del kernel

-v selecciona el modo informativo

-n muestra las direcciones numéricas en lugar de intentar determinar los nombres simbólicos de los hosts. Es útil cuando se intenta determinar cual es la ruta al servidor.

-e usa el formato de netstat para desplegar la tabla de ruteo. -ee genera una línea larga con todos los parámetros de la tabla de ruteo.

del borra una ruta

add agrega una ruta

target es el host o la red de destino. Se puede dar la dirección IP en notación decimal puntuada o el nombre del host o de la red.

-net el destino es una red

-host el destino es un host

netmask NM
Cuando se agrega una dirección a una red, la máscara de red es necesaria.

gw GW Los paquetes son ruteados a través del gateway. NOTA: el gateway especificado debe ser alcanzable primero. Es decir, se tiene que establecer y levantar una ruta al gateway anteriormente. Si se especifica la dirección de una de las interfaces locales, ésta se usará para decidir sobre la interface por la que serán ruteados los paquetes.

metric M
Establece el valor en el campo de métrica en la tabla de ruteo (se usa por los demonios de ruteo).

mss M Establece el tamaño máximo del segmento TCP Maximum Segment Size (MSS) para las conexiones a través de la ruta en Mbytes. El valor por default es el MTU del dispositivo menos cabeceras, o el menor MTU cuando se encuentra la ruta mtu discovery. Este parámetro puede usarse para forzar a que los paquetes TCP sean mas pequeños en el extremo cuando la ruta mtu discovery no funcione (frecuentemente es causado por una mala configuración de los firewalls en el bloque ICMP).

window W
Establece el tamaño de la ventana TCP para conexiones en la ruta a W bytes. Generalmente se usa solamente en redes AX.25 y con controladores que pueden manejar frames back to back.

irtt I Establece el initial round trip (irtt) para conexiones TCP sobre la ruta en I milisegundos (1-12000). Se usa solamente en redes AX.25.

reject instala un bloqueo de ruta, que será forzado a buscar una ruta que falle. Se usa por ejemplo para enmascarar redes antes de usar la ruta por default. No se usa para firewall.

mod, dyn, reinstate
instala o modifica una ruta dinámica. Este parámetro es para propósitos de diagnóstico y se usa por los demonios de ruteo.

dev If obliga a que la ruta sea asociada con el dispositivo específico, como el kernel en lugar de intentar determinar el dispositivo por si mismo (revisando las rutas que existen y las especificaciones del dispositivo, y donde la ruta es agregada). En la mayoría de las redes no es necesario.

Si dev If es la última opción en la línea de comandos, la palabra dev puede ser omitida, ya que se encuentra establecida por default. Si no, no importa el orden de los demás parámetros.

SALIDA
La salida de la tabla de ruteo del kernel esta organizada en columnas.

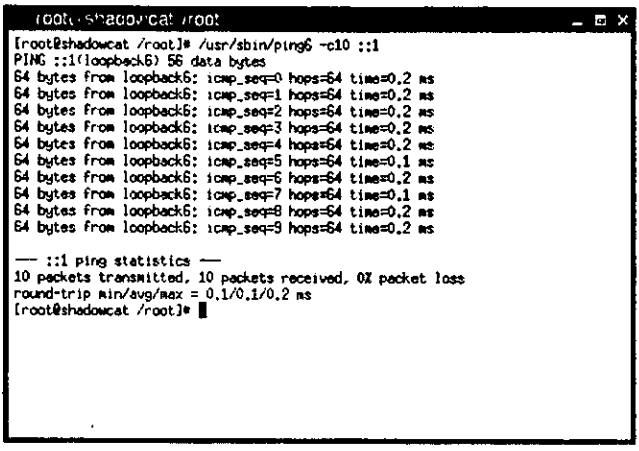
Destination
El destino de la red o host

Gateway
La dirección del gateway o un * si no se usa

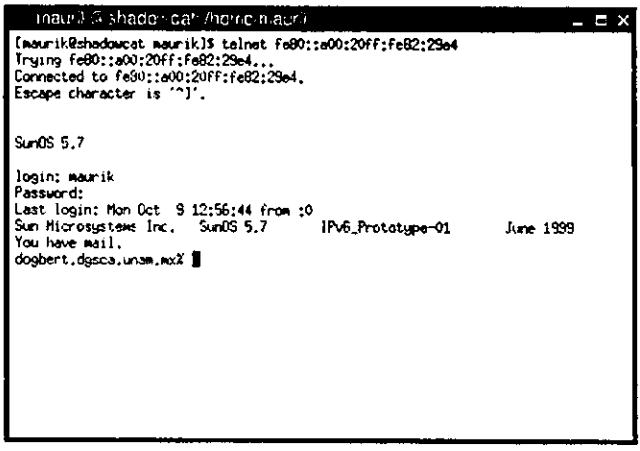
```
       Genmask
              La mascara de red para el destino; 255.255.255.255 para
              destino de host y 0.0.0.0 para la ruta por default.

       Banderas
              U  la ruta esta activa
              H  el destino es un host
              G  se usa un gateway
              R  ruta dinámica
              D  demonio dinámico de ruteo o redireccionamiento
              M  modificación del demonio de ruteo o redireccionamiento
              A  instalado por addrconf
              C  entrada de cache
              !  ruta rechazada

       Metric La distancia al destino (generalmente contada en saltos).
No es usada por kernel recientes, pero puede
              ser necesaria para los demonios de ruteo.

       Ref    Numero de referencias a esa ruta. (no se usa en el
              kernel de linux)

       Use    Cuenta las consultas de la ruta. Dependiendo de el uso
              de -F o -C puede ser de rutas del cache erróneas (-F) o
              exitosas (-C).

       Iface  Interface a la que los paquetes de esa ruta serán enviados

       MSS    Tamaño máximo del segmento para conexiones TCP sobre esta
              ruta.

       Window Tamaño por default de la ventana para conexiones TCP sobre
              la ruta.

       irtt   Initial RTT (round trip time). Es usado por el kernel
              para encontrar los mejores parámetros del protocolo TCP
              sin esperar respuestas.

       HH (solo cache)
              El numero de las entradas de ARP y rutas en el cache que
              se refieren a la cabecera de hardware para el cache
              de la ruta.

       ARP (solo cache)
              Actualiza el cache de la ruta con la dirección de hardware

ARCHIVOS
       /proc/net/ipv6_route
       /proc/net/route
       /proc/net/rt_cache

RELACIONADOS
       ifconfig, netstat, arp, rarp
```

Screenshoots

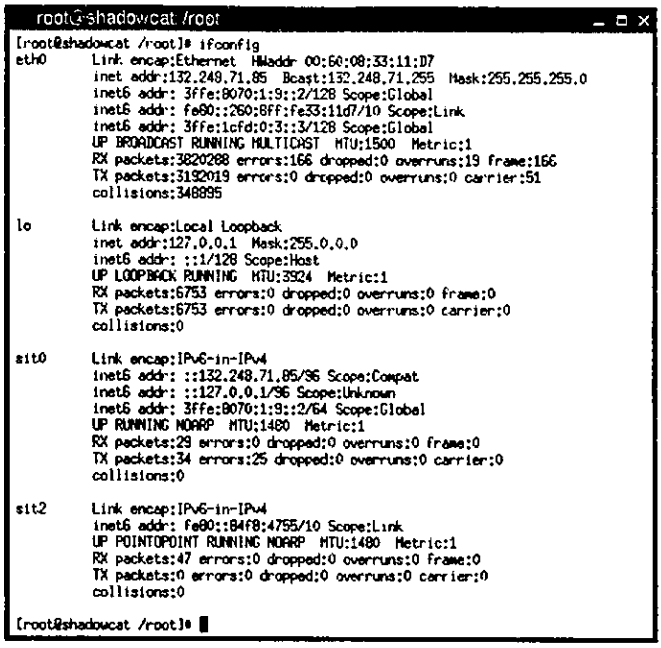
Incluí en esta sección imágenes que tomé en el sistema en el que estuve trabajando; se trata de varias terminales en donde se muestran datos referentes a la configuración del sistema o herramientas de monitoreo de la red del sistema en funcionamiento.



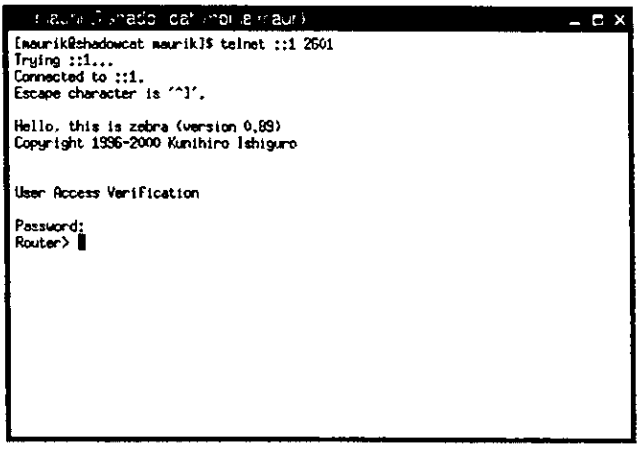
salida de ping



cliente de telnet



salida de ifconfig



usando Zebra

Características de la computadora

Aquí muestro las características con las que cuenta la computadora con la que trabaje en la configuración del sistema.

```
Mauricio Hernández García
maurik@servidor.unam.mx
octubre 24, 2000.


Aquí van las características de hardware de shadowcat.
-------------------------------------------------------

CPU                 Hewlett Packard Modelo Vectra VA
Monitor SVGA        Hewlett Packard 14 pulgadas
Teclado             Hewlett Packard
Mouse               Hewlett Packard
CD ROM              Hewlett Packard

Procesador          Intel Pentium Pro 180 MHz
Memoria RAM         64 MB
Swap                136 MB
Disco Duro          12 G
Tarjeta de red      3com FastEthernet

Particiones  hda1   30 M
             hda5   4.8 G
             hda6   4.8 G
             hda7   1.9 G
             hda8   243 M
```

BIBLIOGRAFÍA

Libros

Black, Uyless. Redes de ordenadores. Protocolos, normas e interfaces. 2ª edición. Ed. RA-MA México, 1995.	Redes de Computadoras	TK5105.5 B5318 1994
Harley Hahn Rick Stout Internet. Manual de referencia. Ed. McGraw-Hill España, 1994.	Internet	TK5105 875157 H34518
Paul G. Sery LINUX Network Toolkit. IDG Books Worldwide USA, 1998.	Linux	QA76.76063 S47
Tackett, Jack Utilizando Linux Prentice Hall México, 1996.	Linux	QA76.76063 T31318 1996
Peter H. Salus Big Book of IPv6 Addressing RFCs Morgan Kaufmann USA, 2000.	IPv6	406/00

Revistas

Internet World en Español Año 5, No. 9 pp 10	*La próxima generación*
Information Week Tecnología y negocios Volumen 1, Número 9 Febrero 2000 pp 36	*Linux se extiende* Scott Leibs
Network Computing México Volumen 1, Número 1 Agosto 1999 pp 14	*¿Ha llegado el momento de Linux?* Greg Shipley
Network Computing México Volumen 1, Número 4 Noviembre 1999 pp 40	*Ipv6, el cambio que todos esperamos* Enrique Oropesa

Network Computing México Volumen 1, Número 5 Diciembre 1999 pp 40	*IPv6, tierra a la vista* Enrique Oropesa
Network Computing México Volumen 1, Número 6 Enero 2000 pp 40	*IPv6, una revisión interna* Enrique Oropesa
PC Magazine en Español Volumen 10, Número 6 Junio 1999 pp 32	*Internet, EPISODIO II*
PC WORLD Año V, Número III Marzo 1999. pp 37	*Ancho de banda en Demanda* Harry McCracken
PC WORLD Año V, Número III Marzo 1999. pp 55	*Avanzando rápidamente hacia el futuro: Internet* Angela Navarrette
Personal Computing México Computación para la pequeña y mediana empresa Año 10, Número 120 Mayo 1998 pp 112	*Historia de Linux en México* Max de Méndizabal
tele.com Negocios y tecnología para proveedores de la Nueva Generación Volumen 1, Número 1 Noviembre 1999. pp 6	*¿Quién domina a los dominios?* Tyya Turner

URL's

http://www.cis.ohio-state.edu/htbin/rfc/rfc0793.html	RFC 793 TCP
http://www.cis.ohio-state.edu/htbin/rfc/rfc791.html	RFC 791 IP
http://www.cis.ohio-state.edu/htbin/rfc/rfc2460.html	RFC 2460 IPv6
http://www.cis.ohio-state.edu/hypertext/information/rfc.html	Internet Requests for comments (RFC)
http://www.ipv6.unam.mx	IPv6
http://www.linux-es.com	Rincón de Linux
http://www.internet2.unam.mx	Internet2
http://www.6forum.com	6Forum
http://www.bieringer.de/linux/IPv6	IPv6 para Linux
http://www.linux.net.mx/lucas	Proyecto LUCAS

http://www.redhat.com	Linux Red Hat
http://bofh.st/ipv6	Referencias a aplicaciones IPv6

Seminarios

Congreso General de Cómputo 99 Octubre 15, 1999. Antiguo Colegio de San Ildefonso Universidad Nacional Autónoma de México.
Seminario Internet2 en México Noviembre 15, 16 y 17 de 1999. Rectoría General Universidad Autónoma Metropolitana
Primer Seminario Nacional de IPv6 Diciembre 10, 1999. Antiguo Colegio de San Ildefonso Universidad Nacional Autónoma de México
Segundo Seminario Nacional de IPv6 Mayo 11, 2000. Palacio de Minería Universidad Nacional Autónoma de México
Primer Coloquio Nacional de Tecnologías Emergentes Mayo 12, 2000. Palacio de Minería Universidad Nacional Autónoma de México